Real-time Monitoring and Optimal Control for Smart Additive Manufacturing

by

Sepehr Fathizadan

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2022 by the
Graduate Supervisory Committee:

Feng Ju, Chair
Teresa Wu
Yan Lu
Ashif Iquebal

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

Additive manufacturing consists of successive fabrication of materials layer upon layer to manufacture three-dimensional items. Several key problems such as poor quality of finished products and excessive operational costs are yet to be addressed before it becomes widely applicable in the industry. Retroactive/offline actions such as post-manufacturing inspections for defect detection in finished products are not only extremely expensive and ineffective but are also incapable of issuing corrective action signals during the building span. In-situ monitoring and optimal control methods, on the other hand, can provide viable alternatives to aid with the online detection of anomalies and control the process. Nevertheless, the complexity of process assumptions, unique structure of collected data, and high-frequency data acquisition rate severely deteriorates the performance of traditional and parametric control and process monitoring approaches. Out of diverse categories of additive manufacturing, Large-Scale Additive Manufacturing (LSAM) by material extrusion and Laser Powder Bed Fusion (LPBF) suffer the most due to their more advanced technologies and are therefore the subjects of study in this work. In LSAM, the geometry of large parts can impact the heat dissipation and lead to large thermal gradients between distance locations on the surface. The surface's temperature profile is captured by an infrared thermal camera and translated to a non-linear regression model to formulate the surface cooling dynamics. The surface temperature prediction methodology is then combined into an optimization model with probabilistic constraints for real-time layer time and material flow control. On-axis optical high-speed cameras can capture streams of melt pool images of laser-powder interaction in real-time during the process. Model-agnostic deep learning methods offer a great deal of flexibility when facing such unstructured big data and thus are appealing alternatives to their physical-related and regression-based modeling counterparts. A configuration of Con-

volutional Long-Short Term Memory (ConvLSTM) auto-encoder is proposed to learn a deep spatio-temporal representation from sequences of melt pool images collected from experimental builds. The unfolded bottleneck tensors are then further mined to construct a high accuracy and low false alarm rate anomaly detection and monitoring procedure.

ACKNOWLEDGEMENTS

I would like to appreciate my family specially my mom for her unconditional love, Dr. Feng Ju and and the dissertation committee members for their great support throughout this journey, and Feifan Wang for his collaboration and friendship.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Additive manufacturing consists of successive fabrication of materials layer upon layer to manufacture three dimensional items. There are diverse categories of additive manufacturing processes including Photopolymer vat processes (PVP), Material jetting processes (MJP), Binder jetting processes (BJP), Extrusion-based processes (EBP), Powder Bed Fusion (PBF), Directed Energy Deposition (DED) and Sheet Lamination Processes (SLP) (Astm (2015)). Out of various additive manufacturing technologies, material extrusion and specifically Fused Filament Fabrication (FFF) (also referred to as Fused Deposition Modeling (FDM)) is a more prevalent process for its availability to practitioners and low cost of operation (Bikas *et al.* (2016)). The technologies of large-scale material extrusion and Laser Powder Bed Fusion (LPBF) however, are new compared to other additive manufacturing technologies. Thus, several key problems are yet to be addressed before they are widely applied in industry. Amongst many issues, the practical application of printing machines is mainly bounded by the poor quality of finished products and excessive operational costs.

The abundance of real-time data collected during the build-span lends itself well to the development of optimal control and monitoring techniques to aid with online parameter tuning and anomaly detection in the process. On the other hand, the unique structure of collected data and high-frequency data acquisition rate pose a challenge toward more traditional and parametric control and process monitoring approaches. Several unrealistic assumptions in the existing methods need to be revised and later incorporated in the proposed methods to properly address the quality and efficiency in the additive manufacturing process.

1.1 Background Information for Large-scale Additive Manufacturing

In FFF, a printer head with an extrusion nozzle attached moves horizontally to deposit the continuous filament of thermoplastic material on the surface. As soon as the previously laid down substrate is cool enough to accept another bead of freshly molten material, it moves vertically to print the next layer on top of the current layer. FFF is usually utilized to fabricate delicate items in small sizes. However, recent developments in technology introduced more advanced machines capable of manufacturing high volume items as large as 30.4m long, 3.0m wide and 1.5m high. Industrial machines such as Big Area Additive Manufacturing (BAAM$^{)}$ developed at Oak Ridge National Laboratory (ORNL) (Duty *et al.* (2017)) and Large Scale Additive Manufacturing (LSAM$^{\text{TM}}$) developed by Thermwood (Thermwood (2019b)) are using feedstock of thermoplastic materials to manufacture voluminous parts. The LSAM machine fabricating a car lower chassis is shown in Figure 1.1. Carbon fiber reinforced thermoplastic material is used for large-scale additive manufacturing to fabricate parts in large size. The process might takes several hours or even days to get finished printing a large-scale product. Furthermore, the finished products have to go through several steps of post-build processing before dispatching. Specifically, the large size of the parts poses a challenge to practitioners on how to find and tune the operational settings. Currently, these settings are mostly determined based on the experience and intuition of the operator.

It has been shown that the current surface temperature of the layer on which the next layer is about to be printed significantly impacts the mechanical properties of the printed product (Sun *et al.* (2008)). For instance, part deformation as the result of weak bonding between layers can happen if the surface temperature is too high (Brenken *et al.* (2018)). On the other hand, cracking and warping can occur

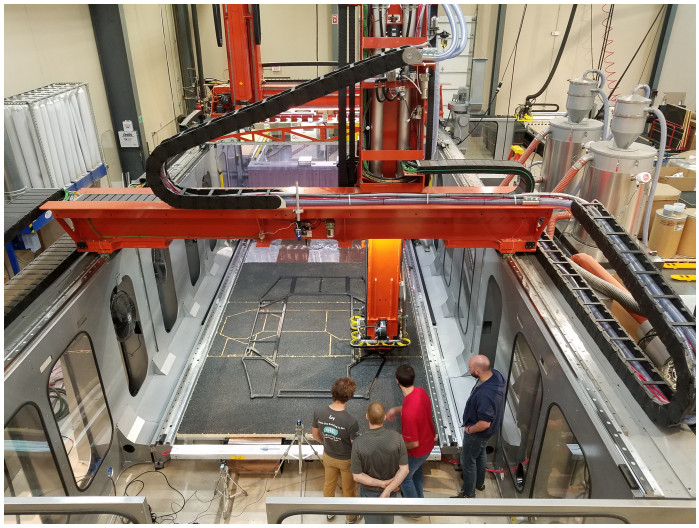


Figure 1.1: LSAM^{TM} Machine

if the surface temperature is less than the glass transition temperature (Compton *et al.* (2017)). Figure 1.2 illustrates the crack defects on a sample print due to improper print surface temperature. Therefore, it is important to keep the print surface temperature within an interval for each layer. The lower and upper bounds of this interval depend on the type of the feedstock materials used by the machine. The print surface temperature and heat dissipation of each location on the surface are impacted by several factors such as airflow, humidity, ambient and deposition temperatures, mechanical properties of the materials and the geometry of the parts.

On the other hand, print surface temperature can be controlled by the means of a few levers such as proper scheduling of the material depositions on the surface and active cooling. The time between two consecutive prints is referred to as layer time. Moreover, for each location on the surface, the print time is defined as the time when it accepts a new deposition. Either too large or too low layer time and print time could lead to improper print surface temperature outside the desired interval. Therefore, it is important to attain an optimal control of the materials depositions on

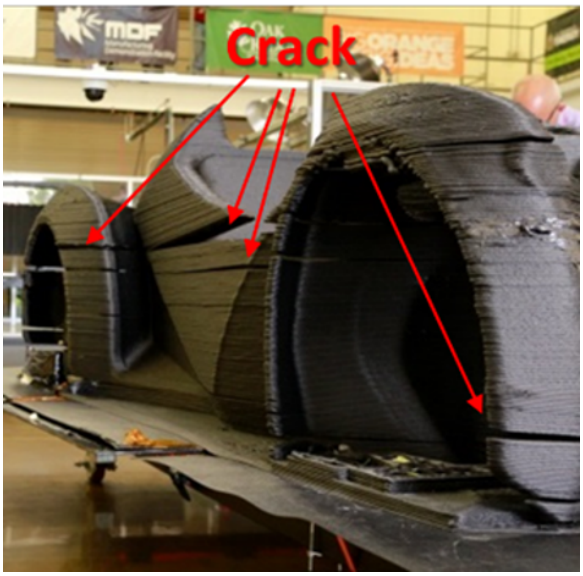


Figure 1.2: Crack Defects Due to Improper Print Surface Temperature

the surface. Currently, a commonly used method is to fix the layer time and printer head speed to simplify the process. However, to follow the dynamics of the process, both layer time and printer head speed can be adjusted and tuned in real time.

1.1.1 Print Surface Thermal Modeling and Layer Time Control for Large-scale Additive Manufacturing

The time between two consecutive prints is referred to as layer time. This chapter aims to provide a robust real-time method that dynamically control the layer time during the build span. The surface's temperature change in large-scale material extrusion is captured by an infrared thermal camera in real-time. A non-linear regression model is formulated and proved to fit the cooling dynamics. To deal with the layer-wise change of cooling dynamics, due to humidity and air flow, the Gaussian process is used to keep the regression model updated. The regression model together with the Gaussian process can predict surface temperature of a part accurately, even in a

dynamic environment. This method to predict the surface temperature is then combined into an optimization model with probabilistic constraints for a robust real-time layer time control. Specifically, more than one position on the surface is monitored and considered in the optimization model, and the resulting layer time for each layer by solving the optimization model has the quality requirement satisfied and improves production efficiency.

1.1.2 Dynamic Material Flow and Machine Motion Control for Large-scale Additive Manufacturing Using Thermal Images

For each location on the surface, the print time is defined as the time when it accepts a new deposition. The material flow and machine motion control through printing with adjustable speed determine the print time for each location. This allows for a greater control over printing compared to the layer time control case where the print time is the same for all location. To this aim, first the thermal images are processed to extract the thermal profiles. Similar to layer time control, the extracted data are translated into a non-linear profile model describing the heat dissipation on the surface. Exploiting the maneuverability characteristics of the printer head while considering its mechanical constraints, a real-time printer head speed control model is formulated as a non-linear mixed integer optimization program to determine the material flow and machine motion. Following deterministic finite-state optimal control and shortest path problem paradigm, a novel algorithm is developed to decide for the optimal printing speed trajectory for each layer. The proposed approach was tested by two case studies including a thin wall specimen and a car lower chassis. The results showed that the method can capture the thermodynamics of the process and achieve a further improvement in both quality and efficiency.

1.2 Background Information for Laser Powder Bed Fusion

LPBF is an additive manufacturing technique that utilizes laser power to melt and fuse the thin layers of spread powder on a build plate. Iterative laser beam scans on the build surface, layer upon layer and dictated by a CAD model, ultimately fabricate the finished product. Melt pools are formed when powder material is heated and reaches the melting temperature. Lack of fusion or over melting phenomena contribute to the development of pores in the solidified material, which in return can degenerate the mechanical properties of the finished parts (Frazier (2014); Gong *et al.* (2014); Sadowski *et al.* (2016)). In particular, extensive studies have shown that the formation of such defects on parts is highly correlated to the morphological characteristics of the melt pool (Tang *et al.* (2017); Cunningham *et al.* (2017); Khairallah *et al.* (2016)). The laser power, scan velocity, and hatching distance are the primary LPBF process parameter acting on melt pool formation and consequently their morphological characteristics. Nevertheless, many random or environmental factors such as material's fluid dynamics and properties, layers thermal conduction and convection, and tools calibration also impact the characteristics of melt pools (Yang *et al.* (2020b); Ciurana *et al.* (2013)). Thus, despite having a set of stable process parameter settings, melt pools must be actively monitored in real-time to detect the anomalies in the process.

On-axis thermal or optical high-speed cameras can capture streams of co-axial melt pool images of laser-powder interaction in real-time during the process. A high image sampling frequency poses a challenge toward conventional process monitoring methods that rely on individual samples for fault detection and alarm management. Furthermore, the scarcity of anomaly events makes it extremely hard to annotate out-of-control samples and draw a high-fidelity baseline for in-control conditions.

One widely used solution is to associate each melt pool image to either a normal or defective spot on the part determined by Non-Destructive Testing (NDT) such as X-ray micro-Computed Tomography (XCT) scanning or destructive inspection methods (Gobert *et al.* (2018); Zhang *et al.* (2019)). However, obtaining reliable labels with either approach is tremendously time-consuming and expensive due to the aforementioned reasons, and the results might suffer from a lack of generalization, especially in job shop process manufacturing flows. On the other hand, a cheaper yet effective alternative is to utilize the correlation between defects and morphological malformation in melt pool images such as splattering, deformation, tiny or oversized melt pool regions (Fathizadan *et al.* (2021); Repossini *et al.* (2017)). Given a set of high-quality features, the proximity in structural properties of features can be explored to distinguish the normal and abnormal events in the process.

1.2.1 Individual Anomaly Detection for Process Variation Management in Laser Powder Bed Fusion

The state-of-the-art methods often rely on oversimplified and hand-crafted features of the melt pool to analyze the images (Yang *et al.* (2020b, 2019a)). These methods fail to preserve the spatial correlation within images and lose the valuable and intricate information inside each sample. A framework to process the melt pool images by a configuration of Convolutional Auto-Encoder (CAE) neural networks is provided. The network's corresponding bottleneck layer learns a deep low-dimensional representation from melt pools while preserving the spatial correlation and complex features intrinsic in the images. As opposed to the manual annotation of data by X-ray imaging or destructive tests, an agglomerative clustering algorithm is applied to these representations to automatically extract the anomalies and annotate the data accordingly. A control charting scheme based on Hotelling's T^2 and

S^2 statistics is then developed to monitor the process's stability by keeping track of the learned representations and residuals obtained from the reconstruction of original images. An experimental build using a spiral concentrating scan pattern with varying laser power was conducted on the Additive Manufacturing Metrology Test bed at the National Institute of Standards and Technology (NIST) (Lane and Yeung (2019)). Testing the proposed methodology on the collected data demonstrates that the method can extract a set of complex features that are inextricable by using hand-crafted feature engineering methods. Finally, through extensive numerical studies, it is shown that the proposed feature extraction and statistical process monitoring scheme is capable of detecting the anomalies in real-time with relatively high accuracy.

1.2.2 Deep Spatio-temporal Anomaly Detection in Laser Powder Bed Fusion

In addition to spatial correlation, the state-of-the-art melt pool control and monitoring methods in additive manufacturing also disregard the rich temporal interdependencies between melt pool images sequences (Yang *et al.* (2020b, 2019a) ; Fathizadan *et al.* (2021)). The previous work on individual anomaly detection methodology is further extended by incorporating both temporal and spatial correlation into the feature extraction part. To this aim, a configuration of convolutional long short term memory auto-encoders is developed to learn a deep spatio-temporal representation from sequences of melt pool images. Unfolded bottleneck tensors and reconstructed sequences are statistically mined to detect the group anomalies in the process. A dual control chart monitoring scheme based on Hotelling's T^2 and residual's variance S^2 is again used as an easy-to-implement and intuitive tool to monitor and control the process in real-time. Deploying a constant laser power, scan velocity, and hatch distance, four nominally identical experimental builds with cylindrical features

were printed on the Additive Manufacturing Metrology Testbed at the National Institute of Standards and Technology (NIST) (Lane and Yeung (2020)). The collected data are used for training, validation, and testing the proposed methodology. The results demonstrate the superiority of spatio-temporal extracted features in detecting anomalies compared to hand-crafted feature engineering methods.

1.3 Organization of the Document

In what follows, we first discuss an overview of relevant studies regarding both LSAM and LPBF in Chapter 2. The experiment settings, thermal profile modeling, and real-time layer time control method regarding LSAM will be explained in Chapter 3. Chapter 4 introduces the pipeline to extract the data from thermal images and material flow and machine motion control through printer head speed control models for LSAM. The structure analysis and procedure to solve each model along with the numerical results by evaluating two case studies will be provided in these two chapters. In Chapter 5, the proposed pipeline to process and analyze the melt pool images from LPBF, including convolutional auto-encoder for deep representation learning, agglomerative clustering for data annotation, and statistical process control for anomaly detection, will be explained. Finally, constituents of the proposed methodology for deep spatio-temporal anomaly detection in LPBF, including convolutional long short term memory auto-encoders for deep spatio-temporal representation learning, hierarchical-based agglomerative clustering for anomaly extraction, and dual control charting scheme for statistical process monitoring, will be delineated in Chapter 6. Given the experiment design including material, part geometry and scan strategy, and laser processing parameters, the numerical results of the proposed approaches after applying on real-world case studies are discussed in the latter two chapters. Finally, Chapter 7 concludes the dissertation by summarizing

the overall work.

Chapter 2

LITERATURE REVIEW

2.1 Improve efficiency and quality through process control and design of experiment in various technologies of additive manufacturing

There exists several studies on how to achieve efficiency and quality through process control, Design of Experiment (DOE) and Response Surface Methodology (RSM) in various technologies of additive manufacturing. They mostly inspect the mechanical or material properties of the specimens by conducting different experiments. Anitha *et al.* (2001) investigated the effect of FFF process parameters such as layer thickness on the quality characteristics of the final products using Taguchi method and analysis of variance. Employing central composite designs, Sood *et al.* (2010) studied the relationship among the printing parameters such as layer thickness and printing orientation. Given the effect of these parameters on the mechanical properties of specimens such as strength, an optimal settings was pursued to improve the quality. Using factorial designs, Asadi-Eydivand *et al.* (2016) investigated the effect of layer time and printing orientation on dimensional accuracy, compressive strength and porosity in BJP. Zeng *et al.* (2012) suggested that process control by temperature monitoring can result in a higher quality. They also concluded that better quality is achieved when temperature is uniformly distributed throughout the surface. In another study on FFF, Mohamed *et al.* (2016) used Q-optimal designs to decide the best parameters setting for reducing the production makespan and feedstock material consumption while maintaining high quality.

Mireles *et al.* (2013) and Mireles *et al.* (2015) developed a decision support system that uses thermal imaging information to detect defects and anomalies in PBF. It was shown that through the use of optimal process control, this tool is capable of stabilizing the temperature and thus increasing the quality. Farzadi *et al.* (2015) evaluated the effect of a limited set of layer printing delays on the mechanical properties of prototypes such as dimensional accuracy and material strength. In another study in JMP, Sitthi-Amorn *et al.* (2015) developed a closed-feedback loop 3D printing platform that incorporates 3D scanning and imaging to calibrate the printer head in real-time. Yao *et al.* (2018) first used multifractal analysis to characterize the defect level of layer-wise thermal images in PBF. Then, they formulated the defect conditions as a Markov process to determine the optimal control policy of whether to perform a corrective action at each layer. Khanzadeh *et al.* (2019) used melt pool thermal imaging to characterize the underlying thermo-physical dynamics of the DED process. Self-Organizing Maps (SOMs) was employed as the next step to predict porosity in thin wall specimen in real-time. Finally, Liu *et al.* (2019) developed an closed-loop quality control system based on FFF thermal images. Coupled with an online defect detection classifier, the system can adjust the material flow rate, extruder temperature, and layer height to minimize the defects in real-time.

2.2 Improve Efficiency and Quality Through Process Control in Fused Filament Fabrication

FFF has drawn tremendous attentions since its introduction in the late 80s because of its wide availability to the practitioners and relative low cost. Several studies have been devoted to achieve high quality and efficiency through tool path planning in FFF process (see Agarwala *et al.* (1996), Han *et al.* (2002), Huang *et al.* (2011)). Han *et al.* (2003)) performed build time analysis to improve process efficiency of FFF. Jin *et al.*

(2019) formulates an optimization model that aims to minimizing the makespan of the printing of the products. Given collision-avoidance constraints, a heuristic procedure was developed to find the optimal solution. By investigating process conditions, Sun *et al.* (2008) proposed lateral geometric path to improve the bonding quality of FFF. However, these approaches and results are not necessarily applicable within the large scale additive manufacturing domain due to different printing and material characteristics. For instance, heat transfer via conduction and convection is not negligible anymore in large-scale additive manufacturing. Given that several factors affect this interaction, new methodologies for data extraction and analysis are required to provide control in real-time based on the dynamics of the process.

2.3 Current Research on Layer Time and Material Flow Control in Large-scale Additive Manufacturing

The building dimension of manufactured products in large-scale additive manufacturing is considerably larger than that of regular additive manufacturing technologies. As the result, significantly larger volume of thermoplastic material in high temperature and with distinct mechanical and thermal characteristics is consumed by the printer nozzle. This renders the cooling process to be notably slower compared to regular additive manufacturing such as FFF, where the printer head keeps depositing the materials without pause and purge. Therefore and in contrast to small-scale additive manufacturing, the effect of heat transfer via conduction and convection on the dynamics of the process is not negligible anymore and needs to be addressed from the automation manufacturing point of view. Furthermore and due to the small scale, the speed of the printer head has been regraded as an irrelevant factor to the monitoring and control of these processes since it is plausible to assume new layers can be deposited almost instantaneously. Instead, the literature and practitioners in

these areas devoted their efforts to investigate the effect of other deemed important experiment settings such as layer thickness, printing orientation and deposition temperature on the mechanical and quality characteristics of the finished products as it was discussed by Section 2.1-2.2. Another important distinction factor between large-scale versus regular additive manufacturing lies in the building span of the operation. Currently, it takes hours to print a single part on a LSAM^{tm} machine, making the proper scheduling of printing operations vital to address this issue.

Additive manufacturing making large sized parts was studied as early as 1990s (Khoshnevis and Dutton (1998)). The majority of early practice of large-scale additive manufacturing was in the construction industry and aimed at an approach to construction automation (Pegna (1997); Khoshnevis *et al.* (2004); Khoshnevis (2004); Labonnote *et al.* (2016); Buswell *et al.* (2018)). Large-scale additive manufacturing for production systems received little attention until recent years. The early version of BAAM machine was developed around 2013 and was as simple as an extruder installed on a robotic arm (Holshouser *et al.* (2013)). A more complete BAAM machine was developed soon after that and successfully fabricated a customized electric vehicle (Love (2015); Talagani *et al.* (2015)). Paper by Kunc *et al.* (2017) shows examples of trim tools, hand layup tools, vacuum assisted resin transfer molding tools and autoclave tools fabricated by the BAAM machine. Another large-scale 3D printer designed for production systems is the LSAM™ machine (Thermwood (2019b,a)). It operates by a similar mechanism and provides similar functionality but has a larger size. Thermoset printing, as an alternative of large-scale additive manufacturing, adopts thermoset polymers, instead of thermoplastic polymers used in the BAAM machine and the LSAM™ machine. One application of this technology is the large scale reactive polymer additive system, also referred to as Thermobot (Kunc *et al.* (2018); Lindahl *et al.* (2018); Romberg *et al.* (2019)). Thermoset polymers behave

viscoelastically at room temperature, and it reduces the dependence on temperature control (Lindahl *et al.* (2018)). The study, presented in this paper, focuses on large-scale additive manufacturing that takes thermoplastic polymers as raw materials and is applied to production systems. The LSAM™ machine is adopted to conduct experiments.

Extensive review studies were conducted by Kim *et al.* (2018) and Colosimo *et al.* (2018) on the quality control efforts in additive manufacturing. As it has been outlined, several issues need special attentions to ensure the reliability of additive manufacturing process, including prediction of optimal printing parameters and mechanical properties, robust real-time monitoring and process control, and high speed fabrication and scale. Likewise, the problem of how to achieve efficiency and quality through manufacturing automation and control of surface temperature in large scale additive manufacturing has almost remained untouched. Guo and Leu (2013) emphasized the need for future models to address material flow and machine motion control in various technologies of additive manufacturing in large scale. Dreifus *et al.* (2017) examined the deposition scheduling of large scale additive manufacturing, but they failed to take print surface temperature into account in the optimization problem. Chesser *et al.* (2019) examined the multi-resolution printing, extrusion diversion, and feed-forward extruder control strategies to improve the geometric quality characteristics of the parts manufactured by BAAM machine. A handful of studies also relied on the use of thermal images to extract and model temperature profiles in large scale additive manufacturing. However, they all suffer from multitude of simplistic assumptions and missing important information in the process. For example, focusing on a limited portion of the printed component, Compton *et al.* (2017) studied the temperature cooling dynamics alongside the vertical-axis for a fixed location on each layer. It has been also assumed that new layers are deposited instantaneously and

that there is no thermal gradient across the surface . This assumption is not certainly true in practice. It is almost impossible to deposit a new layer instantaneously as the printer head can only operate within a limited speed range. Accordingly, the deposition time gap between distant locations causes thermal gradients across the surface. In a most recent study done at ORNL by Borish *et al.* (2019), the simple average of actual temperatures across the surface is taken to decide when to print the next layer. Again, this is not applicable in practice as the distribution of temperature varies a lot across the surface.

2.4 Melt Pool Image Feature Extraction and Process Variation Management in Laser Powder Bed Fusion

Melt pool image analysis and feature extraction methods in the literature can be classified into three major classes: parametric modeling, non-parametric modeling, and deep learning approaches. Among all the studies, only a few make efforts to incorporate the effect of spatio-temporal correlation between data into their modeling approaches. Regarding parametric modeling, Yang *et al.* (2019b) extracted the area of a fitted ellipse around each melt pool using a thresholding constant. Laser power, scan velocity, and neighborhood spatio-temporal scanning effects are selected as input variables to predict the melt pool area. Building upon the previous study, Yang *et al.* (2020a) developed a Layer-wise Neighboring-Effect Modeling (L-NBEM) method that uses a fully-connected neural network to predict the melt pool area. Predictor variables this time are re-engineered to include the data from previous layers' scan and exposure settings. Guo *et al.* (2019) proposed a framework based on a general linear mixed model called Spatio-Temporal Conditional Auto-Regressive (STCAR). The model accounts for spatial and temporal correlations within and between images, respectively. The stability of the process is controlled by monitoring the

STCAR model's estimated parameter using a Hotelling's T^2 control charting scheme. Given the sparse nature of spatio-temporal anomaly events, Yan *et al.* (2021) decomposed the original tensor of melt pool images to various events, including background, and natural and anomaly foregrounds. The parameters of the formulated regression model are estimated by an algorithm of block coordinate descent and monitored by a likelihood ratio test statistics.

Several studies are devoted to non-parametric modeling with handcrafted features in this area. Khanzadeh *et al.* (2019) applied bi-harmonic surface interpolation in the spherical coordinate system to extract thermal features from individual melt pool images. Self-Organizing Maps (SOM) is then utilized to cluster the normal and porous samples while the accuracy of results is tested by actual labels obtained by XCT-scan imaging. In a similar effort, Khanzadeh *et al.* (2016) analyzed melt pool morphology by converting the melt pool boundary data points to a polar system and taking the fitted cubic spline coefficient weights as features. Functional Principal Component Analysis (FPCA) and tensor decomposition are amongst the other feature extraction methods used by Khanzadeh *et al.* (2018a,b) that strive to preserve the spatio-temporal characteristics of melt pool images. Support Vector Machines (SVM), Decision Trees (DT), and Hotelling's T^2 control charting are used as the statistical schemes to classify and detect the porous samples following the melt pool characterizations. The results, however, suffer from large standard deviations in accuracy and F_1 scores. Grasso *et al.* (2017) acquired the vector of intensities for each pixel throughout the layers after unfolding the tensor of melt pool images by Vectorized Principal Component Analysis (VPCA). The vanilla k-means is used to cluster the set of weights and Hotelling's T^2 distances assigned to each frame and each pixel, respectively. In another study conducted by Grasso *et al.* (2018), image segmentation and contour detection is performed to retrieve the melt pool area from infrared melt

pool images. The extracted features are then fed by a bi-variate control charting method based on Hotelling's T^2 statistics to monitor and control the process over time. Gobert *et al.* (2018) is another example of studies with handcrafted-feature engineering where feature extraction is realized by sliding $3D$ convolution kernels on tensors of multi-modality data. The votings from an ensemble of multiple classifiers trained on individual modalities are concatenated to predict the porosity of each x, y, z location on the surface.

The loss of rich spatio-temporal correlation between data as the result of handcrafted feature engineering is the common and inevitable issue of all aforementioned studies. Recently, deep learning methods have emerged in new research streams to address this issue by interlacing feature extraction with model training. Conclusively, the structure of collected data determines the set of advanced features required to achieve superior performance in intended statistical process monitoring, anomaly detection, or labelling tasks (Goodfellow *et al.* (2016)). Nevertheless, the deployment of such deep learning methods within the additive manufacturing domain is at an infancy stage. Amongst the very few studies, Kwon *et al.* (2020) fed the melt pool images to the Convolutional Neural Networks (CNN) to fit a regression model for laser power estimation. The model architecture is capable of capturing the spatial correlation within the melt pool images while it fails to account for the temporal correlation between data samples. Very similar work is done by Yang *et al.* (2019a), where the areas of individual melt pools are predicted by training a CNN model on melt pool images. Zhang *et al.* (2019) extracted the size and location of a porous region by performing thresholding and feature matching on the cross-sectioning and XCT imaging data. Various classification and regression tasks are conducted by feeding the melt pool images to CNN model architectures to predict the locations and sizes of pores. Following an analogous path in this category of additive manufac-

turing research, we develop a deep spatio-temporal feature extraction method that provides a superior anomaly detection performance compared to existing parametric and non-parametric studies.

2.5 Summary of Key Technical Contributions of the Work Toward Large-scale Additive Manufacturing

In summary, the mentioned above works regarding LSAM lack a proper holistic modeling of the correlation between operation parameters and important quality characteristics or efficiency of the process from manufacturing automation point of view. To the best of our knowledge, there exists no established online monitoring framework to utilize such a correlation to provide real-time control specially for large-scale additive manufacturing. The key technical contributions of this work aiming at filling these gaps can be highlighted as follows:

1. A novel methodology has been developed to address the issue of robust real-time monitoring and process control for large-scale additive manufacturing. Specifically, printing layer time and printer head speed control models were proposed to optimize the scheduling of the material depositions on the surface online according to the thermal dynamics of the print surface and mechanical characteristics of the extruder machine. Disregarding such factors, the current practice in LSAM is to implement a fixed layer time and fixed printer head speed strategy. Moreover, the previous studies all lack a realistic perspective of the process leading to multitude of oversimplifying assumptions such as ignoring thermal gradient or assuming instant deposition across the surface.

2. A modeling framework has been developed to characterize the temperature dissipation on the entire surface. Furthermore, a set of digital image processing

steps along with a temperature profile modeling framework assist to extract and model the thermal profiling data with minimum amount of supervision in real-time.

3. To ensure the viability of our developed control models, they were all analysed in great details and the corresponding solving procedures were either provided or developed for each one. Finally, relying on the numerical results from real-world case studies, it was proved that our method can lead to a simultaneous improvement in both the fabrication speed and quality of parts.

2.6 Summary of Key Technical Contributions of the Work Toward Laser Powder Bed Fusion

A prior technical knowledge of the specific AM technology is often necessary to model the physics-based relationships and detect the anomalies in the process. On the other hand, high-dimensional and spatio-temporally correlated co-axial images are widely captured during the laser powder bed fusion process. However, data-driven regression-based methods suffer from inadequacy and lack of generalizations in the feature extraction development phase. A framework to translate the collected data into simple and easy to understand statistical scalars in real-time is developed. Practitioners will be equipped with generalized feature engineering and ultimately control charting methodologies that automate the anomaly detection process without requiring to understand the underlying physical relationships in the process.

Chapter 3

PRINT SURFACE THERMAL MODELING AND LAYER TIME CONTROL FOR LARGE-SCALE ADDITIVE MANUFACTURING

3.1 Analysis of Surface Temperature

3.1.1 Motivation

Fabrication of vehicles through large-scale additive manufacturing has been proved to be possible. However, the gap from the first prototype to mass manufacturing is always underestimated due to a lot of issues related to quality and efficiency. Great effort is being made to keep a consistent quality. As has been discovered, the quality of a final product is sensitive to the print surface temperature, which is referred to as the temperature on surface of part before a new layer is about to be added.

Fig. 3.1 shows the result of an experiment about the strength and layer time. As the layer time increases, the surface temperature decreases. It causes a small strength, which further results in bonding issue like crack. Thus, small print surface temperature is supposed to be prevented. Meanwhile, the print surface temperature cannot be larger than a certain level. Hot thermoplastic materials could be not stiff enough to support a new layer, and the corresponding consequence is deformation of final product.

Improvement of efficiency and quality through a proper print surface temperature can be achieved by manufacturing automation. This work is intended to contribute to this end.

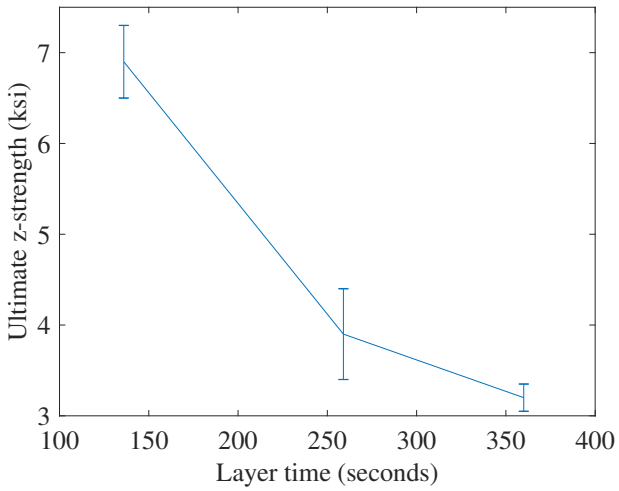


Figure 3.1: The Relationship Between the Strength and Layer Time

3.1.2 Experimental Setup

The machine used to perform large-scale additive manufacturing is the LSAM™ machine, shown in Fig. 1.1. The process of fabricating a part consists of two steps. The part is first printed layer by layer and then trimmed by a CNC router. There are two gantries on a single LSAM™ machine. One is to perform the additive, and the other the subtractive process. The gantry with a print head can be seen at the center of Fig. 1.1. The pelletized thermoplastic material is dried and pneumatically conveyed to the print head. Fig. 3.2 provides a schematic of the print head. The thermoplastic material gets into heated barrel through feed throat. Several heat zones are there in the heated barrel to get the thermoplastic material into a softened state. A nozzle and a compression wheel are installed at the bottom of the print head. The melted material is squeezed out of the nozzle, and then the compression wheel flattens the extruded bead. Several cooling fans are equipped on the two sides of the envelope to expedite the cooling process if needed. This cooling process is monitored by FLIR™ *A*65 infrared camera, which is capable of capturing temperatures in the

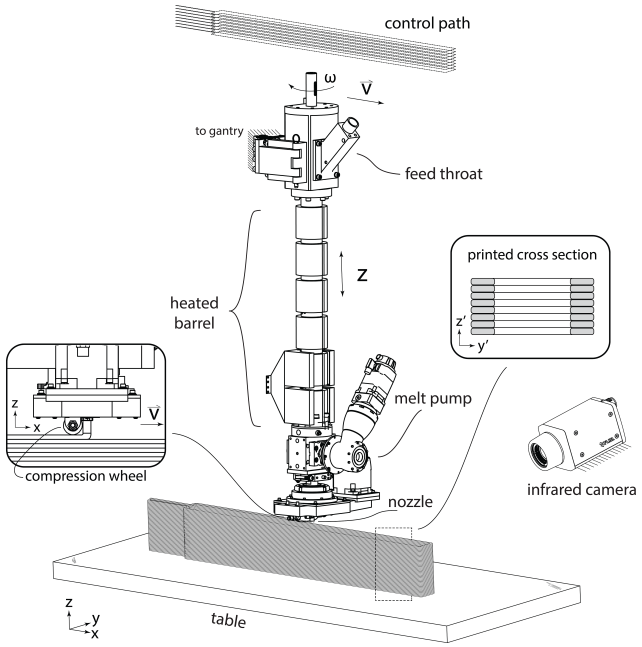


Figure 3.2: Schematic of Print Head for Large-scale Additive Manufacturing

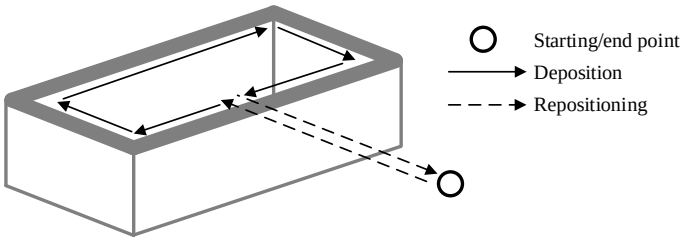


Figure 3.3: Thin Wall and Tool Path

range from -40°C to 550°C with accuracy of ±5°C.

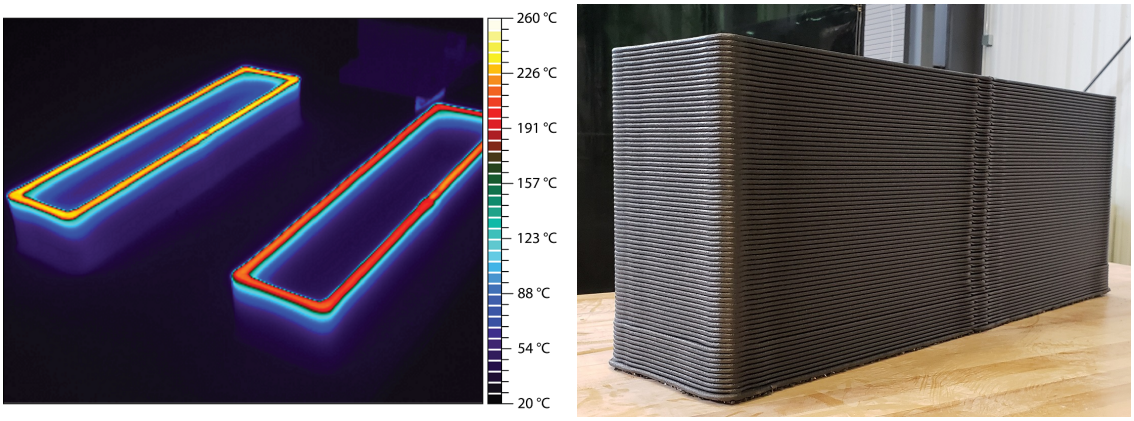
In this study, carbon fiber filled polycarbonate (CF/PC) compound named "SABIC DC004" is used as the thermoplastic material to print thin wall tests. A single-bead-wide layer is added on the surface in a clockwise motion for each layer, shown in Fig. 3.3. Then, the print head moves to the the starting/end point, performing a purging procedure and waiting for printing the next layer. The experiment setting is shown in TABLE 3.1. Deposition temperature, denoted by T_0, is the temperature

Table 3.1: Experiment Setting

Item	Setting
Deposition temperature, T_0	260°C
Ambient temperature, T_{env}	24°C
Upper bound of surface temperature, T_u	150°C
Lower bound of surface temperature, T_l	120°C
The best surface temperature, T_b	135°C
Fixed layer time, t_c	260s
Deposition time per layer, t_d	15s
Speed	22.7cm/s
Bead Width	2.0cm
Thickness	0.5cm
Length of the final product	91.5cm
Width of the final product	20.3cm
Height of the final product	35.6cm

of the thermoplastic material when it is deposited. Ambient temperature, denoted by T_{env}, was around 24 °C, when the experiment was conducted. Based on extensive experiments by material scientists, the upper bound of print surface temperature T_u, the lower bound of print surface temperature T_l and the best surface temperature T_b are set to be 150°C, 120°C and 135°C, respectively. This means that the print surface temperature should be within upper and lower bounds and as close to the ideal print surface temperature as possible to achieve good product quality. The deposition time per layer, denoted by t_d, is the time that the nozzle spends adding a layer. The total amount of time for a layer, including the deposition time and waiting time, is referred to as the layer time, denoted by t_c.

The surface temperature of part is monitored by an infrared camera, and its thermal image is shown in Fig. 3.4a. The frame rate is around 1 frame per second.



(a) Thermal Image

(b) Final Product

Figure 3.4: Test Samples

Fig. 3.4b shows the final product of the thin wall.

3.1.3 Temperature Distribution and Cooling Dynamics

A large volume of thermoplastic material in high temperature is deposited in high speed. It causes the cooling process to be slow, when it is compared to desktop FFF. As it was discussed before, time is the primary factor that controls the current surface temperature. The large-scale of printing products implies that instant deposition of materials on the surface is not possible. Large values for deposition time per layer mean that there are considerable time gaps between positions on surface that are far away from each other. As a result, noticeable thermal gradients can be discerned between them. On the other hand, the geometry and dimension of the printing product and spatial correlation of positions within surface and between layers impact the cooling dynamics of different positions. Therefore, the temperature distribution and cooling characteristics vary from position to position and layer to layer. A more detailed analysis of temperature distribution will be discussed later in Chapter 4 when data extraction procedure from thermal images is outlined.

The analysis of surface temperature aims at developing a predictive model that can accurately describe the characteristics of surface temperature. Positions on surface have different cooling dynamic behaviors, and the cooling dynamic behavior of a position could also be changing mildly over time. A model describing temperature profile is required, when multiple positions are considered to determine the best timing to print the next layer.

3.1.4 Modeling of Temperature Profile

It is complex and unnecessary to formulate a single model of the cooling dynamics for the whole surface, and thus we start with a single position on the surface of the thin wall. Fig. 3.5 shows the observations of surface temperatures at this position for four layers. Each circle in the figure stands for an observation. The horizontal axis and the vertical axis provide the time when it is observed and the surface temperature at that time, respectively. The pattern about how the surface temperature of a single position decreases can be observed in this figure. The surface temperature is close to the deposition temperature, when a new layer is being printed. The cooling rate keeps decreasing, as the surface temperature is decreasing. Several outliers in Fig. 3.5 occur, when the position being monitored is blocked by the print head and not seen by the camera.

According to the paper Wang *et al.* (2019), the surface temperature of a position follows the differential equation shown below.

$$\frac{dT(t)}{dt} = \beta_0 + \beta_1(T(t) - T_{env}), \tag{3.1}$$

where $T(t)$ is the surface temperature at time t, and β_0 and β_1 are unknown parameters.

Remark 1. *Equation (3.1) is an empirical equation with similar form to the New-*

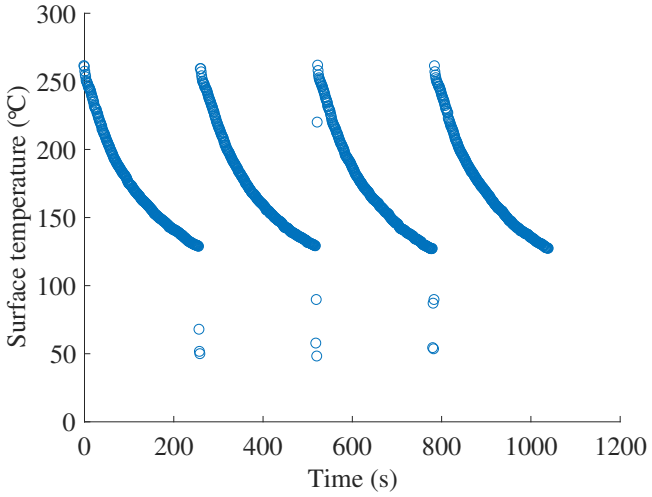


Figure 3.5: The Observations of Surface Temperature of a Single Position

ton's law of cooling. It was observed long time ago that in ideal conditions the rate of temperature change of a body is proportional to the difference between the body temperature and the ambient temperature. However, those ideal conditions are hardly satisfied in practice. Instead of directly using a law in physics, we adopted a statistical model motivated by the Newton's law of cooling, provided by Equation (3.1). We add a constant term β_0 to have a better fit. Parameters are estimated in a data-driven manner, instead of being derived from physical models. This model keeps a simple form and is shown to predict future surface temperature with high accuracy.

By solving Equation (3.1), the future surface temperature can be obtained as follows.

$$T(t) = T_{env} - \frac{\beta_0}{\beta_1} + \left(T_0 - T_{env} + \frac{\beta_0}{\beta_1}\right) e^{\beta_1 t}, \tag{3.2}$$

where t denotes the time since the last time thermoplastic material is added to this position.

Let $\hat{\beta}_0$ and $\hat{\beta}_1$ be the estimates of β_0 and β_1, respectively found by fitting the model from the data. The fitting procedure involves applying the Gauss-Newton iteration

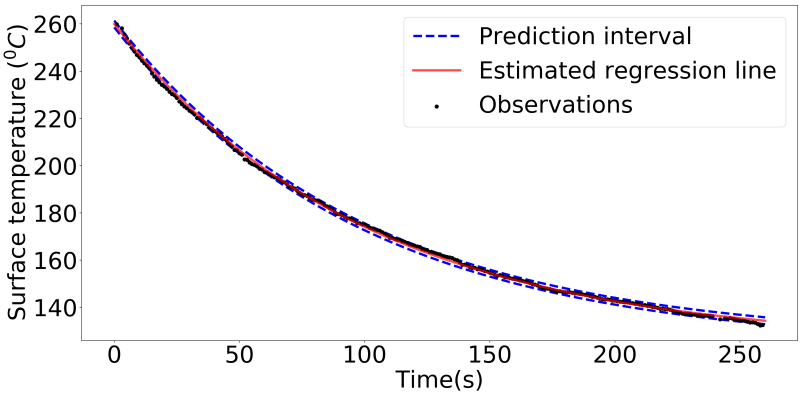


Figure 3.6: Regression Plot

method on the Taylor linearization of function $T(t)$ about β_0 and β_1 (Montgomery *et al.* (2012)). Let μ_i and σ_i, for $i = 0, 1$, denote the mean and standard deviation of $\hat{\beta}_i$, respectively. Given the estimated parameter $\hat{\boldsymbol{\beta}} = \left[\hat{\beta}_0, \hat{\beta}_1\right]^T$, the estimate of $T(t)$, denoted by $\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)$, is expressed as follows.

$$\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right) = T_{env} - \frac{\mu_0}{\mu_1} + \left(T_0 - T_{env} + \frac{\mu_0}{\mu_1}\right) e^{\mu_1 t}. \tag{3.3}$$

Denote $LB_{T(t)}$ and $UB_{T(t)}$ as the lower bound and upper bound of surface temperature for $(1-\alpha)$ one-sided prediction intervals of $T(t)$, respectively. Then, we have

$$\begin{aligned} P\left(T(t) > LB_{T(t)}\right) &= 1-\alpha, \\ P\left(T(t) < UB_{T(t)}\right) &= 1-\alpha, \end{aligned} \tag{3.4}$$

such that

$$\begin{aligned} LB_{T(t)} &= \hat{T}\left(t, \hat{\boldsymbol{\beta}}\right) - t_{\alpha, n-2} se\left(\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)\right), \\ UB_{T(t)} &= \hat{T}\left(t, \hat{\boldsymbol{\beta}}\right) + t_{\alpha, n-2} se\left(\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)\right), \end{aligned} \tag{3.5}$$

where $t_{\alpha,n-2}$ is the $100(1-\alpha)$ percentile of the t-distribution with $(n-2)$ degrees of freedom. The term $se\left(\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)\right)$ in the equations above represents the standard error of $\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)$, which can be found by the delta method as follows. First, the term

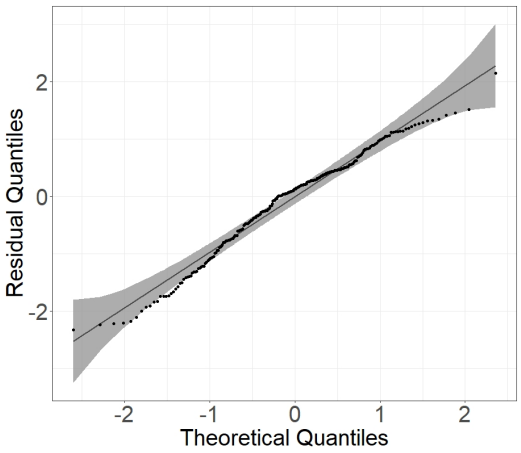


Figure 3.7: Residual Normal Quantile Plot

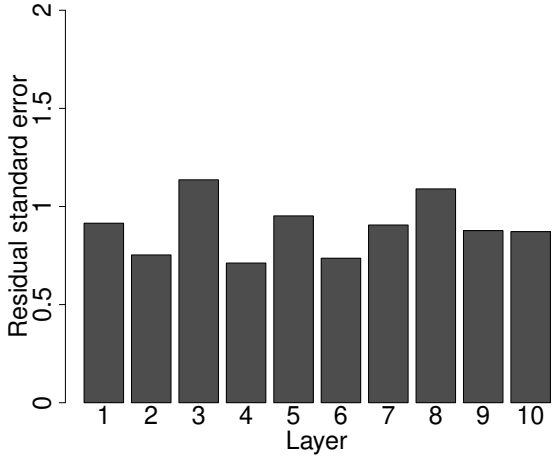


Figure 3.8: Bar Plot of Residual Standard Errors for Ten Layers

Table 3.2: Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob> |t|
β_0	1.00700	0.005942	169.4	$< 2 \times 10^{-16}$ *
β_1	-0.01003	0.00003	-262.7	$< 2 \times 10^{-16}$ *

* shows statistical significance.

$\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)$ is approximated using the Taylor linearization about the parameters:

$$\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right) \simeq \hat{T}\left(t, \boldsymbol{\beta}\right) + \nabla \hat{T}\left(t, \boldsymbol{\beta}\right)\left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right). \tag{3.6}$$

Taking the variance from both sides results in

$$Var\left(\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)\right) \simeq \nabla \hat{T}\left(t, \boldsymbol{\beta}\right) Var\left(\hat{\boldsymbol{\beta}}\right) \nabla \hat{T}\left(t, \boldsymbol{\beta}\right)^T, \tag{3.7}$$

where $Var\left(\hat{\boldsymbol{\beta}}\right)$ denotes the variance-covariance matrix of $\hat{\boldsymbol{\beta}}$. Finally, $se\left(\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)\right)$ can be simply found by taking the squared root of the $\left(Var\left(\hat{T}\left(t, \hat{\boldsymbol{\beta}}\right)\right) + \hat{\sigma}^2\right)$, where $\hat{\sigma}^2$ represents the estimate of the variance of error.

Fig. 3.6 illustrates the resulting fitted model shown by the red line along with observations shown by black dots. The blue ribbon around the fitted curve in Fig.

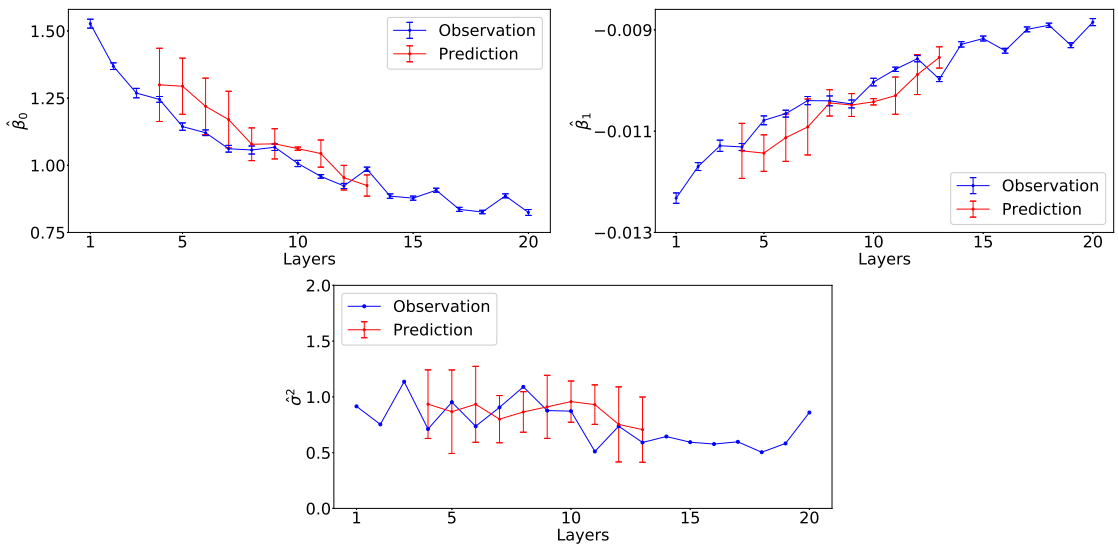


Figure 3.9: The Observations and Predictions of $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\sigma}^2$ for $d = 3$

3.6 illustrates the intersection of two 95% one-sided prediction intervals for $T(t)$. The black dots are all close to the red line. The prediction interval is narrow and covers most of the observations. It suggests that the regression model can describe the cooling process accurately. TABLE 3.2 shows a summary of the model fitting result. We have $\mu_0 = 1.00700$, $\mu_1 = -0.01003$, $\sigma_0 = 0.005942$, and $\sigma_1 = 0.00003$. The estimate of the variance of error $\hat{\sigma}^2$ given by Mean Square of Residuals (MS_{Res}) is equal to 0.76014. The normal quantile plot of residuals are illustrated in Fig. 3.7. If some residuals fall outside the confidence interval, one might suspect the non-normality of the residuals. Some departures from normality can be detected as some of the residuals fall outside the confidence band for normal distribution. Generally, statistical inference in non-linear regression is realized by large-sample or asymptotic theories. This suggests that they are approximately true even if the errors are non-normally distributed (Montgomery *et al.* (2012)).

An additional experiment is conducted, and the observations for ten layers are collected and compared with the predicted surface temperature. Fig. 3.8 is the

bar plot of the residual standard error for ten layers. The residual standard errors are all around 1°C, indicating that the model can fit and predict the future surface temperature accurately.

3.2 Adaptive Model with Bayesian Methods

The printing environment, such as humidity and airflow, could change during the printing process. It thus requires the temperature prediction model to be adaptive to capture the changing cooling dynamics. In this section, we introduce an adaptive model with Bayesian methods in forecasting β_0, β_1 and $\hat{\sigma}^2$. Let $\hat{\beta}_{i,n}$, for $i = 0, 1$, be the estimate of β_i in Equation (3.1) for the nth layer according to the observations of the nth layer. The estimate $\hat{\beta}_{i,n}$ approximately follows the Gaussian distribution. Denote by $\mu_{i,n}$ and $\sigma_{i,n}$ the mean and standard deviation of $\hat{\beta}_{i,n}$, respectively. The estimate $\hat{\beta}_{i,n}$ forms a time series with the index n. Let $\hat{\beta}_{i,N+\tau}(N)$ be the τ-step-ahead forecast made at the end of layer N. Assume the forecast $\hat{\beta}_{i,N+\tau}(N)$ is a random variable following the Gaussian distribution with mean $\mu_{i,N+\tau}(N)$ and standard deviation $\sigma_{i,N+\tau}(N)$.

The Gaussian process is used to conduct the forecasting. Specifically, the last d continued estimates of parameters β_0 and β_1 are taken as input to perform a τ-step-ahead forecast. Here, d refers to the number of previous layers that are considered in forecast, and it is determined according to autocorrelation of parameters. Assume $N \geq d$, and then the τ-step-ahead forecast $\hat{\beta}_{i,N+\tau}(N)$ is of interest and obtained through the historical data $\hat{\beta}_{i,n}$, for $n = N - d + 1, N - d + 2, \cdots, N$. Let

$$\hat{\boldsymbol{\beta}}_i = \begin{bmatrix} \hat{\beta}_{i,N-d+1} & \hat{\beta}_{i,N-d+2} & \cdots & \hat{\beta}_{i,N} \end{bmatrix}^T, \tag{3.8}$$

for $i = 0, 1$. Assume $\begin{bmatrix} \hat{\boldsymbol{\beta}}_i^T & \hat{\beta}_{i,N+\tau}(N) \end{bmatrix}^T$, for $i = 0, 1$, follows the multivariate Gaussian distribution. Specifically,

$$\begin{bmatrix} \hat{\boldsymbol{\beta}}_i \\ \hat{\beta}_{i,N+\tau}(N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{\boldsymbol{m}} \\ \hat{m} \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} + \sigma_n^2 I_{d+1} \right), \tag{3.9}$$

where σ_n^2 denotes noise variance, I_{d+1} is a $(d+1) \times (d+1)$ identity matrix, and

$$\hat{\boldsymbol{m}} = [\hat{m} \quad \hat{m} \quad \cdots \quad \hat{m}]^T, \tag{3.10}$$

$$\hat{m} = \frac{\sum_{j=N-d+1}^{N} \mu_{i,j}}{d}, \tag{3.11}$$

$$\sigma_n^2 = \frac{\sum_{j=N-d+1}^{N} \sigma_{i,j}^2}{d}. \tag{3.12}$$

Matrices K, K_* and K_{**} are represented as follows.

$$K = \begin{bmatrix} k(0) & k(1) & \cdots & k(d-1) \\ k(1) & k(0) & \cdots & k(d-2) \\ \vdots & \vdots & \ddots & \vdots \\ k(d-1) & k(d-2) & \cdots & k(0) \end{bmatrix}, \tag{3.13}$$

$$K_* = \begin{bmatrix} k(\tau+d-1) & k(\tau+d-2) & \cdots & k(\tau) \end{bmatrix}^T, \tag{3.14}$$

and

$$K_{**} = k(0), \tag{3.15}$$

where $k(x)$ is a squared exponential kernel function. Specifically,

$$k(x) = \sigma_f^2 \exp\left(-\frac{x^2}{2l^2}\right), \tag{3.16}$$

where l and σ_f^2 are the hyperparameters that control the smoothness and vertical variation of the approximation to the historical data, and x is the Euclidean distance between vectors of features. To predict β_0 and β_1, x has only one dimension, representing the number of layers. The posterior is given by

$$\hat{\beta}_{i,N+\tau}(N) \mid \hat{\boldsymbol{\beta}}_i \sim \mathcal{N}\left(\mu_{i,N+\tau}(N), \sigma_{i,N+\tau}^2(N)\right), \tag{3.17}$$

where

$$\mu_{i,N+\tau}(N) = K_* \left(K + \sigma_n^2 I_d\right)^{-1} \left(\hat{\boldsymbol{\beta}}_i - \hat{\boldsymbol{m}}\right) + \hat{m}, \tag{3.18}$$

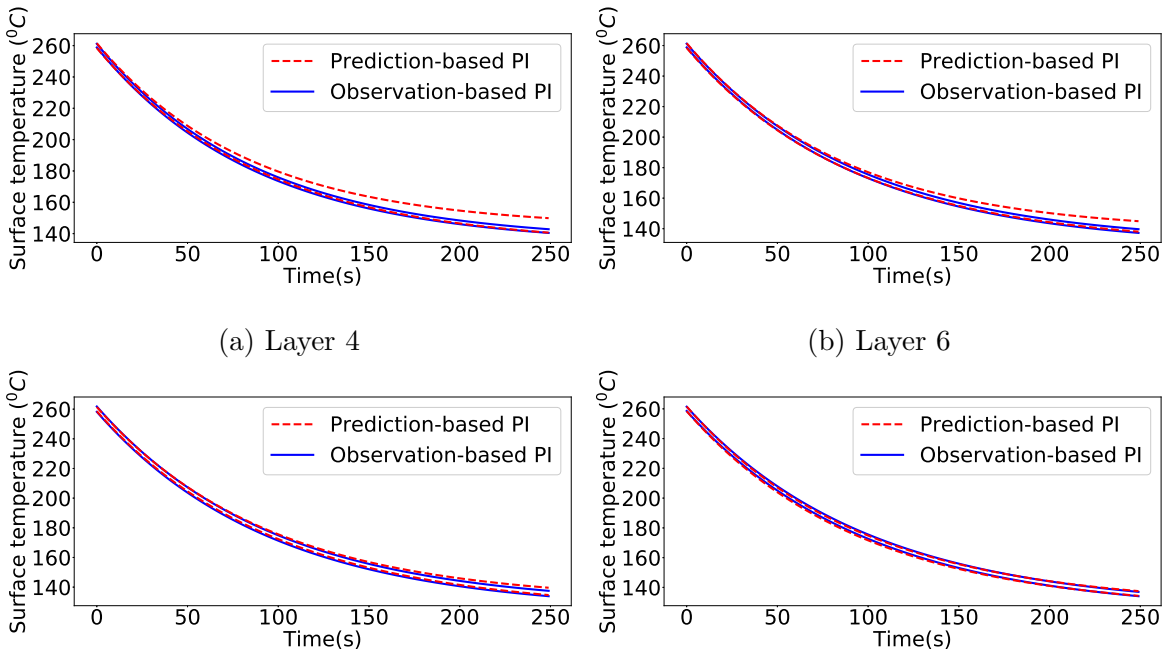
$$\sigma^2_{i,N+\tau}(N) = K_{**} + \sigma_n^2 - K_* \left(K + \sigma_n^2 I_d\right)^{-1} K_*^T. \tag{3.19}$$

Following the similar way, the variance of error, $\hat{\sigma}^2$, is predicted. Let

$$\hat{\boldsymbol{V}} = \begin{bmatrix} \hat{\sigma}^2_{N-d+1} & \hat{\sigma}^2_{N-d+2} & \cdots & \hat{\sigma}^2_N \end{bmatrix}^T, \tag{3.20}$$

where $\hat{\sigma}_i^2$, for $i = N-d+1, N-d+2, \cdots, N$, are observed variances of error. Denote by $\hat{\sigma}^2_{N+\tau}(N)$ the τ-step-ahead forecast of $\hat{\sigma}^2$. Assume $\begin{bmatrix} \hat{\boldsymbol{V}}^T & \hat{\sigma}^2_{N+\tau}(N) \end{bmatrix}^T$ follows the multivariate Gaussian distribution. The Gaussian process is then used to derive $\hat{\sigma}^2_{N+\tau}(N) \mid \hat{\boldsymbol{V}}$, following the same way that predicts $\hat{\beta}_{i,N+\tau}(N)$. In addition to layer number, the prediction of $\hat{\sigma}^2$ also depends on the estimated parameters of β_0 and β_1. Therefore, $\hat{\boldsymbol{\beta}}_i$ are included in the set of features to calculate distance x shown in Equation (3.16) to predict the $\hat{\sigma}^2$.

Fig. 3.9 shows a case where parameter estimates $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\sigma}_2$ are changing over time due to the change of printing environment. Three plots in the figure give the trend of $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\sigma}_2$, respectively. In each plot, the observed means and confidence intervals are presented by blue color, while the one-step-ahead predicted means and confidence intervals are presented by red color. The observed means and confidence intervals of those parameters suggest that, due to the change of the environment, β_0 keeps decreasing, β_1 keeps increasing, and σ^2 is fluctuating in a certain range. A negative correlation of β_0 and β_1 is observed, and it is estimated to be around minus one. The correlation observed from the data will later be used to predict future surface temperature. The Gaussian process is applied to capture the change of parameters and do smoothing for the time series. The optimal values of hyperparameters of the Gaussian process are found by maximizing the marginal log-likelihood of historical



(a) Layer 4

(b) Layer 6

(c) Layer 8

(d) Layer 10

Figure 3.10: Comparison of Predicted Intervals for Observations and Predictions for Different Layers

data. d is set to be three, and thus last three layers are used to make a one-step-ahead prediction. The predicted values of $\hat{\beta}_0$ and $\hat{\beta}_1$ for each layer can capture the trend of observed values of $\hat{\beta}_0$ and $\hat{\beta}_1$ well. There is no obvious increasing or decreasing trend for $\hat{\sigma}^2$, and the predicted value of $\hat{\sigma}^2$ can smooth the time series.

The prediction of $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\sigma}^2$ are applied to Equation (3.5) to find the prediction interval of future surface temperature, and the result is shown in Fig. 3.10. Four plots in Fig. 3.10 are presenting the result for layer 4, layer 6, layer 8 and layer 10, respectively. In Fig. 3.10a, two blue solid lines indicate the upper bound and lower bound of surface temperature of layer 4 based on the observations of layer 4, respectively. In other words, $\hat{\beta}_{0,4}$, $\hat{\beta}_{1,4}$ and $\hat{\sigma}_4^2$ are estimated and used to do predictions. It reflects the real temperature dynamics for this layer. The two red dashed lines in Fig.

3.10a show the predicted upper bound and lower bound, respectively. Specifically, we let N to be three. $\hat{\boldsymbol{\beta}}_i = \begin{bmatrix} \hat{\beta}_{i,1} & \hat{\beta}_{i,2} & \hat{\beta}_{i,3} \end{bmatrix}^T$, for $i = 1, 2$, and $\hat{\boldsymbol{V}} = [\hat{\sigma}_1^2 \quad \hat{\sigma}_2^2 \quad \hat{\sigma}_3^2]^T$, as historical data is used to predict $\hat{\beta}_{i,4}(3)$, for $i = 0, 1$, and $\hat{\sigma}_4(3)$ through the Gaussian process, which then leads to the predicted upper bound and lower bound of surface temperature. In a similar way, we draw upper bound and lower bound for three other layers shown in Fig. 3.10b–3.10d, respectively. It can be observed that the prediction intervals based on parameter predictions for layer 4 and layer 6 are a little wider than the prediction intervals based on observations. Nonetheless, they still maintain a high accuracy following the change of printing environment. On the other hand, the prediction-based prediction intervals for layer 8 and layer 10 are much more accurate as they quite overlap with observation-based prediction intervals.

3.3 Real-time Layer Time Control

3.3.1 Problem Description

Large-scale additive manufacturing is subject to constraints on print surface temperature, which can be neither too high nor too low. A new layer is supposed to be printed on a surface with the surface temperature in the range from T_l to T_u and close to T_b. Currently, a commonly used method is to fix the layer time. The fixed layer time is predetermined based on experiment or experience. However, the cooling dynamics of a single position on the surface could change over time due to the change of printing environment. Therefore, any fixed layer time cannot guarantee to be optimal. A dynamic approach is required to perform real-time layer time control.

The dynamics of surface temperature at different positions are different, and it is represented by different values of parameters in Equation (3.2). We observed that the surface temperature decreases slowly at a corner and fast at a long edge. The best time

to add a layer should be obtained by considering multiple positions comprehensively based on the prediction of their surface temperature subject to other constraints such as quality restrictions. In this section, we introduce a real-time layer time control method. More than one position is monitored and considered in an optimization model to improve both the quality and efficiency.

3.3.2 Model for Layer Time Control

Denote by M the number of positions being monitored. The temperature change of the kth position, for $k = 1, 2, \cdots, M$, is modeled by Equation (3.2) with parameters β_0^k and β_1^k, which are obtained by one-step-ahead forecasting introduced in Section 3.2. For the predicted surface temperature at the kth position $\hat{T}^k(t)$, the time t starts with 0 when thermoplastic material is deposited at the kth position. Since the print speed is fixed, each position shares the same layer time. The optimization model with decision variable t is built as follows.

$$\min f(t) = \sum_{k=1}^{M} \omega_k \left(\hat{T}^k(t) - T_b\right)^2 + \omega_0 t \tag{3.21}$$

s.t.

$$P\left(\hat{T}^j(t) \geq T_l\right) \geq 1 - \alpha, \text{ for } j = 1, \cdots, M, \tag{3.22}$$

$$P\left(\hat{T}^j(t) \leq T_u\right) \geq 1 - \alpha, \text{ for } j = 1, \cdots, M. \tag{3.23}$$

The objective function, given by Equation (3.21), is a weighted summation with weight for each term denoted by ω_i, for $i = 0, 1, \cdots, M$. The term $\sum_{k=1}^{M} \omega_k \left(\hat{T}^k(t) - T_b\right)^2$ measures how much each position has print surface temperature close to the best temperature T_b. The term $\omega_0 t$ is the layer time multiplied with a coefficient. Small values of all those terms are pursued, and it is achieved by minimizing the objective function. The weight ω_i, for $i = 0, 1, \cdots, M$, which shows how much each term is valued, is determined based on domain knowledge. Inequation (3.22) and Inequation

(3.23) are chance constraints. Each inequation, represented by Inequation (3.22), is to guarantee that the print surface temperature for a single position is larger than the lower bound with probability $(1-\alpha)$. Similarly, each position has print surface temperature smaller than its upper bound with probability $(1-\alpha)$, according to Inequation (3.23). The chance constraints can be reformulated as follows.

$$LB_{T(t)} \geq T_l, \text{ for } j = 1, \cdots, M, \tag{3.24}$$

$$UB_{T(t)} \leq T_u, \text{ for } j = 1, \cdots, M. \tag{3.25}$$

Each constraint given by Inequation (3.24) or Inequation (3.25) stands for an interval. The feasible region is the intersection of all such intervals, and it is a closed interval, represented by a lower bound t_l and an upper bound t_u.

Convexity of an optimization model guarantees that the optimal solution can be derived. A proposition for the model's convexity is provided as follows.

Proposition 1. *The optimization model described by Equation (3.21), Inequation (3.22) and Inequation (3.23) is convex, if*

$$t_u \leq \min\left\{t \mid \hat{T}^k(t) \geq \frac{T_{env}}{2} - \frac{\beta_0^k}{2\beta_1^k} + \frac{T_b}{2}, k = 1, 2, \cdots, M\right\}. \tag{3.26}$$

Proof. Define function $f_k(t)$, for $k = 1, 2, \cdots, M$, as follows.

$$f_k(t) = \left(\hat{T}^k(t) - T_b\right)^2. \tag{3.27}$$

The second order derivative of $f_k(t)$ can be expressed as follows.

$$\frac{df_k^2(t)}{dt^2} = 2c_k\left(2\left(T_0 - T_{env} + \frac{\beta_0^k}{\beta_1^k}\right)e^{\beta_1^k t} + \left(T_{env} - T_b - \frac{\beta_0^k}{\beta_1^k}\right)\right), \tag{3.28}$$

where c_k is a positive number given below.

$$c_k = \left(\beta_1^k\right)^2\left(T_0 - T_{env} + \frac{\beta_0^k}{\beta_1^k}\right)e^{\beta_1^k t}. \tag{3.29}$$

By Equation (3.3), the second order derivative in Equation (3.28) can be reorganized as follows.

$$\frac{df_k^2(t)}{dt^2} = 4c_k\left(\hat{T}^k(t) - \frac{T_{env}}{2} + \frac{\beta_0^k}{2\beta_1^k} - \frac{T_b}{2}\right) \tag{3.30}$$

The condition provided by Inequation (3.26) implies $df_k^2(t)/dt^2 \geq 0$, for $k=1,2,\cdots,M$. Thus, function $f_k(t)$ is convex. Function $f(t)$ can be expressed as follows.

$$f(t) = \sum_{k=1}^{M} \omega_k f_k(t) + \omega_0 t, \tag{3.31}$$

which is also convex. The feasible region is convex, so the optimization model is convex. □

The condition provided by Inequation (3.26) is usually satisfied in practice, when sample size is large enough to estimate $\hat{T}^k(t)$. Therefore, the optimization model can be solved by a method for convex optimization problem that can handle nonlinear objective function and constraints. For this purpose, the Sequential Quadratic Programming (SQP) method is used to find the optimal solution. SQP is an iterative method using the Taylor's second order approximations of the objective function with the first order Taylor's approximations of the constraints. Let $b(t) \geq 0$ represent constraints provided in Inequation (3.24) and Inequation (3.25). Define the Lagrangian of the problem as $L(t,\lambda) = f(t) - \lambda^T b(t)$, where vector λ is a Lagrange multiplier. Let t^k and λ^k be the solution and the Lagrange multiplier at iterate k, respectively. Then, a solution to the following quadratic programming subproblem defines an appropriate search direction D^k:

$$\min_{D^k} \nabla f\left(t^k\right)^T D^k + \frac{1}{2}\left(D^k\right)^T \nabla_{tt}^2 L\left(t^k, \lambda^k\right) D^k \tag{3.32}$$

s.t.

$$\nabla b\left(t^k\right)^T D^k + b\left(t^k\right) \geq 0. \tag{3.33}$$

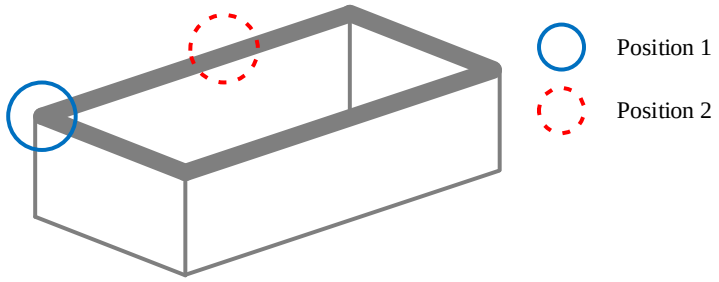


Figure 3.11: Positions Monitored in the Case Study

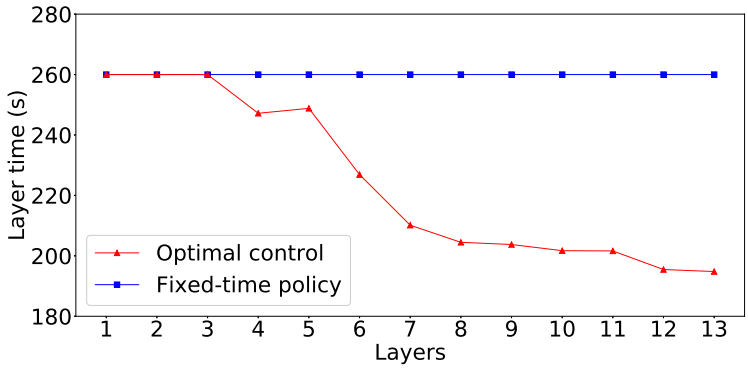


Figure 3.12: Comparison of Layer Time from the Optimal Control of Layer Time and the Fixed Layer Time Strategy

The values of t and λ for iteration $(k+1)$ can be obtained by $t^{k+1} = t^k + D^k$ and the corresponding Lagrange multiplier of the quadratic sub-problem, respectively. The initial values of t and λ, denoted by t^0 and λ^0, are determined by choosing a random value, and the optimal values are found by repeating the process until convergence.

3.4 Case Study

This case study section is dedicated to demonstrating the effectiveness of the developed real-time layer time control method. In this case study, only two positions are monitored, with $M = 2$. In practice, it can be easily extended to a model moni-

toring more than two positions. The locations of the two positions being monitored are shown in Fig. 3.11. Position 1 is at the corner and always has a higher surface temperature, while position 2 is in the middle of an edge and usually has a lower surface temperature. The coefficients ω_i, for $i = 0, 1, 2$, are set to be one.

The data of surface temperature from 13 layers are collected in this case study. Two control methods are applied to that data. One is the real-time layer time control method proposed in this paper, and this method gives the optimal control for layer time. The other is the traditionally used method with a fixed layer time policy, where layer time is fixed to be 260s. The two control methods are compared with respect to two performance measures, layer time and print surface temperature.

Fig. 3.12 shows the comparison of layer time from two control methods. The horizontal axis gives the layer numbers, and the vertical axis represents the layer time. The blue color stands for the traditionally used method, and it is a straight line since the layer time is fixed. The red color gives the layer time of each layer when the optimal layer time control is applied. In this case study, d is set to be three, and thus the two lines in blue color and red color overlap in the first three layers. It starts with the fourth layer that the optimal control of layer time starts reducing layer time, compared to the fixed-time policy. As the printing environment is changing, the cooling rate is increasing. The required layer time is decreasing. The optimal control of layer time can track the change and achieve a better performance. For the last several layers, the layer time is reduced by about 23%.

Fig. 3.13 presents the print surface temperature from two control methods. The horizontal axis shows the layer numbers, and the vertical axis gives the temperature. The straight dashed line in black color marks the best surface temperature, which is 135°C. The two lines in blue color show the print surface temperatures of position 1 and position 2 with fixed-time policy, respectively. Due to the change of printing

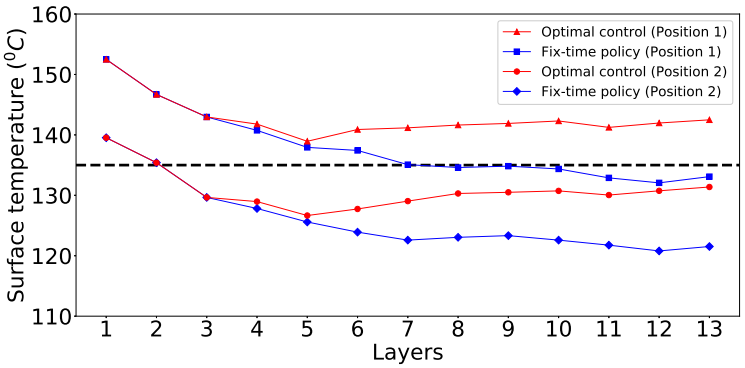


Figure 3.13: Comparison of Print Surface Temperature from the Optimal Control of Layer Time and the Traditionally Used Method with a Fixed Layer Time

environment, the cooling rate is increasing, and thus the two blue lines are decreasing. As the layer time is fixed, it is not capable of adjusting itself to the change of printing environment. In contrast, the lines in red color, representing the optimal layer time control, stay around the best surface temperature. The overlap can be seen for the first three layers, as d is set to be 3. After the third layer, the print surface temperature of position 1 is a little higher than the best surface temperature, while the print surface temperature of position 2 is a little lower than the best surface temperature. A better quality is achieved by comprehensively considering print surface temperature of the two positions.

The values of the objective function of two control methods are presented in Fig. 3.14. The red color and blue color represent the optimal control and the fixed-time policy, respectively. There is an overlap for the first three layers. After the third layer, the optimal control of the layer time results in a smaller value of objective function. Besides, the improvement is getting larger, as the layer number is increasing.

Coefficient ω_i, for $i = 0, 1, 2$, is set according to the trade-off of layer time and print surface temperature. To study the effect of coefficient ω_i, we fix ω_1 and ω_2 and tune ω_0, and the resulting performance measures with different ω_0 are presented in

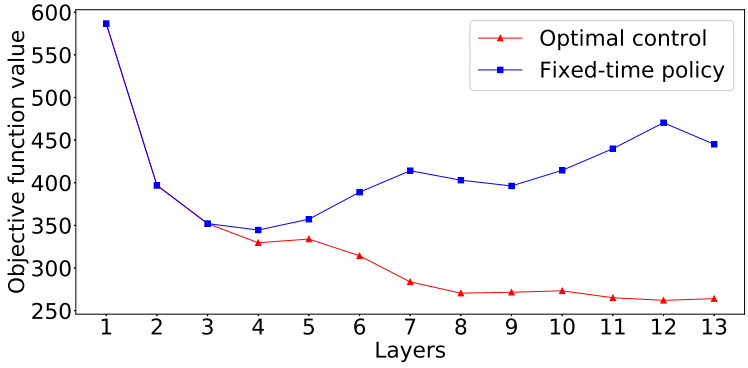


Figure 3.14: Values of Objective Function from the Optimal Control of Layer Time and the Traditionally Used Method with a Fixed Layer Time

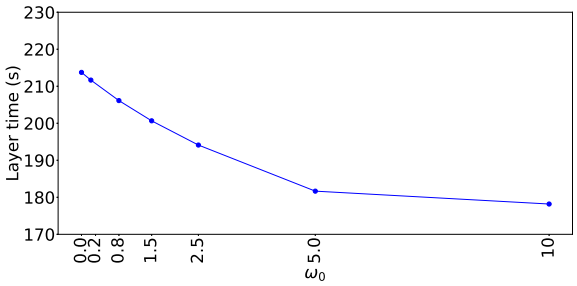


Figure 3.15: The Effect of ω_0 on Layer Time

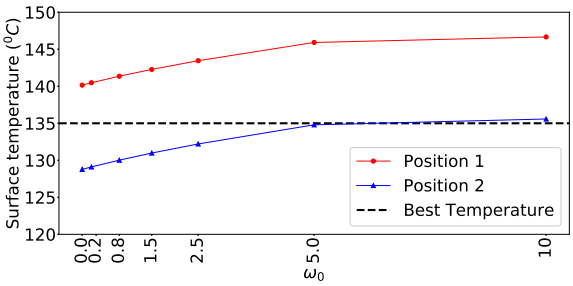


Figure 3.16: The Effect of ω_0 on Print Surface Temperature

Fig. 3.15 and Fig. 3.16. Fig. 3.15 presents the optimal layer time with different ω_0 for layer 8. As ω_0 is increasing from 0 to 10, the layer time is decreasing and finally converges. The parameter ω_0 is the weight for layer time, and thus a larger ω_0 can lead to a smaller layer time. However, due to the constraints of the optimization model, the print surface temperature must be smaller than its upper bound. Thus, the plot shown in Fig. 3.15 will converge. Fig. 3.16 shows the print surface temperature under different ω_0. When ω_0 is equal to 0, the print surface temperature is the only measure considered in the objective function. The print surface temperatures of position 1 and position 2 are on the two sides of the best surface temperature with the same

distance to the best surface temperature. When ω_0 is increasing, more weight is put on the layer time. The two lines in Fig. 3.16 are increasing and converging due to the constraints. Because of the chance constraints, the print surface temperature of position 1 converges to a temperature smaller than the upper bound 150 °C.

Chapter 4

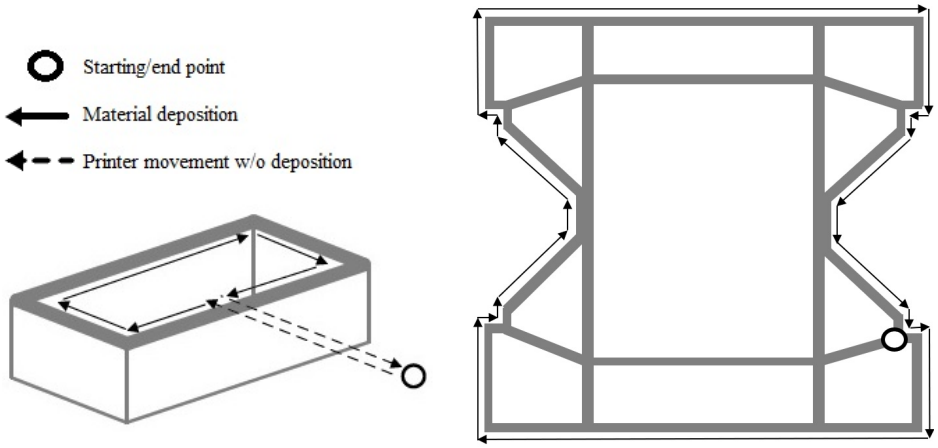
DYNAMIC MATERIAL FLOW AND MACHINE MOTION CONTROL FOR LARGE-SCALE ADDITIVE MANUFACTURING USING THERMAL IMAGES

4.1 Surface Temperature Prediction Model

4.1.1 System Setup

In this study, two separate cases of components, two thin wall tests and a lower chassis of a real car are printed. A carbon fiber filled polycarbonate (CF/PC) compound named "SABIC DC004" is used as the thermoplastic material. For the scope of this study, we just focus on one of the wall tests and also lower edge of the chassis outer border, as will be shown in next section. Each layer on the wall forms a rectangle, and each layer on the chassis forms a complex geometry as is shown in Figure 4.1. For both cases, the printer head starts from the starting point shown by tiny circles in the figures. It then deposits single-bead-wide molten thermoplastic materials on the surface as it traverses along the edges in a clockwise direction. The process is completed once the printer head return to the starting point.

For the walls, the printer head then proceeds to the next wall and repeats the same process. Once the second wall is over, it will return to a purge station -shown by dashed lines in Figure 4.1- and wait there until the surface is cool enough for printing the next layer. However, in case of chassis the entire process consists of printing three segments subsequently: the outer border, the inner box and the edges that connect the box and border together. Note that there is no purge and pause here and the printer keeps printing. That is because the the locations at the beginning of the path are already too cool when the printer head approaches the end of the path.



(a) Thin Wall Test (b) Lower Chassis

Figure 4.1: Printed Components and Their Tool Path

As it was mentioned earlier the focus here is on the outer border so the path for the other two segments are not shown in the figure.

The common experiment settings for both cases are shown in Table 4.1. According to the material characteristics and through extensive experiments, it has been determined that the lower bound, ideal, and the upper bound of the printing surface temperature are $120°C$, $135°C$ and $150°C$, respectively. Therefore, the feasible region for temperature is $[120, 150]$, and the closer it is to $135°C$, the higher the quality is. Table 4.2 provides the dimensions, current fixed layer time denoted by t_c, and deposition time per layer denoted by t_d for each case separately.

4.1.2 Data Extraction from Thermal Images

The schematic of the system setup for capturing the thermal images from car chassis is illustrated by Figure 4.2. Example RGB images taken by the camera for both wall tests and lower chassis along with their colormaps are shown in Figure 6.8. The camera was emissivity corrected in software to account for the slightly lower emissiv-

Table 4.1: Common Experiment Setting

Parameter	Setting
Deposition temperature, T_{dep}	$250°C$
Ambient temperature, T_{env}	$24°C$
Upper bound of surface temperature, T_u	$150°C$
Lower bound of surface temperature, T_l	$120°C$
The ideal surface temperature, T_b	$135°C$
Printer head speed	535 in/min
Printer head maximum Speed, v_u	850 in/min
Printer head minimum Speed, v_l	320 in/min
Printer head maximum acceleration/deceleration, a	5.35 in/s^2
Bead Width	0.8 in
Thickness	0.2 in

Table 4.2: Experiment Setting for the Case Studies

Parameter	case study	
	Wall	Chassis
Length of the final product	36 in	154 in
Width of the final product	8 in	80 in
Height of the final product	14 in	98 in
Layer time, t_c	$250s$	$126s$
Deposition time per layer, t_d	$15s$	$126s$

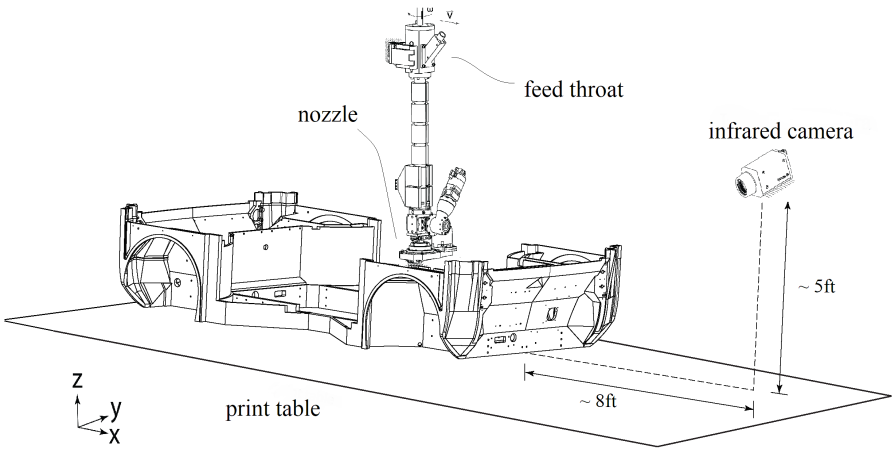
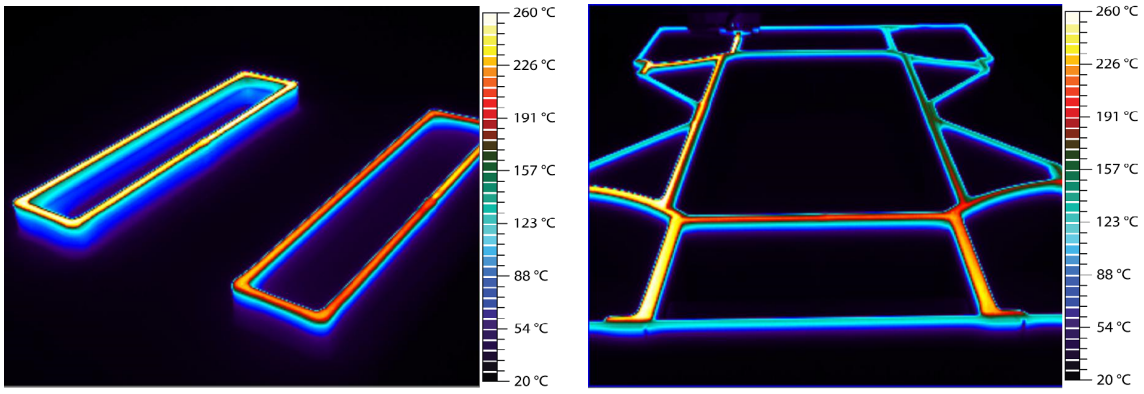


Figure 4.2: System Setup to Capture the Thermal Images from Car Chassis

ity of ~ 0.9 of the CF/PC material. This number was verified using a thermocouple against experiments performed by ORNL using a black body source (Dinwiddie *et al.* (2014)). It should be noted that accuracy of thermal readings might get affected by the distance and angle of locations on the surface relative to the position of camera's thermal sensor. This could potentially lead to undemeasuring surface temperature of distant and off-angle locations. Ideally, installing more cameras and fusing multiple data readings can resolve this issue. For the sake of both simplicity and accuracy in real-time analysis, all the digital image processing tasks are performed on thermal images containing actual temperature values rather than RGB intensities. Given the wall, the region of interest (ROI) consists of congruent rectangles with their upper right corners being truncated out of the frame. Whereas for the lower chassis, it consists of congruent polygons forming the outer border with their left and right lower corners being truncated out of the frame.

For the convenience of future geometric calculations, the first layer in the first image will be perspectively transformed and warped to form a new polygon such that the width and length of the bounding box around ROI become parallel to the



(a) Thin Wall Test (b) Lower Chassis

Figure 4.3: Thermal Images Taken by FLIR^{TM} Camera

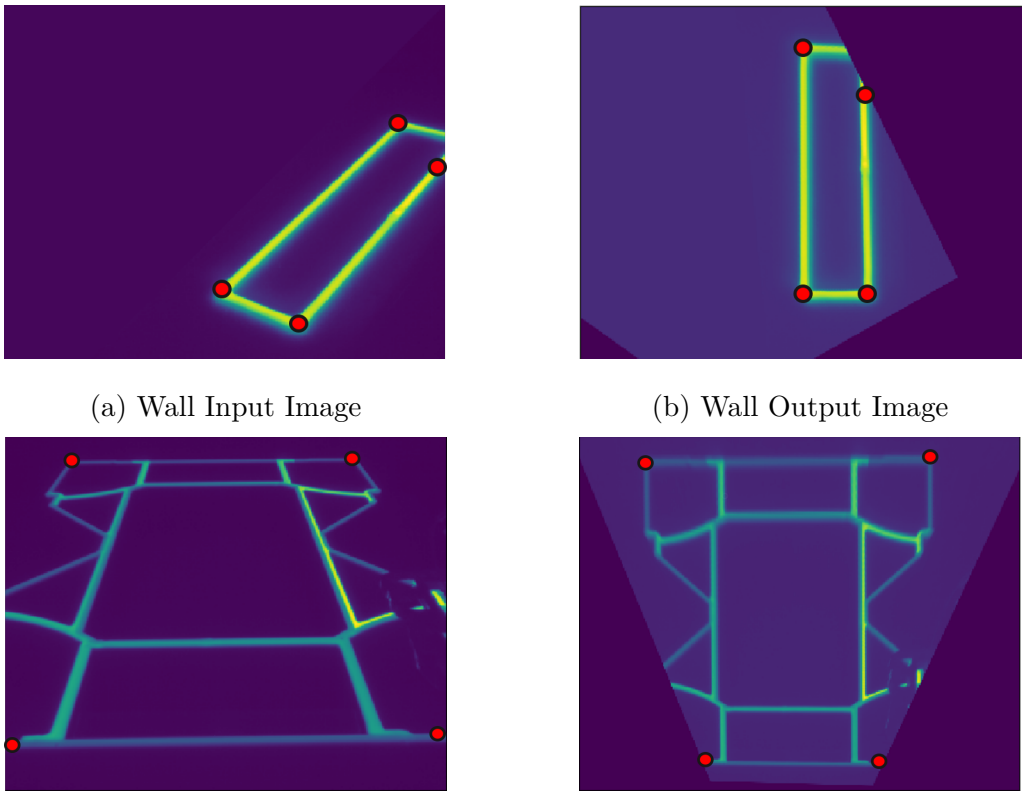
X and Y axis, respectively. In a spatial transformation, each point (x, y) of the reference image is mapped to a point $(u = f_1(x, y), v = f_2(x, y))$ in a new coordinate system of the source image. Specifically, a perspective transform is a type of spatial transform that preserve the collinearity but changes the ratio of distances. Find a set of points $\{R_0, R_1, \ldots, R_{n-1}\}$ in the reference image, and a set of corresponding points $\{S_0, S_1, \ldots, S_{n-1}\}$ in the source image. The matrix representation of perspective transformation from reference to source image with transformation matrix $M \in \mathbb{R}^{3\times3}$, can be expressed as $S = MR$, where the columns in $R \in \mathbb{R}^{3\times n}$ and $S \in \mathbb{R}^{3\times n}$ are the homogeneous coordinate representation of each point:

$$R = \begin{bmatrix} x_0 & \ldots & x_{n-1} \\ y_0 & \ldots & y_{n-1} \\ 1 & \ldots & 1 \end{bmatrix}, \quad S = \begin{bmatrix} u_0 & \ldots & u_{n-1} \\ v_0 & \ldots & v_{n-1} \\ 1 & \ldots & 1 \end{bmatrix}. \tag{4.1}$$

The transformation matrix M can be solved by minimizing the mean squared error as follows:

$$M = SR^T(RR^T)^{-1}. \tag{4.2}$$

The reference images and the transformed images found by warping the reference



(a) Wall Input Image

(b) Wall Output Image

(c) Chassis Input Image

(d) Chassis Output Image

Figure 4.4: Thermal Image Perspective Transformation and Warping

images are shown in Figure 4.4 for both the wall and chassis. The four matching points used to derive the the perspective transformation matrix for each case are shown by tiny red circles in the figures. Once the first round of process is done and the transformation matrix is estimated, it can be used to transform the images from next layers. Nevertheless, as more layers of materials get stacked upon older ones, the location of ROI also shifts upward. Thus, another step is required to register the source images from next layers back to the reference image in real-time without any supervision.

For this purpose, the next step toward data extraction concerns with the image segmentation by thresholding to extract the foreground, here the ROI for the recent

layer, from the background. Image thresholding binarizes the image by assigning one to the pixels who have greater value in intensity than a threshold value (class 1), and assigning the rest to zero (class 0). In this study, we propose Otsu thresholding method (Otsu (1979)) to automatically extract the threshold value. The threshold value in Otsu's method is obtained by finding the best value that maximizes the homogeneity inside each class or equivalently minimizes the within-class variances. After mapping the temperature values on the image to integer values, the optimization problem to seek the threshold value τ by minimizing the sum of weighted within-class variances is defined as follows:

$$\min_{\tau} f(\tau) = w_0(\tau)\sigma_0^2(\tau) + w_1(\tau)\sigma_1^2(\tau), \quad \tau \in [0, T_{dep}], \quad \tau \in \mathbb{Z} \tag{4.3}$$

, where $w_0(\tau) = \sum_{i=1}^{\tau} N(i)$ and $w_1(\tau) = \sum_{i=\tau+1}^{T_{dep}} N(i)$ are the weights of class 0 and 1, respectively, $N(i)$ denotes the number of pixels with corresponding temperature equal to i, and T_{dep} is the deposition temperature as shown in Table 4.1. Denote the mean of pixels inside class 0 and 1 by $\mu_0(\tau)$ and $\mu_1(\tau)$, respectively, and let $\sigma_0^2(\tau)$ and $\sigma_1^2(\tau)$ be the variance of pixels inside each class. Then, we have:

$$\mu_0(\tau) = \sum_{i=1}^{\tau} \frac{iN(i)}{w_0(\tau)}, \quad \mu_1(\tau) = \sum_{i=\tau+1}^{T_{dep}} \frac{iN(i)}{w_1(\tau)}, \tag{4.4}$$

$$\sigma_0^2(\tau) = \sum_{i=1}^{\tau} \frac{i^2N(i)}{w_0(\tau)} - \mu_0(\tau)^2, \quad \sigma_1^2(\tau) = \sum_{i=\tau+1}^{T_{dep}} \frac{i^2N(i)}{w_1(\tau)} - \mu_1(\tau)^2. \tag{4.5}$$

Line search methods such as golden section search can be used to minimize the optimization model given by Equation (4.3).

Given the binarized images, the corners of bounding box around the ROI can be extracted. Let the set $P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ contain the location of retrieved pixels belonging to class 1. Then, the corners of bounding box around ROI denoted by C_j for $j = 1, 2, 3, 4$ can be found as follows:

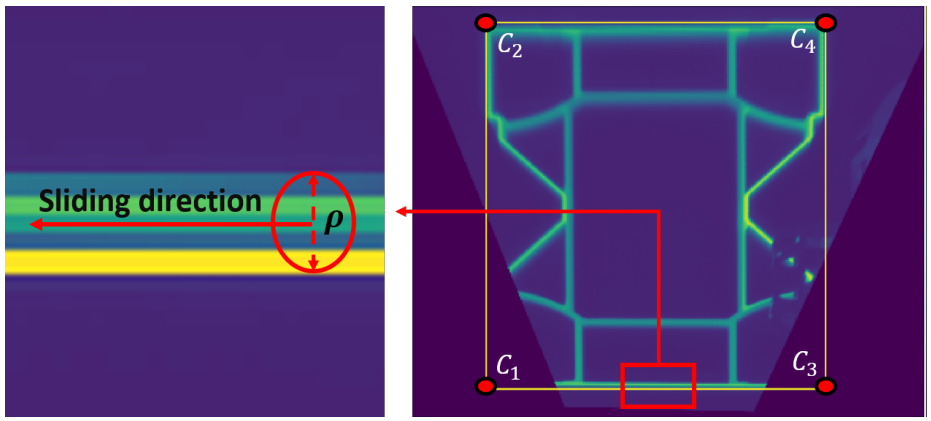


Figure 4.5: Sliding Windows Profile Scan

$$C_1 = (\min_{1\leq i\leq n}\{x \mid (x_i, y_i) \in P\}, \min_{1\leq i\leq n}\{y \mid (x_i, y_i) \in P\}), \tag{4.6}$$

$$C_2 = (\min_{1\leq i\leq n}\{x \mid (x_i, y_i) \in P\}, \max_{1\leq i\leq n}\{y \mid (x_i, y_i) \in P\}), \tag{4.7}$$

$$C_3 = (\max_{1\leq i\leq n}\{x \mid (x_i, y_i) \in P\}, \min_{1\leq i\leq n}\{y \mid (x_i, y_i) \in P\}), \tag{4.8}$$

$$C_4 = (\max_{1\leq i\leq n}\{x \mid (x_i, y_i) \in P\}, \max_{1\leq i\leq n}\{y \mid (x_i, y_i) \in P\}). \tag{4.9}$$

Finally, A methodology is required to obtain the temperature in each location. It has to be robust as there is a high variability of temperature throughout the surface. In addition, the spatial correlation between locations should be taken into account. We propose using sliding circular window profile scan. The window will be sliding in the same direction as the extruder path by an increment of one pixel and take the maximum temperature inside each circle. The diameter of the circular window profile scan denoted by ρ is usually set to be equal to the bead width in the images. A schematic of the method applied on the chassis image is shown in Figure 4.5. The corners of bounding box around ROI are annotated by the tiny red circles.

Following the proposed methods, two sets of raw temperature data for one layer of wall and chassis have been extracted, and their corresponding surface plots are shown in Figure 4.6 and Figure 4.7. The X-axis represents the location of points according

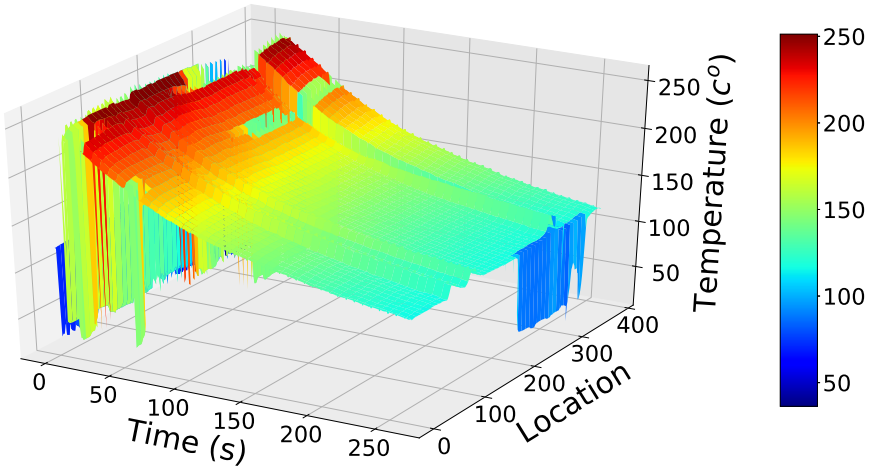


Figure 4.6: Surface Plot of Raw Temperature Data for Wall

to the order of which they have been printed. The Y-axis and Z-axis represent the layer time and surface temperature, respectively. The color bar on the right side demonstrates the relationship between colors on the plot and their corresponding temperature values. In contrast to PBF and PVP processes, image processing and data cleansing pose a challenge to LSAM process. This is because the printer head is quite large, and therefore it obstructs the infrared camera when it traverses to extrude the materials upon the surface. This will leave some cold marks on the images and introduce the thermal surface with multiple instances of discontinuity and interruption. The step by step cleansing procedure to handle the outliers will be discussed in the next subsection.

4.1.3 Modeling of Temperature Profile

The heat dissipation flow on the surface of complex geometries is commonly estimated by computational thermal models such as finite element analysis and graph theoretic approaches (Yavari *et al.* (2019), Cole *et al.* (2020)). In order to investigate the cooling dynamics of the entire temperature field, we deal with each location as an

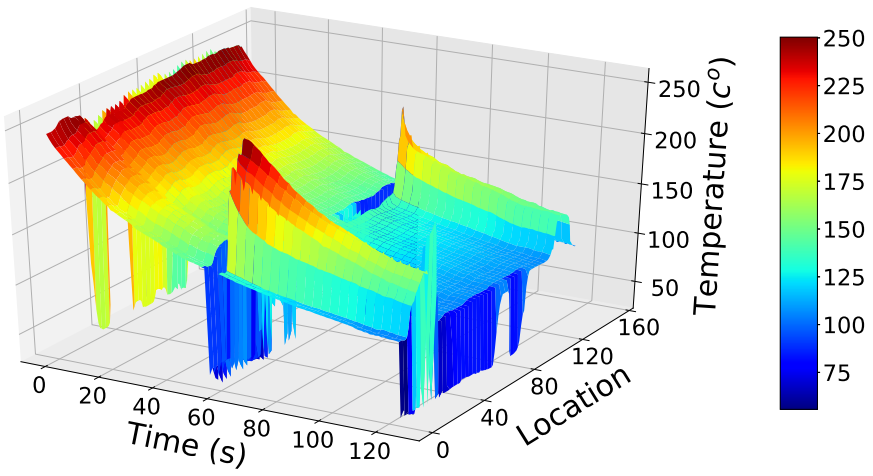


Figure 4.7: Surface Plot of Raw Temperature Data for Chassis

independent profile on the surface. According to Newton's law of cooling, the temperature dissipation on a surface has a linear relationship with the difference of surface temperature and ambient temperature. With some small modifications, Wang *et al.* (2019) proposed using the following equation for each location:

$$\frac{dT^i(t)}{dt} = \beta_0^i + \beta_1^i(T^i(t) - T_{env}), \tag{4.10}$$

where $T^i(t)$ is the surface temperature of location i at time t, T_{env} is the ambient temperature, and β_0^i and β_1^i are unknown regression parameters associated with each location. There exist different methods to introduce the spatial error between locations on the surface to the model (see the Spatial Error Model (SEM) in Fathizadan *et al.* (2017a)). However, this further complicates the real-time framework as we already accounted for it by employing the sliding windows sampling strategy to collect the data.

The fitting procedure involves with estimating parameters β_0^i and β_1^i given T_{env}. Specifically, we reformulate Equation (4.10) to be a linear regression model as follow.

$$\frac{T^i(t + \triangle t) - T^i(t)}{\triangle t} = \beta_0^i + \beta_1^i(T^i(t) - T_{env}), \tag{4.11}$$

where $(T^i(t) - T_{env})$ is the dependent variable, and $\frac{T^i(t+\triangle t)-T^i(t)}{\triangle t}$ is the independent variable. $\triangle t$ denotes the time between two consecutive observations of temperature at location i. $\triangle t$ is small, so $\frac{T^i(t+\triangle t)-T^i(t)}{\triangle t}$ is approximately equal to $\frac{dT^i(t)}{dt}$ in Equation (4.10). The Least Squares Estimation is applied to estimate parameters β_0^i and β_1^i. The estimates of β_0^i and β_1^i are denoted by $\hat{\beta}_0^i$ and $\hat{\beta}_1^i$, respectively. The predicted future surface temperature of location i for $t > t_0^i$ denoted by $\hat{T}^i(t, t_0^i)$, can be obtained as follows:

$$\hat{T}^i(t, t_0^i) = T_{env} - \frac{\hat{\beta}_0^i}{\hat{\beta}_1^i} + \left(T^i(t_0) - T_{env} + \frac{\hat{\beta}_0^i}{\hat{\beta}_1^i} \right) e^{\hat{\beta}_1^i (t - t_0^i)}, \tag{4.12}$$

where t_0^i represents the time that we can record the initial temperature $T^i(t_0)$.

As mentioned in the previous subsection, the extracted thermal data need to be cleaned before further consideration into the analysis. The data cleansing is mainly concerned with handling the outliers as the result of extruder obstruction. The steps are summarized as follows:

1. First the untidy data during the deposition times are simply removed. This has two main benefits. First, the data during deposition has too many interruptions leading to a poor fitted temperature model. Also, note that due to the large size of the parts, there is a noticeable time gap between the deposition time of distant locations. Removing the messy data during deposition makes sure that all locations will be sharing the same timestamp and thus simplifies the analysis.

2. Once the printer is done with the extrusion of materials on the ROI, it might move to other regions inside the frame and obstruct the camera again. For instance, printer can intercept the vision when it proceeds to print other products or segments. Thus, the second step of data cleansing involves with handling the outliers after the initial depositions. Since these outliers are significantly

colder than the rest of the surface, only the temperature data above a small enough threshold value are retained, and the rest are replaced by the polynomial interpolation of surrounding data points.

3. As the last step, the proposed temperature profile modelling and fitting procedure can help to detect and remove the remaining outliers. The Equation (4.12) is iteratively fitted to the data in each location, and any standardized residuals outside the z-score interval $[-3, 3]$ are removed. This process continues until all standardized residuals fall inside the interval.

The resulting estimated temperature surface plots by model fitting and data cleansing are illustrated in Figures 4.8 and 4.9 for the wall and chassis, respectively. The temperatures on the new surfaces now have a higher contrast and the surface itself looks much more smoother compared to the raw data, and therefore reveal the signatures of cooling dynamics for the whole layer more clearly. For instance, three bumps corresponding to the location of corners on the surface of the wall indicate that these regions have slower cooling rate compared to other regions. It is worth noting from the previous section that printing a layer of the chassis itself is composed of printing three segments subsequently without pause: the outer border, the inner box and the edges connecting the box and border together. Finishing the lower edge of the outer border, i.e. our ROI, the printer head redeposits a set of freshly hot materials when it returns to print the connecting edges. This will leave two hot peaks on the temperature surface of the chassis.

Cutting the surface plots across three arbitrary locations on both wall and chassis, the future surface temperatures of the whole layer have been estimated and plotted against the actual observations in Figure 4.10 and 4.11, respectively. In addition, the estimated and actual surface temperature profiles for $t = 70$ for the wall and chassis

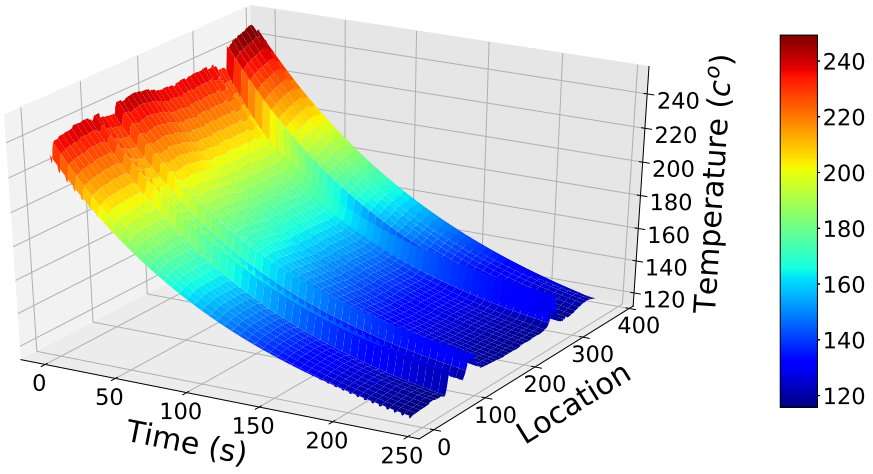


Figure 4.8: Surface Plot of Estimated Temperature Data for Wall

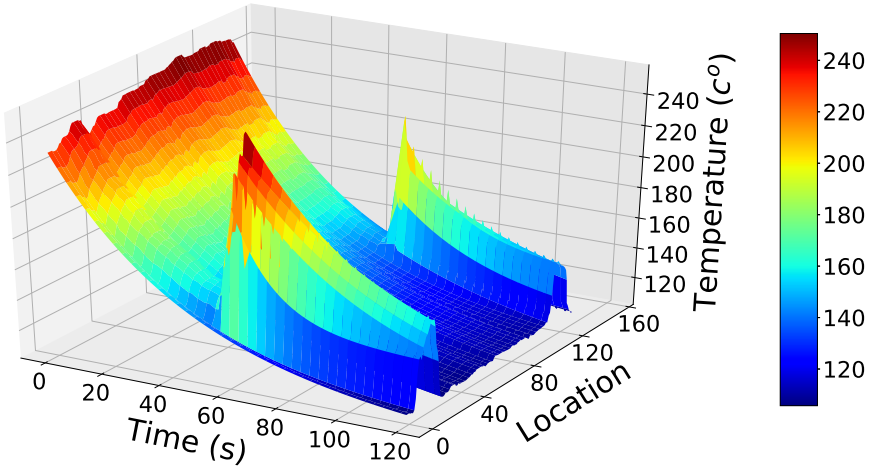
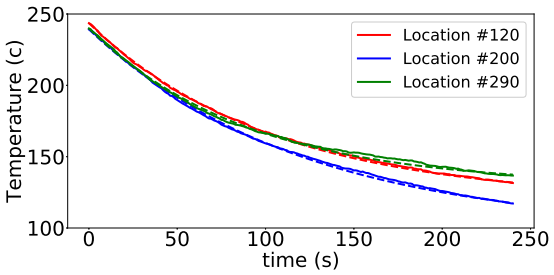
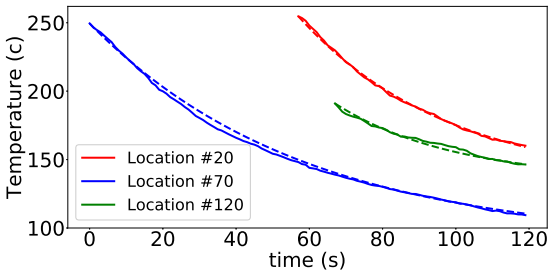


Figure 4.9: Surface Plot of Estimated Temperature Data for Chassis

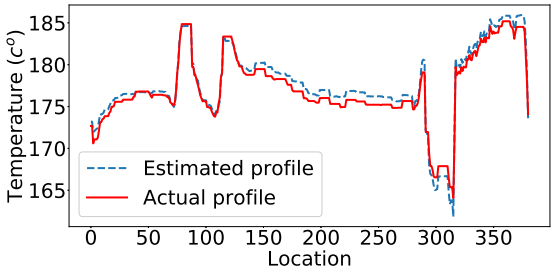


(a) Wall

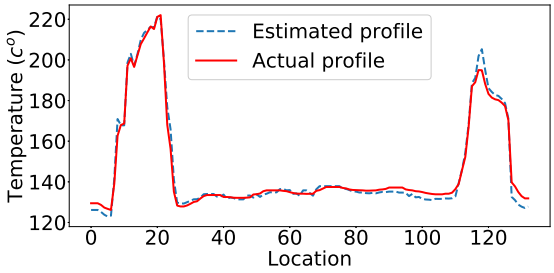


(b) Chassis

Figure 4.10: Temperature vs Time Estimation



(a) Wall



(b) Chassis

Figure 4.11: Temperature vs Location Estimation

have been compared in Figure 4.11 to provide an intuition into the performance of the model considering the entire surface. As it can be seen by the charts, there is a reasonably well agreement between the model predictions and observations across both time and location.

4.2 Real-time Printing Control

4.2.1 Model for Print Time Control

Denote M as the number of locations being monitored. The temperature change of the i^{th} location, for $i = 1, 2, \cdots, M$, is modeled by Equation (4.10) with parameters β_0^i and β_1^i, which can be estimated by one-step-ahead forecasting methods such

as Gaussian processes regression (Quiñonero-Candela and Rasmussen (2005)) and Exponentially Weighted Mean Average (Noorossana *et al.* (2016)). Then the future temperature of the i^{th} location for $t > t_0^i$, $T^i(t, t_0^i)$ can be estimated following the Equation (4.12). Here, we assume a fixed speed v for the printer head during the process. As it was discussed in Subsection 4.1.3, a unique time stamp across all locations is obtained by removing the untidy data during the deposition process. Thus, time t here is a universal variable that can be shared by all locations. The optimization model for print time control with decision variable t is built as follows:

$$\min f(t) = \sum_{i=1}^{M} \omega_i \left(\hat{T}^i \left(t + \frac{i}{v}, t_0^i \right) - T_b \right)^2 + \omega_0 t \tag{4.13}$$

s.t.

$$T_l \leq \hat{T}^i \left(t + \frac{i}{v}, t_0^i \right), \text{ for } i = 1, 2, \cdots, M, \tag{4.14}$$

$$T_u \geq \hat{T}^i \left(t + \frac{i}{v}, t_0^i \right), \text{ for } i = 1, 2, \cdots, M, \tag{4.15}$$

$$t + \frac{i}{v} \geq t_0^i, \text{ for } i = 1, 2, \cdots, M. \tag{4.16}$$

In the objective function above, the first term $\sum_{i=1}^{M} \omega_i \left(\hat{T}^i \left(t + \frac{i}{v}, t_0^i \right) - T_b \right)^2$ measures how much each location has predicted print surface temperature close to the best temperature T_b. The second term $\omega_0 t$ is the print time for the initial location multiplied with a coefficient. Minimizing the objective function is equivalent to getting the print surface temperature of each location close to t_b and minimizing the starting print time simultaneously. The positive coefficients, $\omega_0, \omega_1, \cdots, \omega_M$, are set to balance the trade-off between how much the best temperature is valued and how much a short starting print time is valued. The first two constraints in Equation (4.14) and (4.15) are to ensure the temperature of no location could go below the lower bound T_l or above the upper bound T_u of surface temperature, respectively. Finally, the last constraint in Equation (4.16) is set to force the print time for each

location to occur after its available t_0^i. This property will later come in handy in case of the chassis where some regions accept multiple reprintings during the process.

4.2.2 Model for Printer Head Speed Control

The print time control model introduced in last subsection assumes a constant speed for the printer head. Constant speed is typically used in practice where the printer does not allow the adjustment of printing speed. However, there are some situations where one can leverage the dynamic printer head speed to achieve a more flexible print by gaining better print quality and efficiency. So the decision is to determine not only the timing to print, but also the speed to use for different location. To solve this problem, we propose a new optimization model that controls the print time and printer head speed during the printing process. First, the locations on the surface are grouped into partitions such that locations inside each partition share a similar pattern of cooling dynamics. Then, we adopt a fixed speed within each partition and try to find the best partition to partition speed trajectory. This is a valid strategy for two main reasons. First, it is hard to coordinate and maintain a constantly varying speed profile especially in large parts where the surface consists of hundreds of locations. Second, the control model needs to be solved for each and every layer to provide the optimal control variables in real-time. Therefore, the maximum run time for each model has to be less than the corresponding starting print time which are usually short. Partitioning will help to reduce the number of required decision variables and as the result decrease the run time needed to solve the optimization model.

Let N be total number of partitions on the surface, and denote by t^j and v^j the print time and speed at the start of the j^{th} partition, for $j = 1, 2, \cdots, N$, respectively. Each partition itself consists of multiple locations, and the number of locations within

each partition defines its distance represented by d^j for $j = 1, 2, \cdots, N$. The lower and upper bounds of the printer head speed determined by the mechanical characteristics of the tool are given by v_l and v_u, respectively. The optimization model for simultaneous control of print time t^j and printer head speed v^j is built as follows:

$$\min f(t, v) = \sum_{j=1}^{N} \omega_j^{'} \left[\sum_{k=0}^{d^j} \left(\hat{T}^{j,k} \left(t^j + \frac{k}{v^j}, t_0^j \right) - T_b \right)^2 \right] + \omega_0^{'} t^1 \tag{4.17}$$

s.t.

$$t^{j+1} = t^j + \frac{d^j}{v^j}, \text{ for } j = 1, 2, \cdots, N, \tag{4.18}$$

$$v_l \leq v^j, \text{ for } j = 1, 2, \cdots, N, \tag{4.19}$$

$$v_u \geq v^j, \text{ for } j = 1, 2, \cdots, N, \tag{4.20}$$

$$T_l \geq \hat{T}^{j,k} \left(t^j + \frac{k}{v^j}, t_0^j \right), \text{ for } j = 1, 2, \cdots, N, \quad \text{for } k = 0, 1, \cdots, d^j, \tag{4.21}$$

$$T_u \leq \hat{T}^{j,k} \left(t^j + \frac{k}{v^j}, t_0^j \right), \text{ for } j = 1, 2, \cdots, N, \quad \text{for } k = 0, 1, \cdots, d^j, \tag{4.22}$$

$$t^j \geq t_0^j, \text{ for } j = 1, 2, \cdots, N. \tag{4.23}$$

Similar to the layer time control model, $\sum_{j=1}^{N} \left[\sum_{k=0}^{d^j} \omega_j^{'} \left(\hat{T}^{j,k} \left(t^j + \frac{k}{v^j}, t_0^j \right) - T_b \right)^2 \right]$ measures the temperature difference of each location from the ideal temperature T_b, and the second term $\omega_0^{'} t^1$ is the starting print time multiplied by a coefficient. The temperature distance from T_b and starting print time is therefore simultaneously minimized by solving the model. The positive coefficients $\omega_j^{'}$ for $j = 1, \cdots, N$, determine the value put on each partition. $\omega_0^{'}$ on the other hand determines the importance of the starting print time compared to temperature distance for partitions.

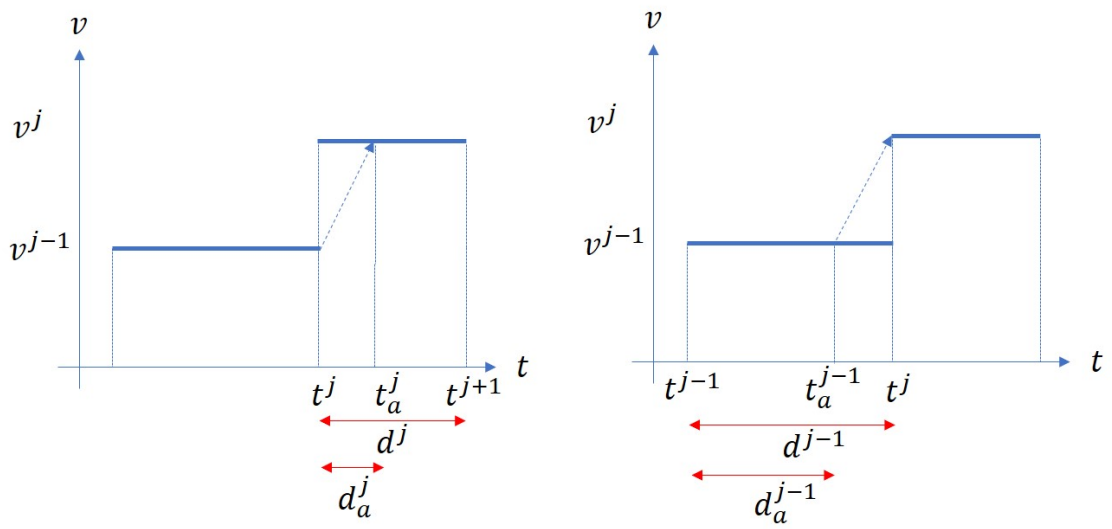
The print time for the next partition is determined by the print time from previous partition plus the time it takes to print that partition. The first constraint in Equation 4.18 establishes such a relationship. Equation (4.19) and (4.20) will keep the speed of the printer head in each partition confined within the tool predefined speed range. The

two constraints given in Equation (4.21) and (4.22) are to ensure the temperature of no location could go below the ideal lower bound or upper bound of surface temperature. Finally, the last constraint in Equation (4.23) is set to force the print time for each partition j, t^j to occur after its corresponding available t_0^j.

4.2.3 Model for Printer Head Speed Control with Acceleration

As it was argued before and investigated by the control model given in the last section, the speed of the printer head is an adjustable parameter that can be tuned in real-time. However, the printer head is not capable of changing the speed instantly. It requires some time to transit from one speed to another. This time is determined by another mechanical constraint that the printer head is subject to, called printer head acceleration/deceleration denoted by a. This imposed time latency might limit the applicability of the speed trajectory obtained from last model. Therefore, in this section we propose another control model that addresses this issue by introducing the acceleration/deceleration factor to the model. We use the term acceleration to point to both acceleration and deceleration throughout the rest of this study.

Consider an optimal speed trajectory obtained by solving the speed control model. Suppose we want to reach the speed of v^j given v^{j-1}. Generally, there are two acceleration strategies to transit from one speed to another. The first strategy is to start accelerating at the intersection of partition $j-1$ and j and then proceed by the fixed speed v_j. This strategy is referred to as feed-forward model and is shown in Figure 4.12a. The second strategy is to first proceed by the fixed speed v_{j-1} and then start accelerating before the end of the partition $j-1$ toward the beginning of the partition j. This strategy is referred to as feed-backward model and is shown in Figure 4.12b. The speed and time are represented by the vertical and horizontal axes in the figures, respectively. The new variables t_a^j and t_a^{j-1} denote the times



(a) Feed-forward Model

(b) Feed-backward Model

Figure 4.12: Printer Head Acceleration Strategies

to stop accelerating in feed-forward model or to start accelerating in feed-backward model, respectively. Interchangeably, t_a^j can be expressed by d_a^j the passed distance during acceleration since t^j. Likewise, t_a^{j-1} can also be expressed by the d_a^{j-1} the passed distance before starting to accelerate since t^{j-1}. Following the feed-forward acceleration strategy, the new printer head speed control model can be formulated as follows:

$$\min f(t,v,d_a) = \sum_{j=1}^{N} \omega_j^{''} \left[\sum_{k=0}^{d_a^j} \left[\left(\hat{T}^{j,k} \left(t^j + \frac{-v^{j-1} + \sqrt{\left(v^{j-1}\right)^2 + 4ak}}{2a}, t_0^j \right) - T_b \right)^2 + \sum_{k=d_a^j+1}^{d^j} \left(\hat{T}^{j,k} \left(t_a^j + \frac{k}{v^j}, t_0^j \right) - T_b \right)^2 \right] \right] + \omega_0^{''} t^1 \tag{4.24}$$

s.t.

$$t^{j+1} = t_a^j + \frac{d^j - d_a^j}{v^j}, \text{ for } j = 1, 2, \cdots, N, \tag{4.25}$$

$$t_a^j = t^j + \frac{\mid v^j - v^{j-1} \mid}{a}, \text{ for } j = 1, 2, \cdots, N, \tag{4.26}$$

$$d_a^j = a\left(t_a^j - t_j\right)^2 + v^{j-1}\left(t_a^j - t_j\right), \text{ for } j = 1, 2, \cdots, N, \tag{4.27}$$

$$v_l \leq v^j, \text{ for } j = 1, 2, \cdots, N, \tag{4.28}$$

$$v_u \geq v^j, \text{ for } j = 1, 2, \cdots, N, \tag{4.29}$$

$$T_l \geq \hat{T}^{j,k} \left(t^j + \frac{-v^{j-1} + \sqrt{\left(v^{j-1}\right)^2 + 4ak}}{2a}, t_0^j \right), \quad \text{for } j = 0, 1, \cdots, N, \text{ for } k = 0, 1, \cdots, d_a^j, \tag{4.30}$$

$$T_u \leq \hat{T}^{j,k} \left(t^j + \frac{-v^{j-1} + \sqrt{\left(v^{j-1}\right)^2 + 4ak}}{2a}, t_0^j \right), \quad \text{for } j = 0, 1, \cdots, N, \text{ for } k = 0, 1, \cdots, d_a^j, \tag{4.31}$$

$$T_l \geq \hat{T}^{j,k} \left(t_a^j + \frac{k}{v^j}, t_0^j \right), \text{ for } j = 1, 2, \cdots, N, \text{ for } k = d_a^j + 1, \cdots, d^j, \tag{4.32}$$

$$T_u \leq \hat{T}^{j,k} \left(t_a^j + \frac{k}{v^j}, t_0^j \right), \text{ for } j = 1, 2, \cdots, N, \text{ for } k = d_a^j + 1, \cdots, d^j, \tag{4.33}$$

$$d_a^j \leq d^j, \text{ for } j = 1, 2, \cdots, N, \tag{4.34}$$

$$d_a^j \geq 0, \text{ for } j = 1, 2, \cdots, N, \tag{4.35}$$

$$t_a^j \geq 0, \text{ for } j = 1, 2, \cdots, N. \tag{4.36}$$

$$t^j \geq t_0^j, \text{ for } j = 1, 2, \cdots, N. \tag{4.37}$$

The first term inside the brackets in Equation (4.24), $\left(\hat{T}^{j,k}\left(t^j + \frac{-v^{j-1}+\sqrt{(v^{j-1})^2+4ak}}{2a}, t_0^j\right) - T_b\right)^2$ aims to minimize the the temperature distances of the locations inside partition j that are involved in the acceleration process from the best temperature T_b. The second term inside the brackets, $\left(\hat{T}^{j,k}\left(t_a^j + \frac{k}{v^j}, t_0^j\right) - T_b\right)^2$, aims to minimize the the temperature distances of the locations inside partition j that are passed by the fixed speed v^j from the best temperature T_b. The final term, $\omega_0^{''}t^1$, is again the starting print time multiplied by a coefficient, and the positive coefficients $\omega_j^{''}$ for $j = 1, \cdots, N$, are used to set the trade-off values among different partitions. Similarly, $\omega_0^{''}$ determines the importance of the starting print time compared to temperature distance for partitions.

The first constraint given by Equation (4.25) schedules the print time for next partition t^{j+1} after the acceleration time t_a^j plus the time needed to reach the end of the current partition after d_a^j with fixed speed v^j. Given two consecutive speeds v^j and v^{j-1}, the second constraint in Equation (4.26) finds the time to stop accelerating t_a^j. The passed distance during acceleration d_a^j is determined by the third constraint provided in Equation (4.27). Similar to last control model, Equations (4.28) and (4.29) will keep the speed of the printer head in each partition confined within the printer predefined speed range. Moreover, Equations (4.30) to (4.31) ensure that temperature of all locations will lie inside the interval defined by the ideal lower bound or upper bound of surface temperature. Equation (4.34) guarantees that the passed distance during acceleration d_a^j is less than or equal to the corresponding partition distance d^j. The two Equations (4.35) and (4.36) are set to force d_a^j and t_a^j to be greater than or equal to zero, respectively. Finally, the last constraint in Equation (4.37) is set to force the print time for each partition j, t^j, to occur after its corresponding available t_0^j.

4.3 Solution Approaches

The three control models introduced in the last section share a common goal which is to increase the efficiency by decreasing the starting print time and improve the quality by controlling the surface temperature. However, the complexity involved with formulation and structure of objective functions and constraints is different for each model. Therefore, separate methodologies are required to find the optimal control variables for each model. In the following sections, we first analyze the characteristics of each model and then provide the suitable procedure to derive the optimal control variables.

4.3.1 Optimal Solution for Print Time Control Model

Using Equation (4.12), the corresponding ts, for which the model constraints given in Inequations (4.14), (4.15) and (4.16) hold can be obtained. After some manipulations, the upper and lower bounds for variable t denoted as t_U and t_L are as follows:

$$t_L = \max_i \left\{ \frac{1}{\beta_1^i} ln \frac{T_u - T_{env} + \frac{\beta_0^i}{\beta_1^i}}{T^i(t_0^i) - T_{env} + \frac{\beta_0^i}{\beta_1^i}} - \frac{i}{v} + t_0^i \right\} \tag{4.38}$$

$$t_U = \min_i \left\{ \frac{1}{\beta_1^i} ln \frac{T_l - T_{env} + \frac{\beta_0^i}{\beta_1^i}}{T^i(t_0^i) - T_{env} + \frac{\beta_0^i}{\beta_1^i}} - \frac{i}{v} + t_0^i \right\} \tag{4.39}$$

The above equations form a bounded interval for variable t which can be represented by $[t_L, t_U]$. This will lead the optimization problem to be reformulated as minimizing the objective function $f(t)$ in Equation (4.13) with bounded decision variable as follow:

$$\min_t f(t) \quad t \in [t_L, t_U] \tag{4.40}$$

The objective function is differentiable and nonlinear but in general, is neither a

convex nor concave function. However, we provide a necessary and sufficient condition that when satisfied guarantees the the convexity of the objective function.

Proposition 2. *The optimization model described by Equation (4.13) - (4.16) is convex, if and only if* $t_U \leq \gamma$*, where* γ *is the numerical solution to the following equation:*

$$\sum_{i=1}^{M} \left(a^i + 2b^i e^{\beta_1^i (t + \frac{i}{v} - t_0^i)} \right) c^i = 0 \tag{4.41}$$

, where $a^i = T_{env} - \frac{\beta_0^i}{\beta_1^i} - T_b$*,* $b^i = T^i(t_0) - T_{env} + \frac{\beta_0^i}{\beta_1^i}$*, and* $c^i = 2\omega_i b^i \left(\beta_1^i\right)^2 e^{\beta_1^i (t + \frac{i}{v} - t_0^i)}$*.*

See appendix (A) for proof.

If Proposition 2 holds, then the optimization problem becomes a convex optimization model, and the local optimal will be the global solution. The univariate Newton method can then be applied to find both γ and the optimal layer time control t^*. Selecting a random initial guess t_0 from inside $[t_L, t_U]$, the following equation is iterated until convergence:

$$t_{k+1} = t_k - \frac{f(t_k)}{f'(t_k)}, \quad \text{for} \quad k = 0, 1, 2, \cdots . \tag{4.42}$$

4.3.2 Optimal Solution for Printer Head Speed Control Model

In contrast to print time control model, the printer head speed control model given by Equations (4.17) - (4.23) can not be further reduced to a more compact model. Moreover, the objective function is differentiable and nonlinear but in general, is neither a convex nor concave function. Providing a condition for the convexity of this problem is fairly complex and thus is out of scope of this study. The optimization model can be solved by a method for numerical optimization that can handle nonlinear objective function and constraints. For this purpose, the Sequential Quadratic Programming (SQP) method is used to find the optimal solution. SQP is an iterative method using the Taylor's second order approximations of the objective function

with the first order Taylor's approximations of the constraints (Nocedal and Wright (2006)). Let $b(t,v) = 0$ represent the constraint provided in Equation (4.18) and $c(t,v) \geq 0$ represent the constraints provided in Inequation (4.19) - (4.23). Define the Lagrangian of the problem as $L(t,v,\lambda) = f(t,v) - \lambda^T (b(t,v) + c(t,v))$, where vector λ is a Lagrange multiplier. Let t^k, v^k and λ^k be the solution and the Lagrange multiplier at iterate k, respectively. Then, a solution to the following quadratic programming subproblem defines an appropriate search direction $D^k = \left[D_t^k, D_v^k\right]^T$:

$$\begin{aligned} \min_{D^k} \; & \nabla f\left(t^k, v^k\right)^T D^k + \frac{1}{2}\left(D^k\right)^T \nabla^2_{tt,vv} L\left(t^k, v^k, \lambda^k\right) D^k \\ & \text{s.t.} \\ & \nabla b\left(t^k, v^k\right)^T D^k + b\left(t^k, v^k\right) = 0. \\ & \nabla c\left(t^k, v^k\right)^T D^k + c\left(t^k, v^k\right) \geq 0. \end{aligned} \tag{4.43}$$

The values of t, v and λ for iteration $(k+1)$ can be obtained by $\left[t^{k+1}, v^{k+1}\right]^T = \left[t^k, v^k\right]^T + \left[D_t^k, D_v^k\right]^T$ and the corresponding Lagrange multiplier of the quadratic subproblem, respectively. The initial values of t, v and λ, denoted by t^0, v^0 and λ^0, are determined by choosing a random value, and the optimal values are found by repeating the process until convergence. The method can be initialized with several random values to ensure it is not stuck in a local minimum and that the global optimal has been found.

4.3.3 Optimal Solution for Printer Head Speed with Acceleration Control Model

The control model given by Equations (4.24) to (4.37) is a non-linear mixed integer program. Computational complexity aside, this problem clearly doesn't fit into any standard structure required by the relevant solving algorithms. For instance, d_a^j that appears at the index of summation in the objective function must be an integer variable. This issue arises in other applications and a general strategy usually is to

break down the original problem into simpler subproblems (Azizi *et al.* (2019)). It is worth noting from section (4.2.2) that the locations on the surface are grouped together to form different partitions. Each partition therefore can serve as a stage for which the optimal controls are to be found for subproblems. The solutions for subproblems together form the overall solution for the main problem.

Following the dynamic systems paradigm, we can derive the state of the system at each stage j. Figure (4.12a) suggests that the state of feed-forward system can be derived by the vector $[t^j, v^{j-1}]$, the print time at the start of the current stage and speed at the end of the previous stage. On the other hand, the control variable comprised of choosing a location to stop accelerating denoted by d_a^j. The set of Equations given by (4.25) to (4.27) provide the system evolution equations, and the rest are simply the problem constraints on the upper and lower bound of the variables. The additive cost of transition from one stage to another is composed of the summation of differences between surface and best temperature during acceleration with varying speed, and after acceleration with constant speed. Accordingly, the problem can be treated as a deterministic finite-state optimal control or alternatively a shortest-path (SP) problem.

Define the nodes that describe the state of the system at stage j by the tuple (j, t^j, v^{j-1}), such that $j \in \{1, 2, \ldots, N\}$, $t^j \in [t_l, t_u]$, and $v^{j-1} \in [v_l, v_u]$. The values of t_l and t_u are the lower and upper bounds of the print time search region. Denote the print time for the first and last location on the surface found by solving the speed control model by lt_f and lt_l, respectively. Then t_l and t_u can be derived as follows:

$$t_l = lt_f - \delta, \tag{4.44}$$

$$t_u = lt_l + \delta. \tag{4.45}$$

The proper choice of δ is important since too wide search region results in the

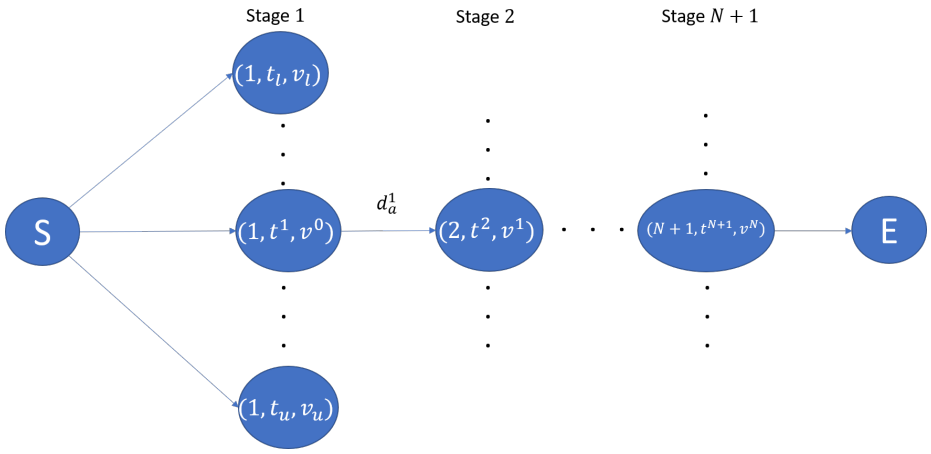


Figure 4.13: Shortest Path Representation of the Speed with Acceleration Control Model

explosion of the problem, and too narrow region leads to suboptimality. Moreover, v_l and v_u are the lower and upper bounds of the printer head speed determined by the mechanical characteristics of the tool as shown by Table 4.1. On the other hand, choosing a value for control variable d_a^j from the set $\{0, 1, \ldots, d^j\}$, results in landing to a child node for which the occurred cost is described above. Moreover, S and E define the artificial origin and terminating nodes, respectively. Obviously, the state variables are continuous, and therefore their corresponding state space is infinite. Furthermore, the control space is relatively large which can exponentially increase the run time needed to solve the model. A common practice is to transform the state and control space into a grid of discrete values with sufficiently small increments (Bertsekas (1995)). The discertization increments for t^j, v^{j-1}, and d_a^j is set to be equal to inc_t, inc_v, and inc_d, respectively. A schematic of the SP problem is shown in Figure (4.13).

A greedy approach to solve this problem is to first solve the speed control model and then take the optimal starting print time and speed of the first partition as the

state of the first stage for speed control with acceleration model. Trying different controls, one can optimize this subproblem and also find the landing child node for the next stage. The same process continues until all subproblems are optimized separately. This method is pretty fast but it doesn't take the optimality of the overall problem into account. Thus, it usually fails to provide the global optimal. Here, we propose a label correcting algorithm called "Dijkstra A^* with greedy initialization" that is specifically tailored to solve the SP for printer head speed control with acceleration model. Let $dist_p$ be a label that maintains the most recent shortest path from S to node p, and a_{pq} be the the cost of moving from parent node p to child node q, i.e. the cost of arc (p, q). The value of $dist_E$ and the set of candidate nodes to enter the shortest path are stored in separate variables called $UPPER$ and $OPEN$, respectively. Using the solution from greedy algorithm, we can construct a initial good path represented by $P = (S, p_1, \cdots, p_k, \cdots, E)$. Then, the label of all nodes in the greedy path are initially assigned to their distance from S except for node S where its label is always equal to 0. The label of the rest of the nodes in the system are also initially assigned to ∞, and the initial $OPEN$ list contains the nodes p whose $dist_p$ are less than ∞.

As for the A^* part of the algorithm, each node p requires a positive underestimate of distance from itself to the terminating node E denoted by h_p to further bolster the test to enter the $OPEN$ list. It is easy to show that any solution found by the speed control with acceleration model is also feasible for the speed control model but the vice versa is not correct due to extra acceleration restrictions. Therefore, any solution found by the speed control with acceleration control must have a greater value for objective function than that for the speed control model. We utilize this fact to derive an underestimate for each node.

Proposition 3. *The value of objective function obtained by solving the speed*

Algorithm 1: Dijkstra A* with Greedy Initialization

```
Greedy initialization;
while OPEN is not empty do
    Find the node p with minimum value of label from OPEN, i.e.
     p = arg min{dist_i | i ∈ OPEN};
    Remove the node p from OPEN;
    for Each child q of p do
        if dist_p + a_pq + h_q < UPPER then
            dist_q = dist_p + a_pq;
            Assign p to be the parent of q;
            if q ≠ E then
                Place q in OPEN;
            else
                UPPER = dist_p + a_pq;
            end if
        end if
    end for
end while
```

control model for (j, t_l, v_l)*, the first node at each stage* j*, is the underestimate of all other nodes* (j, t^j, v^{j-1}) *for* $t^j \in [t_l, t_u]$ *and* $v^{j-1} \in [v_l, v_u]$ *at that stage, i.e.:*

$$f^*(t, v \mid j, t_l, v_l) \leq f^*(t, v \mid j, t^j, v^{j-1}), \ \ for \ j = 1, 2, \cdots, N \tag{4.46}$$

,where $f^*(t, v \mid j, t^j, v^{j-1})$ *is the optimal objective function of speed control model from partition* j *onward given that* t^j *seconds have been passed so far and the initial speed is* v^{j-1}*.*

See appendix (B) for proof.

Therefore, running the speed control model only for the first node at each stage will provide the underestimates for the remainder of nodes at that stage. The step by step procedure of the algorithm is summarized in Algorithm 1.

4.4 Case Study and Numerical Results

4.4.1 Optimal Control for Case Studies

The optimal controls obtained by solving the proposed optimization models are discussed in this section for both wall test and lower border of the chassis. Refer to Figures (4.14) and (4.15) for a demonstration on how locations with common dynamic characteristics are grouped together to form the partitions on the estimated heatmap of wall and chassis, respectively. The radius of the sliding windows profile scan introduced in section (4.1.2) is set to be 5 pixels. Therefore, only the locations by increments of 5 pixels in each partition on the surface will be evaluated for all three models. This not only ensures a fair comparison of the impact of different control models on the value of objective function but also significantly reduces the run time needed to solve the speed control with acceleration model. This also means that inc_d has to be set to 5. Furthermore, the minimum and maximum speed and maximum acceleration/deceleration of the printer head, i.e. v_l, v_u and a, are found to be about 15 and 40 pixels/s, and 15 pixels/s^2 after conversion, respectively. In this study, every positive coefficient ω, $\omega^{'}$, and $\omega^{''}$ for all three models are set to 1. Also, note that for the sake of simplicity, we assume a continuous surface by ignoring the truncated area at the upper right hand side of the wall as it is always possible to capture an image from the entire surface.

First, the print time and speed control models are solved for both case studies.

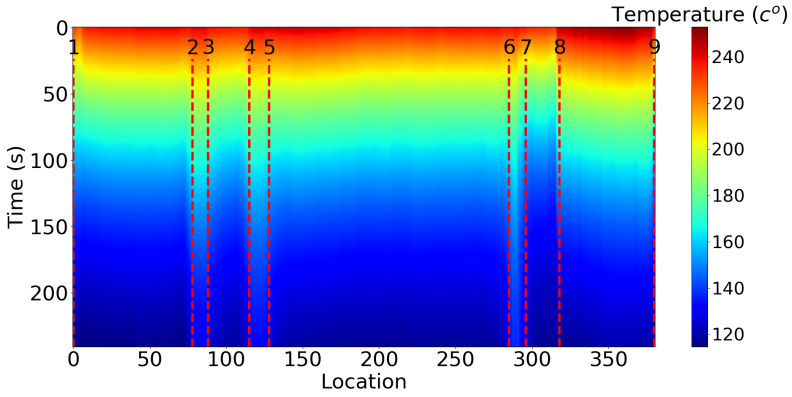


Figure 4.14: Partitioned Estimated Heatmap for Wall

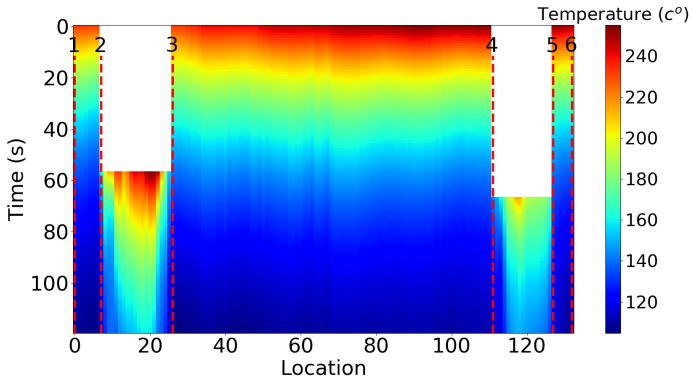


Figure 4.15: Partitioned Estimated Heatmap for Chassis

To this aim, first Propositions 2 is tested for the potential convexity of the print time control model for both wall and chassis. In case of wall, the bounded interval for t and γ are found to be $[142.60, 203.97]$ and 239.49 respectively. Meaning the print time control model for the wall is a convex problem as t_u is less than γ. On the other hand for the chassis, all of the constraint concerning the temperature, i.e. Equations (4.14) and (4.15) for the layer time and Equations (4.21) and (4.22) for the speed control models are violated due to the reprintings in partitions 2 and 4. Nevertheless, the value of γ found to be 100.68, meaning the print time control model is convex for

time values less than 100.68. In the cases where there exists a fear of non-convexity, the corresponding solving algorithms should be repeated with several random initial guesses. Then, the solution with the best objective function can be considered as the global optimal.

Once the optimal controls for speed control model are found, they are used to decide for the lower and upper bounds of print time search region needed to solve for the speed with acceleration control model. They are also used by the greedy algorithm to augment the Dijkstra A^* with the initialization step. Specifically, the values of t_l, t_u and inc_t are set to 165, 195, and 0.5 seconds respectively for the wall and 88, 103, and 0.25 seconds respectively for the chassis. The value of inc_v is set to be 1 pixels/s for both cases and following the radius of the sliding windows profile scan, the value of inc_d is set to be 5 pixels. Figures 4.16 and 4.17 provide the optimal print time, speed trajectory and location to stop accelerating found by solving the print time, speed and speed with acceleration control models for the wall and chassis respectively. As is expected, in both cases the speed control with acceleration model is showing a similar pattern to the speed control model in terms of print time. On the other hand, as it is obvious from the plots the optimal speed trajectory found by speed with acceleration model is a smoothed version of that for the speed control model. It is not feasible to reach very high or very low speeds when there is no sufficient distance between two consecutive partitions. As it is shown by Table 4.2, the current fixed layer times are equal to $250s$ and $126s$ for the wall and chassis, respectively. However, the solutions provided by the control models suggest that the fabrication speed can be boosted up by about 30% and 25% for the wall and chassis, respectively. This is a substantial saving in time when considering that the finished products are composed of several layers.

Moreover, Figures 4.18a and 4.18b present the value of objective function after

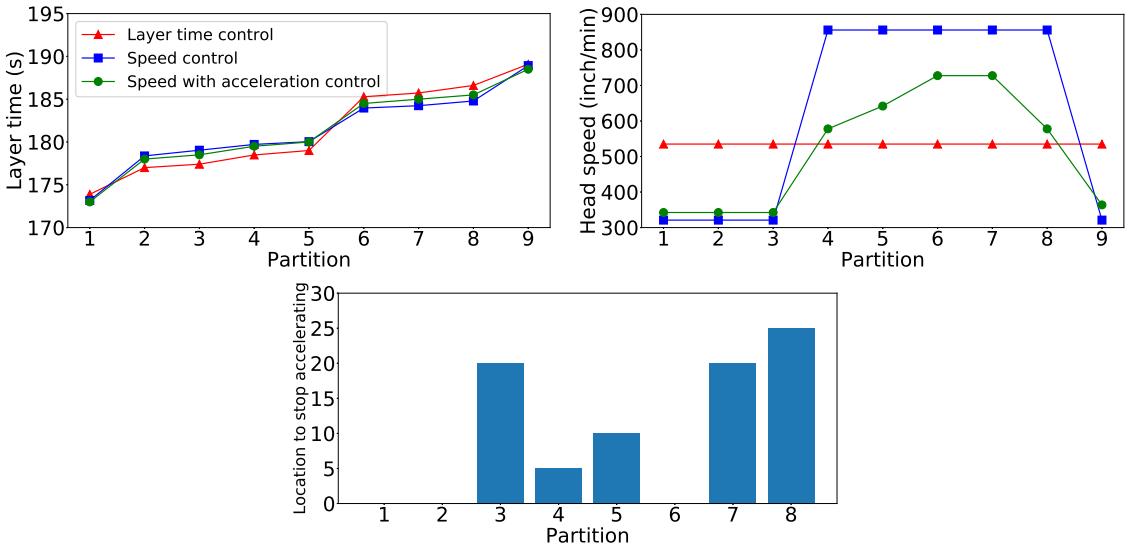


Figure 4.16: Optimal Controls for the Layer Time, Speed and Speed with Acceleration Control Models for the Wall

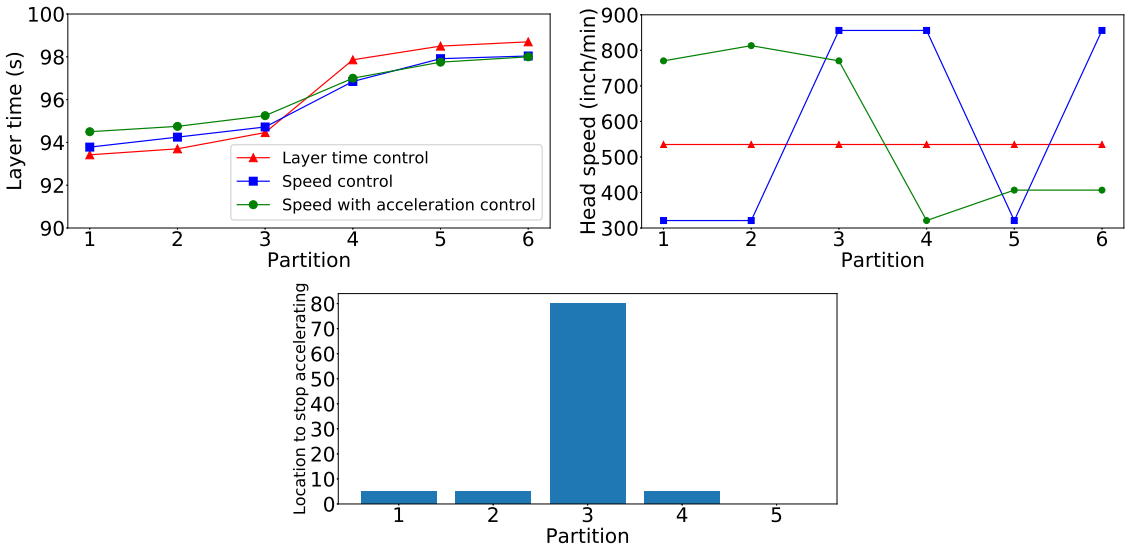
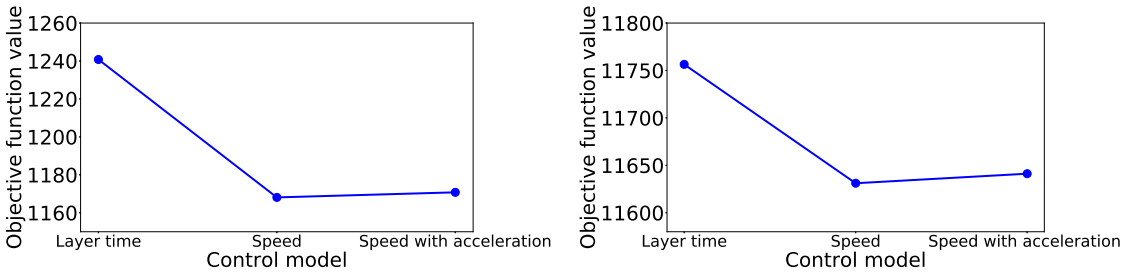


Figure 4.17: Optimal Controls for the Layer Time, Speed and Speed with Acceleration Control Models for the Chassis



(a) Wall

(b) Chassis

Figure 4.18: Optimal Value of Objective Function for Control Models

Table 4.3: Solving Run Time for SP and DP Methods

Solving algorithm	Run time (s)	
	Wall	Chassis
Dijkstra A^* with greedy initialization	139.80	21.05
Backward DP	252.21	24.50

employing the optimal control for each model for the wall and chassis, respectively. Evaluating the objective functions given by either of the models reveals that fixed starting print time policy is highly inefficient. The value of objective function by employing corresponding fixed starting print time for the wall and chassis found to be equal to 21316.83 and 17358.83, respectively. As can be observed by both plots, a significant drop in objective function compared to the current fixed-time strategy is achieved simply by the print time control model following by a more moderate improvement when speed control is applied. This indicates that the control models are capable of bringing the temperature of many locations close to the ideal temperature and keep them within the predefined thermal interval what that is crucial to achieve a good quality product.

4.4.2 Discussion

Formulating the speed with acceleration control model as a deterministic finite-state optimal control or interchangeably a shortest-path (SP) problem, gives rise to employment of other algorithms designed to solve such problems. Out of the SP algorithms family, we investigated the Small-Label-first (SLF) and generic Dijkstra method. The results showed that generic Dijkstra is outperforming the SLF in terms of the solving run time. Therefore, Dijkstra was selected for further tuning by equipping it with the proposed A^* and greedy initialization features.

As an alternative to SP, we also utilized the backward Dynamic Programming (DP) algorithm. Let $J_j(x_j)$ denote the cost of moving forward at stage j by its corresponding state vector $x_j = [t^j, v^{j-1}]$. Moreover, let $g_j(x_j, u_j)$ be the additive cost of transition from stage j to stage $j+1$ by taking control variable $u_j = d_a^j$ as described by subsection 4.2.3. The backward Bellman equations minimizing the total cost of the system are given as follows:

$$J_{N+1}^*(x_{N+1}) = g_{N+1}(x_{N+1}) \tag{4.47}$$

$$J_j^*(x_j) = \min_{u_j \in U_j(x_j)} \{g_j(x_j, u_j) + J_{j+1}^*(x_{j+1} = f_j(x_j, u_j)\}, \text{ for } j = 0, 1, \ldots, N \tag{4.48}$$

, where $f_j(x_j, u_j)$ is the set of system evolution equations given by Equations (4.25) to (4.27), and $U_j(x_j) = \{0, 1, \ldots, d^j\}$ is the set of available policies for x_j. The printer head speed with acceleration control model was solved via backward DP and its performance in terms of run time was compared with Dijkstra A^* with greedy initialization. The configurations of the PC used to run the models are Intel (R), Core ™ i7, 8,750 CPU, 2.20 GHz, and 16.0 GB of RAM. Table 4.3 provides the run time needed to solve the problem with each method for the wall and chassis. As it is obvious from the table, the proposed SP method is outperforming the DP for both wall and chassis. This is expected as in the DP algorithm every node and arc will

participate in the computation where by the proposed method a noticable portion of the nodes and arcs not relevant to the shortest path are eliminated. It is worth to mention that it is not possible to use DP as a real time solver for wall because its run time exceeds the corresponding optimal starting print time.

Chapter 5

INDIVIDUAL ANOMALY DETECTION FOR PROCESS VARIATION MANAGEMENT IN LASER POWDER BED FUSION

5.1 Methodology

The framework of the proposed methodology is illustrated in Figure 5.1. The pre-processed tensor of melt pool images are fed to the CAE, and the bottleneck tensor from latent space is extracted. This encoded deep representation tensor is then flattened and deployed by a method of agglomerative clustering to annotate the data as either normal or anomaly. In the next step, the CAE is retrained on the normal samples, and the new set of extracted encoded vectors are used to construct the phase-I of the proposed multivariate control charting method. The deep representations of the set of previously discovered anomalies are then employed by the phase-II control charting to evaluate the method's performance in terms of anomaly detection. The three constituents of the framework, including CAE neural networks to process the melt pool images, agglomerative clustering algorithm to annotate the data, and the control charting scheme to monitor the process's stability and perform anomaly detection, are delineated by the next subsections, respectively.

5.1.1 Deep Representation Learning for Feature Extraction from Melt Pool Images

Assume a set of N grey-scale images each with a dimension of $d_1 \times d_2$ and let's denote the flattened vector of i^{th} image by $\vec{x_i} \in \mathbb{R}^{1\times P}$, for $i = 1, 2, \ldots, N$, where $P = d_1 \times d_2$. The vector of weights associated with each neuron j in the hidden layer is represented by $\vec{\alpha_j} = [\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jP}]$, for $j = 1, 2, \ldots, M$, where M is the total

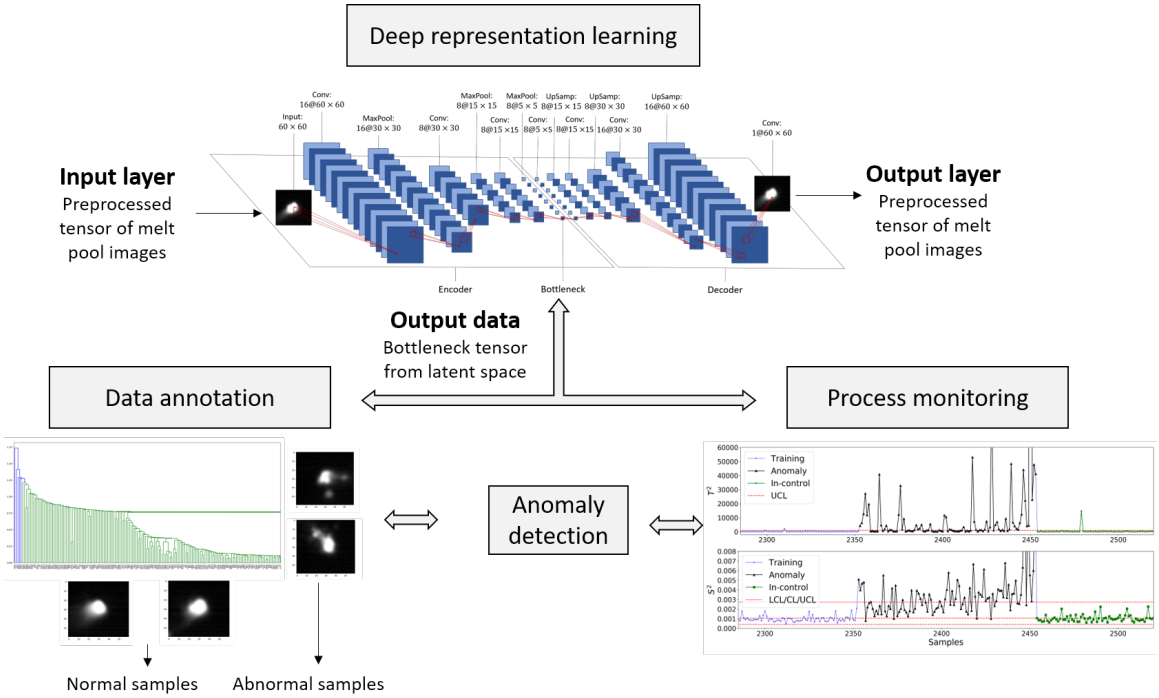


Figure 5.1: Framework of Proposed Deep Representation Learning for Anomaly Detection Methodology

number of neurons in the hidden layer. Furthermore, let $\vec{\beta_k} = [\beta_{k1}, \beta_{k2}, \ldots, \beta_{kM}]$ be the vector of weights associated with each neuron k in the final output layer, for $k = 1, 2, \ldots, K$, where K is the total number of neurons in the hidden layer or equivalently the number of classes. The output of each output unit k for the input i, $\hat{y}_{ik}$, is the probability of assigning each input vector $\vec{x_i}$ to class k. The model utilizes the outputs $\hat{y}_{ik}$ to benchmark its performance against y_{ik}, the ground truth for the sample inputs. The criteria for benchmarking is given by a loss function, which is pursued to minimized. Table 6.7 provides the expression forms of the minimization of two common loss functions, mean squared error and cross entropy.

Stochastic gradient descent is used to train the neural network on large data sets. Without the loss of generality, consider the loss function after dropping the $1/N$. Denoted by $\mathcal{L}(x; \alpha, \beta)$, it is comprised of sum of errors for each sample,

	Loss function	
	Mean squared error	Cross entropy
Expression form	$\min_{\alpha,\beta} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - \hat{y}_{ik})^2$	$\min_{\alpha,\beta} -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} ln(\hat{y}_{ik})$

Table 5.1: Common Loss Functions for Training Neural Networks

i.e. $\mathcal{L}(x; \alpha, \beta) = \sum_{i=1}^{N} \mathcal{L}(x_i; \alpha, \beta)$, where $\mathcal{L}(x_i; \alpha, \beta)$ is equal to $\sum_{k=1}^{K} (y_{ik} - \hat{y}_{ik})^2$ or $\sum_{k=1}^{K} y_{ik} ln(\hat{y}_{ik})$ in case of mean squared error or cross entropy, respectively. Then, given the weights at time τ, the updated weights at time $\tau + 1$ can be found by moving in the direction of the negative gradient. The size of the movement is a small non-negative number called learning rate, and is denoted by γ. Specifically, $\vec{\alpha}_j$ and $\vec{\beta}_k$ weights are updates as follows:

$$\begin{bmatrix} \alpha_{j1}^{\tau+1} \\ \vdots \\ \alpha_{jp}^{\tau+1} \\ \vdots \\ \alpha_{jP}^{\tau+1} \end{bmatrix} = \begin{bmatrix} \alpha_{j1}^{\tau} \\ \vdots \\ \alpha_{jp}^{\tau} \\ \vdots \\ \alpha_{jP}^{\tau} \end{bmatrix} - \gamma \begin{bmatrix} \frac{\partial \mathcal{L}(x_i;\alpha,\beta)}{\partial \alpha_{j1}^{\tau}} \\ \vdots \\ \frac{\partial \mathcal{L}(x_i;\alpha,\beta)}{\partial \alpha_{jp}^{\tau}} \\ \vdots \\ \frac{\partial \mathcal{L}(x_i;\alpha,\beta)}{\partial \alpha_{jP}^{\tau}} \end{bmatrix}, \quad \begin{bmatrix} \beta_{k1}^{\tau+1} \\ \vdots \\ \beta_{km}^{\tau+1} \\ \vdots \\ \beta_{kM}^{\tau+1} \end{bmatrix} = \begin{bmatrix} \beta_{k1}^{\tau} \\ \vdots \\ \beta_{km}^{\tau} \\ \vdots \\ \beta_{kM}^{\tau} \end{bmatrix} - \gamma \begin{bmatrix} \frac{\partial \mathcal{L}(x_i;\alpha,\beta)}{\partial \beta_{k1}^{\tau}} \\ \vdots \\ \frac{\partial \mathcal{L}(x_i;\alpha,\beta)}{\partial \beta_{km}^{\tau}} \\ \vdots \\ \frac{\partial \mathcal{L}(x_i;\alpha,\beta)}{\partial \beta_{kM}^{\tau}} \end{bmatrix}. \tag{5.1}$$

The value of partial derivatives of loss of each sample with respect to α_{jp} and β_{km} is computed by the chain rule for derivatives and error backpropagation. The process can be iterated until convergence. Adaptive learning rate and momentum such as "Adam" (Kingma and Ba (2014)), "Adadelta" (Zeiler (2012)) and "RMSprop" (Tieleman and Hinton (2012)) is used to avoid a local minima. Finally, as the number of samples is relatively large in practice, a batch of samples is used to update the parameters as opposed to one sample at a time procedure in Equation 5.1.

The explosion in the number of weights as image data gets large in dimension can lead to excessive computational overheads. Furthermore, there is also a loss in spatial

information when images are flattened for the feed forwarding process. As a remedy, Convolution Neural Networks are developed to locally connect the features on the input image to the nodes inside the hidden layers. CNNs consist primarily of three different types of layers: convolutions, pooling, and fully connected layers. Fully connected layers are discussed above, so the details for convolutional and pooling layers are given next. A matrix known as a kernel is slid over the input matrix in the convolutional layers to create a feature map by performing convolution operation (Goodfellow *et al.* (2016)). Let K and S be the $f_1 \times f_2$ kernel and feature map matrix, respectively. The feature map can be obtained following the 2-D convolution operation denoted by '$*$' on the two-dimensional image X as follows:

$$S(i,j) = (X * K)(i,j) = \sum_{m=0}^{f_1-1} \sum_{n=0}^{f_2-1} X(i+m, j+n) K(m,n). \tag{5.2}$$

Then, a non-linear activation function h such as sigmoid function (σ), hyperbolic tangent ($tanh$), or Rectified Linear Unit ($ReLU$) (Bishop (2006)) is applied to the feature map to obtain the final output, i.e., $h(S)$. In practice, more than one kernel is applied to the images so as to extract different types of features. The number of kernels is called filter or kernel depth. Therefore, the final feature map forms a $3D$ tensor composed of several feature map matrices.

Next, a pooling layer is applied to the feature map produced by the previous layer to downsample the data while preserving spatial invariance. The notion behind pooling layers is relatively straightforward. Max pooling, the most common type of pooling, is to simply take the maximum value from a given array or matrix. In this case, the feature map matrix is partitioned into $w \times u$ submatrices, and the maximum values inside each submatrix are taken to form a new matrix denoted by D:

$$D(i,j) = \max\{S(i+m, j+n) \mid m \in [0,w], n \in [0,u]\}. \tag{5.3}$$

It is worth noting that the convolutions and max-pooling layers are stacked on top of each other indefinitely to gain better results in practice.

Finally, the structure behind the proposed CAE can be demystified as follows. Specifically, Auto-Encoders are a method of unsupervised learning that learn a compact representation of the unlabeled input data. As the name suggests, it is composed of two components: an encoder and a decoder. The encoder f learns a mapping from input data X to a compact latent space H, i.e. $f : X \rightarrow H$, and the decoder g that learns a mapping back from latent space to a reconstructed version of original input data $\tilde{X}$, i.e. $g : H \rightarrow \tilde{X}$. Remember from above that the ground truth for each sample is represented by $\vec{y_i} = [y_{i1}, y_{i2}, \ldots, y_{iK}]$, where $y_{ik} = 1$, if sample i belongs to the class k and the rest are simply zero. The Auto-Encoders have exactly the same structure and training process as fully connected neural networks with the difference that $\vec{y_i}$ is simply replaced with $\vec{x_i}$. In the case of CAEs, just as similar to CNNs, the data are not flattened, and therefore, the input and output matrices are mapped to each other while the spatial correlation is preserved. The layer which holds the data in the latent space is called the bottleneck layer. The bottleneck layer outputs an encoded tensor of customized dimensions for each X_i denoted by H_i. The model also has several hyper-parameters, including the number of layers, kernel's and pooling's size and depth, that need to be tuned based on cross-validation or test data, as will be discussed in subsection 5.2.2.

5.1.2 Individual Anomaly Detection in Melt Pool Images by Clustering

The final learned representation vector can be fed to other methods of unsupervised or supervised learning for further analysis. As discussed before, manual annotation of melt pool images is time-consuming and expensive. A variation of clustering algorithms is employed to automatically annotate the melt pool images

as either normal or anomaly to tackle this challenge. Anomalies are usually defined as individuals or groups of samples isolated from normal samples by their distinguishable characteristics. Agglemoartive clustering tends to merge the samples into clusters in a hierarchical fashion. At the beginning of the process, each sample is a cluster of its own. The children clusters are then sequentially combined by the shortest cophenetic distance (linkage) into larger clusters until all elements end up being in the same parent cluster. An immediate intuition from this procedure is that normal samples tend to merge faster and with shorter cophenetic distances while appending the anomaly samples are postponed until the end of the process and when cophenetic distances between clusters are large. Furthermore, anomalies tend to form singleton clusters of their own, while normal samples are grouped together in clusters with large populations. These characteristics make the agglomerative clustering a perfect candidate for the anomaly detection, and thus automatic annotation of melt pool deep representation flattened vectors.

The definition of linkage is what differentiates between the different agglomerative clustering methods. The most widely used linkage functions include single, complete, average, and Ward. Denote the vectors of flattened encoded tensors by $\vec{h_i} \in \mathbb{R}^{1\times Q}$, for $i = 1, 2, \cdots, N$, where Q is the product of dimensions of the corresponding tensors. Then, the following linkage functions can be defined:

$$l_{single}(C_i, C_j) = min\{dist(\vec{h_i}, \vec{h_j}) \mid \vec{h_i} \in C_i; \vec{h_j} \in C_j\}, \tag{5.4}$$

$$l_{complete}(C_i, C_j) = max\{dist(\vec{h_i}, \vec{h_j}) \mid \vec{h_i} \in C_i; \vec{h_j} \in C_j\}, \tag{5.5}$$

$$l_{average}(C_i, C_j) = \sum_{\vec{h_i} \in C_i} \sum_{\vec{h_j} \in C_j} \frac{dist(\vec{h_i}, \vec{h_i})}{\mid C_i \mid\mid C_j \mid}, \tag{5.6}$$

$$l_{Ward}(C_i, C_j) = \sqrt{\frac{2 \mid C_i \mid\mid C_j \mid}{\mid C_i \mid + \mid C_j \mid}}.dist\left(\bar{C}_i, \bar{C}_j\right), \tag{5.7}$$

where $l_k(C_i, C_j)$ denotes the type k linkage between two cluster C_i and C_j. $\mid C_i \mid$

is the cardinality of cluster C_i. $\bar{C}_i$ is the centroid of cluster C_i and is computed as $1/ \mid C_i \mid \sum_{\vec{h_i} \in C_i} h_i$. The measure of proximity or similarity between two flattened encoded vectors $\vec{h_i}$ and $\vec{h_j}$ is denoted by $dist(\vec{h_i}, \vec{h_j})$. The proximity measure also varies in type and include cosine similarity, Manhattan or Euclidean distance, and correlation. For the sake of consistency in computations between different linkage types, Euclidean distance is used as the primary measure of proximity in this study, i.e. $dist(\vec{h_i}, \vec{h_j}) = \|\vec{h_i} - \vec{h_j}\|_2$. The steps of the agglomerative clustering algorithm are summarized in Algorithm 2. The outputs of the process are summarized by means of a visualization tool called dendrogram, where the hierarchical structure of parent and child clusters are demonstrated by u-shape links. The height of the links in the dendrogram determine the cophenetic distance between two clusters, which can be also used to decide for the best number of clusters and linkage types visually.

5.1.3 Statistical Process Monitoring for Anomaly Detection

As the last step, a multivariate profile monitoring scheme is deployed to perform control charting on the flattened encoded vectors. To this aim, the anomaly images detected by the discussed methods are labeled as out-of-control samples and are discarded out of the phase-I statistical process monitoring where the parameters are estimated, and control limits are built based on normal samples. The anomaly vectors are later employed to evaluate the out-of-control performance of the control charting method. The multivariate profile monitoring scheme consists of a Hotelling's T^2 control chart to monitor the deep representation vectors and a Shewhart control chart for monitoring error variance the of residuals ϵ denoted by σ^2. The first step toward designing the control chart method in phase-I analysis, involves estimating the mean vector, variance-covariance matrix and upper control limits. Denoting the $(N_1 - 1) \times Q$ successive difference matrix by V, it can be calculated as follows:

Algorithm 2: Hierarchical Agglomerative for Clustering Melt Pool Images

Input: The set of flattened encoded vectors $\{\vec{h_1}, \vec{h_2}, \cdots, \vec{h_N}\}$;

type k linkage function $l_k(C_i, C_j)$

for $i = 1, \cdots, N$ **do** // loop over the data

  $C_i = \{\vec{h_i}\}$; // each object is the cluster of its own

end for

$C = \{C_1, C_2, \cdots, C_N\}$; // super set includes all clusters

while $\mid C \mid > 1$ **do** // loop until super set has only one object

  $C_1^*, C_2^* \leftarrow \arg\min\{l_k(C_i, C_j) \mid C_i \in C; C_j \in C\}$; // pick best clusters to merge

  $C \leftarrow \{C \setminus C_1^*\} \setminus C_2^*$; // remove them from super set

  $C \leftarrow C \cup \{C_1^* \cup C_2^*\}$; // add their union to the super set

end while

$$V = \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_{N_1-1} \end{bmatrix}, \tag{5.8}$$

where $\vec{v_i} = \vec{h}_{i+1} - \vec{h_i}$, $i = 1, 2, \ldots, N_1 - 1$, and N_1 is total number of normal samples. Then, the variance-covariance matrix Σ and the mean μ are estimated as follows:

$$\vec{\mu} = \frac{\sum_{i=1}^{N_1} \vec{h_i}}{N_1}, \tag{5.9}$$

$$\Sigma = \frac{V^T V}{2(N_1 - 1)}. \tag{5.10}$$

The computational relationships for ith sample are defined as follows (Williams *et al.* (2007)):

$$T_i^2 = [\vec{h_i} - \vec{\mu}]^T \Sigma^{-1} [\vec{h_i} - \vec{\mu}], \qquad i = 1, 2, \ldots, N_1 \tag{5.11}$$

The upper control limit for the above statistics given a normal assumption, is approximated using $UCL = \chi^2_{\alpha,Q}$, where $\chi^2_{\alpha,Q}$ is the $100(1-\alpha)^{th}$ percentile of chi-squared distribution with Q degrees of freedom. However, as the normal assumption is usually violated in practice, the non-parametric kernel density estimator is fitted on the data to estimate the empirical probability density functions (Martin and Morris (1996); Fathizadan *et al.* (2017b)). Denote the Gaussian kernel function as $\Gamma(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$ and its bandwidth by b, then the estimated density at point T_i^2 is measured by:

$$\hat{f}(T_i^2; b) = \frac{1}{n} \sum_{j=1}^{N_1} \Gamma \left(\frac{T_i^2 - T_j^2}{b} \right), \tag{5.12}$$

where the optimal value for b is determined by a $K - fold$ cross validation that maximizes the log likelihood of training data. Denote the hold out set at the k^{th} $fold$ by S_k, then the optimal value for b is obtained as follows:

$$b = \arg\max \left(\frac{\sum_{k=1}^{K} \log \prod_{i \in S_k} \hat{f}(T_i^2; b)}{K} \right). \tag{5.13}$$

The $100(1-\alpha)^{th}$ percentile of the estimated probability density function determines the upper control limit of the control chart.

Note that the flattened encoded vectors store a low-dimensional representation that preserves only the main patterns in the original images. The residual variance of the process can be further investigated to monitor any potential unaccounted variability. The computational relationship for ith sample is calculated as follows:

$$S_i^2 = \frac{\vec{e}_i^T \vec{e}_i}{P - 1}, \qquad i = 1, 2, \ldots, N_1 \tag{5.14}$$

,where the $P \times 1$ vector of the estimated residuals is given by $\vec{e}_i = \vec{x}_i - \vec{\tilde{x}}_i$, and $\vec{\tilde{x}}_i$ is the flattened vector of i^{th} decoded image reconstructed from the bottleneck layer. Again, the nonparametric kernel density estimator estimates the empirical probability

density function of residual variance statistics. The $100(\alpha/2)^{th}$ and $100(1-\alpha/2)^{th}$ percentiles of the estimated probability density function determine the lower and upper control limit, and $\bar{S^2} = \sum_{i=1}^{N_1} S_i^2/N_1$ is the central control limit of the control chart, respectively. The two control charts are performing concurrently. Therefore, considering an overall false alarm probability of α, the false alarm probability of each individual control chart can be determined by $\alpha^{'} = 1 - \sqrt{1-\alpha}$.

5.2 Case Study and Results

5.2.1 Experiment Setting

An experiment is conducted on an open-architecture laser powder bed fusion system - the Additive Manufacturing Metrology Testbed (AMMT) at the National Institute of Standards and Technology (NIST). Figure 5.2 illustrates a schematic set up of the testbed with five major subsystems: a build platform, a powder recoating system, a laser positioning optical system, and two process monitoring modules (Yang *et al.* (2019a)). One of the process monitoring systems is a high-speed camera-based coaxial melt pool monitoring system. The system is designed to allow full control of laser scans as well as synchronized data acquisitions. It uses a real-time controller to set galvo position and laser power as well as trigger the acquisition of melt pool images (Lane *et al.* (2016)).

In this experiment, a total of 3×4 Inconel 625 parts are manufactured on a 100 mm $\times$100 mm build platform (Lane and Yeung (2019)). Each part has the same nominal 10 mm $\times$10 mm $\times$5 mm geometry but built with a unique scan strategy – a combination of various scan paths, laser power, and scan velocity. An in-house developed AM software is used to program the scan strategy into a series of position commands with a time interval of 10 μs (100 KHz). The high-speed coaxial melt pool

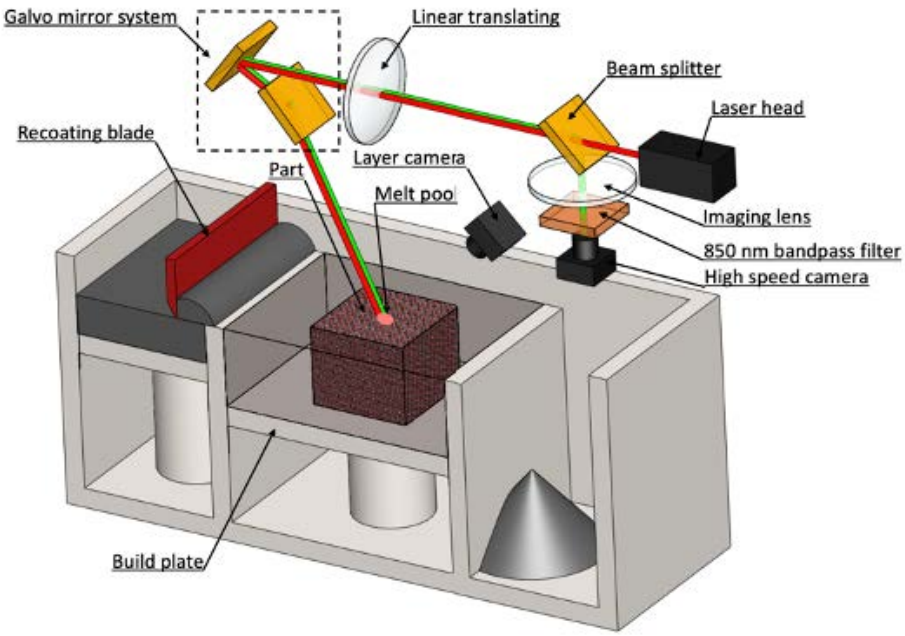


Figure 5.2: Schematic of the AMMT Experiment Setup (Yang *et al.* (2019a))

camera is triggered at 2 kHz (500 μs of sample interval). However, the monitoring system only captures melt pool images for one of the 12 parts at each layer due to the camera memory limit. Hence, every part has around 3000 melt pool images every other 12 layers.

In this study, part 8 is selected because of its richness of the melt pool characteristics. Figure 5.3 shows the scan strategy of this part, named "island spiral concentrating scan strategy". Each layer on the part is composed of four adjacent islands that are printed sequentially. The laser scans each island, spirally following the numbering order shown in the figure. The part has 250 layers, each with 20 μm thickness. The machine reduces the scan speed when the laser turns direction. The machine also reduces the laser power for scan speed to avoid high energy input. Moreover, to avoid high heat concentration and introduce island shape variance, the machine would rotate the centroid angle at each layer. The range of laser power and

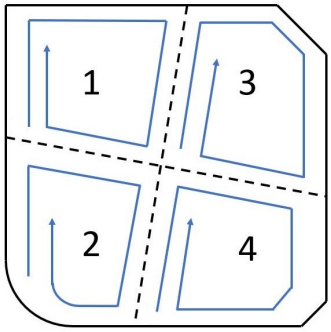


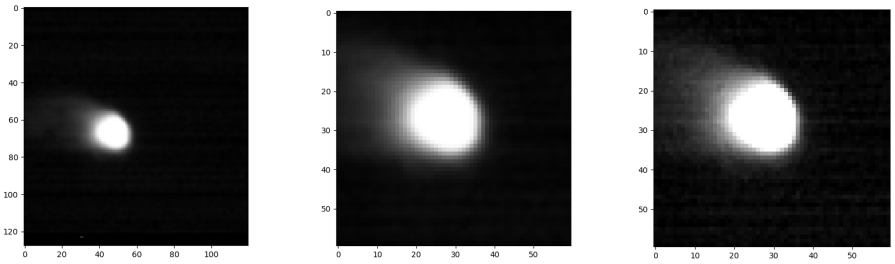
Figure 5.3: Schematic of the Experimental Build and the Scan Strategy

scan speed is from 0 to 234.83 W and 0 to 900 mm/s, respectively. Part 8 has melt images captured at layer 8, 20, and 32, etc. We chose the images from layer 8 for the deep representation learning and control chart method validation.

5.2.2 Deep Representation Learning of Melt Pool Images

The melt pool images from layer 8 of the process have been collected. The original tensor of the data-set has shape $3769 \times 128 \times 120$, which means there are 3766 total images each with a resolution of 128×120. An example of an original image is shown in Figure 6.6a. The original images are cropped to the biggest common frame that includes the heated area zone's information. The cropped images now have a resolution of 60×60, as shown in Figure 6.6c. As the last step toward image pre-processing and following the convolution operation in Equation 5.2, a $2D$ sharpening kernel matrix is slid twice on each image to represent the local features in each image. The example image after the sharpening process is illustrated in Figure 6.6b.

Next, the images are randomly split into training and testing data by a proportion of 75% and 25%, respectively. Thus a total of 2824 images are used for training the CAE, and the rest 942 images are utilized to validate the model and prevent potential overfitting. The images are also normalized to have pixel intensities in $[0, 1]$ interval



(a) Original Image (b) Cropped Image (c) Sharpened Image after $2D$ Convolution

Figure 5.4: Image Pre-processing on Melt Ppool Images

before feeding to the CAE. The proposed CAE architecture is comprised of 14 layers, out of which the first and last layers contain the input and output images, respectively. The summary of the model layers, excluding the input and output layers, is shown in Table 5.2. Layers 2 to 7 and layers 8 to 13 in the model form the encoder and decoder, and are represented as '*operation_e#*' and '*operation_d#*', respectively in the table. The compact deep representation from each image is stored in the feature map tensor from the encoder's last layer, i.e., '$MaxPool\_e6$'. This layer is called the bottleneck layer. The architecture of the resulting feature map tensors is illustrated in Figure 5.5. In the figure, the number of slices and the corresponding dimension for each slice in each layer is shown before and after @ sign, respectively. The deep representation tensor has a dimension of $8 \times 5 \times 5$.

The CAE model is compiled using the following settings: optimizer 'Adadelta' with a batch of size 32, binary cross-entropy as the loss function, and training on 200 epochs. Figure 6.7 demonstrates the learning trend for both train and test data resultant from the training process. The y-axis in the figure displays the loss function and x-axis is the number of epoches. Ten arbitrary learned encoded deep representation of melt pool images are obtained from the bottleneck layer, and each is reshaped to

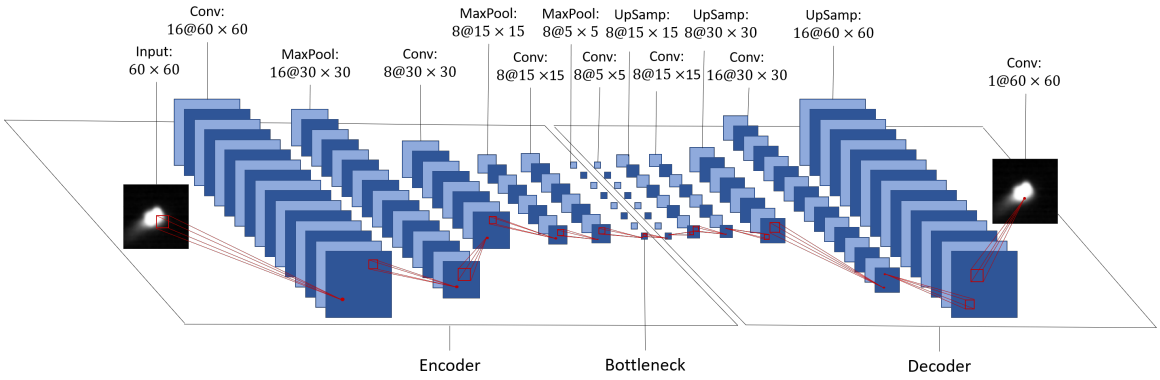


Figure 5.5: The Architecture of Input Images, Feature Map Matrices and Output Images

a matrix of shape 40×5 so that it could be displayed as an image. The third row in Figure 5.7 depicts these vectors. Moreover, the first and second row provides the corresponding original images, and decoded images reconstructed using information from the bottleneck layer, respectively. As shown by the figure, the encoded vectors are capable of downsampling the original image while preserving the melt pool's local characteristics. Furthermore, the encoded vectors provide a compact deep representation that can be fed to other methods of supervised or unsupervised learning models for further analysis. For this aim, the encoded tensors of the shape $8 \times 5 \times 5$ are flattened to form 200×1 vectors. Therefore the original data set is downsized to a total of 3766 samples, each with 200 features.

5.2.3 Individual Anomaly Detection by Clustering

This step aims to annotate the melt pool images as either normal or anomaly. The hierarchical-based agglomerative clustering with single linkage given by Algorithm 2 was performed on the flattened encoded vectors, and the resulting dendrogram is illustrated in Figure 6.9. The x-axis shows the samples, and the y-axis provides the

Table 5.2: Configuration of the Convolutional Auto-encoder

Type	Layer name	Operation	Number of filers	filter size	padding
Encoder	$conv2D\_e1$	$Convolution + ReLU$	16	3×3	1
	$MaxPool\_e2$	$Maxpooling$	1	2×2	0
	$conv2D\_e3$	$Convolution + ReLU$	8	3×3	1
	$MaxPool\_e4$	$Maxpooling$	1	2×2	0
	$conv2D\_e5$	$Convolution + ReLU$	8	3×3	1
	$MaxPool\_e6$	$Maxpooling$	1	3×3	0
Decoder	$conv2D\_d1$	$Convolution + ReLU$	8	3×3	1
	$UpSamp\_d2$	$UpSampling$	1	2×2	0
	$conv2D\_d3$	$Convolution + ReLU$	8	3×3	1
	$UpSamp\_d4$	$UpSampling$	1	2×2	0
	$conv2D\_d5$	$Convolution + ReLU$	16	3×3	1
	$UpSamp\_d6$	$UpSampling$	1	3×3	0

cophenetic distance to merge the clusters sequentially. The dendrogram's visual inspection reveals several parent clusters with their approximate partitionings specified by the vertical dashed lines in the figure. Investigating the images inside each cluster also reveals that they share common characteristics. For instance, the angle of the tail of the melt pools is the same inside each cluster. However, the first and the second clusters display quite different behaviors in the dendrogram. First, the number of samples in the first cluster is low. Second, the samples in this cluster tend to merge at a relatively large cophenetic distance compared to the rest of the samples in the data set, meaning they tend to form the singleton clusters of their own. Lastly, and on the contrary, the samples inside the second cluster are merged at a relatively small cophenetic distance, and their numbers are also large. These points suggest that the

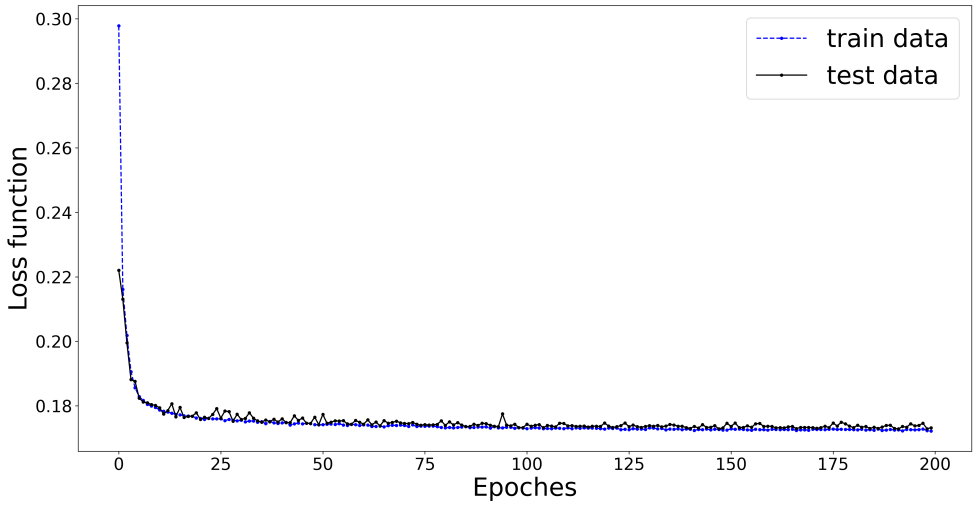


Figure 5.6: The Train and Test Learning Curves to Extract the Deep Representations from Melt Pool Images

data set can be partitioned into three major distinctive clusters: cluster C_1 composed of a few singleton clusters, cluster C_2, and cluster C_3 composed of remaining clusters.

Truncating the dendrogram at the cophenetic distance of 0.70, results in a total of 94 singleton clusters, two doubleton clusters, one tripleton cluster, one cluster with 530 samples, and one cluster with 3138 samples. One can confirm that the 97 singleton and near singleton clusters belong to C_1, the cluster with 530 samples is C_2, and the cluster with 3138 samples is, in fact, C_3. Three arbitrary images inside each of these clusters are selected and displayed in Figure 5.9. The left-hand side, middle, and right-hand side panel contain images in C_1, C_2, and C_3 clusters, respectively. From now on, clusters C_1, C_2, and C_3 will be referred to as "anomaly", "noisy", and "normal" clusters, respectively. This also allows for the construction of multi-variate control charting method as described before, after removing the samples inside anomaly and noisy clusters.

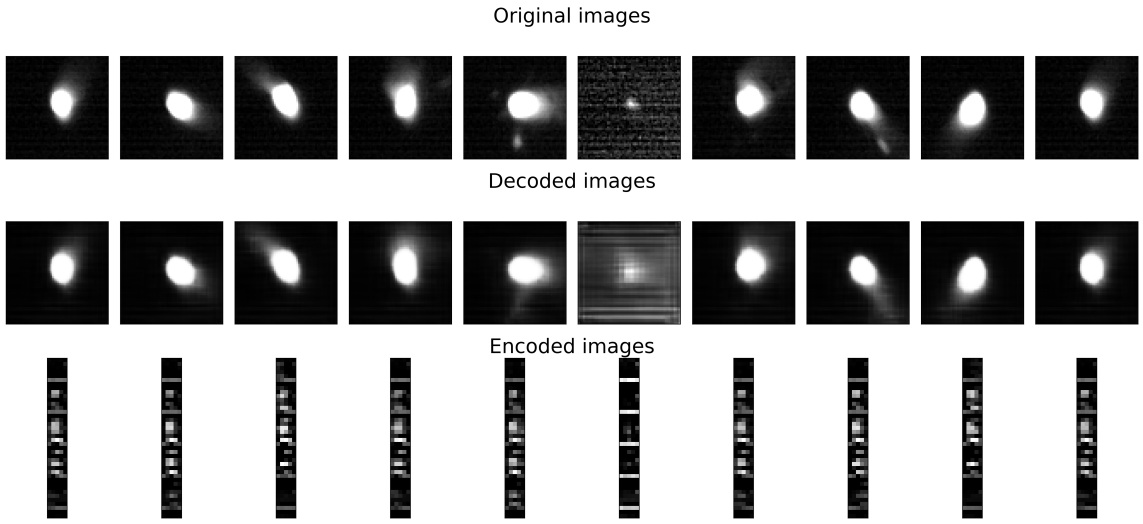


Figure 5.7: 10 Arbitrary Melt Pool Images

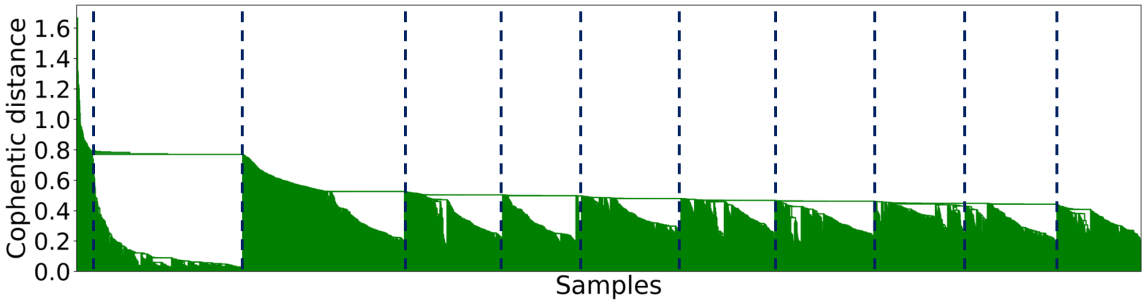


Figure 5.8: Dendrogram for Hierarchical-based Agglomerative Clustering with Single Linkage

5.2.4 Statistical Process Monitoring and Anomaly Detection Performance

Given a set of normal melt pool image obtained by the methodologies described in the last subsections, one can proceed to perform a control charting scheme to monitor the stability of the process and also detect the anomalies and out of control situations in real-time. To this aim, first, the normal data set is randomly split into training and testing sets by a proportion of 75% and 25%, respectively. Then, the same

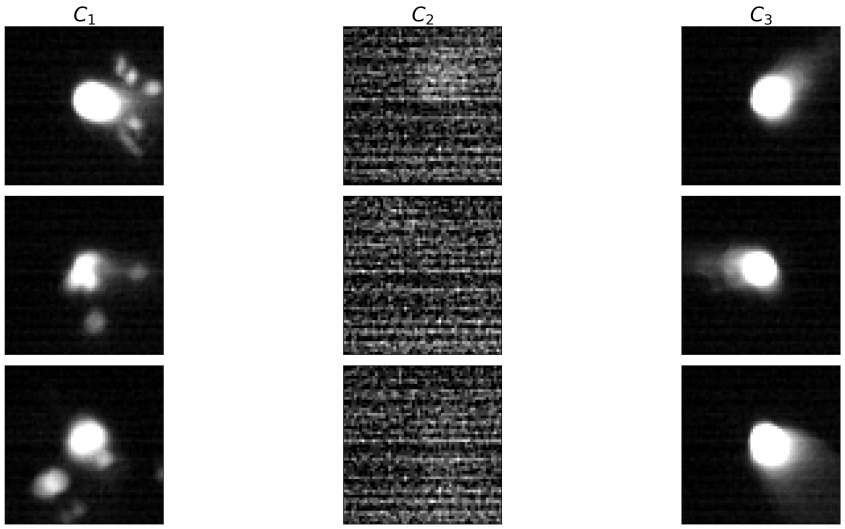


Figure 5.9: Sample Melt Pool Images Inside C_1:anomaly, C_2:noisy, and C_3:normal Clusters

configuration of CAE proposed in Table 5.2 is retrained and validated this time on the new training and testing data, respectively. Next, the flattened encoded tensors of melt pool images belonging to the training data are used to construct the phase-I control charting scheme proposed in subsection 5.1.3. In this phase and in regards to Hotelling's T^2 control chart, the mean vector μ and variance-covariance matrix Σ given by Equations 5.9 and 5.10 and the value of UCL are estimated all based on the training data. LCL, CL, and UCL values pertaining to residual variance monitoring are also estimated using the original and decoded training images.

In phase-II monitoring, the estimated parameters are used to calculate the statistics for monitoring data and decide if the process is either in control or out of control. Basically, a decision for out of control situation is made when either of T^2 or S^2 control charts issue an out of control alarm. This happens when either monitoring statistics exceeds their corresponding control limits. On the other hand, if both statistics fall

inside the control limits, the process can be considered in control. In order to acquire the monitoring data, the melt pool images in the noisy cluster are all discarded. The testing data and images in anomaly clusters are labeled as "normal" and "anomaly" data, respectively, and their encoded and decoded tensors are obtained by running them through the trained CAE. The former and the latter are then employed to evaluate the control charting method's performance in terms of triggering false alarms and true anomaly detection, respectively. The common practice to this aim is involved with forming a confusion matrix that stores the values for True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). In the context of the control chart monitoring, a sample can come under one of the following categories:

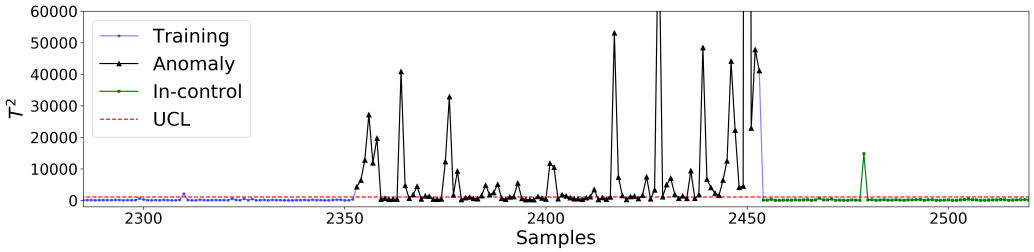
- TP: if it is anomaly and detected as out of control
- FP: if it is normal but detected as out of control
- TN: if it is normal and detected as in control
- FN: if it is anomaly but detected as in control

Using the elements of confusion matrix, a set of criteria are defined as follows to asses the quality of process monitoring:

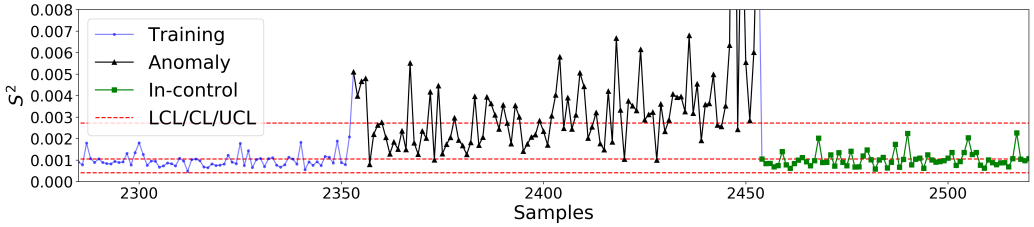
$$Accuracy = \frac{\sum(TP+TN)}{\sum(TP+FP+TN+FN)}, \quad Specificity = \frac{\sum(TN)}{\sum(TN+FP)},$$
$$Sensitivity = \frac{\sum(TP)}{\sum(TP+FN)}, \quad precision = \frac{\sum(TP)}{\sum(TP+FP)},$$
$$F_1 score = 2\frac{Sensitivity.Precision}{Sensitivity + Precision},$$

Table 5.3: Performance Measurement Criteria for Different Training Splits and Type-I Error Probability Of $\alpha^{'}$

Measurement	Type-I error probability of $\alpha^{'}$									
	0.0010	0.0064	0.0119	0.0173	0.0228	0.0282	0.0336	0.0391	0.0445	0.0500
Accuracy	93.10%	94.58%	95.16%	**95.40%**	**95.38%**	95.24%	95.07%	94.87%	94.67%	94.43%
	(0.0067)	(0.0040)	(0.0032)	**(0.0025)**	**(0.0035)**	(0.0061)	(0.0061)	(0.0060)	(0.0064)	(0.0069)
Specificity	**99.35%**	98.78%	98.30%	97.93%	97.47%	96.99%	96.53%	96.05%	95.63%	95.22%
	(0.0024)	(0.0034)	(0.0034)	(0.0050)	(0.0056)	(0.0076)	(0.0081)	(0.0081)	(0.0082)	(0.0081)
Sensitivity	47.55%	64.88%	73.79%	78.74%	82.10%	84.68%	86.76%	88.74%	90.22%	**91.31%**
	(0.0633)	(0.0473)	(0.0467)	(0.0287)	(0.0248)	(0.0301)	(0.0199)	(0.0194)	(0.0179)	**(0.0138)**
Precision	**92.93%**	90.00%	87.47%	85.74%	83.34%	81.02%	78.92%	76.87%	75.19%	73.57%
	(0.0322)	(0.0283)	(0.0220)	(0.0326)	(0.0329)	(0.0405)	(0.0391)	(0.0371)	(0.0361)	(0.0344)
F_1score	62.64%	75.24%	79.91%	81.99%	**82.63%**	**82.70%**	82.57%	82.30%	81.95%	81.44%
	(0.0567)	(0.0287)	(0.0234)	(0.0100)	**(0.0118)**	**(0.0216)**	(0.0190)	(0.0179)	(0.0190)	(0.0212)



(a) Hotteling's T^2 Control Chart to Monitor the Flattened Encoded Vectors



(b) S^2 Control Charts to Monitor the Variance of Residuals

Figure 5.10: Anomaly detection by statistical process monitoring on top of the convolutional auto-encoders

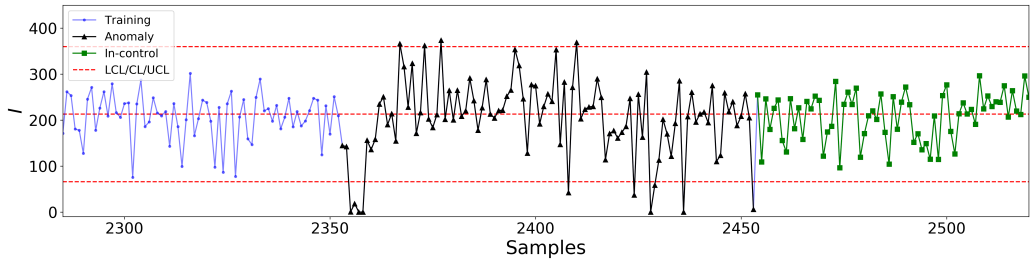
where *accuracy* measures the ratio of overall true classifications to the number of samples, $specificity$ measures the ratio of true classification of normal samples to the number of normal samples, *sensitivity* measures the ratio of true classification of anomalies to the number of anomalies, *precision* measures the ratio of true classification of anomalies to the number of classification of anomalies, and finally $F_1 score$ is the harmonic mean of *sensitivity* and *precision*.

As it was discussed by Section 5.1.3, the $100(1-\alpha)^{th}$ percentile of the estimated probability density function for measured statistics will be used to determine the control limits. This means there is a 100α probability that the statistics go above the control limits and get detected as out-of-control (rejected) while they are in fact in control (true null hypothesis). The larger the α, the smaller the control limits and thus more alarms will be issued by the charts and vice versa. This will also impact

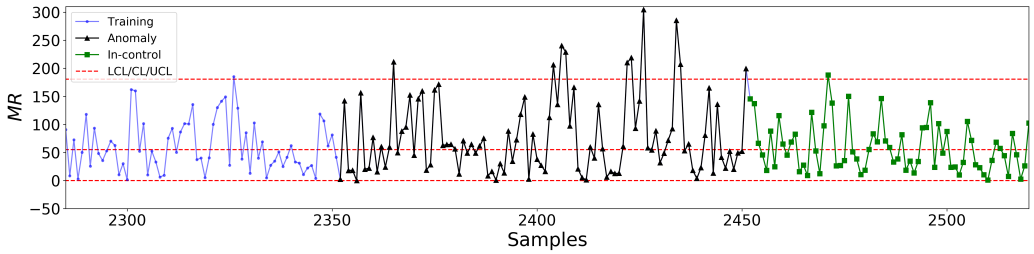
the out-of-control detection power of control charts that is represented by $1-\beta$ error, where β is the probability of accepting a sample while it is out-of-control. It is common to choose the type-I error as a number between 0.001 to 0.05, which means the corresponding control charts will issue false alarm warnings 0.1% and 5% of the time. 10 equally spaced points inside this interval are chosen to test and find the best control limits and therefore the best trade-off between chart's false alarm rate and out-of-control detection power. By selecting 10 different random proportions of the data from the normal cluster as training data for each of the control chart configuration, 100 trials were conducted. Table 5.3 provides the results for performance measurements criteria. The standard deviation of results is given in parenthesis below each entry. The highest scores for each measurement have been emboldened in the table. All the measurements are either monotonically increasing or decreasing with respect to $\alpha^{'}$, except for *accuracy* and $F_1 score$ that reach their maximum at $\alpha^{'} = 0.0228$. On rare occasions where the true classification of normal samples is preferable to the practitioner, the high score of $Specificity$ can be used to select the $\alpha^{'}$. Figure 5.10 illustrates a slice of phase-II T^2/S^2 control charting scheme constructed based on $\alpha^{'} = 0.0228$. It includes the entire samples from anomalies and covers a portion of training and test data. The data are sorted according to their labels to provide a better visual intuition into the control charting method's performance regarding the true classification of samples.

5.2.5 Comparison with Hand-crafted Features

As it can be observed by the right-hand side images in Figure 5.9, the deep representation of melt pool images allows for the detection of eccentric types of anomalies that are not detectable by the handcrafted feature engineer methods. For example, two melt pool images might encompass the same amount of bright pixels or share the



(a) I Control Chart to Monitor the Melt Pool Areas



(b) MR Control Charts to Monitor the Moving Ranges

Figure 5.11: Anomaly Detection by Statistical Process Monitoring on Top of the Melt Pool Areas

same melt pool area, but the splashing effect could still distinguish them from each other. The NBEM model (Yang *et al.* (2020b)) described briefly in section 2 is an example study that uses the area of fitted ellipses on the thresholded images of melt pools as the feature of interest. It should be noted that this work and NBEM study both share the same experiment setting and data set of melt pool images. In order to extract the features according to Yang *et al.* (2020b), the threshold value of 150 is chosen to segment the melt pools from surrounding areas.

Similar to the procedure described in subsection 5.2.4, normal melt pool images are employed to construct the phase-I control charting and estimate the parameters. However, in this case, a univariate monitoring scheme is deployed. It is composed of a I and a MR chart to monitor the individual observations and their moving range, respectively, (Montgomery (2007)). The individual observations represented by x_i

Table 5.4: Performance Measurement Criteria Comparison Between NBEM and Proposed Method for Different Training Splits

Model	Accuracy	Specificity	Sensitivity	Precision	$F_1 score$
NBEM	89.20%	98.38%	17.89%	59.77%	27.32%
	(0.0045)	(0.0052)	(0.0029)	(0.0798)	(0.0079)
Deep representation learning	95.38%	97.47%	82.10%	83.34%	82.63%
	(0.0035)	(0.0056)	(0.0248)	(0.0329)	(0.0118)

are simply the measured melt pool areas, and their moving range can be calculated as $MR_i = |x_i - x_{i-1}|$. The UCL, CL, and LCL for the I chart to monitor x_is is obtained as follows:

$$\begin{aligned} UCL &= \overline{x} + 3\frac{\overline{MR}}{d_2} \\ CL &= \overline{x} \\ LCL &= \overline{x} - 3\frac{\overline{MR}}{d_2}, \end{aligned} \tag{5.15}$$

where $\overline{x} = \frac{\sum_{i=1}^{N_1} x_i}{N_1}$, $\overline{MR} = \frac{\sum_{i=2}^{N_1} MR_i}{N_1 - 1}$, and d_2 is a predetermined constant equal to 1.128. Moreover, the UCL, CL, and LCL for the MR chart can be obtained as follows:

$$\begin{aligned} UCL &= D_4\overline{MR} \\ CL &= \overline{MR} \\ LCL &= D_3\overline{MR}, \end{aligned} \tag{5.16}$$

where D_3 and D_4 are predetermined constants equal to 0 and 3.267, respectively. Figure 5.11 illustrates the phase-II $I - MR$ control charting scheme constructed based on the above control limits. It includes the entire samples from anomalies and

covers the same portion of training and test data as Figure 5.10. Similarly, the data are sorted according to their label to provide better visualization. As it is obvious by the comparison between the two figures, the control charting scheme based on the proposed deep representation learning approach outperforms the method based on melt pool areas in terms of anomaly detection. The set of performance measurement criteria was calculated for 100 trials, and the results are provided in the first row of Table 5.4. The second row in the table provides the results based on $\alpha^{'} = 0.0228$ for the proposed deep representation learning method. The low values for performance measurement criteria, especially *sensitivity*, *precision*, and $F_1 score$ compared to the proposed method, prove that simple characterization of melt pools by their area results in losing valuable information vital to maintaining a quality process monitoring and control.

Chapter 6

DEEP SPATIO-TEMPORAL ANOMALY DETECTION IN LASER POWDER BED FUSION

6.1 Methodology

The proposed methodology adapts the framework in Chapter 5 for spatio-temporal anomaly detection using sequences of melt pool images and consists of the following three primary components:

1. Following a sliding windows sampling strategy, the $3D$ stream of pre-processed melt pool images is reshaped into samples of sequential data. The resulting $4D$ tensor is then fed to an auto-encoder model to extract the deep spatio-temporal bottleneck features from each sequence.

2. Next, the bottleneck tensors are unfolded into $1D$ vectors and deployed by an agglomerative clustering algorithm. By relying on the structural similarity of bottleneck vectors, each sequence is annotated as either a normal or anomaly sample.

3. The $1D$ unfolded bottleneck vector are further downsized to scalars and deployed by a multivariate statistical control charting scheme. The control charting method provides an intuitive visualization tool for real-time monitoring of the process's stability and detecting anomalous spatio-temporal events. The phase-I of control charting scheme employs random portions of normal unfolded bottleneck vectors to estimate the in-control baseline for phase-II monitoring.

The overall performance of the framework is evaluated in phase-II by benchmarking the accuracy of anomaly detection and false alarm rate.

The following subsections strive to shed light on the underlying structure of the above components and their interconnection in further detail.

6.1.1 Deep Spatio-temporal Representation Learning for Feature Extraction from Melt Pool Sequence

A co-axial optical camera captures gray-scale melt pool images in time sequence. Denoting the total number of frames by m and assuming a sliding time window of length l, the sequences of melt pool images can be represented by the tensor $\mathcal{S} \in \mathbb{R}^{(m-l+1)\times l\times n_1\times n_2}$, where $n_1 \times n_2$ is the dimension of each image. Given the current data structure, a high-quality feature extraction method must seize the temporal correlation between time-indexed samples as well as spatial correlation within each image. Recurrent neural networks (RNN) have been extensively used to model and preserve the temporal correlation between samples in sequential and time series data (Sutskever *et al.* (2014)). As opposed to general-purpose neural networks, these models are trained to minimize the error backpropagation through time. However, the accumulation of near-zero derivatives through time causes the vanilla RNN models to suffer from vanishing gradient problem and lose the long-range temporal interdependencies between time-indexed samples (Pascanu *et al.* (2013)). As a remedy, Long Short Term Memory (LSTM) networks are introduced to tackle this issue by the inclusion of memory states in cells. The memory states are equipped with two gates for a control mechanism that allows the past information to be confined or forgotten within the current cell. In order to avoid the loss of spatial information, we use Convolutional LSTM neural networks (ConvLSTM) as opposed to fully connected LSTM where the input data are readily unfolded and fed as $1D$ vectors (Shi *et al.* (2015)).

Figure 6.1 lays out the structural schematic of one ConvLSTM cell for a sample at time t. Let's denote the corresponding cell's memory state and hidden state by $3D$ tensors $\mathcal{M}_t$ and $\mathcal{H}_t$, respectively, and its input sample by the $3D$ tensor $\mathcal{X}_t \in \mathbb{R}^{fn_1 \times fn_2 \times fn_3}$, where $fn_1 \times fn_2 \times fn_3$ is the dimension of feature map after performing convolutional operation on each original image. The hidden state $\mathcal{H}_t$, stores the spatio-temporal features of the whole sequence up to the time t. Moreover, the input, forget, and output $3D$ gate tensors are given by $\mathcal{I}_t$, $\mathcal{F}_t$, and $\mathcal{O}_t$, respectively. As it can be seen by the lower left corner of Figure 6.1, convolutional operations are performed by sliding kernel patches on the current input image as well as the previous hidden state to extract the local features and preserve the spatial correlation within the cells through time. The relationships between cell's input, states, and gates at time t are defined as follows:

$$\mathcal{F}_t = \sigma\left(\mathcal{W}_{xf} * \mathcal{X}_t + \mathcal{W}_{hf} * \mathcal{H}_{t-1} + \mathcal{W}_{mf} \odot \mathcal{M}_{t-1} + B_f\right), \tag{6.1}$$

$$\mathcal{I}_t = \sigma\left(\mathcal{W}_{xi} * \mathcal{X}_t + \mathcal{W}_{hi} * \mathcal{H}_{t-1} + \mathcal{W}_{mi} \odot \mathcal{M}_{t-1} + B_i\right), \tag{6.2}$$

$$\mathcal{O}_t = \sigma\left(\mathcal{W}_{xo} * \mathcal{X}_t + \mathcal{W}_{ho} * \mathcal{H}_{t-1} + \mathcal{W}_{mo} \odot \mathcal{M}_t + B_o\right), \tag{6.3}$$

$$\mathcal{M}_t = \mathcal{F}_t \odot \mathcal{M}_{t-1} + \mathcal{I}_t \odot tanh\left(\mathcal{W}_{xm} * \mathcal{X}_t + \mathcal{W}_{hm} * \mathcal{H}_{t-1} + B_m\right), \tag{6.4}$$

$$\mathcal{H}_t = \mathcal{O}_t \odot tanh\left(\mathcal{M}_t\right), \tag{6.5}$$

where $*$ and $\odot$ represent convolution and Hadamard product operations, respectively. Moreover, σ and $tanh$ are the element-wise sigmoid and hyperbolic tangent activation functions given as $\sigma(u) = 1/(1+e^{-u})$ and $tanh(u) = (e^u - e^{-u})/(e^u + e^{-u})$ to map the values between $[0, 1]$ and $[-1, 1]$, respectively. The training of the model includes estimating the $4D$ tensors $\mathcal{W}$ and matrices B. The forget and input gates in Equation (6.1) and (6.2) determine the amount of information from current input and previous hidden and memory states to be erased and contributed, respectively. The cell's memory state is updated by Equation (6.4). The output gate is obtained

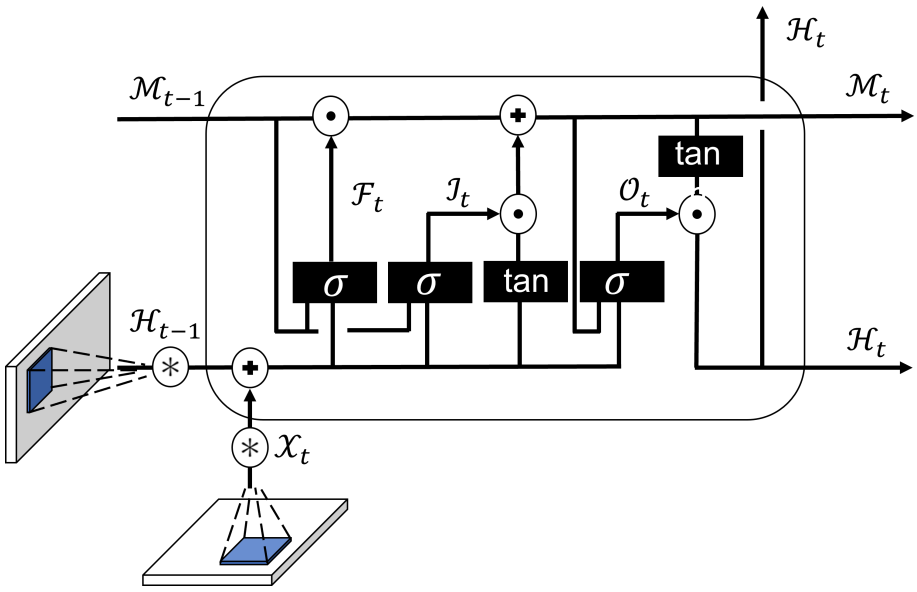


Figure 6.1: Structural Schematic of a ConvLSTM Cell for Sample at Time t

by the tensor manipulation of current input and memory as well as previous hidden state as given by Equation (6.3). Finally, the spatio-temporal feature of the cell based on the output gate and memory state can be extracted by Equation (6.5). The tensor dimension of weights is determined based on the dimension of input data, and number of kernels and their filter size. Moreover, the weights are trained in a similar manner to RNN using backpropagation through time. Considering a sequence to sequence training paradigm and a single ConvLSTM unit, if we denote the standardized output sample at time t by $3D$ tensor $\mathcal{Y}_t$, then the cross-entropy loss function to minimize for each sequence sample of size l is defined as: Loss(X, Y; W, B) = $\sum_{t=1}^{l} \sum_{All\ Elements} -\mathcal{Y}_t \odot log(ReLU(\mathcal{W}_{hq} * \mathcal{H}_t + B_q))$, where log and $ReLU$ are element-wise logarithmic and Rectified Linear Unit operators, respectively. Stochastic batch gradient descent is used to update the weights in each error propagation step where the value of partial derivatives of loss with respect to each weight is computed by the chain rule for derivatives. For instance, given a non-negative learning rate of γ, W_{xo}

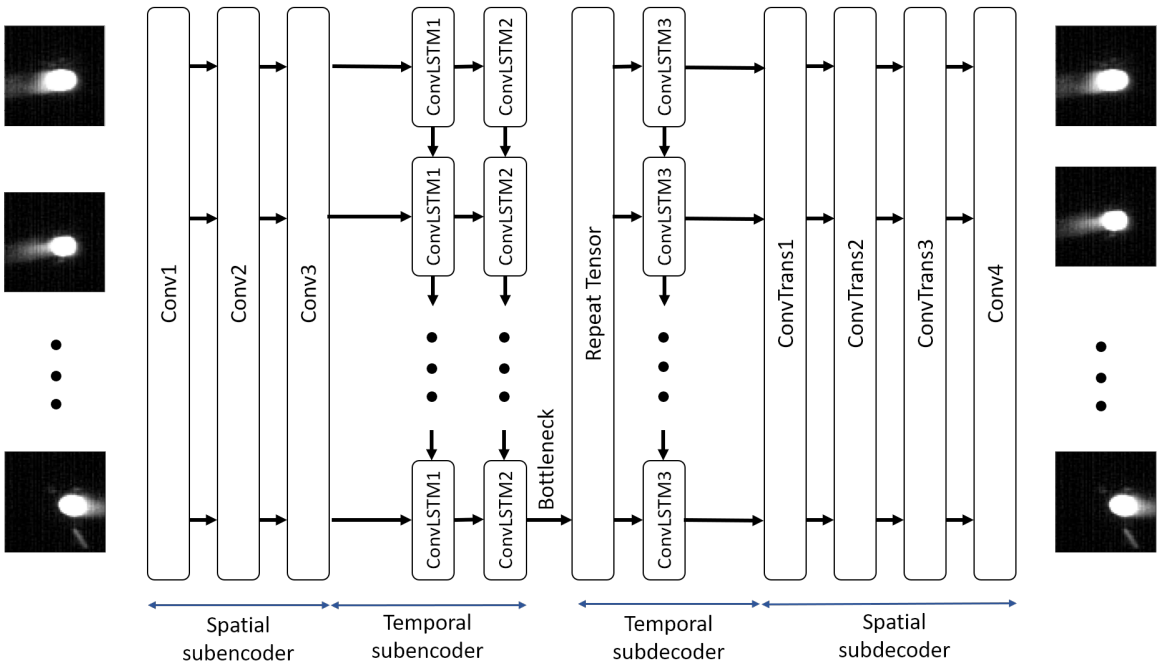


Figure 6.2: Proposed Spatio-temporal Auto-encoder Model Architecture

in Equation (6.3) can be updated as follows:

$$\mathcal{W}_{xo}^{\tau+1} = \mathcal{W}_{xo}^{\tau} - \gamma \frac{\partial Loss(\mathcal{X}, \mathcal{Y}; W, B)}{\partial \mathcal{W}_{xo}^{\tau}} \quad \tau = 0, 1, \cdots \tag{6.6}$$

$$= \mathcal{W}_{xo}^{\tau} - \gamma \sum_{t=1}^{l} \frac{\partial Loss(\mathcal{X}, \mathcal{Y}; W, B)}{\partial \mathcal{H}_t} \frac{\partial \mathcal{H}_t}{\partial \mathcal{O}_t} \frac{\partial \mathcal{O}_t}{\partial \mathcal{W}_{xo}} \quad \tau = 0, 1, \cdots \tag{6.7}$$

To learn a compact spatio-temporal representation from the unlabeled sequences of melt pool images, we develop a spatio-temporal auto-encoder architecture by feeding the same sequence of input data as the output to the model. The encoder f and decoder g component of the auto-encoder are then designed and trained to learn a mapping from input data X to a compact spatio-temporal latent space H, and from H to a reconstructed version of original input data $\tilde{X}$. These mappings can be summarized by $f : X \rightarrow H$ and $g : H \rightarrow \tilde{X}$ for encoder and decoder, respectively. The overall architecture of the proposed model is shown by Figure 6.2. Specifically, the encoder component is composed of two subencoders: 1) a spatial subencoder

and 2) a temporal subencoder stacked on top of the spatial subencoder. The spatial subencoder learns the $3D$ spatial feature tensors from each original $2D$ melt pool image by deep convolutional layers. Next, the resulting tensors are sequentially fed to the ConvLSTM layers of the temporal subencoder to learn the temporal correlation between time-indexed samples while preserving the spatial features. The last cell's hidden state of the final ConvLSTM layer in the temporal subencoders stores the deep spatio-temporal bottleneck tensor for each sequence. These tensors are represented by $\mathcal{H}_i^{st} \in \mathbb{R}^{b_1 \times b_2 \times b_3}$ for $i = 1, 2, \cdots, m-l+1$, where b_1, b_2, and b_3 are custom dimensions that their size are determined by the layers' characteristics and model architecture. The repeat vector layer simply replicates the bottleneck tensor l times to provide a suitable sequence of length l for the decoder. Mirroring the encoder, the decoder component also consists of a spatial and a temporal subdecoder. The temporal subdecoder is placed right on top of the bottleneck layer, and the spatial decoder is stacked afterward to reconstruct the original image sequence using deep convolutional transpose layers. Refer to Section 6.2.2 for greater detail into the design characteristics of each layer and training parameters of the proposed model configuration.

6.1.2 Spatio-temporal Anomaly Detection in Melt Pool Image Sequence by Clustering

The trained bottleneck tensors from previous section are deployed here to perform the data annotation and anomaly detection procedures. As discussed before, the manual annotation of high-frequency streams of melt pool images contaminated with scarce anomalous data is a prohibitive process. Clustering algorithms, on the other hand, rely on the structural similarity of high-quality but low-dimensional feature vectors to automatically extract the normal and anomalous labels for original data. Refer to Chapter 5 Section 5.1.2 for details regarding the agglomerative cluster-

ing algorithm. In the previous chapter we introduced various linkage functions and used the single linkage to perform anomaly detection. In this chapter, however, the anomaly melt pools of interest are temporally correlated and occur in a short amount of time right after each other in groups of samples. Therefore, the single linkage can no longer detects them. The average linkage function relies on distances of more than one sample to join child clusters C_i and C_j. The cophenetic distance measured by this linkage is defined as follows:

$$l_{average}(C_i, C_j) = \sum_{\vec{h_i} \in C_i} \sum_{\vec{h_j} \in C_j} \frac{dist(\vec{h_i}, \vec{h_i})}{\mid C_i \mid\mid C_j \mid}. \tag{6.8}$$

In Equation (6.8), $\vec{h_i} \in \mathbb{R}^{1 \times Q}$, for $i = 1, 2, \cdots, m - l + 1$, denote the vectors of flattened encoded tensors extracted from bottleneck tensors $\mathcal{H}_i^{st}$ and Q is the product of dimensions of the bottleneck tensor which is equal to $b_1 \times b_2 \times b_3$.

6.1.3 Statistical Process Monitoring for Anomaly Detection

The normal samples retained by the clustering method are deployed in this section to construct the phase-I multivariate profile monitoring scheme. Phase-I consists of estimating the unknown parameters to determine the baseline control limits. The annotated anomaly sequence samples are held out as validation data set to evaluate the out-of-control performance of the control charting when fine-tuning the hyperparameters of the model. Analogous to Chapter 5 Section 5.1.3, a Hotelling's T^2 control chart is used to monitor the flattened encoded vectors of the deep representation, while a Shewhart control chart is used to monitor the variance of residuals ϵ denoted by σ^2. It should be noted that the vector of i^{th} sample's estimated residual $\vec{e_i} \in \mathbb{R}^{1 \times P}$ is given by taking the difference $\vec{x_i} - \vec{\tilde{x}}_i$, where $\vec{x_i}$ and $\vec{\tilde{x}}_i$ are the sample's sequence corresponding unfolded original and decoded tensors with $P = l \times n_1 \times n_2$.

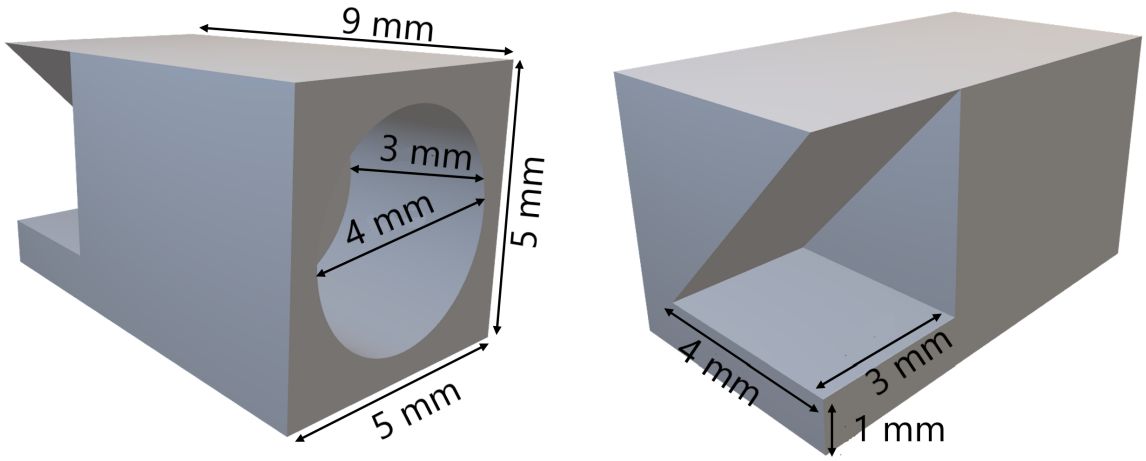


Figure 6.3: 3D Model of the Experimental Specimen

6.2 Case Study and Results

6.2.1 Experiment Setting

The Additive Manufacturing Metrology Testbed (AMMT) at the National Institute of Standards and Technology (NIST) is the primary platform developed to study powder bed fusion processes. AMMT is a fully customized metrology instrument that enables flexible control and measurement of the L-PBF process (Lane *et al.* (2016)) that equips a high-speed co-axial camera to capture an in-situ melt pool images every 100 μs. Similar to the typical commercial L-PBF machines, AMMT uses galvo scanner mirrors to control the laser path. It also utilizes a custom-designed dynamic translating lens for laser flat-field correction, instead of an f-theta lens. This custom scanning lens is also optimized to enable diffraction-limited image quality for the melt pool monitoring camera. Using custom-developed scan strategy design software, the machine can realize precise laser control for beam power and position at 100 KHz frequency (Yeung *et al.* (2018)). The melt pool monitoring images are $120\,pixels \times 120\,pixels$ with a resolution of 8 $\mu m/pixel$.

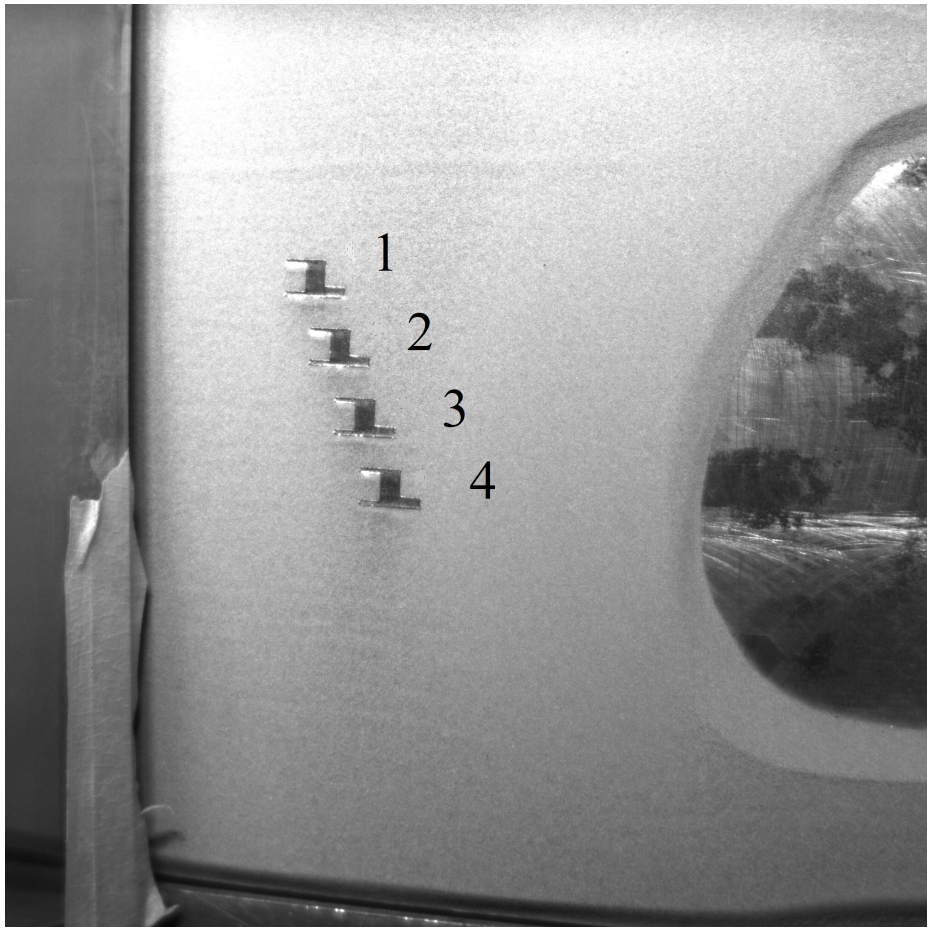


Figure 6.4: Part Placement in the Building Chamber

The experiment for this study creates four specimens on a wrought nickel alloy 625 ($IN625$) substrate with identical geometry and size. Figure 6.3 demonstrates the geometrical features of the experimental specimen, including the bounding box of size $5mm \times 9mm \times 5mm$, the 45^o overhang feature, and the cylinder cavity. The powder material is $IN625$ that is a mixture of virgin and recycled powder. 250 layers at 20 μm thickness per layer construct the finished part. Figure 6.4 captures the part layout in the build chamber, where the numbers mark the four specimens. The laser power and scan pattern for layer 1 of a specimen is illustrated in Figure 6.5. The build employs constant laser power (195 W) and scan speed (800 mm/s) for the infilling

area. All layers use stripe scan pattern with skywriting (Lane and Yeung (2020)). This strategy allows the laser beam to maintain a constant scan speed in the infilling area. The controller turns off the laser at the end of each track immediately to avoid over-melting. The speed reduction can only happen on the over-shooting area, which provides steady energy density to the entire part. This experiment uses a pre-contour path to melt the outline of each layer by a lower laser power of 100 W. To minimize the heat accumulation effect, the experiment deploys a 90^o rotation strategy. In other words, the laser beam has a different starting point and scan direction between layers. Refer to Lane and Yeung (2020) for more detailed information regarding the experiment settings.

6.2.2 Deep Representation Learning of Melt Pool Image Sequence

The video streams of melt pool images of layer 60 for the first 2 specimens have been collected. After removing the noisy images, the total number of collected frames for each specimen is equal to 3167, and each melt pool image has a resolution of $120\,pixels \times 120\,pixels$ as described in Subsection 6.2.1. A sliding time window of length 10 divides the collected data for each specimen into sequential samples, each containing 10 consecutive melt pool images. After the sequential resampling, the initial data set is a tensor of size $3158 \times 10 \times 120 \times 120$ for each specimen. An example of a raw melt pool image is shown in Figure 6.6a. First, a $2D$ sharpening kernel matrix is slid twice on each image to highlight the underlying spatial features further, as shown by Figure 6.6b. To discard the unused pitch-black regions surrounding the melt pools, the sharpened images are cropped to their largest common frame which encompasses the heated area zone. The resultant 60×60 example image after cropping that serves as the final image for our analysis is shown in Figure 6.6c. The tensors of two specimens are concatenated together to form a $6316 \times 10 \times 60 \times 60$

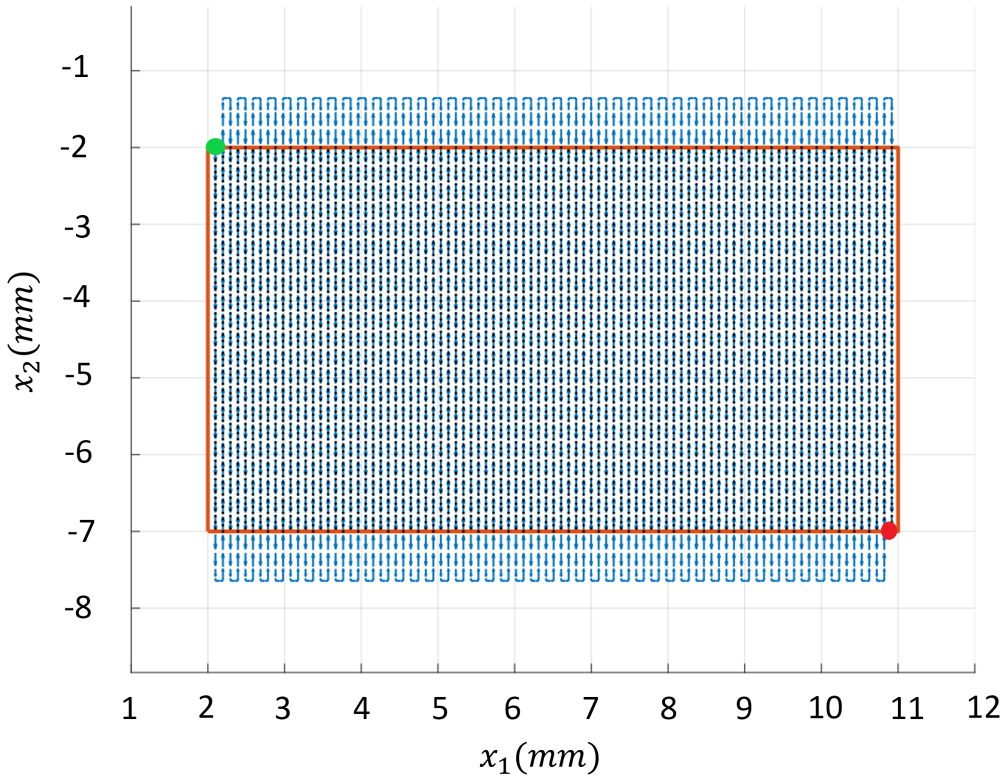
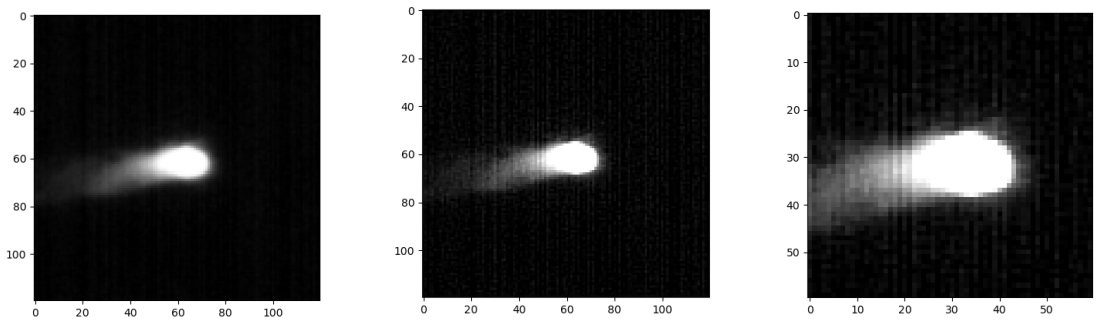


Figure 6.5: Scan Pattern for the Layer 1 of the Experiment (Yang *et al.* (2021)). The orange curve is outline of the part of this layer created during the pre-contour, which is scanned with a lower laser power of 100 W. Green and red dots represent the start and finish position, respectively. The dashed arrow indicates the laser moving direction. Note, the laser beam does have an active path outside the part outline as the laser power is set to zero.

tensor. The pixel intensities of images are normalized in $[0, 1]$ interval to stabilize the model training and prediction behavior. Moreover, to keep an eye on the model's performance against unseen data and to avoid potential overfitting in training, the input data set is randomly split into training and validation segments by a proportion of 75% and 25%, respectively. Therefore, a total of 4737 melt pool image sequences take part in training the proposed auto-encoder while the rest 1579 sequences are



(a) Raw Image

(b) Processed Image after Sharpening with $2D$ Convolution

(c) Cropped Sharpened Image

Figure 6.6: Image Pre-processing on Melt Pool Images

used for validation. The architecture of the proposed spatio-temporal auto-encoder for feature extraction from melt pool sequences was first introduced and shown by Section 6.1.1 and Figure 6.2, respectively. Table 6.1 summarizes the design characteristics for each layer of the model, including operation, size and number of filters, activation function, and stride parameter. Moreover, the dimensions of the feature map tensors are shown in the last column of the table. The spatial subencoder is composed of three stacking convolutional layers. In addition to convolution operations, these layers also downsample the slices of their input tensors by a scale of value equal to the stride parameter. The stride parameter determines the number of patch-to-patch pixels to skip when convolutional filters are slid throughout the image. As the next step in model architecture, the sequential output tensors of spatial subencoder are subsequently fed into two ConvLSTM layers of temporal subencoder. The $ConvLSTM2$ can be seen as the bottleneck layer of the model architecture, in which its final recurrent cell stores the bottleneck tensor $\mathcal{H}^{st}$. As can be seen by the table, the bottleneck tensor is a compact deep spatio-temporal representation extracted from each sequence of melt pool image samples and has a dimension of $5 \times 5 \times 64$.

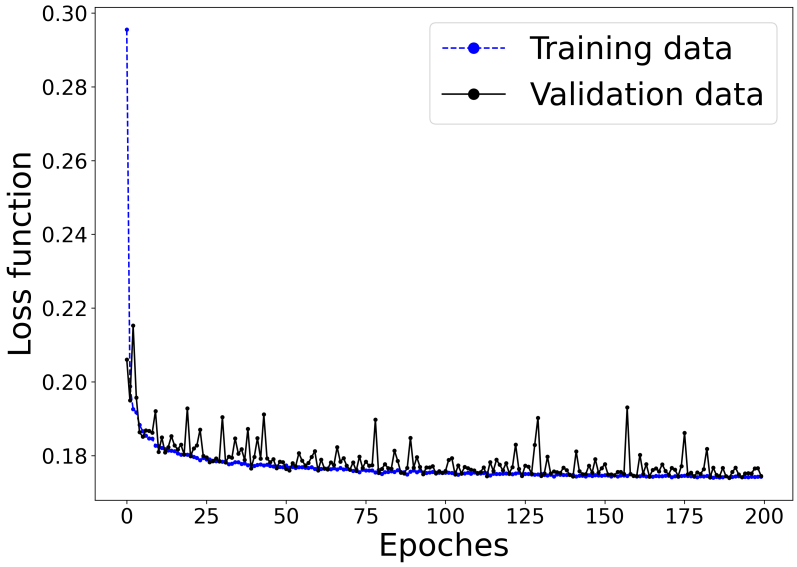


Figure 6.7: The Train and Validation Learning Curves for Spatio-temporal Auto-encoder

This is equivalent to a dimension reduction ratio of about 95% in the input sequence.

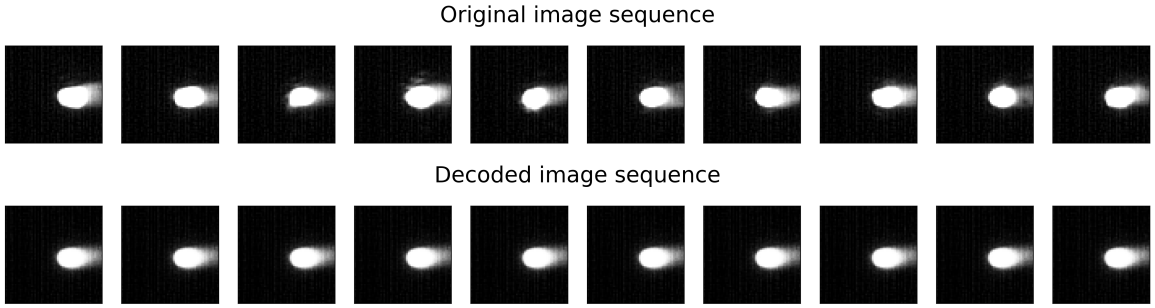
The configured spatio-temporal auto-encoder is compiled and trained with cross-entropy loss function introduced in Equation (6.7) and 'Adadelta' with a batch size of 128 samples as the optimizer engine. Figure 6.7 demonstrates the learning trend for both training and validation data after 200 epochs. The y-axis in the figure represents the loss function, and the x-axis shows the number of epochs. After some manual investigations into the validation data, three samples from different melt pool image sequence categories are selected and fed as input to the trained model. Figure 6.8a, 6.8b, and 6.8c illustrates those representative samples from "normal", "normal with individual melt pool anomalies", and "abnormal" categories, respectively. The first row in each subplot shows the original sequence with its images sorted from left to right according to their chronological appearance. Moreover, the second rows have the corresponding reconstructed samples based on decoding the information from

encoded bottleneck tensors. As can be seen by the figures, the encoded bottleneck tensors are capable of downsampling the original image sequence while preserving the within spatial characteristics and between temporal correlation between melt pools.

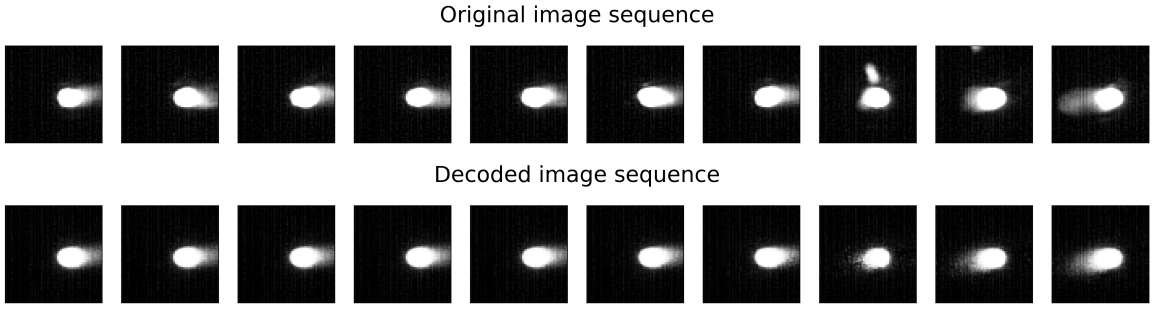
A further investigation of the figures also reveals several important insights into the performance of the deep spatio-temporal auto-encoder and patterns captured by the bottleneck tensors. First, the spattered lump on top of the main melt pool shown in the third from the last image of the original sequence in Figure 6.8b has been deducted in the corresponding decoded image in the second row. This means the encoding-decoding procedure, and thus the extracted representations are robust against individual melt pool anomalies. This is a major improvement over the existing methods that rely on single non-temporal melt pool anomalies to detect out-of-control events in the process. On the other hand, in Figure 6.8c, the tiny twin melt pool to the left of the main pool is consistently present across the original images and therefore has been captured by the reconstructed sequence. These points imply that the bottleneck tensors obtained from the original sequences in Figure 6.8a and 6.8b have a similar structure while the one captured by Figure 6.8c is substantially dissimilar to those. The anomaly detection procedure shown in the next sections will rely on this behavior of the bottleneck tensors to accurately and with low false alarm rates find the out-of-control spatio-temporal events in the process.

6.2.3 Anomaly Detection and Data Annotation by Clustering

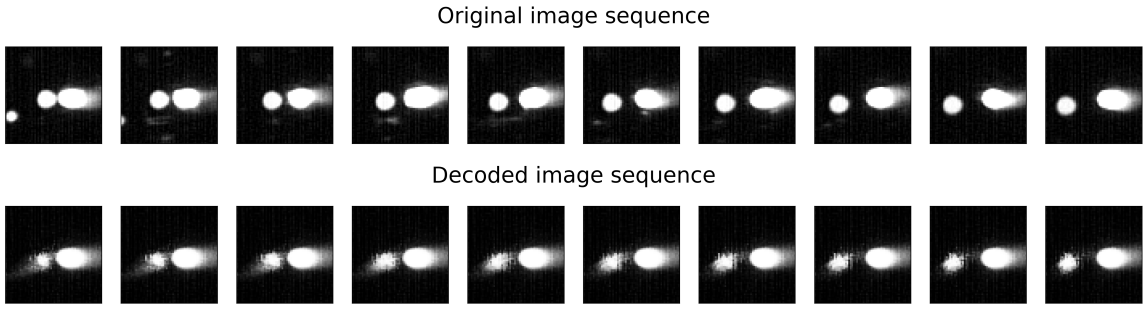
As shown by the previous section, the extracted bottleneck tensor $\mathcal{H}^{st}$ has a dimension of $5 \times 5 \times 64$ which after unfolding results in the flattened encoded vector $\vec{h_i}$ of size 1600. The encoded vectors provide a compact deep representation to be fed to the agglomerative clustering introduced in Section 6.1.2 and given by Algorithm 2 in order to determine the spatio-temporal anomaly samples. The resulting dendrogram



(a) Normal Melt Pool Image Sequence



(b) Normal Melt Pool Image Sequence with Anomaly in a Single Melt Pool



(c) Abnormal Melt Pool Image Sequence

Figure 6.8: Melt Pool Image Sequence Reconstruction by the Proposed Deep Spatio-temporal Auto-encoder

trained on the entire samples from two specimens is illustrated in Figure 6.9. The x-axis in the figure represents the sequence samples, and the y-axis shows the cophenetic distance to merge the clusters sequentially. The dendrogram's visual inspection reveals the existence of two super-clusters. Each super-cluster itself consists primarily of one large parent cluster and a few less populated child clusters, as shown by the vertical black dashed lines on the figure.

Truncating the dendrogram at a cophenetic distance above a certain threshold will create the desired partitioning as described above. One example of such thresholding at the cophenetic distance of 13 is shown by the horizontal red line in Figure 6.9. After sorting according to their population, the resultant clusters are illustrated by Figure 6.10 in the form of a bar chart. Clusters 9 to 22 fit well into our definition of anomaly provided in Section 6.1.2 as they merge with other clusters by considerably larger cophenetic distances and also form near-singleton clusters of their own. Figure 6.11 represents 5 arbitrary sequence samples that fall inside these anomaly clusters to provide better insights into the structure of detected anomalies by the method. As can be observed, only samples with spatio-temporal abnormalities are detected, and sequences with single anomalies in melt pools remain in the normal clusters. It is worth noting that changing the threshold values for the clustering method can result in different clusters with varying populations of normal and abnormal samples. In general, the exact thresholding value depends on the user's preference and degree of conservatism. Nevertheless, a hyperparameter fine-tuning step is provided in Section 6.2.4 to further guide the practitioners in this matter.

Table 6.1: Configuration of the Spatio-temporal Auto-encoder

Coding type	Layer name	Operation	Filters	Filter size	Stride	Output dimension
Input	-	-	-	-	-	$10 \times 60 \times 60 \times 1$
Spatial	$Conv1$	$Convolution + ReLU$	16	5×5	3	$10 \times 20 \times 20 \times 16$
subencoder	$Conv2$	$Convolution + ReLU$	8	3×3	2	$10 \times 10 \times 10 \times 8$
	$Conv3$	$Convolution + ReLU$	8	2×2	2	$10 \times 5 \times 5 \times 8$
Temporal	$ConvLSTM1$	$Convolution + LSTM + ReLU$	128	2×2	1	$10 \times 5 \times 5 \times 128$
subencoder	$ConvLSTM2$	$Convolution + LSTM + ReLU$	64	2×2	1	$5 \times 5 \times 64$
Temporal	$Repeat\ Tensor$	$Expand\ dimension$	-	-	-	$10 \times 5 \times 5 \times 64$
subdecoder	$ConvLSTM3$	$Convolution + LSTM + ReLU$	128	2×2	1	$10 \times 5 \times 5 \times 128$
	$ConvTrans1$	$Convolution\ transpose + ReLU$	8	2×2	2	$10 \times 10 \times 10 \times 8$
Spatial	$ConvTrans2$	$Convolution\ transpose + ReLU$	8	3×3	2	$10 \times 20 \times 20 \times 8$
subdecoder	$ConvTrans3$	$Convolution\ transpose + ReLU$	16	5×5	3	$10 \times 60 \times 60 \times 16$
	$Conv4$	$Convolution + sigmoid$	1	5×5	0	$10 \times 60 \times 60 \times 1$
Output	-	-	-	-	-	$10 \times 60 \times 60 \times 1$

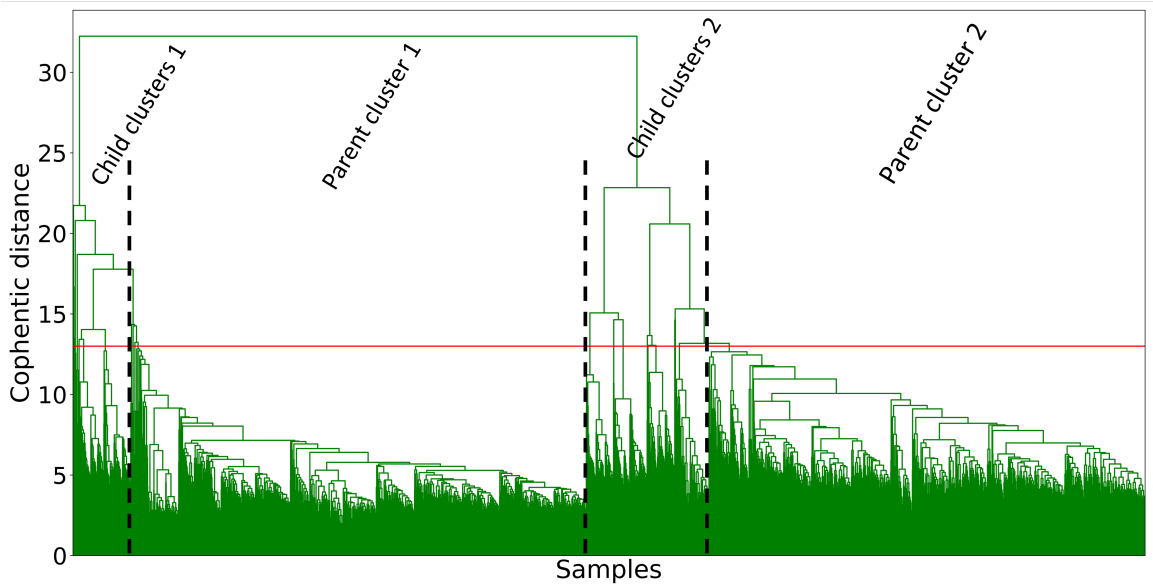


Figure 6.9: Dendrogram for Hierarchical-based Agglomerative Clustering with Average Linkage

6.2.4 Hyperparameter Fine-tuning and Anomaly Detection Performance

The procedure to fine-tune the monitoring hyperparameters based on validation data, including threshold values for clustering and type-I error for constructing control charting schemes, is described in this section. After construction, the control charts are utilized in conjunction with the spatio-temporal auto-encoder to detect anomalies in unseen test data in real-time. The bottleneck tensors tend to preserve the main patterns in the original sequences and thus might fail to detect anomalies caused by the unaccounted variability in the feature extraction procedure. Therefore, the residual's variance control charting introduced in Section 6.1.3 with its computational statistics given by Equation (5.14) can further assist with finding the anomalies. In general, there is a strong correlation between the two methods in terms of detecting anomaly samples. We build upon this fact to fine-tune the clustering cophenetic distance threshold and type-I error simultaneously. To this aim, first, ten equally

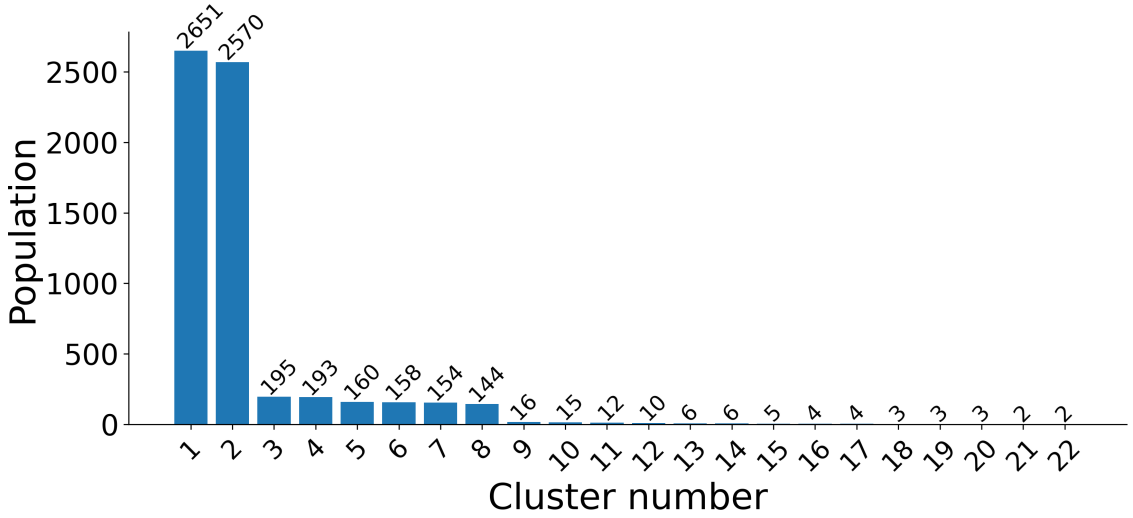
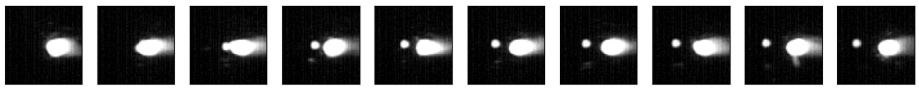


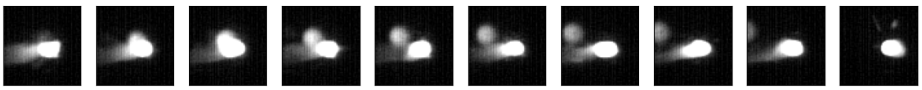
Figure 6.10: Bar Chart of Clusters and Their Population Formed by the Agglomerative Algorithm at the Cophenetic Distance Of 13

spaced discrete values inside intervals $[8, 16]$ are selected as candidate thresholds for clustering. The samples inside the resulting clusters of size less than 40 are counted toward anomalies, and the rest fall inside the normal category. The obtained normal samples are then randomly split into training and validation normal data by a proportion of 75% and 25%, respectively. The training normal samples are retained to construct the phase-I control charting scheme given a type-I error of $\alpha^{'}$. Ten equally spaced points inside interval $[0.005, 0.05]$ have been selected to study $\alpha^{'}$ in this work.

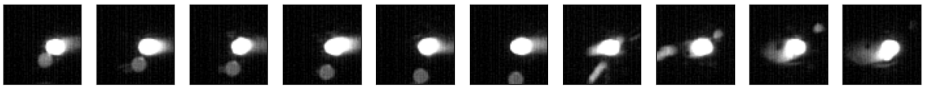
More specifically, the phase-I for Hotelling's T^2, consists of using training normal data to estimate the mean vector μ and variance-covariance matrix Σ, as given by Equations (5.9) and (5.10), respectively. The parameter estimates are used to determine the UCL given a value for α. It should be noted that as the inverse of the variance-covariance matrix Σ^{-1} exhibits unstable behavior in the presence of large dimensions, $\vec{h_i}$ are further downsized to vectors of 400 dimension using Principle Components Analysis (PCA). On the other hand, and with respect to residual's variance



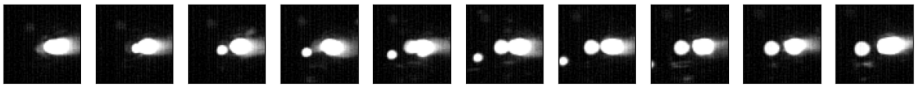
(a) Sequence at $t = 1357$



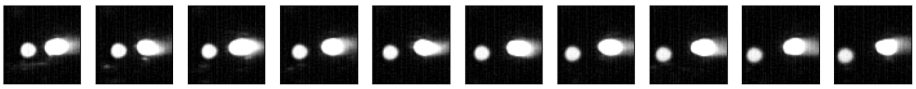
(b) Sequence at $t = 1705$



(c) Sequence at $t = 2263$



(d) Sequence at $t = 5960$



(e) Sequence at $t = 5970$

Figure 6.11: Examples of Spatio-temporal Anomalies Detected by Clustering on Top of the Auto-encoding Procedure

S^2, phase-I only consists of estimating the UCL for each α value using original and decoded sequence images as described in Section 6.1.3. The optimal threshold values for clustering and type-I error are fine-tuned by benchmarking the performance of each combination against the validation data in phase-II monitoring. The validation data set consists of validation normal samples and all abnormal samples. In phase-II, a sample is considered out-of-control when either the T^2 or the S^2 monitoring statistics exceed their corresponding control limits. On the other hand, the process can be considered in-control if all statistics fall inside the control limits.

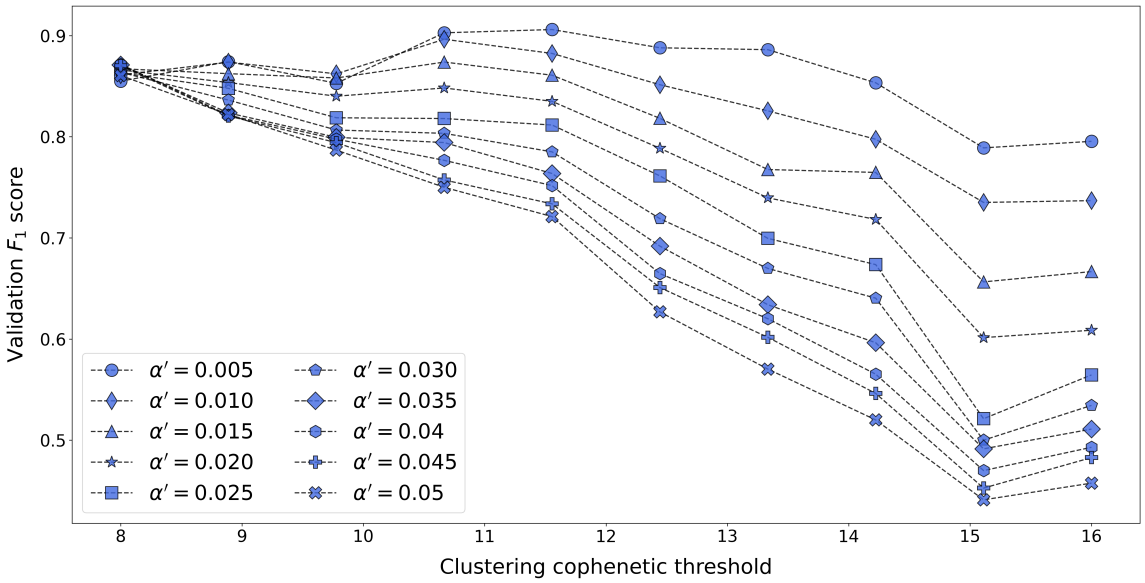


Figure 6.12: Validation $F_1 score$ for Different Clustering Thresholds and Type-I Error

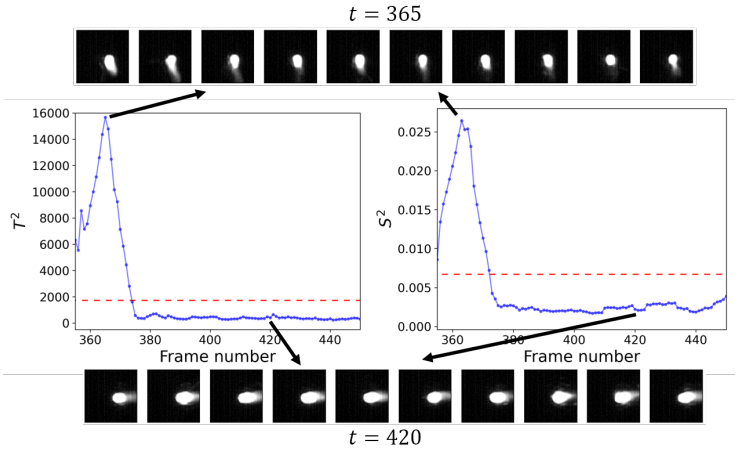
The benchmarking can be realized by placing each sample under TP, FP, TN, and FN categories analogous to Chapter 5 Section. 5.2.4. Eventually, the quality of phase-II monitoring can be measured using the following conventional machine learning performance evaluation metrics similar to that of Chapter 5:

$$recall = \frac{\sum(TP)}{\sum(TP + FN)}, \tag{6.9}$$

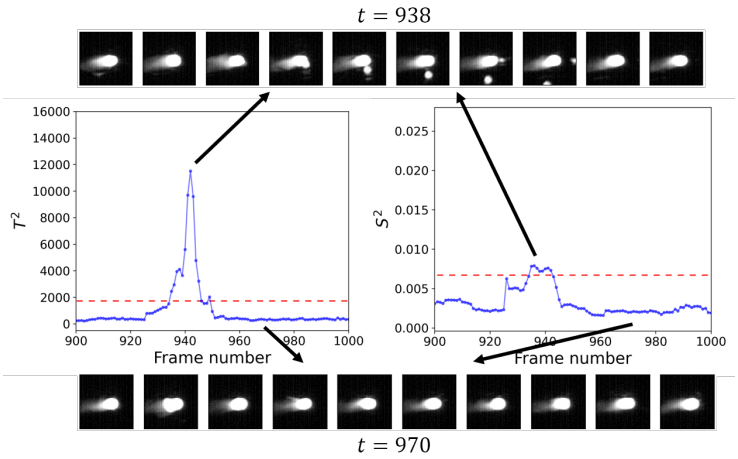
$$precision = \frac{\sum(TP)}{\sum(TP + FP)}, \tag{6.10}$$

$$F_1 score = 2\frac{Recall.Precision}{Recall + Precision}, \tag{6.11}$$

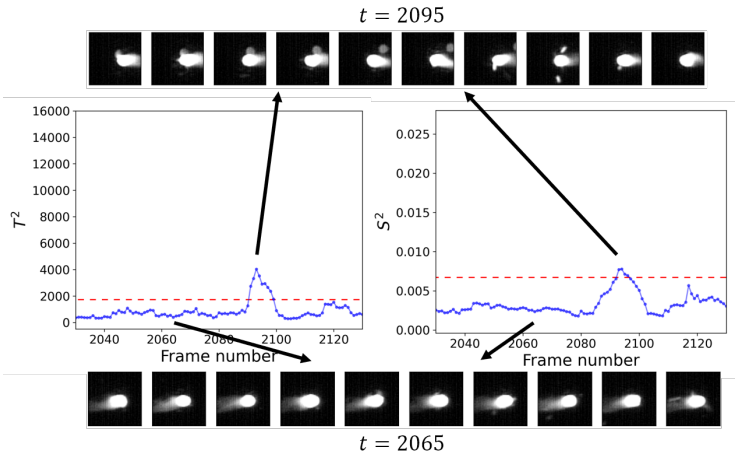
Accuracy is not considered in the list of measured metrics because the data is highly imbalanced. Figure 6.12 provides the validation $F_1 score$ results after 100 trials with different cophenetic distance threshold and $\alpha^{'}$ combinations. The cophenetic threshold and type-I error $\alpha^{'}$ equal to 11.55 and 0.005, respectively result in the highest $F_1 score$ of about 90.60%. Therefore, they have been chosen as the final hyperparameters combinations for the model.



(a) Melt Pool Sequence for Laser Power Equal to 100 W versus 195 W.

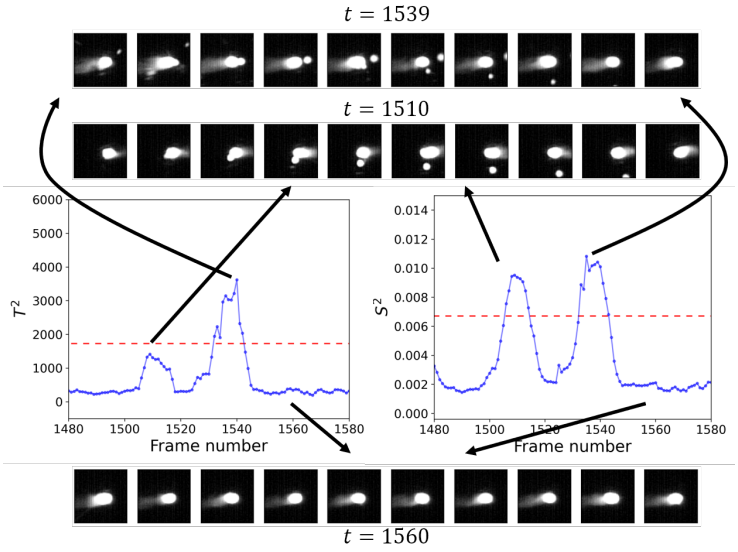


(b) Melt Pool Sequence for Laser Power Equal to 195 W.

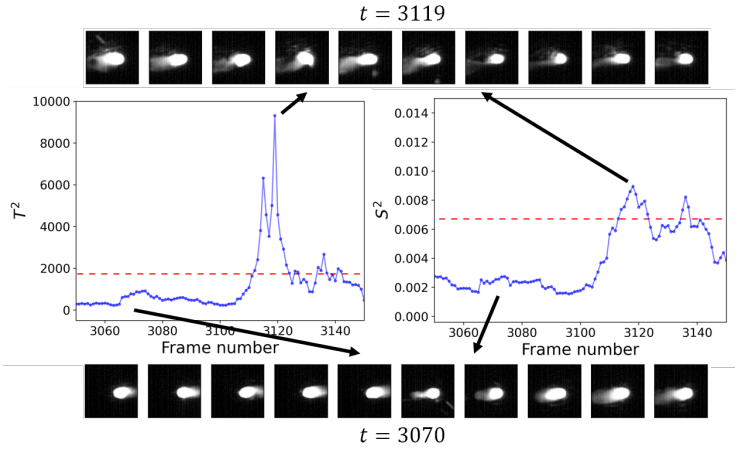


(c) Melt Pool Sequence for Laser Power Equal to 195 W.

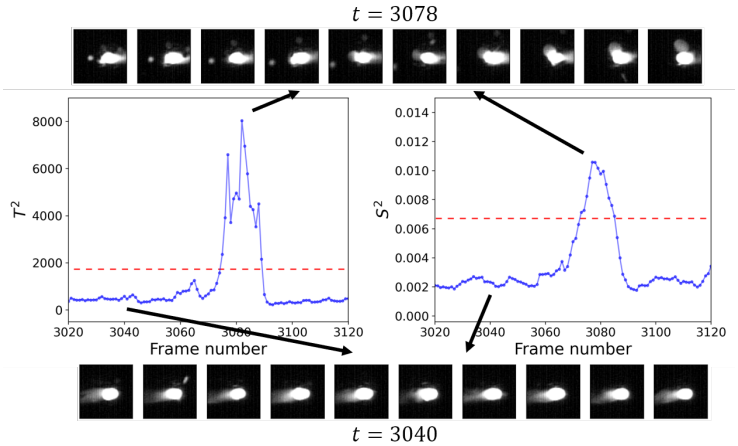
Figure 6.13: Anomaly Detection by T^2 and S^2 Control Charts for Specimen #4 at Layer 60.



(a) Melt Pool Sequence for Specimen #1 at Layer 120.



(b) Melt Pool Sequence for Specimen #3 at Layer 160.



(c) Melt Pool Sequence for Specimen #4 at Layer 200.

Figure 6.14: Anomaly Detection by T^2 and S^2 Control Charts for Different Specimens and Layers

Finally, the generalization power of the constructed control charts is tested against unseen data. To this aim, first, several short video clips containing both normal and abnormal events –from other two specimens and from the layers of experiment that are not used in the training and validation data– are collected. After passing through the image pre-processing pipeline, including sharpening, cropping, and discarding noisy image operations, the video clips are resampled into sliding windows of size 10 sequential images. Next, the samples are fed by the pre-trained auto-encoder, and resulting spatio-temporal bottleneck tensors $\mathcal{H}^{st}$ and reconstructed sequence $\bar{X}$ are retained. Finally and following the control charting procedure, T^2 and S^2 monitoring statistics for each sequence is computed and plotted on the corresponding control charts. Figure 6.13 illustrates the T^2 and S^2 charting results for specimen #4 at layer 60. Remember from Section 6.2.1 that printing a layer is composed of two passes: a contour pass with a lower laser power equal to $100\,W$ to print the border, and an infill pass with a higher laser power equal to $195\,W$ to print the part core and cylindrical features. The former pass is not included in the training data and therefore should be detected as an anomaly in the process if monitored by the method. An example video clip containing the laser power transition pass in time is shown in Figure 6.13a. As detected by the charts and shown by the sequence at time frame 365, melt pool sequences with low laser powers mainly feature smaller melt pool regions. On the other hand and after the start of the infill pass, the control charts go back to issue normal signals.

Figure 6.13b and similarly Figure 6.13c represent several scenarios of spatio-temporal anomalies and normal samples during the process at different times. As it was previously discussed, T^2 control chart might fail to detect those abnormal sample whose faulty signature patterns haven't been captured by the auto-encoding procedure. In those scenarios, the S^2 statistics monitors the samples residual's variance to

assist the T^2 control chart. Figures 6.14a shows an example event for specimen #1 at layer 120, in which the first spatio-temporal anomaly instance at time frame 1510 is detected by S^2 while T^2 indicates a normal situation. Figures 6.14b and 6.14c contain testing samples from specimen #3 at layer 160, and specimen #4 at layer 200, respectively. The majority of detected anomalies so far are mainly characterized by the presence of splattering effects around the main melt pools across the image sequence. The control charts in Figures 6.14b and 6.14c illustrate the power and robustness of the method in detecting a wide-range of rather different anomaly events where the morphological characteristics of the main pool itself have been compromised in the process.

Chapter 7

CONCLUSION

The scope of the dissertation is defined based on developing novel process-specific monitoring and control methods across two types of additive manufacturing processes. As delineated by Chapter 1, LSAM and LPBF technologies have been recently developed but are still under investigation for their lack of viability in practice. The unified objective of the study includes providing monitoring and optimal control schemes that mitigate the quality risks and increase the efficiency of the processes. Investigating the critical assumptions that existing methods often overlooked or failed to model, the proposed frameworks provide data-driven approaches to preprocess, model and transform the data into easy-to-implement proactive actions.

Large-scale additive manufacturing uses carbon fiber reinforced thermoplastic material to fabricate parts in large size. Print surface temperature is an important factor that impacts the product quality. A proper layer time results in a proper print surface temperature, leading to an acceptable product quality. However, layer time control with both quality and efficiency considered has not received sufficient attention. In this work, in-situ surface thermal modeling and real-time layer time control for large-scale additive manufacturing are investigated. Specifically, a regression model together with the Gaussian process is proved to predict surface temperature of parts accurately, even in a dynamic environment. The regression model is then combined into an optimization model to derive the optimal layer time for each layer to better satisfy the quality requirement and also improve the efficiency. The proposed method is useful for practitioners of large-scale additive manufacturing and is capable of being embedded in the control system of the large-scale 3D printer to improve the process

through manufacturing automation.

The second piece of this work toward monitoring and control of LSAM process investigates the set of images captured by thermal cameras for insights into the thermodynamics of the process. Given two case studies, it was shown how different locations subject to different exogenous and environmental factors, geometry of parts, and positioning can exhibit different heat dissipation. Specifically, a non-linear profile model was introduced and fitted to the thermal profile data of each location to capture its corresponding cooling dynamics. The resulting thermal gradient across the surface motivated the introduction and development of several real-time control models including a print time control and a printer head speed control model. Each model was then studied in details and the proper procedures were developed to solve the models to reach the optimal control variables. Finally. the numerical result were provided to prove the effectiveness and applicability of the proposed methods.

The prominent signatures of Laser Powder Bed Fusion (LPBF) processes can be captured by in-situ sensor data, for example, streams of co-axial melt pool images collected during the printing process. In contrast to physics-based modeling approaches, data-driven methods exhibit stable and robust process monitoring performance if trained with sufficient data. On the other hand, parametric and regression-based methods rely on unrealistic assumptions in their development phase, which can be easily violated in practical and high-dimensional settings. Moreover, reliance on handcrafted feature engineering methods deteriorates their performance and generalization power in comparison with model agnostic deep learning methods that integrate feature engineering with model architecture training. Specifically, a configuration of Convolutional Auto-Encoder (CAE) neural networks processes the collected melt pool images to learn a low-dimensional but deep representation from data. The extracted features are plugged into an agglomerative clustering algorithm to tag the anomalies

and automatically annotate the data due to the burdensome and expensive manual annotation process. Subsequently, a Hotelling's T^2 and S^2 control charting scheme is developed to monitor the process's stability by keeping track of the learned representations and residuals obtained from the reconstruction of original images. Finally, the numerical results from a real-world case study were provided to prove the proposed method's effectiveness, applicability, and accuracy.

Anomaly detection based on individual melt pool monitoring suffers from an over-inflated false alarm rate. To overcome this, we propose a configuration of convolutional long short term memory auto-encoders composed of spatial and temporal subencoder/subdecoder components to model the intricate temporal correlation between melt pool video frames as well as the spatial characteristics within each melt pool region. After unfolding, the extracted bottleneck tensors are plugged into an agglomerative clustering algorithm with average linkage to annotate the group anomalies and collect the normal samples. The training portion of normal samples is used to construct the phase-I of the proposed dual control charting method, and the rest are used as validation samples to fine-tune the hyperparameters of the model. The performance of the method against unseen test data demonstrates its applicability and effectiveness to detect the anomalies and distinguish the normal events in real-world experiments.

7.1 Directions for Future Research

The directions for future studies by expanding the current scope of this study are given as follows:

1. The developed methodology in this study strives to be fairly model agnostic and data dependent. Nevertheless, by leveraging physical-related models as prior data, adaptive Bayesian models can be adopted to construct hybrid decision

support frameworks.

2. Typically, more than one source of data is captured during the building spans of various additive manufacturing processes. In addition, data sampling rate and input dimension varies across different modalities. It is worth investigating the feature matching and multi-modality data fusion methodologies to unify all sources of data and bolster the accuracy of proposed control and monitoring schemes.

3. Despite defining and utilizing several sliding windows sampling strategies to incorporate spatio-temporal correlation of data into account, the data from different layers assumed to be independent in this study. The opposite will introduce further complications to the feature extraction methods by adding an extra dimension to the tensor of captured data. A future direction would involve investigating the layerwise framework that employ between layer correlation.

4. Monitoring methods can only provide real-time predictions about upcoming samples after analyzing the historical data. We can expect to receive accurate signals that determine when the process is in or out of control and whether it requires intervention to get back on track. This means repetitive pauses in the process for further investigation into the cause of anomaly incidents. In other words, by relying on historical data as opposed to hard-to-derive physical-related modeling, we can predict the system response using the system's predictor or state variables. On the other hand however, given the current and historical state of the system, real-time optimal control methods strive to provide a prescriptive analysis by finding the control variables that optimize a reward or loss function of interest. This will guarantee that the system is always moving and correcting itself in the right direction without any interrup-

tion. As an instance, consider an autonomous driving vehicle, where state is captured by images from the road and environment, control is the magnitude of acceleration and steering wheel position, and reward is defined as crash or normal drive. Similarly, let's define the state of LPBF system as an embedding of the melt pool images and represent its vector by $\vec{x}$. The control vector denoted by $\vec{u}$ can be provided by experiment parameter settings such as laser speed, hatch distance and laser power. Finally, the loss/reward function r can be defined based on a mechanical characteristics related to the formation of defects or X-CT scan labels. Inevitably, the system evolution equations appears to be almost impossible to be found by a closed-form equation. Fortunately, Deep Reinforcement Learning (DRL) methods can provide a ground to learn the system evolution equations using the historical data for state, control, and reward functions. Deep Q-learning technique in particular provides an appropriate match where the stacks of state and control, i.e., $\vec{x}$ and $\vec{u}$ are fed to a neural network and to learn the reward function r.

REFERENCES

Agarwala, M. K., V. R. Jamalabad, N. A. Langrana, A. Safari, P. J. Whalen and S. C. Danforth, "Structural quality of parts processed by fused deposition", Rapid prototyping journal (1996).

Anitha, R., S. Arunachalam and P. Radhakrishnan, "Critical parameters influencing the quality of prototypes in fused deposition modelling", Journal of Materials Processing Technology **118**, 1-3, 385–388 (2001).

Asadi-Eydivand, M., M. Solati-Hashjin, A. Farzad and N. A. A. Osman, "Effect of technical parameters on porous structure and strength of 3d printed calcium sulfate prototypes", Robotics and Computer-Integrated Manufacturing **37**, 57–67 (2016).

Astm, I., "Astm52900-15 standard terminology for additive manufacturing—general principles—terminology", ASTM International, West Conshohocken, PA **3**, 4, 5 (2015).

Azizi, A., S. Parsaeefard, M. R. Javan, N. Mokari and H. Yanikomeroglu, "Profit maximization in 5g+ networks with heterogeneous aerial and ground base stations", IEEE Transactions on Mobile Computing (2019).

Bertsekas, D. P., *Dynamic programming and optimal control*, vol. 1, No. 2 (Belmont, MA: Athena scientific, 1995).

Bikas, H., P. Stavropoulos and G. Chryssolouris, "Additive manufacturing methods and modelling approaches: a critical review", The International Journal of Advanced Manufacturing Technology **83**, 1-4, 389–405 (2016).

Bishop, C. M., *Pattern recognition and machine learning* (springer, 2006).

Borish, M., B. Post, A. Roschli, P. Chesser, L. Love, K. Gaul, M. Sallas and N. Tsiamis, "In-situ thermal imaging for single layer build time alteration in large-scale polymer additive manufacturing", Tech. rep., Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States) (2019).

Brenken, B., E. Barocio, A. Favaloro, V. Kunc and R. B. Pipes, "Fused filament fabrication of fiber-reinforced polymers: A review", Additive Manufacturing (2018).

Buswell, R. A., W. L. de Silva, S. Jones and J. Dirrenberger, "3d printing using concrete extrusion: A roadmap for research", Cement and Concrete Research **112**, 37–49 (2018).

Chesser, P., B. Post, A. Roschli, C. Carnal, R. Lind, M. Borish and L. Love, "Extrusion control for high quality printing on big area additive manufacturing (baam) systems", Additive Manufacturing **28**, 445–455 (2019).

Ciurana, J., L. Hernandez and J. Delgado, "Energy density analysis on single tracks formed by selective laser melting with cocrmo powder material", The International Journal of Advanced Manufacturing Technology **68**, 5-8, 1103–1110 (2013).

Cole, K. D., M. R. Yavari and P. K. Rao, "Computational heat transfer with spectral graph theory: Quantitative verification", International Journal of Thermal Sciences **153**, 106383 (2020).

Colosimo, B. M., Q. Huang, T. Dasgupta and F. Tsung, "Opportunities and challenges of quality engineering for additive manufacturing", Journal of Quality Technology **50**, 3, 233–252 (2018).

Compton, B. G., B. K. Post, C. E. Duty, L. Love and V. Kunc, "Thermal analysis of additive manufacturing of large-scale thermoplastic polymer composites", Additive Manufacturing **17**, 77–86 (2017).

Cunningham, R., S. P. Narra, C. Montgomery, J. Beuth and A. Rollett, "Synchrotron-based x-ray microtomography characterization of the effect of processing variables on porosity formation in laser power-bed additive manufacturing of ti-6al-4v", Jom **69**, 3, 479–484 (2017).

Dinwiddie, R. B., V. Kunc, J. M. Lindal, B. Post, R. J. Smith, L. Love and C. E. Duty, "Infrared imaging of the polymer 3d-printing process", in "Thermosense: Thermal Infrared Applications XXXVI", vol. 9105, p. 910502 (International Society for Optics and Photonics, 2014).

Dreifus, G., K. Goodrick, S. Giles, M. Patel, R. M. Foster, C. Williams, J. Lindahl, B. Post, A. Roschli, L. Love *et al.*, "Path optimization along lattices in additive manufacturing using the chinese postman problem", 3D Printing and Additive Manufacturing **4**, 2, 98–104 (2017).

Duty, C. E., V. Kunc, B. Compton, B. Post, D. Erdman, R. Smith, R. Lind, P. Lloyd and L. Love, "Structure and mechanical behavior of big area additive manufacturing (baam) materials", Rapid Prototyping Journal **23**, 1, 181–189 (2017).

Farzadi, A., V. Waran, M. Solati-Hashjin, Z. A. A. Rahman, M. Asadi and N. A. A. Osman, "Effect of layer printing delay on mechanical properties and dimensional accuracy of 3d printed porous prototypes in bone tissue engineering", Ceramics International **41**, 7, 8320–8330 (2015).

Fathizadan, S., F. Ju and Y. Lu, "Deep representation learning for process variation management in laser powder bed fusion", Additive Manufacturing **42**, 101961 (2021).

Fathizadan, S., S. T. A. Niaki and R. Noorossana, "Using independent component analysis to monitor 2-d geometric specifications", Quality and Reliability Engineering International **33**, 8, 2075–2087 (2017a).

Fathizadan, S., S. T. A. Niaki and R. Noorossana, "Using independent component analysis to monitor 2-d geometric specifications", Quality and Reliability Engineering International **33**, 8, 2075–2087 (2017b).

Frazier, W. E., "Metal additive manufacturing: a review", Journal of Materials Engineering and Performance **23**, 6, 1917–1928 (2014).

Gobert, C., E. W. Reutzel, J. Petrich, A. R. Nassar and S. Phoha, "Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging.", Additive Manufacturing **21**, 517–528 (2018).

Gong, H., K. Rafi, H. Gu, T. Starr and B. Stucker, "Analysis of defect generation in ti–6al–4v parts made using powder bed fusion additive manufacturing processes", Additive Manufacturing **1**, 87–98 (2014).

Goodfellow, I., Y. Bengio and A. Courville, *Deep learning* (MIT press, 2016).

Grasso, M., A. Demir, B. Previtali and B. Colosimo, "In situ monitoring of selective laser melting of zinc powder via infrared imaging of the process plume", Robotics and Computer-Integrated Manufacturing **49**, 229–239 (2018).

Grasso, M., V. Laguzza, Q. Semeraro and B. M. Colosimo, "In-process monitoring of selective laser melting: spatial detection of defects via image data analysis", Journal of Manufacturing Science and Engineering **139**, 5 (2017).

Guo, N. and M. C. Leu, "Additive manufacturing: technology, applications and research needs", Frontiers of Mechanical Engineering **8**, 3, 215–243 (2013).

Guo, S., W. G. Guo and L. Bian, "Hierarchical spatial-temporal modeling and monitoring of melt pool evolution in laser-based additive manufacturing", IISE Transactions , just-accepted, 1–34 (2019).

Han, W., M. A. Jafari, S. C. Danforth and A. Safari, "Tool path-based deposition planning in fused deposition processes", Journal of manufacturing science and engineering **124**, 2, 462–472 (2002).

Han, W., M. A. Jafari and K. Seyed, "Process speeding up via deposition planning in fused deposition-based layered manufacturing processes", Rapid Prototyping Journal **9**, 4, 212–218 (2003).

Holshouser, C., C. Newell, S. Palas, L. J. Love, V. Kunc, R. F. Lind, P. D. Lloyd, J. C. Rowe, C. A. Blue, C. E. Duty *et al.*, "Out of bounds additive manufacturing", Advanced Materials and Processes **171**, 3 (2013).

Huang, X., C. Ye and Y. Huang, "Tool path planning based on endpoint build-in optimization in rapid prototyping", Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science **225**, 12, 2919–2926 (2011).

Jin, Y., H. A. Pierson and H. Liao, "Toolpath allocation and scheduling for concurrent fused filament fabrication with multiple extruders", IISE Transactions **51**, 2, 192–208 (2019).

Khairallah, S. A., A. T. Anderson, A. Rubenchik and W. E. King, "Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones", Acta Materialia **108**, 36–45 (2016).

Khanzadeh, M., L. Bian, N. Shamsaei and S. M. Thompson, "Porosity detection of laser based additive manufacturing using melt pool morphology clustering", Solid Freeform Fabrication pp. 1487–1494 (2016).

Khanzadeh, M., S. Chowdhury, M. Marufuzzaman, M. A. Tschopp and L. Bian, "Porosity prediction: Supervised-learning of thermal history for direct laser deposition", Journal of manufacturing systems **47**, 69–82 (2018a).

Khanzadeh, M., S. Chowdhury, M. A. Tschopp, H. R. Doude, M. Marufuzzaman and L. Bian, "In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes", IISE Transactions **51**, 5, 437–455 (2019).

Khanzadeh, M., W. Tian, A. Yadollahi, H. R. Doude, M. A. Tschopp and L. Bian, "Dual process monitoring of metal-based additive manufacturing using tensor decomposition of thermal image streams", Additive Manufacturing **23**, 443–456 (2018b).

Khoshnevis, B., "Automated construction by contour crafting—related robotics and information technologies", Automation in construction **13**, 1, 5–19 (2004).

Khoshnevis, B. and R. Dutton, "Innovative rapid prototyping process makes large sized, smooth surfaced complex shapes in a wide variety of materials", Materials Technology **13**, 2, 53–56 (1998).

Khoshnevis, B., H. Kwon and S. Bukkapatnam, "Automated construction using contour crafting", in "IIe Annual Conference. Proceedings, Houston, TX, USA", (2004).

Kim, H., Y. Lin and T.-L. B. Tseng, "A review on quality control in additive manufacturing", Rapid Prototyping Journal (2018).

Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980 (2014).

Kunc, V., A. A. Hassen, J. Lindahl, S. Kim, B. Post and L. Love, "Large scale additively manufactured tooling for composites", in "15th JAPAN International SAMPE Symposium and Exhibition. Japan", (2017).

Kunc, V., A. Lee, M. Mathews, J. Lindahl *et al.*, "Low cost reactive polymers for large scale additive manufacturing", (2018).

Kwon, O., H. G. Kim, M. J. Ham, W. Kim, G.-H. Kim, J.-H. Cho, N. I. Kim and K. Kim, "A deep neural network for classification of melt-pool images in metal additive manufacturing", Journal of Intelligent Manufacturing **31**, 2, 375–386 (2020).

Labonnote, N., A. Rønnquist, B. Manum and P. Rüther, "Additive construction: State-of-the-art, challenges and opportunities", Automation in Construction **72**, 347–366 (2016).

Lane, B., S. Mekhontsev, S. Grantham, M. Vlasea, J. Whiting, H. Yeung, J. Fox, C. Zarobila, J. Neira, M. McGlauflin *et al.*, "Design, developments, and results from the nist additive manufacturing metrology testbed (ammt)", in "Solid Freeform Fabrication Symposium, Austin, TX", pp. 1145–1160 (2016).

Lane, B. and H. Yeung, "Process monitoring dataset from the additive manufacturing metrology testbed (ammt):" three-dimensional scan strategies"", Journal of Research of the National Institute of Standards and Technology **124**, 1–14 (2019).

Lane, B. and H. Yeung, "Process monitoring dataset from the additive manufacturing metrology testbed (ammt): Overhang part x4", Journal of Research of the National Institute of Standards and Technology **125**, 1–18 (2020).

Lindahl, J., A. Hassen, S. Romberg, B. Hedger, P. Hedger Jr, M. Walch, T. DeLuca, W. Morrison, P. Kim, A. Roschli *et al.*, "Large-scale additive manufacturing with reactive polymers", Tech. rep., Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States) (2018).

Liu, C., A. C. C. Law, D. Roberson and Z. J. Kong, "Image analysis-based closed loop quality control for additive manufacturing with fused filament fabrication", Journal of Manufacturing Systems **51**, 75–86 (2019).

Love, L. J., "Utility of big area additive manufacturing (baam) for the rapid manufacture of customized electric vehicles", Tech. rep., Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States). Manufacturing … (2015).

Martin, E. and A. Morris, "Non-parametric confidence bounds for process performance monitoring charts", Journal of Process Control **6**, 6, 349–358 (1996).

Mireles, J., S. Ridwan, P. A. Morton, A. Hinojos and R. B. Wicker, "Analysis and correction of defects within parts fabricated using powder bed fusion technology", Surface Topography: Metrology and Properties **3**, 3, 034002 (2015).

Mireles, J., C. Terrazas, F. Medina, R. Wicker and E. Paso, "Automatic feedback control in electron beam melting using infrared thermography", in "Proceedings of the Solid Freeform Fabrication Symposium", (2013).

Mohamed, O. A., S. H. Masood and J. L. Bhowmik, "Mathematical modeling and fdm process parameters optimization using response surface methodology based on q-optimal design", Applied Mathematical Modelling **40**, 23-24, 10052–10073 (2016).

Montgomery, D. C., *Introduction to statistical quality control* (John Wiley & Sons, 2007).

Montgomery, D. C., E. A. Peck and G. G. Vining, *Introduction to linear regression analysis*, vol. 821 (John Wiley & Sons, 2012).

Nocedal, J. and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

Noorossana, R., S. Fathizadan and M. R. Nayebpour, "Ewma control chart performance with estimated parameters under non-normality", Quality and Reliability Engineering International **32**, 5, 1637–1654 (2016).

Otsu, N., "A threshold selection method from gray-level histograms", IEEE transactions on systems, man, and cybernetics **9**, 1, 62–66 (1979).

Pascanu, R., T. Mikolov and Y. Bengio, "On the difficulty of training recurrent neural networks", in "International conference on machine learning", pp. 1310–1318 (PMLR, 2013).

Pegna, J., "Exploratory investigation of solid freeform construction", Automation in construction **5**, 5, 427–437 (1997).

Quiñonero-Candela, J. and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression", Journal of Machine Learning Research **6**, Dec, 1939–1959 (2005).

Repossini, G., V. Laguzza, M. Grasso and B. M. Colosimo, "On the use of spatter signature for in-situ monitoring of laser powder bed fusion", Additive Manufacturing **16**, 35–48 (2017).

Romberg, S., C. Hershey, J. Lindahl, W. Carter, B. G. Compton and V. Kunc, "Large-scale additive manufacturing of highly exothermic reactive polymer systems", Tech. rep., Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States) (2019).

Sadowski, M., L. Ladani, W. Brindley and J. Romano, "Optimizing quality of additively manufactured inconel 718 using powder bed laser melting process", Additive Manufacturing **11**, 60–70 (2016).

Shi, X., Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting", arXiv preprint arXiv:1506.04214 (2015).

Sitthi-Amorn, P., J. E. Ramos, Y. Wangy, J. Kwan, J. Lan, W. Wang and W. Matusik, "Multifab: a machine vision assisted platform for multi-material 3d printing", ACM Transactions on Graphics (TOG) **34**, 4, 1–11 (2015).

Sood, A. K., R. K. Ohdar and S. S. Mahapatra, "Parametric appraisal of mechanical property of fused deposition modelling processed parts", Materials & Design **31**, 1, 287–295 (2010).

Sun, Q., G. Rizvi, C. Bellehumeur and P. Gu, "Effect of processing conditions on the bonding quality of fdm polymer filaments", Rapid Prototyping Journal **14**, 2, 72–80 (2008).

Sutskever, I., O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks", arXiv preprint arXiv:1409.3215 (2014).

Talagani, M., S. DorMohammadi, R. Dutton, C. Godines, H. Baid, F. Abdi, V. Kunc, B. Compton, S. Simunovic, C. Duty *et al.*, "Numerical simulation of big area additive manufacturing (3d printing) of a full size car", SAMPE Journal **51**, 4, 27–36 (2015).

Tang, M., P. C. Pistorius and J. L. Beuth, "Prediction of lack-of-fusion porosity for powder bed fusion", Additive Manufacturing **14**, 39–48 (2017).

Thermwood, *An all new technology for large scale 3D printing of thermoplastic polymers*, see also URL `http://www.thermwood.com/lsam_home.htm` (2019a).

Thermwood, *Large scale additive manufacturing*, see also URL `http://thermwood.com /lsam/brochures/lsam_2017_brochure/ index.html` (2019b).

Tieleman, T. and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude", COURSERA: Neural networks for machine learning **4**, 2, 26–31 (2012).

Wang, F., F. Ju, K. Rowe and N. Hofmann, "Real-time control for large scale additive manufacturing using thermal images", in "2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)", pp. 36–41 (IEEE, 2019).

Williams, J. D., W. H. Woodall and J. B. Birch, "Statistical monitoring of nonlinear product and process quality profiles", Quality and Reliability Engineering International **23**, 8, 925–941 (2007).

Yan, H., M. Grasso, K. Paynabar and B. M. Colosimo, "Real-time detection of clustered events in video-imaging data with applications to additive manufacturing", IISE Transactions **0**, 0, 1–28, URL `https://doi.org/10.1080/24725854.2021.1882013` (2021).

Yang, Z., Y. Lu, S. Li, J. Li, Y. Ndiaye, H. Yang and S. Krishnamurty, "In-process data fusion for process monitoring and control of metal additive manufacturing", in "2021 ASME International Design Engineering Technical Conferences Computers and Information in Engineering", (ASME, 2021).

Yang, Z., Y. Lu, H. Yeung and S. Kirshnamurty, "3d build melt pool predictive modeling for powder bed fusion additive manufacturing", in "International Design Engineering Technical Conferences and Computers and Information in Engineering Conference", vol. 83983, p. V009T09A046 (American Society of Mechanical Engineers, 2020a).

Yang, Z., Y. Lu, H. Yeung and S. Krishnamurty, "Investigation of deep learning for real-time melt pool classification in additive manufacturing", in "2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)", pp. 640–647 (IEEE, 2019a).

Yang, Z., Y. Lu, H. Yeung and S. Krishnamurty, "From scan strategy to melt pool prediction: A neighboring-effect modeling method", Journal of Computing and Information Science in Engineering **20**, 5 (2020b).

Yang, Z., L. Yan, H. Yeung and S. Krishnamurty, "From scan strategy to melt pool prediction: A neighboring-effect modeling method", in "International Design Engineering Technical Conferences and Computers and Information in Engineering Conference", vol. 59179, p. V001T02A010 (American Society of Mechanical Engineers, 2019b).

Yao, B., F. Imani and H. Yang, "Markov decision process for image-guided additive manufacturing", IEEE Robotics and Automation Letters **3**, 4, 2792–2798 (2018).

Yavari, M. R., K. D. Cole and P. K. Rao, "Design rules for additive manufacturing–understanding the fundamental thermal phenomena to reduce scrap", Procedia Manufacturing **33**, 375–382 (2019).

Yeung, H., B. M. Lane, M. Donmez, J. C. Fox and J. Neira, "Implementation of advanced laser control strategies for powder bed fusion systems", Procedia Manufacturing **26**, 871–879 (2018).

Zeiler, M. D., "Adadelta: an adaptive learning rate method", arXiv preprint arXiv:1212.5701 (2012).

Zeng, K., D. Pal and B. Stucker, "A review of thermal analysis methods in laser sintering and selective laser melting", in "Proceedings of Solid Freeform Fabrication Symposium Austin, TX", vol. 60, pp. 796–814 (2012).

Zhang, B., S. Liu and Y. C. Shin, "In-process monitoring of porosity during laser additive manufacturing process", Additive Manufacturing **28**, 497–505 (2019).

APPENDIX A

PROOF OF PROPOSITION 1 IN CHAPTER 4

Proof. The objective function given by Equation (4.13) can be expressed as follows:

$$f(t) = \sum_{i=1}^{M} \left[\omega_i \left(a^i + b^i e^{\beta_1^i (t + \frac{i}{v} - t_0^i)} \right)^2 \right] + \omega_0 t$$

, where $a^i = T_{env} - \frac{\beta_0^i}{\beta_1^i} - T_b$ and $b^i = T^i(t_0^i) - T_{env} + \frac{\beta_0^i}{\beta_1^i}$ are negative and positive numbers, respectively. Taking the first order derivative we have:

$$\frac{\partial f}{\partial t} = 2 \sum_{i=1}^{M} \left[\omega_i \left(a^i + b^i e^{\beta_1^i (t + \frac{i}{v} - t_0^i)} \right) b^i \beta_1^i e^{\beta_1^i (t + \frac{i}{v} - t_0^i)} \right] + \omega_0$$

The second order derivative can be then obtained as:

$$\frac{\partial^2 f}{\partial t^2} = \sum_{i=1}^{M} \left(a^i + 2 b^i e^{\beta_1^i (t + \frac{i}{v} - t_0^i)} \right) c^i$$

, where $c^i = 2 \omega_i b^i \left(\beta_1^i\right)^2 e^{\beta_1^i (t + \frac{i}{v} - t_0^i)}$ is also a positive number.

Theorem 1. *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable over an open domain. Then, f is convex if and only if $\nabla^2 f(x) \succeq 0$ for all $x \in dom(f)$.*

Theorem (1) implies the objective function given by Equation (4.13) is convex if and only if the following equation holds:

$$\sum_{i=1}^{M} \left(a^i + 2 b^i e^{\beta_1^i (t + \frac{i}{v} - t_0^i)} \right) c^i \geq 0$$

Suppose γ is the numerical solution to $\sum_{i=1}^{M} \left(a^i + 2 b^i e^{\beta_1^i (t + \frac{i}{v} - t_0^i)} \right) c^i = 0$. Then, the above equation holds true if $t \leq \gamma$. The feasible region is also a convex set. As the result, the layer time control model is convex if and only if $t_U \leq \gamma$. □

APPENDIX B

PROOF OF PROPOSITION 2 IN CHAPTER 4

Proof. The optimal speed control model from each partition onward is independent from the speed at previous stage. Thus by substituting v^{j-1} with an arbitrary speed such as v_l, we have:

$$f^*(t, v \mid j, t^j, v^{j-1}) = f^*(t, v \mid j, t^j, v_l), \text{ for } j = 1, 2, \cdots, N \tag{B.1}$$

Furthermore, if t^j seconds have been passed so far, then we have:

$$f^*(t, v \mid j, t^j, v_l) = f^*(t + t^j, v \mid j), \text{ for } j = 1, 2, \cdots, N \tag{B.2}$$

Consider the case where $t^j = t_l$ and assume t^* is the optimal layer time for $f^*(t, v \mid j)$. The value of δ in Equation (4.44) is chosen small enough such that:

$$f^*(t, v \mid j, t_l, v_l) = f^*(t + t_l, v \mid j) = f^*(t, v \mid j), \text{ for } j = 1, 2, \cdots, N \tag{B.3}$$

That is because the optimal value of new variable $t + t_l$ accommodates itself by taking a value equal to t^*. In general we have $t = t^* - t^j$. As the value of t_j increases, the value of t decreases such that $f^*(t, v \mid j, t^j, v^{j-1})$ will be kept equal to $f^*(t, v \mid j)$. At a certain point it reaches a value where no further decrease in t is possible because it has to be non-negative. This is when the optimal value of the objective function starts to increase. Thus following Equations (B.1) - (B.3) for $t^j \in [t_l, t_u]$ and $v^{j-1} \in [v_l, v_u]$, we have:

$$f^*(t, v \mid j, t_l, v_l) \leq f^*(t, v \mid j, t^j, v^{j-1}), \text{ for } j = 1, 2, \cdots, N \tag{B.4}$$

□