October 2018

MODEL-BASED PREDICTIVE ANALYTICS FOR ADDITIVE AND SMART MANUFACTURING

Zhuo Yang
University of Massachusetts Amherst

Recommended Citation

Yang, Zhuo, "MODEL-BASED PREDICTIVE ANALYTICS FOR ADDITIVE AND SMART MANUFACTURING" (2018). *Doctoral Dissertations*. 1398.
https://scholarworks.umass.edu/dissertations_2/1398

MODEL-BASED PREDICTIVE ANALYTICS FOR ADDITIVE AND SMART MANUFACTURING

A Dissertation Presented

by

ZHUO YANG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2018

Mechanical and Industrial Engineering

MODEL-BASED PREDICTIVE ANALYTICS FOR ADDITIVE AND SMART MANUFACTURING

A Dissertation Presented

by

ZHUO YANG

Approved as to style and content by:

____

Sundar Krishnamurty, Chair, MIE

____

Ian R. Grosse, Member, MIE

____

Daeyoung Kim, Member, MS

____

Yan Lu, Member, NIST

____

Sundar Krishnamurty, Department Head

Department of Mechanical and Industrial Engineering

DEDICATION

To my dad, the best mechanical engineer in my heart.

ACKNOWLEDGMENTS

I would like to sincerely thank Professor Sundar Krishnamurty, my advisor and mentor during my entire research career at UMass Amherst. Without his support and guidance, I never would have overcome the obstacles that I faced throughout my research. He gave me a great deal of professional advice that helped me to learn the path to be a qualified researcher. He treats us as family members just as he said we are all important parts of his research family.

I would like to extend my thanks to Professor Ian Grosse, who guided, directed and extended the opportunity for my first project with the e-Design research group. He shared his advanced knowledge on finite element analysis. This was the first time I realized that research is more than solving problems. Beyond the research, he extended his helpful guidance to my daily life. This help included correcting my English and telling me about the American culture, which deeply warmed the heart of this student from a different country.

I would also like to thank my other committee members. Dr. Yan Lu, from the National Institute of Standard and Technology (NIST), spent a lot of time and energy to guide me on my research, introduced interesting topics, pointed out directions, and shared the latest publications. Her advanced experience in metamodeling and additive manufacturing were very valuable to my work. Special thanks must be extended to Professor Daeyoung Kim from the Mathematics and Statistics Department. His professional advice helped to guide this work with a valuable interdisciplinary perspective.

My words cannot describe how many thanks I want to give to Dr. Douglas Eddy. Dr. Eddy and I have collaborated for nearly four years on research projects related to additive manufacturing. During this time, he helped me to prepare almost every report, publication and conference presentation. His comments, suggestions and edits provide invaluable contributions to this dissertation.

Funding from NIST to address the need for this research made this work possible. I would like to thank all of the collaborators from NIST for their interest in this work and support of this project. Particularly, I would like to thank Dr. Paul Witherell at NIST. He contributed many suggestions to guide the direction of the metamodeling methods developed in this work. I would like to thank all of my labmates and classmates who enriched my experience and added to this work in various ways.

I am deeply grateful to my family who have supported and encouraged me throughout my studies at UMass. My parents always stood behind me and empowered me to reach my goals. Finally, I would like to dedicate the rest of this section to my wife Siwen Yang. She is the person who deserves the most praise. She left her hometown, friends and family to stay with me in a foreign country. This is not anything that is easy to do, and can be painful. This all required her to make a big sacrifice for me. Most especially, the Chinese foods made by my wife was the best medicine to ease my homesickness. These foods were also the perfect physical fuel to boost my energy to get this job done.

ABSTRACT

MODEL-BASED PREDICTIVE ANALYTICS FOR ADDITIVE AND SMART MANUFACTURING

SEPTEMBER 2018

ZHUO YANG, B.S., BEIJING UNIVERSITY OF TECHNOLOGY

PhD., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Sundar Krishnamurty

Qualification and certification for additive and smart manufacturing systems can be uncertain and very costly. Using available historical data can mitigate some costs of producing and testing sample parts. However, use of such data lacks the flexibility to represent specific new problems which decreases predictive accuracy and efficiency. To address these compelling needs, in this dissertation modeling techniques are introduced that can proactively estimate results expected from additive and smart manufacturing processes swiftly and with practical levels of accuracy and reliability. More specifically, this research addresses the current challenges and limitations posed by use of available data and the high costs of new data by tailoring statistics-based metamodeling techniques to enable affordable prediction of these systems.

The result is an integrated approach to customize and build predictive metamodels for the unique features of additive and smart manufacturing systems. This integrated approach is composed of five main parts that cover the broad spectrum of requirements. A domain-driven metamodeling approach uses physics-based knowledge to optimally select the most appropriate metamodeling algorithm without reliance upon statistical data. A maximum predictive error updating method iteratively improves predictability from a given dataset. A grey-box metamodeling approach combines statistics-based black-box and physics-based white-box models to significantly increase predictive accuracy with less expensive data overall. To improve computational efficiency for large datasets, a dynamic metamodeling method modifies the traditional Kriging technique to improve its efficiency and predictability for smart manufacturing systems. Finally, a super-metamodeling method optimizes results regardless of problem conditions by avoiding the challenge with selecting the most appropriate metamodeling algorithm.

To realize the benefits of all five approaches, an integrated metamodeling process was developed and implemented into a tool package to systematically select the suitable algorithm, sampling method, and combination of models. All the functions of this tool package were validated and demonstrated by the use of two empirical datasets from additive manufacturing processes.

CONTENTS

Page

CHAPTER

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1
INTRODUCTION

The English word manufacturing first appeared in 1683 and was derived from the Latin term of *manu factus*, meaning made by hand[1]. Over thousands of years, humans have produced products from a myriad of raw materials. Wood, stone, metal or anything available to humans have been used to manufacture products. Manufacturing techniques improved gradually over thousands of years until the Industrial Revolution began in England in the 1750s. Since then, the reliance on manual labor has been rapidly replaced by machines, electrical devices and computers. In recent decades, manufacturing became smarter. Since 2011, the term Industry 4.0 has been used to represent the new age of manufacturing techniques, which is well known by its alternative name of smart manufacturing. Smart manufacturing systems include new technologies such as cyber-physical systems, the internet of things, cloud computing, etc [2, 3, 4].

Behind the big picture of the manufacturing revolution, novel manufacturing frontiers, such as advanced and additive manufacturing (AM), are replacing conventional processes in many industries. AM, also well known as 3D Printing, has overturned the conventional wisdom of traditional subtractive manufacturing. AM gives designers numerous opportunities to produce parts with highly complex geometry and/or functional material properties, which were previously near impossible to produce.

However, challenges accompany these emerging capabilities. Engineers responsible for additive and smart manufacturing processes need to fully control the processes, and at the very least predict the performance prior to production. Predictive metamodels are widely used to solve manufacturing problems. Engineers can build computer simulation and statistical models to estimate process parameters, product quality, and manufacturing costs. A metamodel, which is a model of a model, uses a black-box approach to estimate responses from a given complex system. A typical metamodeling process includes data collection, sampling strategy, algorithm selection, parameter optimization, and model validation. Predictive metamodeling methods prescribed procedures and techniques, but some questions remain for these current challenges. Can we solve a new problem using existing knowledge? How can we choose the most appropriate technique based on any given conditions? How should we improve traditional methods to use for specific domains? Can we develop new methods that are suitable for additive and smart manufacturing scenarios?

Additive and smart manufacturing technology has evolved rapidly over the past two decades. Thus, building traditional metamodels in these areas has many challenges. Issues related to the unique features of additive and smart manufacturing include not enough data, highly complex systems and large process uncertainty. This dissertation focuses on developing novel metamodeling methods that can improve model predictability while minimizing cost.

This dissertation will attempt to address the following key questions derived in regard to the additive and smart manufacturing domains:

- How can all accessible information and techniques be integrated to improve predictability?
- How can existing information be used more effectively to create more accurate models at lower costs?
- How can physics-based knowledge and statistics-based information be combined?

- Can we modify traditional metamodeling methods to accommodate additive and smart manufacturing features?

The rest of this dissertation is organized as follows. Chapter 2 reviews and summarizes the relevant literature about additive and smart manufacturing. Chapter 3 introduces the common metamodeling techniques that are frequently employed in this dissertation. These two chapters cover the background for this work. Chapter 4 provides an overview of the metamodeling approaches developed in this work. Chapters 5 through 9 introduce the approaches developed by this research to address the questions posed above. Chapter 10 introduces an algorithm that integrates these metamodeling methods for applicable use. A case study that uses metal additive manufacturing data is presented in this chapter to demonstrate the ease and benefits of using this integrated approach and associated tool package. The last chapter summarizes the dissertation and discusses the potential future works.

CHAPTER 2
ADDITIVE AND SMART MANUFACTURING

Smart manufacturing system (SMS) are defined by the advent of new technologies that promote rapid and widespread information flow within the systems and surrounding its control [5]. In contrast to other technology-based manufacturing paradigms, SMS merges information and communication technologies and combine features of earlier manufacturing paradigms [6]. Industry is, and will continue to be, increasingly reliant on data and predictive analytics to improve overall process efficiencies [7, 8]. The cyber-physical system (CPS) that are deployed in smart manufacturing presents unique features on increasingly reliant on data, interface of tools, data sharing, knowledge integration, and etc. as illustrated in Figure 2.1. CPS is sometimes called 4^{th} generation manufacturing, and it is considered to be supered than many former manufacturing generations such as mechanization and steam power in the first, assembly line and electricity in the second, or computer and automation in the third. These features make it increasing possible as access to technology improves.

2.1 Smart Manufacturing

The Smart Manufacturing Ecosystem encompasses a broad scope of systems in the manufacturing business including production, management, design, and engineering functions as illustrated in Figure 2.2 [6]. As shown in the figure, the dimensions of production (green), production system (blue), and business (orange) dominate

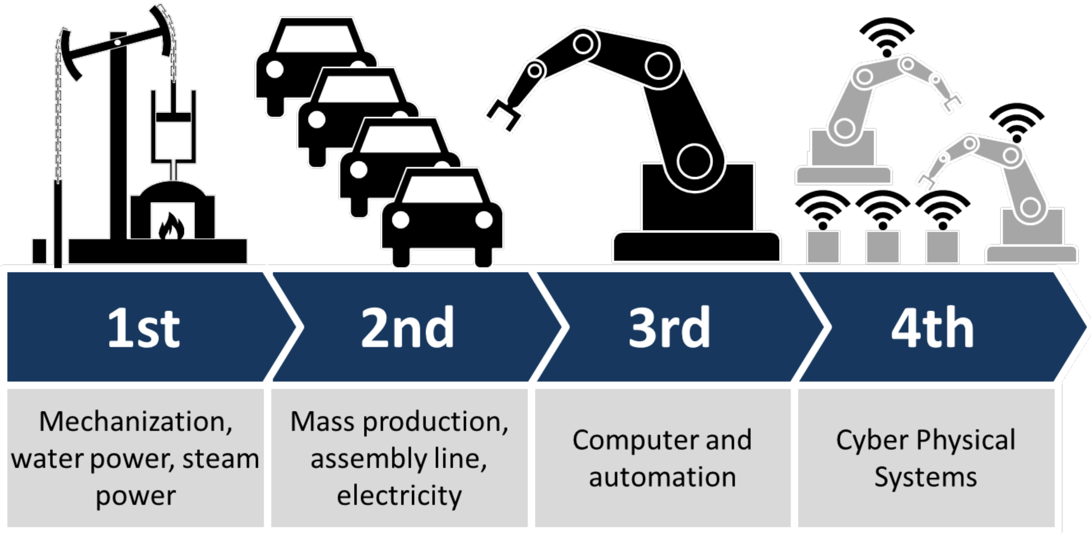

1st
2nd
3rd
4th
Mechanization, water power, steam power
Mass production, assembly line, electricity
Computer and automation
Cyber Physical Systems

Figure 2.1: Industry 4.0, ”Christoph Roser at AllAboutLean.com”

their own lifecycle. Three dimensions are interacted and integrated together by their own functions. For example, the production system tends to focus on design, deployment, operation and decommissioning of an entire production facility. On the other hand, the business cycle is responsible for addressing the functions of supplier and customer interactions. The integration of manufacturing software applications along each dimension helps to enable advanced controls at the shop floor and optimal decision-making at the plant and enterprise. The combination of these perspectives and the systems that support them establish the ecosystem for manufacturing software systems [6].

Such concepts were selected to be reviewed in this section due to their importance to smart manufacturing. CPS, internet of things, big data and cloud manufacturing would be further discussed in the rest of this section. AM, however, would be discussed independently in Section 2.2 as it is one of the most critical topics in my research.

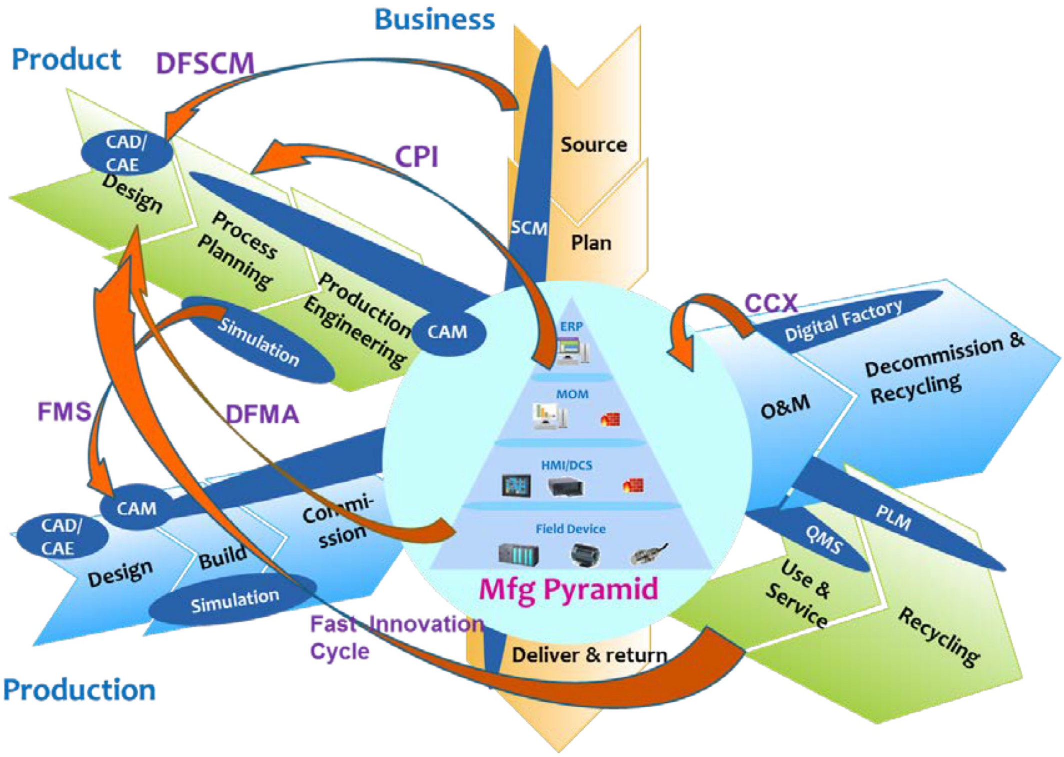


Figure 2.2: Smart Manufacturing Ecosystem [6]

2.1.1 Cyber-physical System

The term cyber-physical systems (CPS) refers to the systems with integrated computational and physical capabilities that can interact with humans through many new modalities. CPS aims to integrate knowledge and engineering principles across the computational and engineering disciplines [9]. More specifically, it integrates the embedded computer systems and networks that monitoring and/or controlling the physical processes. In different to traditional embedded systems, CPS is usually designed as a network of interacting elements with physical input and output instead of as standalone devices [10].

In many years of traditional embedded systems history, such methods and tools were developed by system engineers for solving systems control problems as the time and frequency domain methods, state space analysis, stochastic control, and etc. [11, 12, 13]. Meanwhile, computer engineers have developed a variety of powerful programming languages, real-time computing techniques, visualization methods, modeling formalisms, and verification tools for big breakthrough in this area [14, 15]. These methods and tools are now playing important roles in CPS. However, physical and software components are deeply intertwined in CPS.

In current industrial practices, CPS is integrated with production, logistics and services into smart manufacturing with significant economic potential [16]. It have resulted in higher availability and afford-ability of sensors, data acquisition systems and computer networks. These have further resulted in the continuous generation of high volume data which is known as Big Data due to the ever growing use of sensors and networked machines [17]. Under some environment, CPS can be further transformed to a tool for managing Big Data and leveraging the interconnectivity of machines to reach such goals as intelligent, resilient and self-adaptable machines [18].

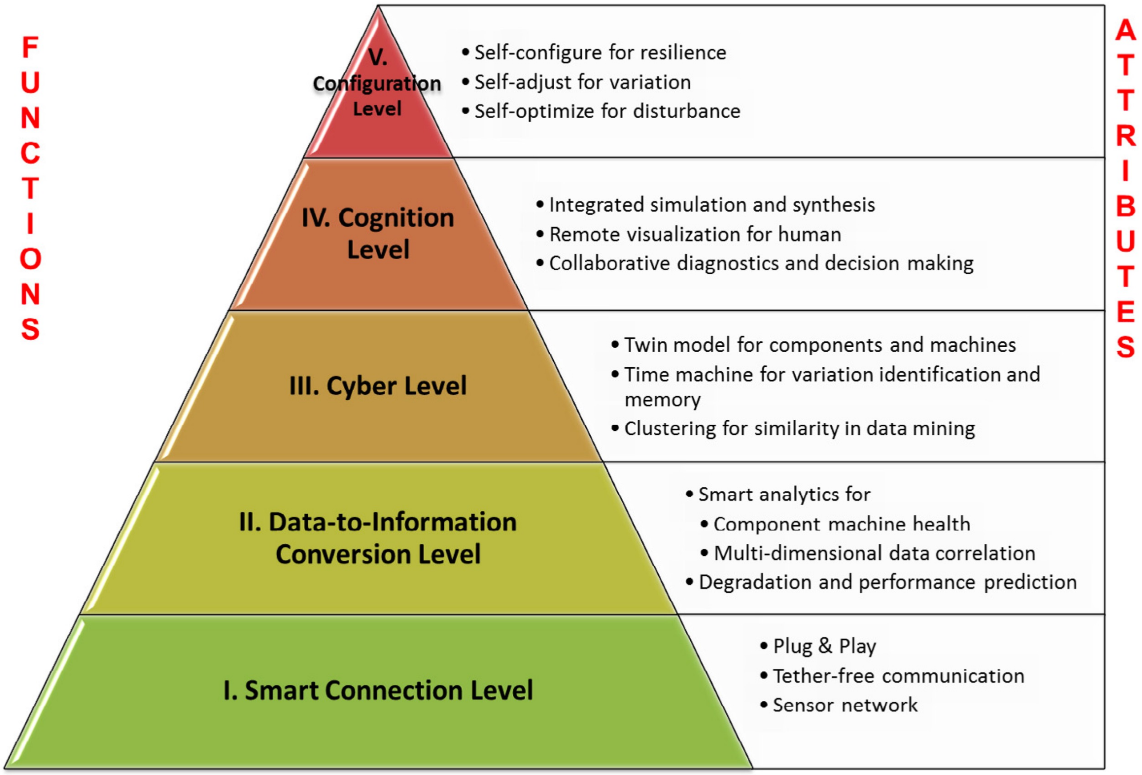

FUNCTIONS
ATTRIBUTES
V. Configuration Level
• Self-configure for resilience
• Self-adjust for variation
• Self-optimize for disturbance
IV. Cognition Level
• Integrated simulation and synthesis
• Remote visualization for human
• Collaborative diagnostics and decision making
III. Cyber Level
• Twin model for components and machines
• Time machine for variation identification and memory
• Clustering for similarity in data mining
II. Data-to-Information Conversion Level
• Smart analytics for
• Component machine health
• Multi-dimensional data correlation
• Degradation and performance prediction
I. Smart Connection Level
• Plug & Play
• Tether-free communication
• Sensor network

Figure 2.3: 5C architecture for implementation of Cyber-Physical System [19]

Figure 2.3 illustrates the detail of a 5-level CPS structure, namely the 5C architecture, suggested by Lee et al's recent publication [19]. The lowest level, called smart connection, is the first step in developing CPS application which by acquiring accurate and reliable data from machines and their components. The second level, Data-to-Information, brings self-awareness to machines by data conversion such as prognostics in health management. The Cyber level acts as central information hub in this architecture as it has absorbed information from every connected machine in the network. At this level, specific analytics are used to extract additional information which aims to provide better insight over status of individual machines. Many data analytics are involved in the Cyber level, such as future performance prediction based on historical information. It requests more accurate and efficient predictive methods to achieve better CPS performance, which is the one major topic of this dissertation. Cognition as the fourth level is responsible for generation of thorough knowledge of the monitored system. It would affect the correct decision and task optimization. For this level, proper info-graphics are necessary to completely transfer acquired knowledge to the users. Configuration, the top level of this architecture, is the feedback from cyber space to physical space and acts as supervisory control to make machines self-configure and self-adaptive.

CPS has many applications that are usually involved with sensor-based communication enabled autonomous systems such as the wireless sensor networks monitoring environment, autonomous automotive systems, medical monitoring, process control systems, distributed robotics, etc. [20, 10, 21]. For these control systems to work properly, CPS usually undergoes verification by extensive simulation, addressing modeling uncertainties, and eliminating random disturbances. However, the integration of various subsystems, while keeping the system functional and operational, has been time-consuming and costly [9]. One example in metal additive manufacturing is a laser sintering control system relies on such different subsystem components of different physical environments as heat source, material supplying, melting, solidification, and post processing subsystems [22]. Each component may have its own software and hardware. Quality of the final part depends on the performance of every integrated components. The increasing complexity of components and the use of different technologies can affect the CPS accuracy and efficiency. According to current state of metal AM product quality, manufactures need a more reliable and cost-effective communicating system for this multi-physics process. The keys for this to work are: 1) accurate data processing and prediction to optimize control parameters; 2) quick response and communication networks that able real-time monitoring and controlling; and 3) insurance of safety, stability, and performance while minimizing computational cost.

2.1.2 Internet of Things

The term of ”the Internet of Things” was coined by Kevin Ashton of Procter and Gamble, later MIT’s Auto-ID Center, in 1999 [23]. Similar to CPS, the Internet of Things (IoT) also involves a lot of works in data communication. The difference is CPS presents a higher combination and coordination between physical and computational elements [24]. On the other hand, IoT is the inter-networking of physical devices, connected devices, buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data [25]. Through unique addressing schemes, these Things are able to interact with each other and cooperate with their neighbors to reach common goals [26].

If say CPS has created a higher combination for multi-physical environment and computational systems. The IoT is the one that allow involved objects to be sensed and/or controlled across exiting network infrastructure for inter-operation [27, 28]. It was believed will play leading role in the near future for industrial, business and private users. It has already involved in the fields such as automation, industrial manufacturing, logistics, business/process management, intelligent transportation, assisted living, and e-health [29]. In these applications, IoT can create opportunities for more direct integration of the CPS system, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention [30]. Figure 2.4 shows the components and layers of IoT for different application domains. These applications involve number of works in data networking and would affect the system performance.

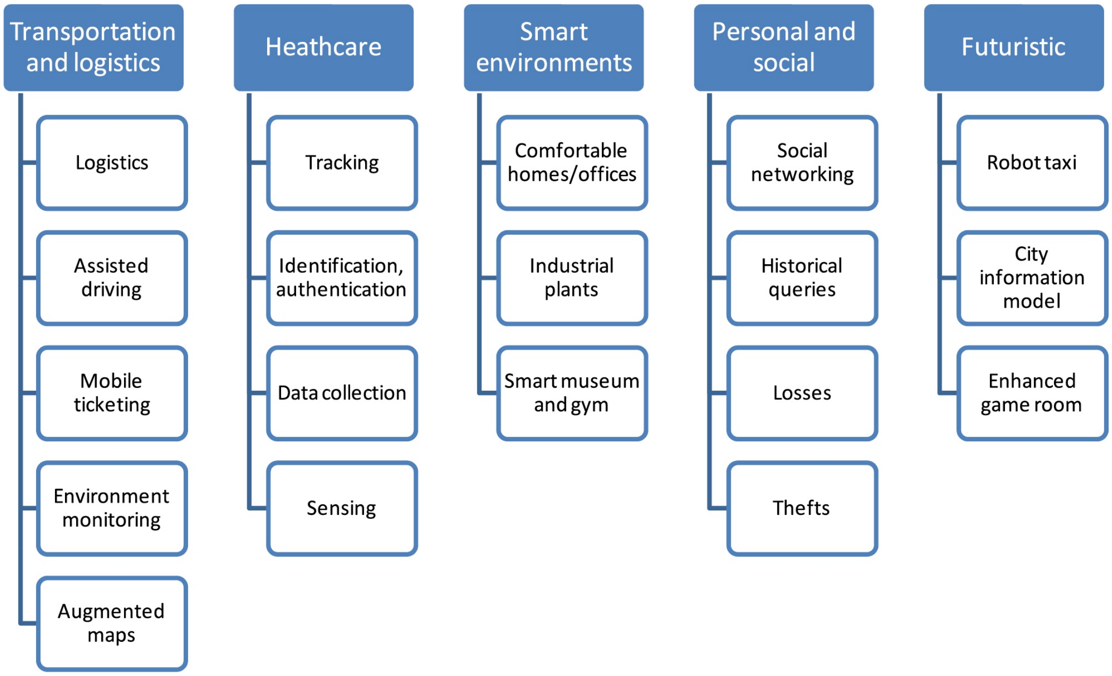


Figure 2.4: Applications domains and relevant major scenarios of IoT [29]

In Transportation domain, advanced cars, trains with roads and/or rails are becoming more instrumented with sensors, actuators, and processing power. Even roads and transported goods themselves, which are the transportation media, are equipped with tags and sensors that send important information to traffic control sites and transportation vehicles to better route the traffic, help in the management of the depots, provide the tourist with appropriate transportation information, and monitor the status of the transported goods. In assisted driving, for example, advanced transportation medias can provide the passengers better navigation and safety based on the important information that collected from the sensors, actuators and processing power. Assistant parking function relies on the data collected from the sensors around the vehicle. Collision avoidance systems can detect the imminent crash and make decision based on the data from radar and laser [31]. For traffic management, more accurate information about traffic pattern can help for better planing.

In Healthcare domain, IoT technologies can solve such issues as tracking of objects and people, identification and authentication of people, and automatic data collection and sensing [32]. Nowadays, doctors can make decision of treatment method based on real-time monitoring data. This process relates to networking between data collection, sensing, and prediction.

Other domains such as Smart environments, personal and social, futuristic applications also can receive benefits from IoT. Smart environments can help in improving the automation in industrial plants by reading the necessary data. Social networking also closely relates to the IoT as the massive data processing that involved in.

Generally, IoT is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine communications and covers a variety of protocols, domains, and applications such as those components in Figure 2.4 [33]. In fact, IoT has different meanings and definitions that may confuse people to follow its major concept. From different orientation, it may include different components [29]. In ”Thing” oriented perspective, it usually refers to variety devices such as monitoring implants, wireless sensors and actuators, streaming live cameras, and other smart items [34]. From this perspective, function of these devices are collecting useful data [35]. However, in ”internet” oriented visions, IoT refers to data flow between ”things”. It builds the connection and allow data communicating in the system. Another oriented vision of IoT is about ”semantic”. In this perspective, IoT refers to a lot of semantic technologies, reasoning over data, and semantic execution environments.

Under either perspective, Iot indicates massive works in data processing. A challenge for producers of IoT applications is to clean, process and interpret the vast amount of data which is gathered by the sensors. There is a solution proposed for the analytics of the information referred to as Wireless Sensor Networks [36]. Accurate and efficient data analytical methodology would benefit to the IoT system especially for those irregular datasets such as defect and big datasets [36]. During this process, data would be collected, stored, and analyzed in certain components. In different to conventional analytical algorithms as typical solution, artificial intelligence (AI) algorithms becomes more and more popular in recent decade. People are developing new AI algorithms for the needs of IoT features [36].

2.1.3 Big Data and Cloud Manufacturing

In every industry and every part of the world, senior leaders always perusing getting full value from the massive amounts of information they already have within their organizations [37]. As mentioned in previous sections, data processing is a key component in CPS and IoT for smart manufacturing. New technologies allow the users to collect more data than ever before. In other words, datasets grow rapidly because they are increasingly gathered by cheap and numerous information-sensing Internet of things devices such as wireless sensors, mobile devices, monitoring equipments, and etc. [38]. Big-data analytics enables continuous innovation and process improvement of manufacturing systems, and has been recognized as a key enabler of smart manufacturing. With a cloud-computing infrastructure, manufacturers gain the ability to access software and real-time data at lower cost and to respond quicker to customer issues. [6].

The terminology Big data indicates the datasets are so large or complex that traditional data processing application software is inadequate to deal with them [39]. After the term was invented and used for couple of years. Lately, big data tends to refer to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of dataset [40, 41]. The analytics of data intends to explain where we are, trace how we got here, and offer an urgently prediction to the benefits and dangers that in the future [42]. However, the term big in big data does not only refer to the volume of data. It also relates to other critical attributes of it such as data variety and data velocity [37]. These three aspects, together, construct the completed definition of big data (Figure 2.5).

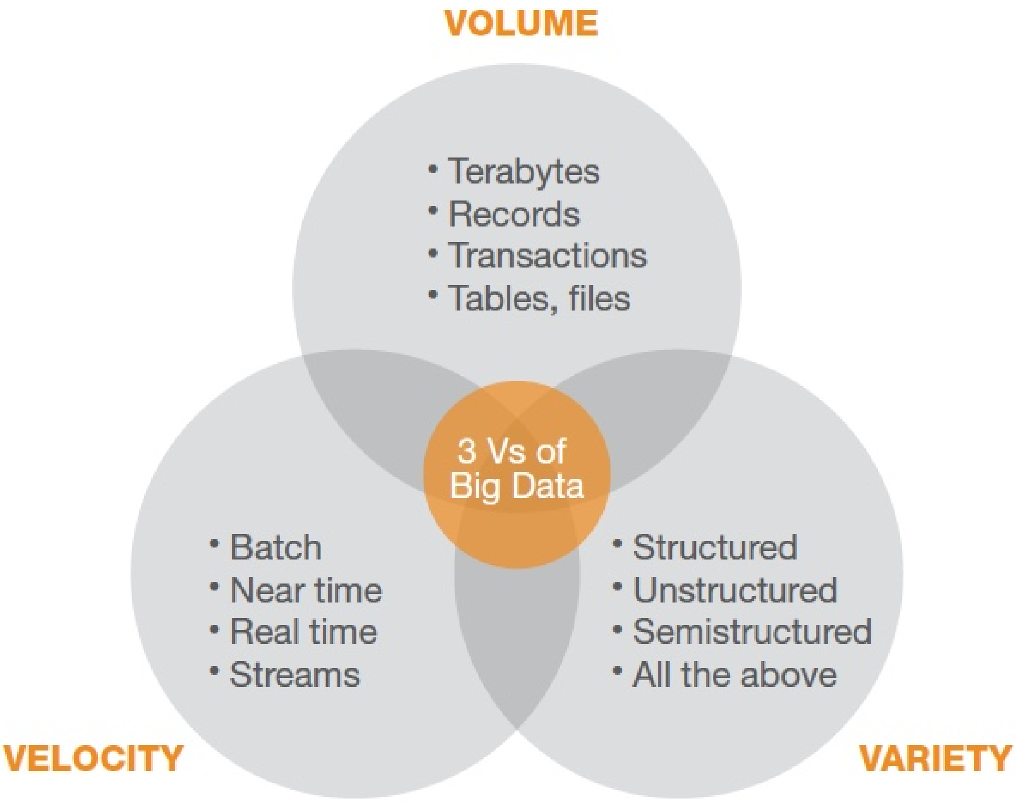


Figure 2.5: The three Vs of big data [37]

Volume, obviously, indicates the size of the data in storage which is the primary attribute of big data. Variety, for example, can correlate to the dimensions or sources of datasets. The great variety of sources, especially from the new technologies such as Web data, can make the data extremely large. As shown in 2.5, it covers both traditional structured data and such unstructured data as text and human language. The variety accompany with volume make data even bigger. Another aspect is so called data Velocity. It usually represents the feeding speed of data processing. For example, it could represent the streaming data from sensors of manufacturing robots and machines or clickstream data from websites. The analytics of these real time data can help the companies make quick responses and decisions to optimize their profit.

Cloud manufacturing is a new manufacturing paradigm developed from existing advanced manufacturing models and enterprise information technologies under the support of smart manufacturing such as IoT and cloud computing [43]. It aims to realize the full sharing and circulation, high utilization, and on-demand use of various manufacturing resources and capabilities by providing safe and reliable, high quality, cheap and on-demand used manufacturing services for the whole lifecycle of manufacturing [44]. The concept of cloud manufacturing refers to big manufacturing that includes the whole lifecycle of a product (e.g. design, simulation, production,test,maintenance) [45].

Cloud manufacturing system mainly includes three category users: provider, operator, and consumer. As shown in Figure 2.6, the providers are those users own and provide the manufacturing resources and abilities that involved in the whole life cycle of manufacturing process. The operators, instead, operate the platform to deliver services and functions other users. The consumers are those who purchase the use of the manufacturing cloud service from the operator on an operational expense basis according to their needs. This process is always based on the knowledge. During this process, different users can search and invoke the qualified MCSs from

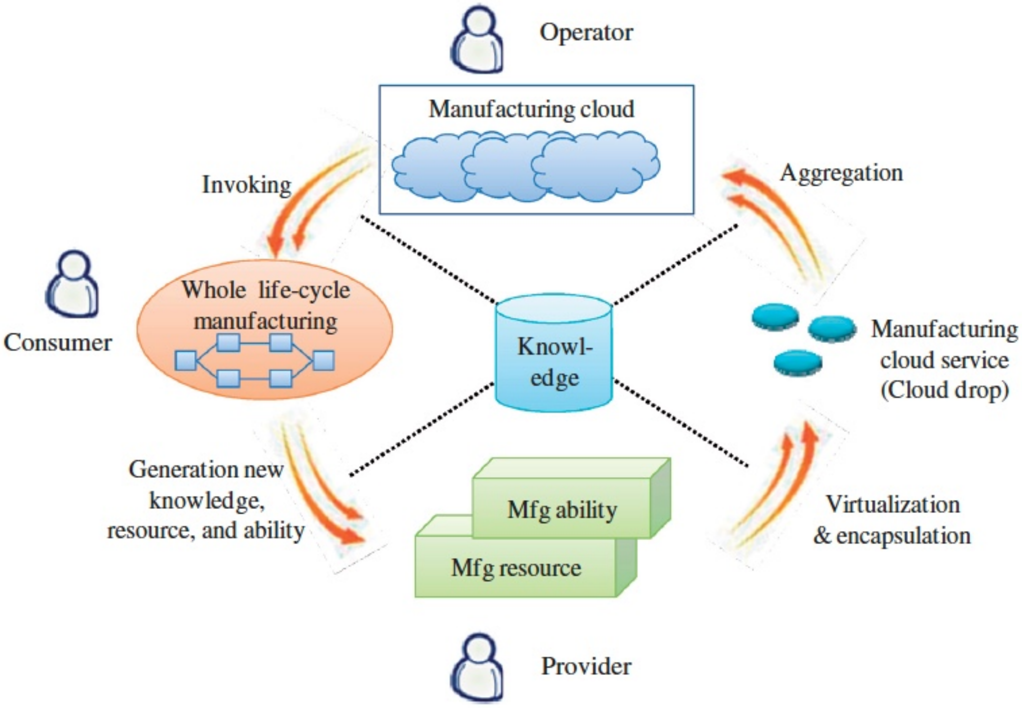

Operator
Manufacturing cloud
Invoking
Aggregation
Consumer
Whole life-cycle manufacturing
Knowl-edge
Manufacturing cloud service (Cloud drop)
Generation new knowledge, resource, and ability
Mfg ability
Mfg resource
Virtualization & encapsulation
Provider

Figure 2.6: Abstract running principle for cloud manufacturing system [44]

a related manufacturing cloud according to their needs, and assemble them to be a virtual manufacturing environment or solution to complete their manufacturing task involved in the whole life cycle of manufacturing processes under the support of cloud computing, service-oriented technologies, and advanced computing technologies.

In general, cloud computing is changing the way industries and enterprises do their businesses. Technology and standards for big data and cloud manufacturing will allow many types of advanced analysis and other functions to be provided on a service basis, thereby making them more readily accessible to manufacturers. [6]

2.2 Additive Manufacturing

Additive manufacturing (AM), which also be well known as rapid prototyping or 3D printing, is an advanced manufacturing technique invented in recent decades. The earliest AM equipment and material were developed in the 1980's for creating models and prototype parts layer by layer [46, 47]. It represents a reverse of the traditional manufacturing method from subtractive to additive. Traditional adding material techniques such as injection molding, casting and welding that remain require pre-shaping mold or parts. However, AM machines finish the building process from nothing to the final part in imported a three-dimensional Computer-Aided Design (3D CAD) data [48]. It is the first time that engineers are able to create almost any shape that can hardly be made by traditional machining methods.

Various materials were introduced to the AM machines for creating parts. Both plastic and metal materials are compatible to AM processes, which give the engineers more design options. Various printing methods were invented and commercialized to the engineers to satisfy their design requirements such as photopolymerization, powder bed fusion, direct energy deposition, material and binder jetting and sheet lamination [48, 49].

AM techniques are exciting and have promising applications in such different domains as aerospace, medical devices, and heavy industry [48]. However, these often multi-physical processes are still not fully understood or controlled [50]. Thus cause issues in the AM parts and processes such as failure printing, low quality, poor life cycle, and large uncertainties. These limitations restrict further application of AM techniques. A well developed analytic and predictive method for AM is needed to overcome these issues, which is one of the primary goal of this research.

The aim of this section it to provide a clear background of AM in order to develop more accurate and efficient AM predictive modeling method. In this section, we briefly review the general AM techniques to discuss different AM processes and materials. Metal powder bed fusion process as the main interest of my research would be further discussed in an subsection with more detail. In this section, we would focus on discussing the major AM processes an parameters.

2.2.1 Additive Manufacturing Techniques

2.2.1.1 AM Process Chain

As shown in Figure 2.7, a generalized AM process usually has eight key steps. Different AM technologies may handle the sequence differently. The eight key steps in the process sequence [48] are:

- Conceptualization and CAD
- Conversion to STL/AMF
- Transfer and manipulation of STL/AMF file on AM machine
- Machine setup
- Build
- Part removal and cleanup
- Post-processing of part
- Application

The first step is a general step for any product development which aims to come up with an idea for how the product will look and function. A generic AM process must start with 3D CAD information, just as its alternative name 3D printing. The second step would transform the CAD model into AM format that the machine is able to read. To date, almost every AM technology uses the STL file format [48], where STL was derived from StereoLithograhy - the first commercial AM technology from 3D Systems in the 1990s [51]. At this step, all construction data and modeling history are eliminated but only the surfaces of the artifact are modeled by series of triangular facets are kept. Now the AM machines are able to read the CAD. However, one should always set the minimum triangle offset to be smaller than the resolution of the AM machine. Otherwise, there would be high possibility of error during the printing process. Once the STL file has been created and repaired, it can be sent directly to the target AM machine. After verification of part, the AM systems usually can usually let the user view and manipulate the part. Typically manipulation includes reposition the part or change of the orientation to allow it to be built at a specific location within the machine. This step would significantly correlate to the final part quality such as relative density [52]. These types effect would be discussed in Section 1.2.3. The next step is about machine setup for both software and physical preparation. Sometimes, the user need to manually setup the building parameters such as the laser power, scan speed, hatch spacing, and etc. When doing physical preparation, the operator needs to make sure sufficient build material is loaded into the machine to complete the build.

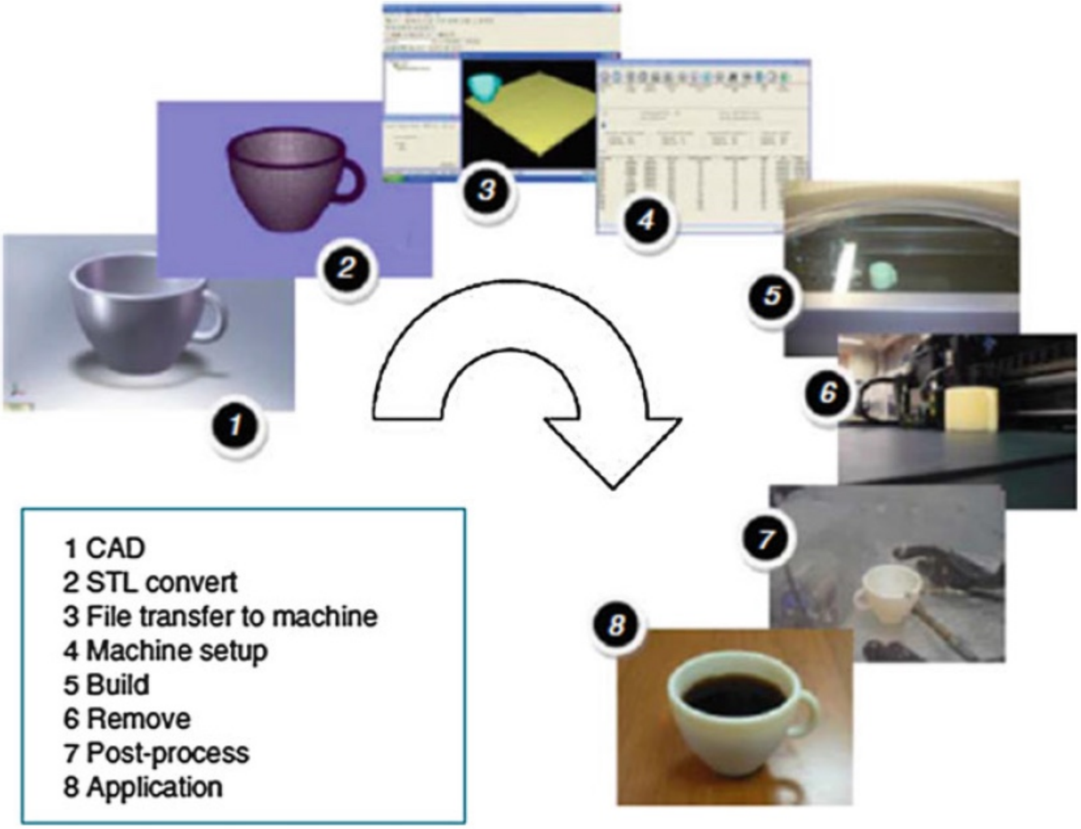


Figure 2.7: The eight stages of the AM process [48]

After these steps are completed, it is now actually going to part building process. Step 5 *build* is usually done by the AM machine automatically as long as no error is detected. During this step, the part is built layer by layer in different ways until the entire process is finished.

Ideally, the output from the AM machine should be ready for use with minimal manual intervention. However, the AM parts sometimes require significant amount of post-processing before they are ready for use. Removal and cleanup together is the initial post-processing step, which need to remove the support material or surrounding powders. Before going to the actual post-process, the part needs to be completely separated from the build platform.

The post-processing may involve abrasive finishing, like polishing and sandpapering, or application of coatings. In addition, some post-processing may involve chemical or thermal treatment of the part to achieve final part properties. For example, the temperature induced phase separation for polymer-ceramic composite powders can increase the alumina concentration significantly by reducing the particle size [53]. Once every above mentioned steps are done, the parts are then ready to use, which can proceed to the application level. Though AM parts are strictly built by these steps, they may not behave according to standard material specifications due to the unique of AM technology [48]. This is a major reason that AM need data and knowledge to well predict the material properties.

2.2.2 Major AM Processes

This section will somes major AM processes, which includes extrusion-based system, material jetting process, binder jetting process, directed energy deposition process, and powder bed fusion (PBF) processes. This review work aims to better understand the physics of general AM processes since the proposed predictive modeling method should compatible with different AM systems. However, as this research was initially founded to study Metal PBF, a detailed review of this AM process would be presented in Section 2.2.3.

2.2.2.1 Extrusion-based Processes

Extrusion-based system is the most popular AM commercial system on current market. While there are other techniques for creating the extrusion, heat is normally used to melt bulk material in a small and portable chamber. The material is pushed through by a tractor-feed system, which creates the pressure to extrude [48]. The general procedure of extrusion-based system is followed as:

- Loading of material
- Liquification of the material
- Application of pressure to move the material through the nozzle
- Extrusion
- Plotting according to a predefined path and in a controlled manner
- Bonding of the material to itself or secondary build materials to form a coherent solid structure
- Inclusion of support structures to enable complex geometrical features

To date, the most common extrusion-based AM technology is fused deposition modeling (FDM) produced and developed by Stratasys [48]. Figure 2.8 shows the basic FDM process. The filament of plastic feeds the machine and be extruded after melted by the print head. The most popular material that used by FDM is acrylonitrile butadiene styrene (ABS) [47]. Other materials such as polycarbonate (PC) is also used. It is believed that parts made by FDM are among the strongest for any polymer-based AM process [48].

FDM is a process in which no chemical post-processing is required, no resins to cure, less expensive machine and materials, resulting in a more cost effective process [54, 55]. However, the low build speed and somewhat bad finishing are its key drawbacks. Speed can be increased by increasing the print head velocity, though it usually reduce the mechanical properties [56]. The correlation between the user

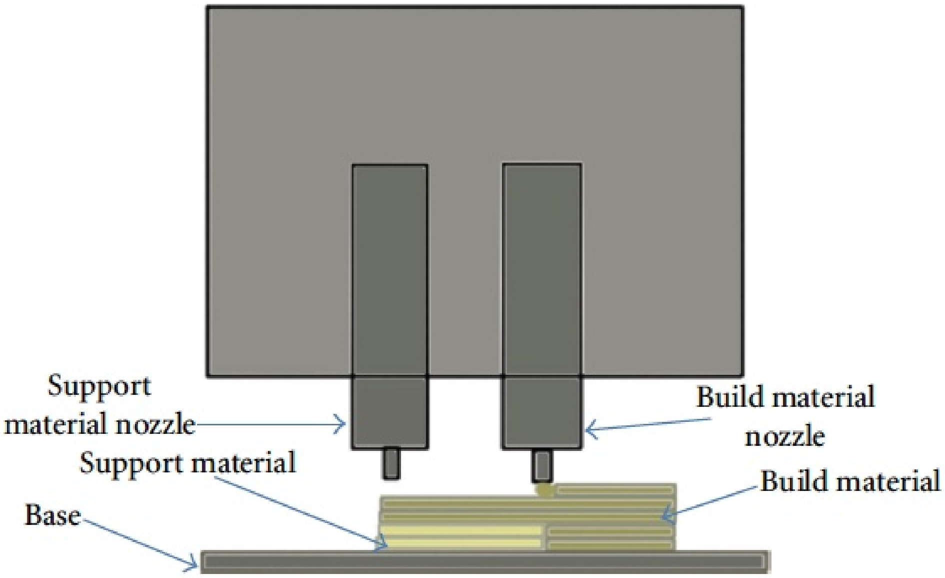


FIGURE 8: Fused deposition modeling.

Figure 2.8: FDM process [47]

inputs as speed and material cost to the outputs as accuracy, roughness, and other mechanical properties is complex [57]. Thus, a accurate predictive model may help the user find the optimal design parameters when using FDM machines. The low material cost but long print time may affect the sampling strategy based on the model developers opinion, which should be considered into future method development.

2.2.2.2 Material Jetting Process

Printing as a three-dimensional building method was first demonstrated in the 1980s with patents related to the development of Ballistic Particle Manufacturing, which involved simple deposition of "particles" of material onto an article [48]. The first commercially successful technology was introduced in 1994, which printed a basic wax material that was heated to a liquid state [58].

During material jetting process, all of the part material is dispensed from a print head which is similar to typical 2D inkjet techniques. The head moves in the x and y axes depositing a photopolymer which is cured by ultraviolet lamps immediately lamps after each layer is finished. Figure 2.9 shows Polyjet process of Stratasys 3D printer [59]. The produced parts by this process can have very high resolution as its layer thickness may achieve 16 μm [47]. Support structures may be needed for certain geometries, which are built in a gel-like material, which is removed by hand and water jetting [59]. Polyjet process is marked as a low cost, high speed, scalability, ease of building parts in multiple materials, and the capability of printing colors [48]. However, the parts produced by this process may be weaker than by stereolithography and selective laser sintering [60].

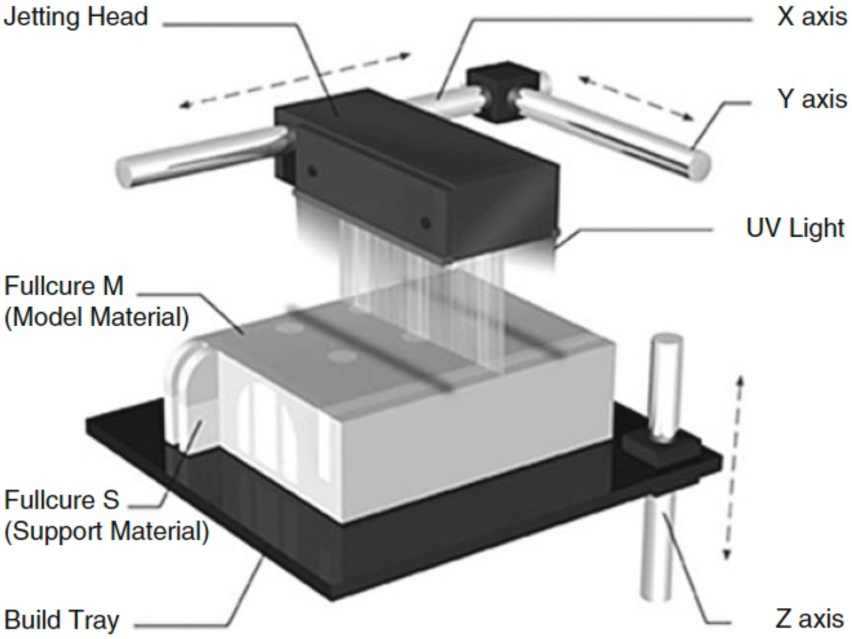


Figure 2.9: Polyjet process [59]

Such studies found the mechanical properties of printed parts by Polyjet process are correlated to printing orientations (Horizontal, Inclined and Vertical) [61, 62]. In these studies, the stress-strain properties and tensile strength vary by different chamber setup methods. Engineers who using Polyjet as a manufacturing method to produce their product would like to know the material properties effect during the designing procedure. Predictive models can help them to address these issues.

2.2.2.3 Binder Jetting Process

In contrast to material jetting that producing the part from a plain chamber, in the binder jetting process, binder or other additive is printed onto a powder bed which forms the bulk of the part. Binder jetting methods were developed in the early 1990s [63]. During this process, the binder is printed onto a powder bed to form part cross sections (Figure 2.10). It can handle a wild range of materials from polymer composite, sand, metals, and ceramic materials [48]. Binder jetting technology can be divided into the following steps: printing, curing, de-powdering, sintering, infiltration, annealing and finishing [64].

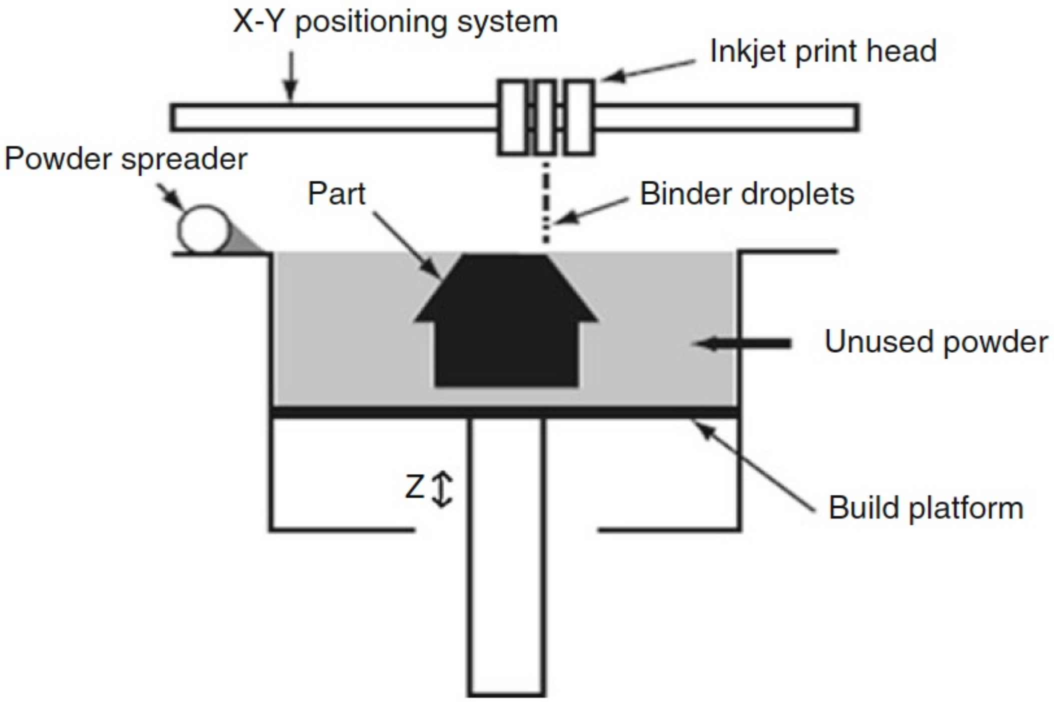


Figure 2.10: Binder jetting process [48]

The binder jetting processes share many of the advantages of material jetting relative to other AM processes. The unique advantages of this process are: 1) fast forming speed since only a small fraction of the total part volume is dispensed through the print heads; 2) it enables the powder and binder materials be combined ; 3) it can achieve higher strength of products as it is compatible with multiple materials.

However, besides the advantages, parts built by binder jetting tend to have poorer accuracies and surface finishes compare to material jetting [48]. As shown in Figure 2.11, the surface roughness of the truss structure is fairly poor [65]. The part might be fail if the cross-section is not properly designed. To avoid this condition, simulation based topology optimization can be applied, which is also a useful application of predictive modeling technique.

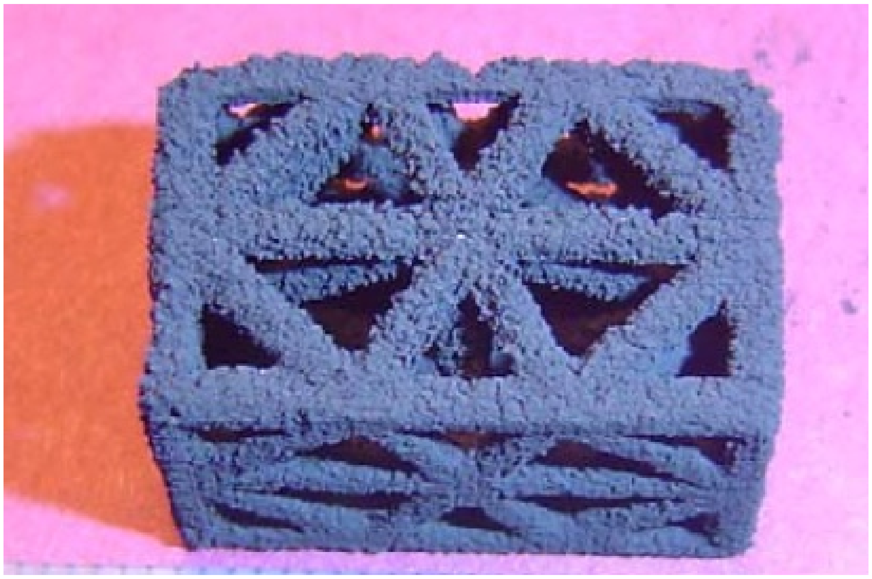


Figure 2.11: Green part by binder jetting process [65]

2.2.2.4 Direct Energy Deposition Processes

Directed energy deposition (DED) processes enable the creation of parts by melting material, mostly used for metal powder, as it is being deposited. Unlike powder bed fusion process in which only laser or binder would be delivered by the print head, the deposition head of DED machine feed both material and laser to the chamber. By DED processes (Figure 2.12), powders are conveyed through the powder delivery nozzle onto the build surface [66]. Meanwhile, a focused laser beam consist the powders. Once particles are deposited, the laser beam provides sufficient thermal energy to melt the particles along the deposition path. Then the melt pool which indicates the heat affect zone is created with different penetration depth. This process is repeated layer by layer to create a solid three dimensional component. DED processes provide engineers a way to produce fully dense parts with highly controllable microstructure and functionally graded components. However, the parts produced by DED is limited by its poor resolution and surface finish. It is difficult to approach accuracy better than 0.25 mm and surface roughness of less than 25 μm [48].

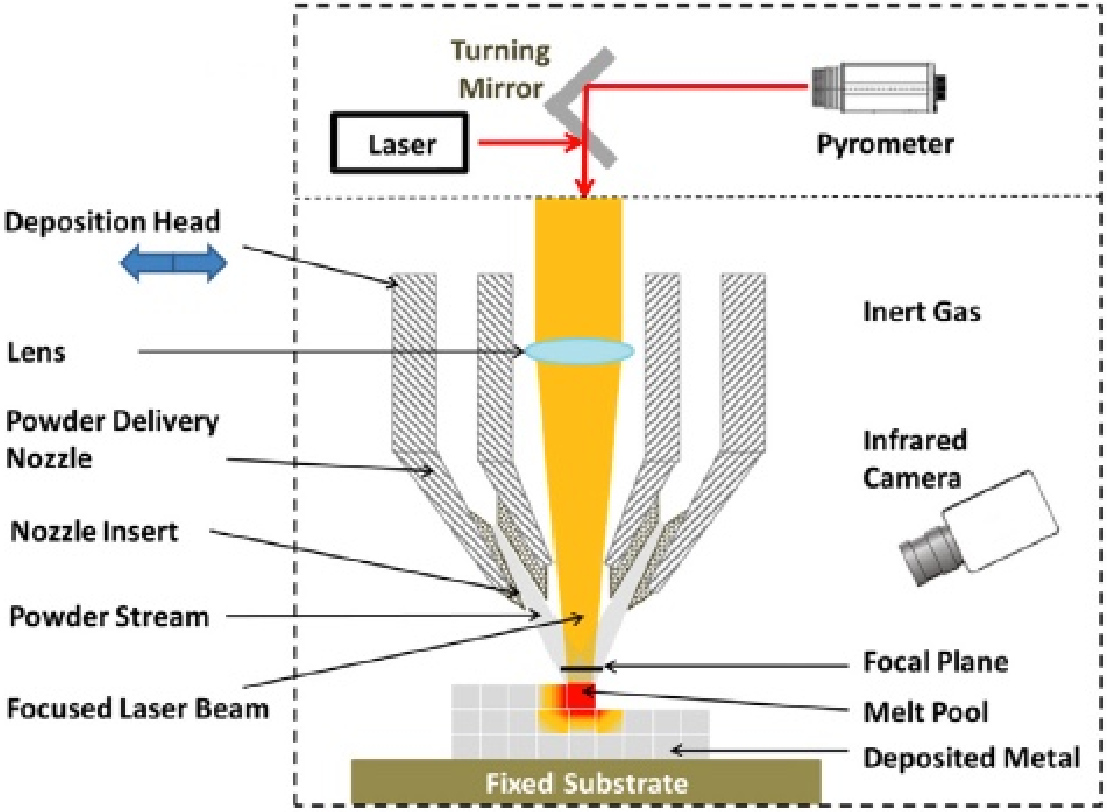


Figure 2.12: Direct Energy Deposition process [66]

The DED process consists of various operating/process parameters. Such operating parameters are user controllable such as laser/substrate relativevelocity (traverse speed), laser scanning pattern, laser power, laser beam diameter, hatch spacing, particle/powder feed rate and inter-layer idle time [66]. However, DED has certain number of random parameters that are not fully controlled as particle size and chamber humidity. All involved parameters can affect the building process and finally change the part properties. The multi-physical environment of DED is very difficult to be simulated and analyzed through traditional methods. Thus, the black-box typed metamodeling method can be a appropriate candidate to provide cost efficient solution.

2.2.3 Metal Powder Bed Fusion Process

The powder bed fusion (PBF) processes were among the first commercialized AM processes [48]. Selective laser melting (SLM) is the alternative name of PBF. These two terms together indicate the key characteristics of this AM process, powder and laser melting. PBF, in principle, can melt and reform any types powder materials. Polymers, metals, ceramics, and their components are all compatible with PBF process. However, in this research, we focus on the metal powder PBF, which the project founded for.

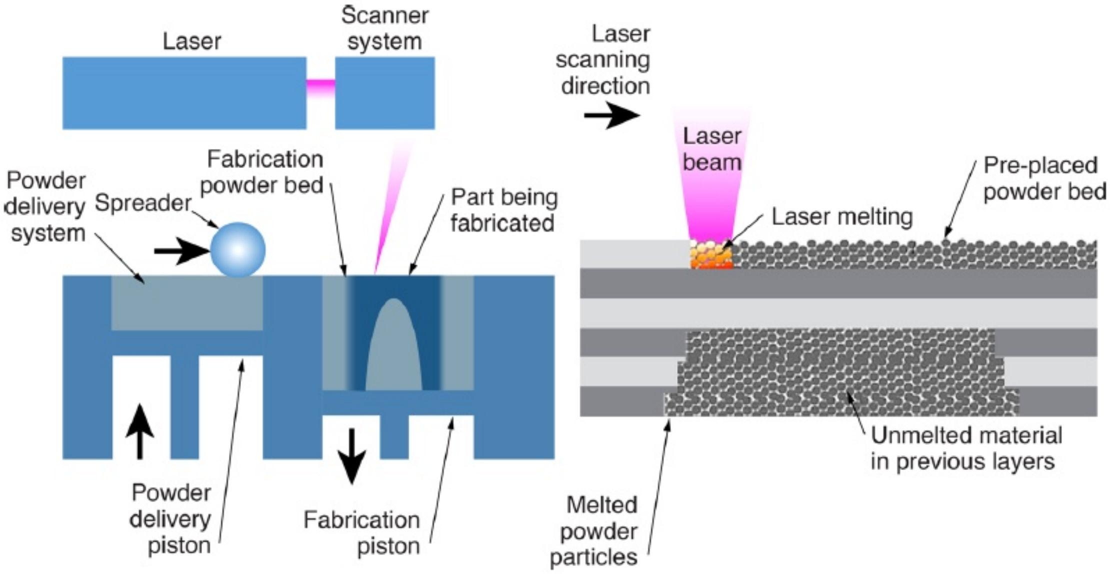


Figure 2.13: Schematic overview of laser PBF process [67]

Typical metal PBF systems have been illustrated in Figure 2.13. Generally, it includes two components: the powder delivery and energy delivery systems. The former one is responsible for supplying resource material. The latter one delivers a focused spot of energy to all points of the build platform [67]. For each layer forming, a scrolling spreader would spread a thing layer of powder on the platform. Once the thin layer has been appropriated preheated, a focused laser beam is directed onto the powder bed and is moved along its path way. It thermally fuses the material to form the cross-section. After this step, the surrounding powder that not fused by the laser remains loose and serves as support for subsequent layers. Preparing for next layer forming, the fabrication position moves down to let the spreader scroll back. Then the top is flatted and the redundant powder is removed. Doing this way, the powder and energy delivery systems can form the part by repeating the above process. Usually, it request a cool-down period to bring the part to a low-enough temperature that they can be handled and exposed to ambient temperature and atmosphere [48]. Finally, the parts are removed from the powder bed, loose powder is cleaned off the parts, and further finishing operations, if necessary, are performed.

Metal PBF process involves a number of subprocess and phase changes. For example, the solid and loose powder firstly change to melted liquid due to the thermal energy provided from the laser. Before forming the next layer, the melted material is solidified, which is usually the case. Witherell and co-authors classified the four major components of PBF process model: Heat source model, heat absorption model, melt pool formation model, and solidification model (Figure 2.14). These models can provide a general understanding of PBF process and how each individual part is composed together.

2.2.3.1 Heat Source Model

This submodel provide the energy source of the PBF process. A laser or election beam move on its designed track on the powder layer. It fabricates the interaction between the beam spot and the powder particles. The laser power, spot

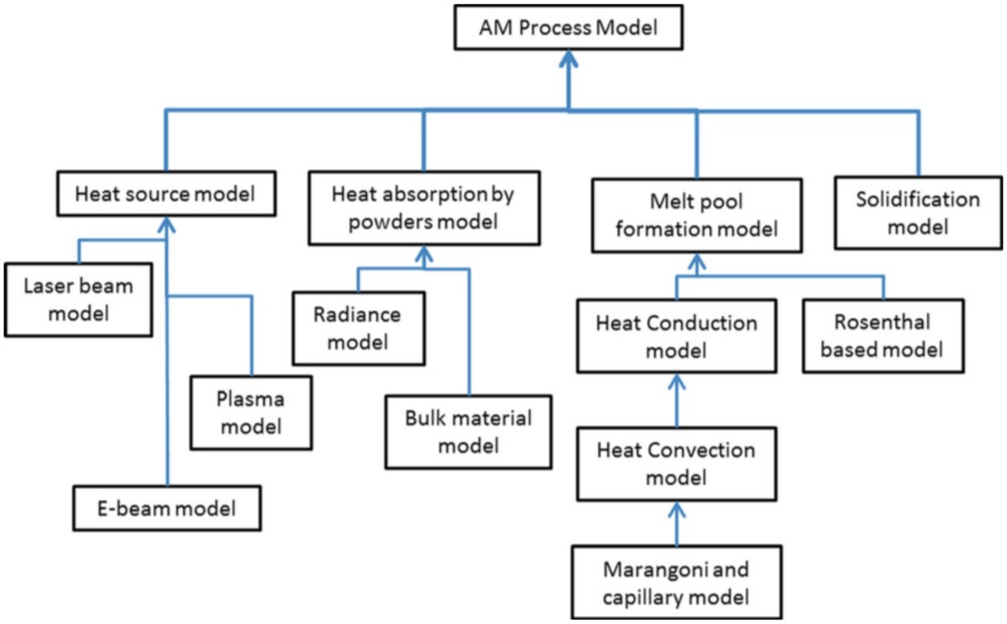


Figure 2.14: Classification of PBF process models[22]

size, and scan speed are key factors to affect the energy input [68]. The heat source model can be further divided into laser beam model, plasma model, and E-beam model based on the different types of energy resources.

2.2.3.2 Heat absorption model

Two factors would decide the energy input to the powder. One is the amount of energy resource that provided in heat source model. The other one is the energy absorption behavior of the powder which depends on many factors such as beam power density, wavelength and pulse duration, irradiation time (or scanning speed), absorptivity of the powder, powder heat conductivity, and oxidation of the powders [69, 70]. Simply speaking, heat absorbed by the powder depends on the heat capacity, temperature difference, and the amount of power [71]. It can also be further divided into the radiance model and bulk material model.

2.2.3.3 Melt Pool Formation Model

The thermal energy from the former two models can cause the first phase change in PBF process, solid powder to metal liquid. Once the melt pool is formed, other material properties such as viscosity and buoyancy effects can lead the following re-solidification process [72]. The formation and re-solidification of the melt pool determines the quality of the scan track. Key factors include melt pool size, temperature distribution and history, fluid flow inside the pool, and cooling time, in addition to material properties (both composition and particle size distribution) [22]. Note, the reheating process may happen in former layers formation due to thermal transmission. Based on current study, melt pool formation model can be further divided into heat conduction model, Rosenthal based model, heat convection model, and Marangoni and capillary model [22].

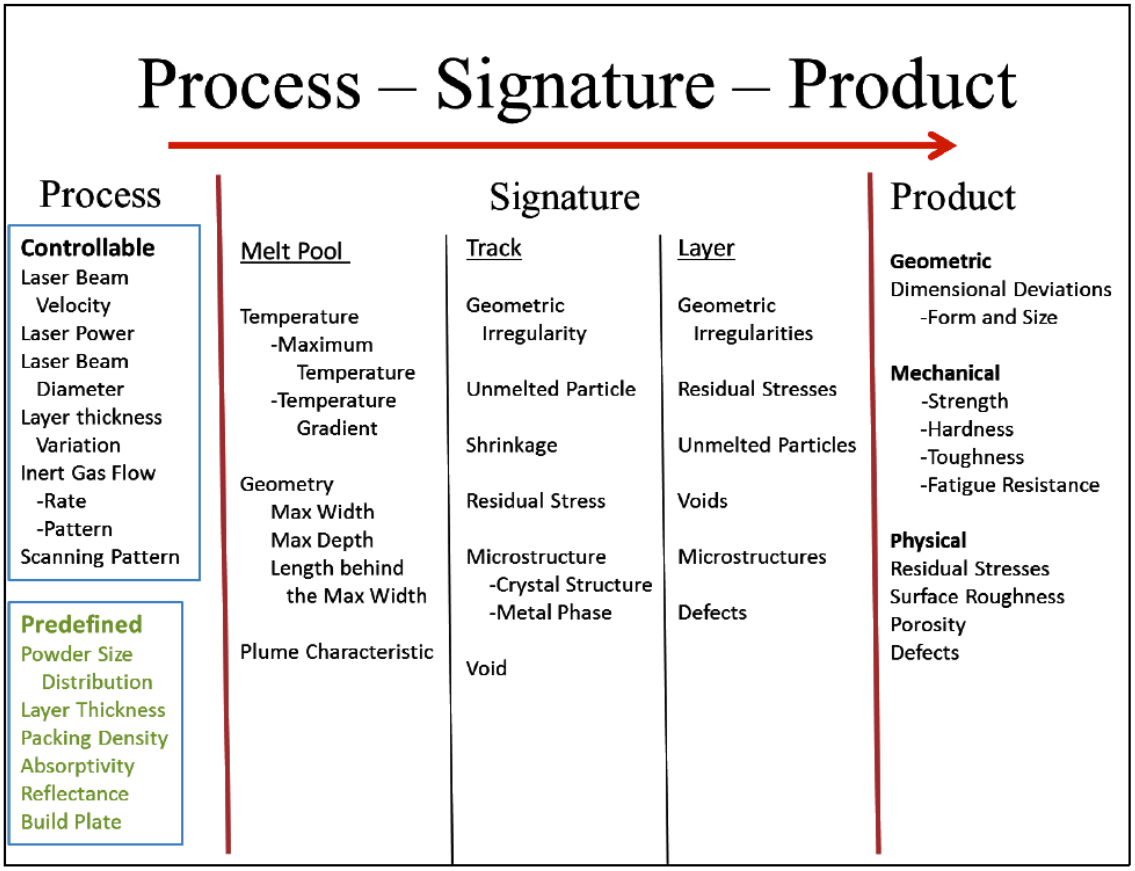
2.2.3.4 Solidification Model

As the energy beam moves away from a certain point in the powder bed, the molten material becomes cool and solidifies. The internal stress accumulation in solidification, a result of heating and cooling, affects the quality of a manufactured part. These accumulation may cause residual stress and other mechanical defects, which might be reduced through post-processing heat treatment [73].

2.2.3.5 Process Parameters of Metal PBF Process

Metal PBF processes involve multi-physical environments. The parameters involved can be generally divided into input, output, control, and environment domains [22].

The input parameters correlate to the properties of powder, material, and laser. The most important powder parameters are: relative density, particle shape and size, shape distribution, thermal conductivity, absorptivity reflectivity, emissivity, diffusivity, and etc. Such material parameters can be: Viscosity, surface tension, capillary force, conductivity, convectivity, specific heat, melting temperature, evaporation temperature, and etc. For the laser, Mode (continuous wave, pulsed), wavelength, intensity profile, average power, peak power, beam quality (how well the beam can be focused), polarity are mainly be considered.



Process – Signature – Product

Process

Controllable
Laser Beam
Velocity
Laser Power
Laser Beam
Diameter
Layer thickness
Variation
Inert Gas Flow
-Rate
-Pattern
Scanning Pattern

Predefined
Powder Size
Distribution
Layer Thickness
Packing Density
Absorptivity
Reflectance
Build Plate

Signature

Melt Pool
Temperature
-Maximum
Temperature
-Temperature
Gradient
Geometry
Max Width
Max Depth
Length behind
the Max Width
Plume Characteristic

Track
Geometric
Irregularity
Unmelted Particle
Shrinkage
Residual Stress
Microstructure
-Crystal Structure
-Metal Phase
Void

Layer
Geometric
Irregularities
Residual Stresses
Unmelted Particles
Voids
Microstructures
Defects

Product

Geometric
Dimensional Deviations
-Form and Size

Mechanical
-Strength
-Hardness
-Toughness
-Fatigue Resistance

Physical
Residual Stresses
Surface Roughness
Porosity
Defects

Figure 2.15: Parameters of PBF processes[74]

The output parameters mentioned here are mostly about the final part properties such as surface roughness, geometric dimension, porosity, residual stress, strength, and etc. There are also some intermediate output parameters that can affect the final output such as melt pool dimensions and melting conditions. Figure 2.15 categorizes and lists the process parameters, process signatures, and product qualities for PBF processes [74]. Such user controllable parameters can significantly affect the product qualities as laser power, scan speed (beam velocity), and layer thickness. Parameters of process signature are sometimes can be used as critical intermediate parameters that correlate to the final product. Most of current physics-based simulation model are focusing on this area. The product includes the AM part qualities that designers care about. However, predicting them can be very difficult. Dimensional deviation, mechanical properties such as strength and hardness have large amount of uncertainties. This becomes to the biggest challenge in AM predictive models. Improve AM predictive model with limited resource under uncertainty is another goal of this research.

CHAPTER 3
METAMODELING TECHNIQUES

3.1 Physics-based Modeling

Predictive modeling is a technique that using existing knowledge to predict outcomes of any type of unknown event. The prediction process can be simple or complex. Engineers can use classical physical theory to predict the structural outcomes of their design. For example, the small deflections of a beam can be predict with its loading conditions using Euler-Bernoulli beam theory. Now ages, finite element analysis (FEA) becomes to a popular numerical method for structural analysis, heat transfer, fluid flow and etc. For example, Yang and co-authors FEA model can predict the pelvic organ dislocation by mimic the abdominal conditions [75].

Along with the boost of computational power in recent decades, physics-based predictive models become to more complex. The models are not limited in one physical environment. In AM domain, for example, a simulation model can integrate such sub-processes as the heat absorption, melting and fusion. Ma and co-authors construct a FEA model to predict the melt pool temperature that simulating the heating and melting conditions during PBF process [76]. A more complex model of metal PBF process was created by Lawrence Livermore National Laboratory that predicts the melt pool data and some manufactured properties [77]. These models are considered as typical physics-based model since their prediction is built on the physical knowledge. According to their complexity, these types of model usually request high computational cost. When knowledge is not fully available, it is almost impossible to set up the model. These limitations encourage the users turn their sight into other less expensive methods 3.1 [78].

Metamodeling technique is an alternative method to make prediction. In stead of using physical knowledge, this method using statistical power to derive the solution. Metamodeling approaches offer an excellent foundation to build predictive analytics as they can establish robust models to realistically represent physical phenomena without intrinsic knowledge of the complex system [79, 80]. Section 3.2 would review the popular metamodeling techniques that may used by this dissertation.

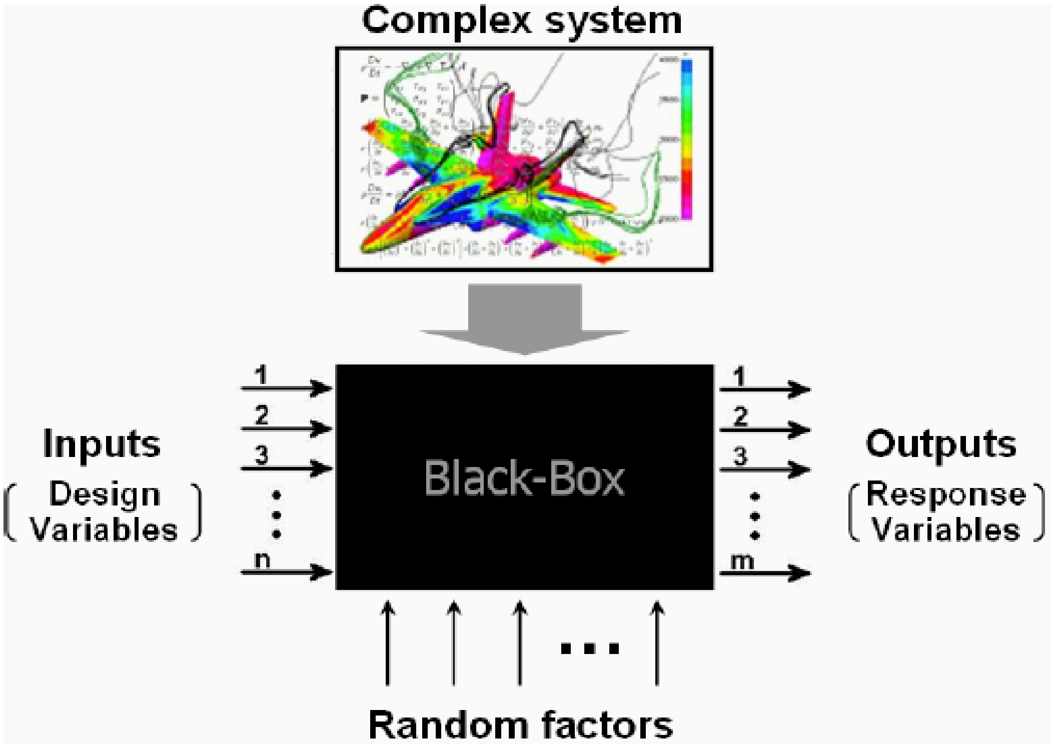


Figure 3.1: Black-box approach for solving complex system[78]

3.2 Metamodeling technique

Metamodels, also known as surrogate models, construct a model of a model to understand complex systems [81] using black-box approach. Unlike physics-based or numerical simulation models that often require detailed knowledge of internal processes, metamodeling techniques focus on the input/output relationships [82]. This data-driven approach is constructed using data which can provide fast approximations of the objects and has been used for design optimization, design space exploration, sensitivity analysis, what-if analysis and real-time engineering decisions [83]. Metamodels can significantly reduce the cost of organizing knowledge for a poorly understood system. Black-box based metamodeling is a useful approach for complex system

There are number of metamodeling techniques that designed for different situation, e.g., Polynomial Regression (PR), Kriging, artificial neural network (ANN), support vector machine (SVR), and radial basis function (RBF). In this section, we would review PR, Kriging, and ANN methods due to their importance in current research. Other metamodeling methods might be covered and employed in the future.

3.2.1 Polynomial Regression

Polynomial Regression (PR) is a form of regression in which a n^{th} order polynomial is modeled to represent the relationship between the independent variable x and the dependent variable y. PR were originally developed to analyze the results of physical experiments and create empirical models of the observed response values [84]. It has been used for more than two hundred years since it was first published in 1815 [85]. Least squares method is usually used to fit PR models. The typical form of PR model is:

$$y(\bar{x}) = f(\bar{x}) + \epsilon \tag{3.1}$$

where $y(\bar{x})$ represents the unknown function, $f(\bar{x})$ is a known polynomial function of $\bar{x}$ derived statistically, and ϵ is random error assumed to be normally distributed. $\bar{x}$ is the set of the system's independent input variables. The response of PR model can be presented as:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_i^2 + \cdots + \beta_m x_i^m + \epsilon_i (i = 1, 2, \ldots, n) \tag{3.2}$$

Which can be expressed in matrix form:

$$\begin{Bmatrix} y_1 \\ \\ y_2 \\ \\ y_3 \\ \\ \vdots \\ \\ y_n \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^m \\ \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ \\ 1 & x_3 & x_3^2 & \ldots & x_3^m \\ \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{bmatrix} \begin{Bmatrix} \beta_0 \\ \\ \beta_1 \\ \\ \beta_2 \\ \\ \vdots \\ \\ \beta_n \end{Bmatrix} + \begin{Bmatrix} \epsilon_1 \\ \\ \epsilon_2 \\ \\ \epsilon_3 \\ \\ \vdots \\ \\ \epsilon_n \end{Bmatrix} \tag{3.3}$$

Its pure matrix notation is:

$$\overrightarrow{y} = X\overrightarrow{\beta} + \overrightarrow{\epsilon} \tag{3.4}$$

Where X is the vector of input variables and $\overrightarrow{y}$ is the response vector. $\overrightarrow{\beta}$ is the parameters vector and $\overrightarrow{\epsilon}$ represents the random errors.

The vector of the estimated PR coefficients can be calculated through ordinary least squares estimation:

$$\overrightarrow{\hat{\beta}} = (X^T X)^{-1} X^T \overrightarrow{y} \tag{3.5}$$

PR model provides the advantage of generating a mathematical function that can easily compute the data location predicted by that equation. However, since PR deploys curve-fitting techniques between the data points, it can tend to smooth out such regions without data to lessen predictive accuracy for highly nonlinear responses of systems [78].

3.2.2 Kriging

3.2.2.1 Overview

The Kriging method has traditionally performed well for many complex problems with high dimensionality and nonlinearity [86, 87, 88]. The theoretical basis for the method was developed by the French mathematician Georges Matheron in 1960. The fundamental assumption of point estimation in Kriging is the estimating point (unknown point) can be represented by observed points (known points) based on spatial correlation [89]. The estimation process is completed by so called spatial correlation functions [90, 91], or the variogram. The variogram is commonly defined as the variance of the difference between field values at two locations across realization of the field [92]. The general form of a kriging estimation for an unknown predicted value of a point Z_E for a single outcome is [91]:

$$Z_E = \bar{Z} + \sum_{i=1}^{n} \lambda_i (Z_i - \bar{Z}) \tag{3.6}$$

where $\bar{Z}$ represents the regional mean value of the response and λ_i is the distance-correlated weight value, which is determined by the computation of spatial correlation. To date, the Kriging method has multiple forms such as simple kriging, ordinary kriging, regression kriging, and etc. [93, 94]. The fundamental difference of Kriging methods is the different assumption of the regional mean, $\bar{Z}$. For example, simple Kriging assumes the model has zero regional mean (i.e. $\bar{Z} = 0$). In contrast, ordinary Kriging is established by the assumption that there will be an unknown constant mean value which can produce the minimum residual estimative error [95].

3.2.2.2 Distance and variance-covariance matrices

Kriging as a spatial statistical method builds its foundation on the observed data points and their correlation. The estimating process first utilizes the given data to construct the distance matrix that the entire Kriging process relies upon. This is critical because distances between data points serve as references of correlation magnitudes. The correlation will determine how much influence is given by one existing data point to others. Another critical matrix, called the variance-covariance matrix, is derived from the distance matrix. It quantifies the spatial correlation for computing the weight factors λ_i in the above equation. Weight factors are the final determinant of the predictive result. For n observed points, the Euclidean distance between points z_i and z_j can be written as $d(i,j) = dis(z_i, z_j)$. As a result, the distance matrix can be formulated as:

$$D = \begin{bmatrix} d(1,1) & d(1,2) & d(1,3) & \dots & d(1,n) \\ d(2,1) & d(2,2) & d(2,3) & \dots & d(2,n) \\ d(3,1) & d(3,2) & d(3,3) & \dots & d(3,n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d(n,1) & d(n,2) & d(n,3) & \dots & d(n,n) \end{bmatrix} \tag{3.7}$$

Since $i = j$, $d(i,j)$ indicates the distance from the original point to itself, the diagonal of this matrix is constantly equal to 0. i.e.

$$D = \begin{bmatrix} 0 & d(1,2) & d(1,3) & \dots & d(1,n) \\ d(2,1) & 0 & d(2,3) & \dots & d(2,n) \\ d(3,1) & d(3,2) & 0) & \dots & d(3,n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d(n,1) & d(n,2) & d(n,3) & \dots & 0 \end{bmatrix} \tag{3.8}$$

Similarly, the vector representing the distance of the estimating point, which predicts the value of Z_E, to all existing points can be formulated as:

$$A = \begin{Bmatrix} dis(Z_E, Z_1) \\ dis(Z_E, Z_2) \\ dis(Z_E, Z_3) \\ \vdots \\ dis(Z_E, Z_n) \end{Bmatrix} \tag{3.9}$$

Note, the distance matrix is always a diagonal matrix since 0 represents zero distance from one point to itself. Once the distance matrix is well established, each correlation $var(Z_i, Z_j)$ can be derived based on the correlation function and its corresponding $dis(Z_i, Z_j)$. The strongest correlation value corresponds to the distance of any point from itself. For demonstrative purposes, we neglect any nugget effect that could reduce correlation [92] and assume a constant regional mean. The variance-covariance matrix (C) is formulated as:

$$C = \begin{bmatrix} var(Z_1, Z_1) & var(Z_1, Z_2) & var(Z_1, Z_3) & \dots & var(Z_1, Z_n) \\ var(Z_2, Z_1) & var(Z_2, Z_2) & var(Z_2, Z_3) & \dots & var(Z_2, Z_n) \\ var(Z_3, Z_1) & var(Z_3, Z_2) & var(Z_3, Z_3) & \dots & var(Z_3, Z_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ var(Z_n, Z_1) & var(Z_n, Z_2) & var(Z_n, Z_3) & \dots & var(Z_n, Z_n) \end{bmatrix} \tag{3.10}$$

Accordingly, the variance-covariance vector for an estimated point to predict Z_E can be formulated as:

$$B = \begin{Bmatrix} var(Z_E, Z_1) \\ var(Z_E, Z_2) \\ var(Z_E, Z_3) \\ \vdots \\ var(Z_E, Z_n) \end{Bmatrix} \tag{3.11}$$

Based on the matrix C and vector B, a weight factor Λ can then be derived from these matrices. For example, the original ordinary Kriging weights may be estimated by:

$$\Lambda = C^{-1}B \tag{3.12}$$

Where $\Lambda = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n]$. In addition to approaching the minimum estimation error, extensive calculus is also required for deriving the optimal weights[96, 97].

3.2.2.3 Spatial correlation function

The fundamental assumption in kriging method is that the correlation between points can be estimated by spatial correlation functions (SCF) [98]. The SCF for an n dimensional problem in Gaussian distribution form is:

$$R(\theta, x_i, x_j) = \prod_{l=1}^{n} exp(-\theta(x_{i,l} - x_{j,l})^2) \tag{3.13}$$

where $x_{i,l}$ is the l^{th} component of the i^{th} vector x_i and $x_{j,l}$ is the l^{th} component of the j^{th} vector x_j [98]. $R(\theta, x_i, x_j)$ depends on the location of points x_i and x_j, and the correlation parameter, θ. The optimal correlation parameter can potentially improve the predictive accuracy by providing more accurate correlation between all given data points [99]. For example, the popular DACE toolbox uses the modified Hooke and Jeeves (H-J) algorithm [100], which is efficient but unable to provide the true optimum [101]. The Levenberg-Marquardt (L-M) method can efficiently find the local optimum by employing a scoring method to calculate the Hessian matrix for optimization [102]. Alternatively, Forrester and Keane use the genetic algorithm (GA) to approach the global optimal correlation parameter without considering computational cost [103], which could be incompatible with very large data sets. Optimization of the correlation parameter is a critical process for improving predictive accuracy. Table 3.1 lists the regular formation of correlation functions [91].

3.2.3 Artificial Neural Network

Artificial Neural Network (ANN) is a computational algorithm that mimics the central nervous system [104] and has been widely used for solving problems with complicated structures. A typical ANN model consists of an input layer, hidden layers, and an output layer [105]. Each layer consists of "neurons" that are connected across layers to transmit and deduce information. The optimal number of neurons

Table 3.1: Kriging correlation functions, $d_j = |x_i - x_j|$

Type	$\mathbf{R(\theta, x_i, x_j)}$
Gaussian	$exp(-\theta_j d_j^2)$
Exponential	$exp(-\theta_j d_j)$
Linear	$max 0, 1 - \theta_j d_j$
Spherical	$1 - 1.5\sigma + 0.5\sigma^2$
Cubic	$1 - 3\sigma^2 + 2\sigma^3$
...	...

and hidden layers may differ, and depends on the complexity of the problem. The structure of a simple ANN model [106] is shown in Figure 3.2, where x_1 to x_3 are input parameters, $u_1 - u_4$ are the neurons in the single hidden layer, and outputs y_1 and y_2 are located in output layer. $w(i, h)$ represents the weight factor for the link between input i to hidden node h. $x(h)$ represents the weight factor from hidden node h. Here, N is the total number of input variables. The estimated output $\hat{y}_i$ is produced by:

$$\hat{y}_i = \sum_{j=1}^{N} w_i(j) x(j) \tag{3.14}$$

The optimal weight factors can be derived after training the neural network using actual output y_i:

$$w_i(j) \leftarrow w_i(j) + \mu(\hat{y}_i - y_i) x_i(j) \tag{3.15}$$

when using backpropagation:

$$x(h) \leftarrow x(h) + \mu \sum_{p=1}^{n} (y_p - \hat{y}_i) v_p(h) \tag{3.16}$$

where μ is the step size of each iteration. Here, $v_p(h)$ is the actual value of the h_{th} hidden node, using the sigmoid function, is given by:

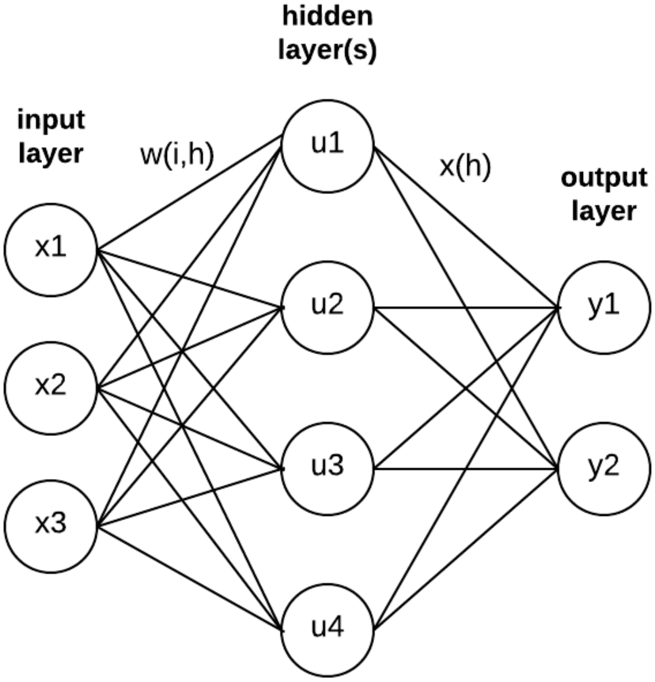


Figure 3.2: Typical structure of a simple ANN model

$$v_p(h) = \frac{1}{1 + e^{-v_p^*(h)}} \tag{3.17}$$

where $v * (h)$ is equal to:

$$v_p^*(h) = \sum_{j=1}^{I} w(j,h)u_p(j) \tag{3.18}$$

Each weight $w(i,h)$ is updated using:

$$w(i,h) \leftarrow w(i,h) + \mu \sum_{p=1}^{n} (y_p - \hat{y_p})x(h)v_p(h)(1 - v_p(h))u_p(i) \tag{3.19}$$

where $\hat{y_p}$ is the predictive result from neural network and y_p is observed value from given data.

3.3 Uncertainties of Modeling Metal PBF Process

This section would mainly discuss about the uncertainty in Metal PBF process on the origin and propagation. Uncertainty can be generated from four major sources: modeling assumptions, unknown simulation parameters, numerical approximations, and measurement error in calibration data [107]. PBF processes involves multiple physical phenomena. The melting and solidification process are occurring cycling. Such uncertainty sources in PBF processes are [108, 109, 76]:

- Particle-level dynamics neglected in continuum models
- Active laser power inaccurately distributed on the powder bed
- Inadequate choice of environment conditions that ignoring the path-to-path and layer-to-layer interactions

Some uncertainties are highly correlated to the input variables of PBF simulation and experiment [67]. For example, the laser beam size may not consist during the building period as the heat can affect the optics. The properties of the powder bed may vary due to the inconsistent distribution of powder particles. This may originally caused by the powder or may occur after powder reuse. The calibration of the laser power, speed, and beam size may change over time. All these variations will influence the properties of the part being built. Thus, generate a certain number of uncertainties.

The uncertainties generated from previous steps may also raise the uncertainty in following steps, which so called uncertainty cumulation. For example, the temperature evaluation will result in an uncertainty of the measurement of melt pool width, depth, and length. Better understanding of measurement uncertainty assists system controller design by identifying the necessary level of precision required to attain the goals of the control system [74].

CHAPTER 4
PREDICTIVE ANALYTICS FOR AM AND SMS

The primary goal of this research is to build high fidelity predictive models for AM and SMS using preexisting datasets. Building on background covered in the previous chapters, the following five chapters introduce several metamodeling methods developed to advance the state of the art in various ways. This research addresses several major challenges. First, it is difficult to select an appropriate metamodeling algorithm with limited knowledge or information. Secondly, inflexible preexisting datasets are often not compatible with advanced sampling methods such as space filling sampling (SFS) or sequential infilling sampling (SIS). Third, a domain-specific metamodel can only partially represent an overall system. Forth, traditional metamodeling methods such as Kriging have limited capabilities for SMS applications. Finally, traditional metamodeling approaches are often not well suited for non-ideal data. This chapter provides an overview of these issues and potential solutions.

The main process of creating a metamodel is shown in Figure 4.1. Each step has unique challenges when using previous AM or SMS datasets. To begin, one needs to analyze the problem and select the best metamodeling technique. Input and outputs variables need to be identified at that stage. Constraints such as the sample size and computational cost should be specified. The appropriate sampling method needs to next be selected. Design of experiment (DOE) techniques are generally deployed as a basis for empirical studies [110]. For computer experiments, SFS and SIS become more practical [111, 63]. For the step of algorithm selection, generating

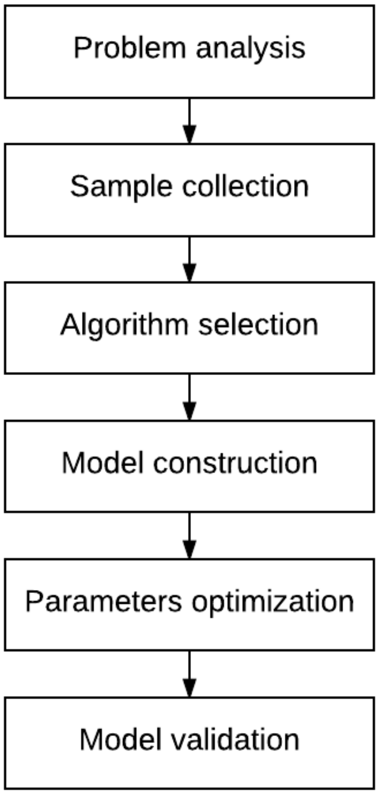


Figure 4.1: General steps for building metamodel

and testing metamodels from all candidate algorithms may be done if the data is simple and fast to use. Alternatively, one can eliminate the non-practical candidates based on prior knowledge and information. Model accuracy and efficiency are key criteria to optimize during model construction. Parameter optimization can improve a metamodel when additional data is available beyond that used to build the initial metamodel. Finally, model validation measures the performance of a built metamodel. Table 4.1 lists the current potential limitations of each step for the AM and SMS domains.

Table 4.1: Limitations of conventional metamodeling approaches in AM and SMS domains

Step	**Action**	**Challenges**
Problem analysis	Define variables	Given conditions do not match what is desired
	Set conditions	
Sample collection	DOE, SFS, or SIS	Optimal sampling strategy does not match the historical data
	Collect data	Data points requested does not exist
Algorithm selection	Analyze the unknown system	Not enough knowledge and/or information
Model construction	Improve efficiency	Traditional methods not suited for AM or SMS data
	Find convergence	Overestimate or underestimate due to non-ideal data
Parameters optimization	Use additional data	Unable to modify or expand historical data
Model validation	Need new data in given design space	Unable to replicate historical experiments

A significant challenge with reusing a dataset is that the given data points may not have desirable locations. The given data is collected based on a specifically designed sampling method which matches the conditions of the original problem. Conditions such as the design variables, sample size, experimental conditions, and initial uncertainties can significantly affect the metamodel performance. As a result, the metamodel built by prior data may not completely suit a new problem. For example, design variables among datasets might be inconsistent. The data collected from different prior experiments can be partially overlapped or nonoverlapped when combining multiple datasets. The desired data points may not exist in the given dataset. In addition, it is hard to select an appropriate metamodeling algorithm if any important information is missing. The desired predictability could be unobtainable without enough data. For the issue of not enough data, collecting additional data points may provide the solution. However, it is difficult to completely mimic the original experimental conditions. Thus, a technique is needed to combine data or models generated under inconsistent conditions. A very big dataset can reduce the model efficiency if a conventional metamodeling method is deployed to remedy situations of disparate or inadequately small sample sizes.

The following five chapters aim to address these issues by developing specifically targeted metamodeling methods. In Chapter 5, a domain-driven metamodeling selection algorithm uses limited knowledge to select the most appropriate method before actually building the metamodel. Chapter 6 introduces the Maximum Predictive Error Updating (MPEU) Method which can improve the sampling strategy when using small sized previous data. The grey-box metamodeling method introduced in Chapter 7 addresses the issue of inconsistent experimental conditions between multiple prior datasets. Chapter 8 modifies the traditional Kriging method to improve the model predictability and efficiency for SMS datasets. An algorithm integration method called a ”super-metamodeling method” is presented in Chapter

9. It maximizes the metamodel predictive accuracy by optimally combining different metamodeling algorithms.

These methods can work independently to address each of the issues listed in Table 4.1. They can also solve the overall suite of problems when integrated together in a comprehensive process. To achieve this goal, we systematically developed a usable toolbox developed in Matlab that integrates these methods via various function calls that are demonstrated to be robust and easy to use. This tool package is introduced and discussed in Chapter 10.

CHAPTER 5

DOMAIN-DRIVEN METAMODELING APPROACH

5.1 Overview

The first step to build a metamodel is to select the appropriate metamodeling algorithm. The selected algorithm should be able to provide higher predictive accuracy and better computational efficiency. This process could be done statistically after complete the data collection through characterizing the data features. This method enable the model developers have deep insight to the data. However, the working sequence can limit the application of this method. One important limitation is the model developers may find they have run more experiments than what needed after establish the metamodel. This is because of the lack of general idea to the unknown system in advance the DOE process.

Test samples of metal AM are usually very expensive. So AM user would need a model selection method which can be operated prior the DOE process. Recent studies have shown advantages to utilizing metamodeling techniques to mimic, analyze, and optimize system input-output relationships in AM. In this chapter, we would investigate the way to use the physics-based knowledge and historical data to select the most appropriate metamodeling method for new problem. We would address a key challenge in applying such metamodeling methods, namely the selection of the most appropriate metamodel. This challenge is addressed with domain-specific

AM information, derived from physics, heuristics and prior knowledge of the process. Domain-specific input/output models and their interrelationships are studied as a basis for a domain-driven metamodeling approach in AM. A metamodel selection process is introduced that evaluates global and local modeling performances, with different AM datasets, for three types of surrogate metamodels PR, Kriging, and ANN. A salient feature of this approach is its ability to seamlessly integrate domain-specific information in the model selection process. The approach is demonstrated with the aid of a metal PBF case study and the results are discussed.

5.2 Background

Previous modeling selection frameworks have used information about data, called data features, to identify the optimal modeling algorithm for certain types of problem. ”Data features” in this context are considered characterizations of how each parameter impacts the responses in a given domain. For example, Rice et al. proposed a model can use extracted characteristics and performance measurements such as normalized root mean square error (NRMSE) and maximum relative error magnitude (MREM) from a given dataset to select the optimal modeling algorithm from a set of candidates [112]. Cui et al.'s energy model recommendation framework uses dual performance evaluation criteria and criteria reduction methods to implement a meta-learning procedure for modeling algorithm selection [83]. While these well-established methods work for some general problems or problems in specific domains, they usually require significant work in data feature characterization.

This chapter aims to address AM predictive modeling challenges by constructing a specific-to-AM framework that looks to efficiently and accurately identify optimal metamodeling methods for given problems, prior to deeply investigating data features related to a specific domain. The term “domain” in this chapter indicates the topic area to which the parameters apply. For example, laser power is a parameter in the AM thermal domain and powder density resides in the AM material domain, etc. In the following sections, we will first discuss the performance of metamodels with different domain-inspired AM input/output parameters. Unlike Rice's model that uses data features for algorithm recommendation, the method proposed in this chapter focuses on using correlations between AM parameters to efficiently identify data features. An AM input/output correlation chart was developed to visually present the nonlinearity of different combinations of parameters. We introduce the domain-driven framework in Section 5.4. For demonstrative purpose, a case study based on AM datasets is presented in Section 5.5 The benefits of using this domain-driven approach are explained and further discussed in these sections.

Section 5.3 discusses parametric correlations in AM models based on a detailed literature review. The remaining sections are built upon the findings from Section 5.3. The three candidates presented in this paper are used to demonstrate the framework. The fundamental structure of the framework and an illustrative case study with empirical data from an AM process are introduced in Sections 5.4 and Section 5.5. Section 5.6 concludes with a comprehensive discussion of the early framework.

5.3 Analysis of the Correlation between Input/Output of PBF Process

This section investigates a metamodel selection method that leverages pre-existing knowledge of parametric relations instead of pure data analysis. For an identical system under the same conditions, one might perceive that the basic relation between inputs and outputs is unchanged. The following beam bending example is introduced to explain this hypothesis.

In the general case of any beam, when an analysis of beam bending is desired many of its characteristics may be unknown. For example, consider a large beam made of lattice of different orders of magnitude (Figure 5.1) with a load applied. Measurements of loading conditions can be taken, but not enough information can be obtained to perform a complete analysis due to the presence of too many variables.

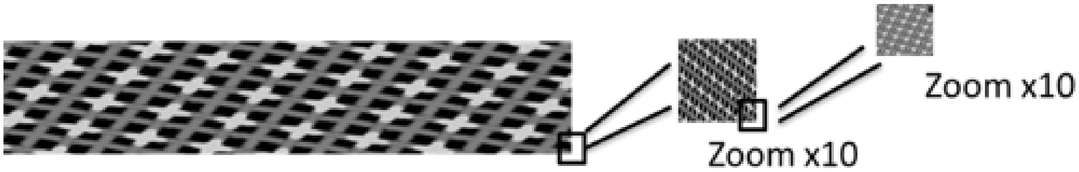


Figure 5.1: Beam created with lattice

However, given the loading and constraints, we can observe that the case behaves like a bending beam. While only partial measurements can be taken at micro and meso scales, we can make the hypothesis that it will perform according to beam theory at the meso and macro levels, thus making domain-specific observations. These observations allow us to extrapolate measured values, based on a combination of measured results and expected macro performance.

A similar situation is found in metal PBF processes in that they are very complex and involve a large number of parameters (more than 50) [48, 113]. The input/output relation is difficult to discern using theoretical analysis. If one considers every single PBF parameter, there are millions of combinations of different models. Thus, we introduce a domain-driven framework that uses past model performance to predict the appropriate modeling method for a new problem. The rest of this section will summarize parametric relations between AM parameters from the literature to construct the prerequisite knowledge for development of a model selection framework.

Figure 5.2 shows 8 inputs and 6 outputs that have high occurrence in recent PBF literature. The relation between these parameters is currently marked as unknown due to lack of information.

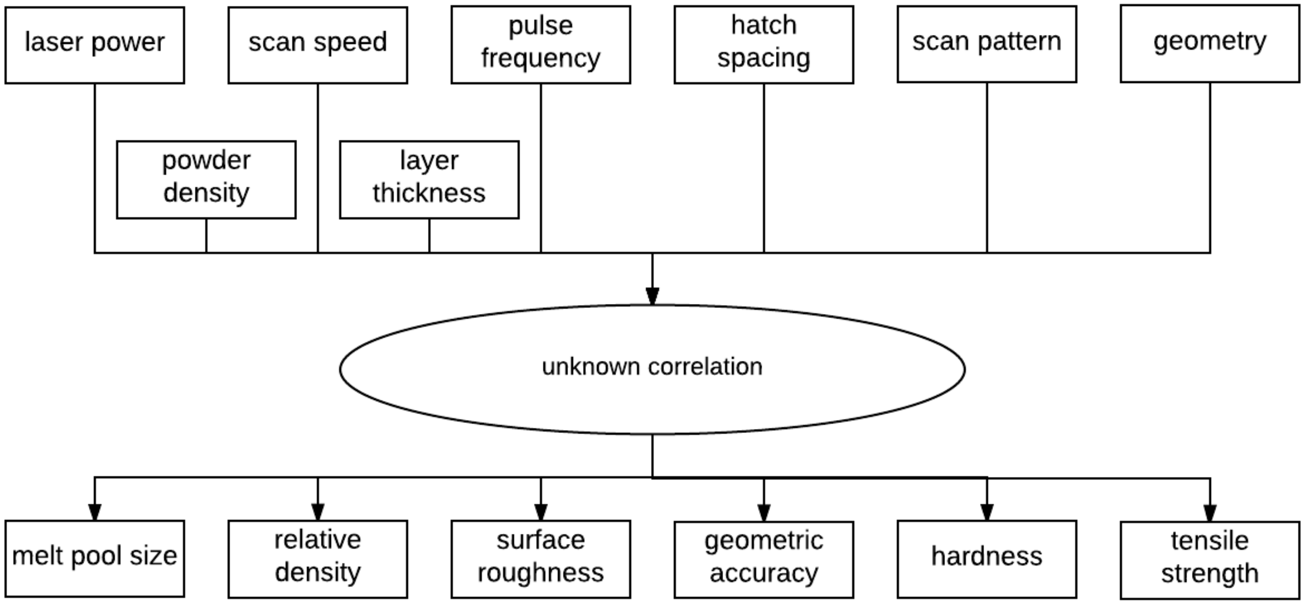


Figure 5.2: Infrastructure of principal AM inputs/outputs

For single input cases, PBF parameters such as laser power, scan speed, and hatch spacing have highly linear correlations to certain outputs [114, 115]. For example, Tang et al (2003)'s metal laser sintering experiment indicates the surface roughness and tensile strength is linearly increased with laser power [116]. Similarly, tensile strength decreases monotonically with higher scan speed with other parameters held constant. From the same study, however, surface roughness was not linearly related to scan speed. With reduced layer thickness, scan speed and roughness instead had a slight nonlinear relation. Another study, using different materials but the same laser parameters, found scan speed and layer thickness have a linear relation to relative density when varied individually [117]. Such similar results using multiple materials suggest that relative density is linearly related to laser power, scan speed, and layer thickness [118]. Intermediate outputs such as penetration depth (not included in Figure 5.2 have also been shown to exhibit a linear relation with some parameters. Kruth et al (2003)'s laser sintering experiment found higher scan speeds produced a linearly decreasing layer thickness [119]. Some inputs, such as laser pulse frequency, have been found to have a highly nonlinear relation to relative density [120].

Compared to the single input/output problem, the relations between variables become considerably more complicated when studying multiple input parameters. Tang et al (2003) found that the surface roughness is not linearly related to a combination of laser power and hatch spacing [116]. Similarly, when considering the relation of laser power and layer thickness to surface hardness, the relation is also nonlinear [32]. A similar, nonlinear relation is also observed in Morgan et al (2004)'s empirical result [120]. The simple linear relation between scan speed and relative density becomes significantly more complicated when pulse density is also varied. This evidence seems to imply that more variables generate larger uncertainties in PBF due to an increase in unknown interactions. However, other outputs such as the tensile strength are not that sensitive to the same combinations of inputs. The observed relation remains linear under a combination of factors. Thus, more inputs can increase PBF problem complexity but do not necessarily indicate increasing nonlinearity of the input/output relation.

This proposed method focuses on the development of a general algorithm recommendation framework based on input/output parameters before deeply investigating physical interactions between PBF parameters. As such, it is necessary to understand the general factors that cause the uncertainty in PBF processes. Beaman et al (1997) first introduced the concept of energy density for AM, which is described by the following Equation [121]:

$$E_\rho = \frac{4P}{\pi r^2} \frac{2r}{v} \frac{2r}{s} \tag{5.1}$$

where E_ρ is Energy density, P is laser power, r is beam radius, v is scan speed, and s is hatch spacing.

This equation indicates higher laser power, lower scan speed, and closer hatch spacing produce higher energy density. More energy delivered to the powder usually means better melting conditions. Improved melting conditions will result in lower porosity and thus higher relative density. For example, Meier et al (2008)'s experiment with the metal laser sintering process shows the relative density increases from 69% to 99% with a power increase from 30W to 90W with other parameters held constant [122]. Another study [123] concluded higher energy density tends to produce a continuous melting track against irregular melt shape. These findings imply the linear relations may be more likely if the involved input/output parameters can be related to an overall energy density dependency.

In contrast, nonlinear relations were found in studies on outcomes related to part microstructure. Meier et al (2008) found surface roughness is not monotonically increased with scan speed, with the optimal roughness obtained in the middle of the range of scanning speeds tested [122]. Other research suggests that microstructure varies throughout the entire PBF part, and thus can be considered a local rather part-wide property. Wang et al (2012) found the hardness tests at different locations/directions in the same AM part produce different results [124]. Similarly, studies of thermal conditions indicate that variation of thermo-physical properties of AM parts are complicated [125].

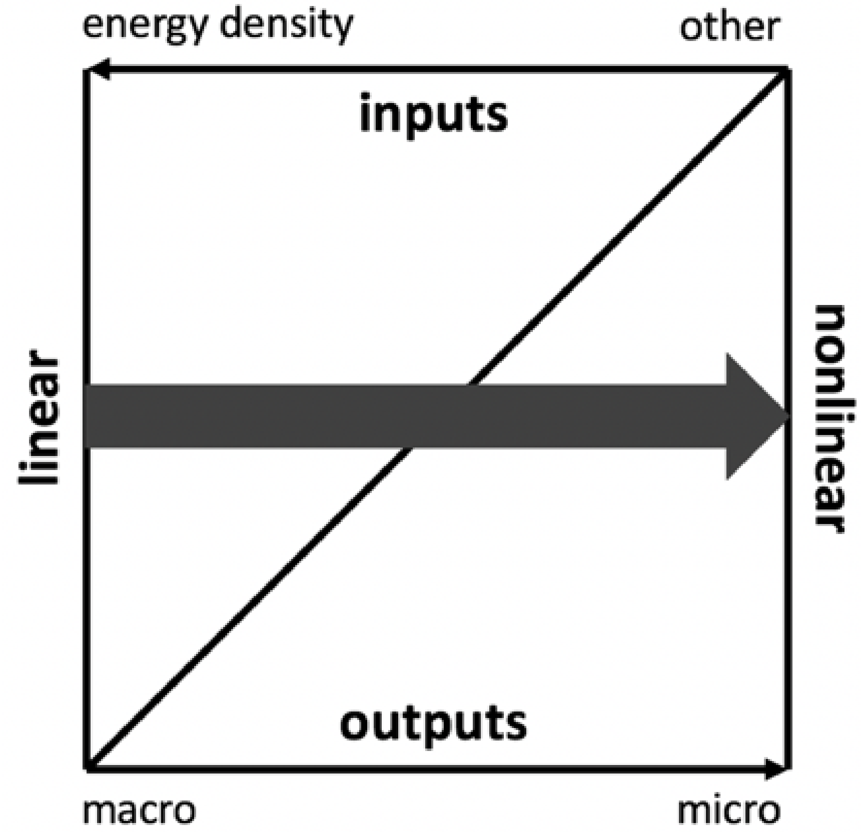


Figure 5.3: Hypothetical relation of input/output correlations

Figure 5.3 summarizes the hypothetical relation of input/output correlations observed in the literature. The thick arrow in the middle of the figure represents the relation from linear to highly nonlinear. The arrow (right to left) on the top represents whether the input parameters can be classified as related to energy density or not. The bottom arrow (left to right) represents whether the outputs are in macro-scale or micro-scale. The observations performed in this section pertain to linearity of input/output relationships. Figure 5.4 illustrates the more general case to which these techniques may be applied.

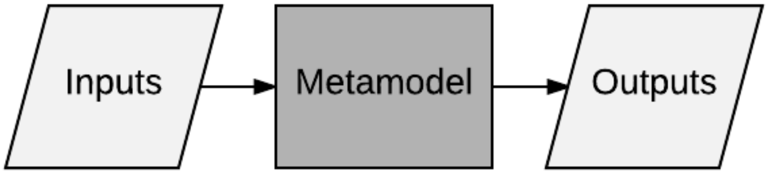


Figure 5.4: General case to model input/output relationships

Figure 5.3 can be used to summarize past literature results for the particular case of PBF. As discussed, laser power, layer thickness, and scan speed are input parameters that relate to energy density. Pulse frequency is located in the upper right corner since it is unrelated to energy density. For outputs, relative density is considered a high macro scale property as it depends on part width rather than local porosity. Surface roughness, however, relates more to AM microstructure. For the problem that involves the parameters in Figure 5.5(a), the link between inputs and outputs intersects with the bold arrow on the left. Figure 5.5(b) indicates pulse frequency and surface roughness have a highly nonlinear relation. Figure 5.5(c) and Figure 5.5(d) demonstrate the limitations in past research.

Figure 5.5 visually summarizes the relation between different combinations of PBF parameters observed in the literature. However, it must be stressed that, like the literature, it only summarizes some of the parameters of interest, and might not be sufficient to guide metamodel selection, especially in the case of indeterminate parameter sets. A more rigorous mathematical solution is thus needed, which will be introduced in next subsection.

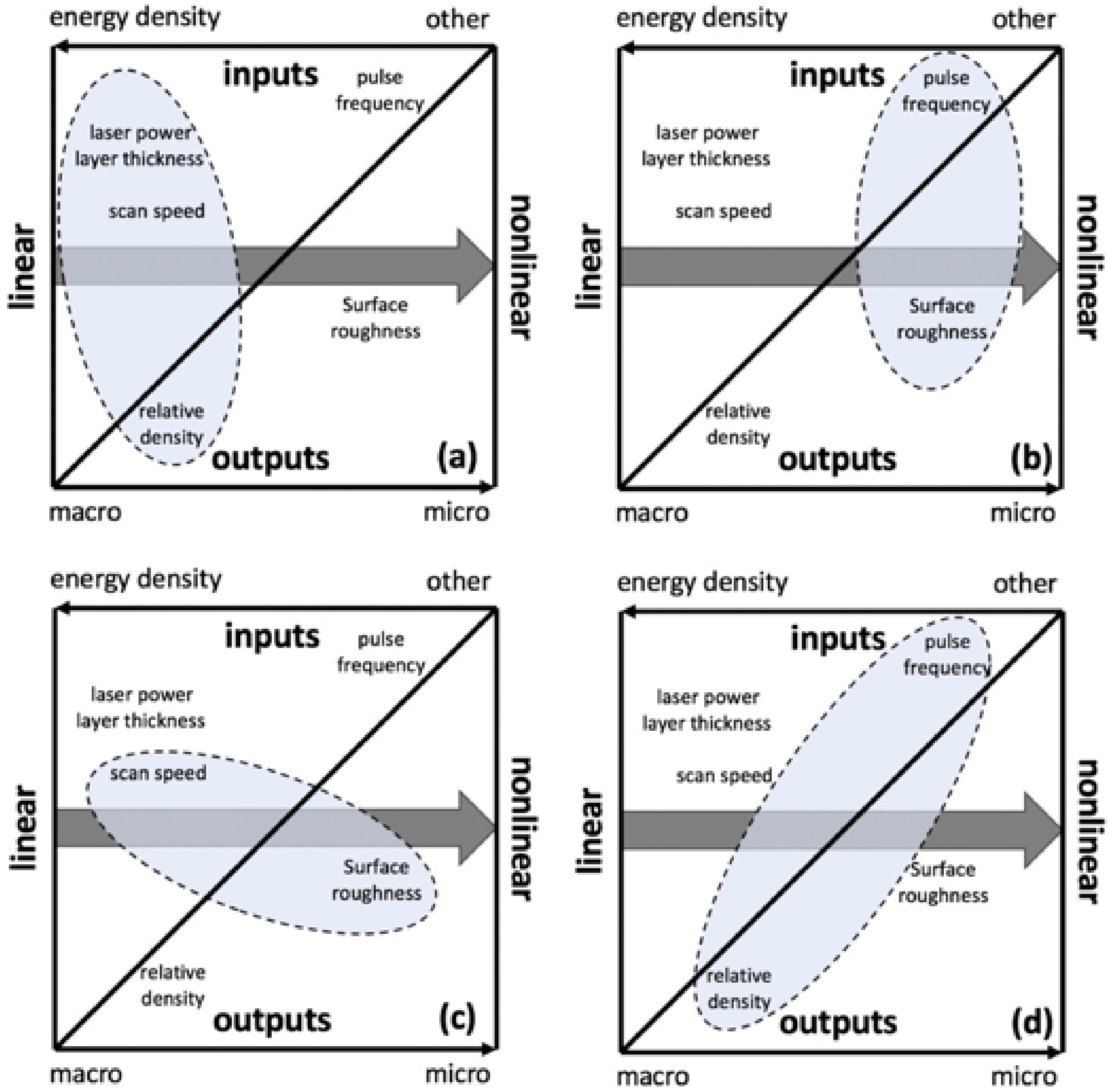


Figure 5.5: The use of AM input/output correlation chart

5.4 Domain-Driven Model Recommendation Framework

An exhaustive search, which is also known as the generate and test method, is the most general problem solving technique for systematically enumerating all possible candidate algorithms and selecting the most appropriate candidate based on a set of criteria [126]. While it is a global optima algorithm, it is also extremely inefficient, especially for those problems with abundant candidates and/or large input datasets. In such cases, it may be more efficient to incorporate prior knowledge into the algorithm selection process.

Many selection or recommendation techniques were developed to improve the efficiency of exhaustive search. Rice's model [112], for example, can recommend the best candidate for a new instance based on previous model selection knowledge. It includes four spaces: the problem space P represents the datasets of learning instances; domain space F contains the characteristics; algorithm space A includes all candidate algorithms; performance space Y is the measured performance of instance P for each algorithm in A [112]. Rice's model compares characteristics of a new instance to all previous examples and then assesses the suitability of each algorithm based on a set of rules or a selection algorithm. The model findings can be used to select the optimal algorithm from a given problem. Once the solution is derived, the performance in the new instance is added to the performance space Y, updating the model with a new point. In this way, a user can avoid exhaustively testing each candidate algorithm for a new instance [83].

The proposed domain-driven method is built upon Rice's algorithm selection method. However, instead of using data-features to characterize the new dataset, this proposed AM framework uses AM knowledge to indicate a possible optimal option from candidate algorithms. This approach is fundamentally different than Rice's method in that the result can be independent of the unfixed data strategy and rely on the relatively fixed knowledge of the physics of the problem. The AM characteristics used are the relations between input/output parameters discussed in previous subsection. The general workflow of the proposed framework is shown Figure 5.6. It requires sufficient knowledge to commence the selection process. Knowledge construction consists of collecting existing datasets, classifying the instances, and computing the performance of each candidate algorithm on each dataset. The extracted information is then fed back into the current knowledge model and the system predicts a possible optimal metamodeling algorithm. Before proceeding to model construction with actual data, the newly calculated solution updates the knowledge model. Details of each critical step are discussed in the rest of this section.

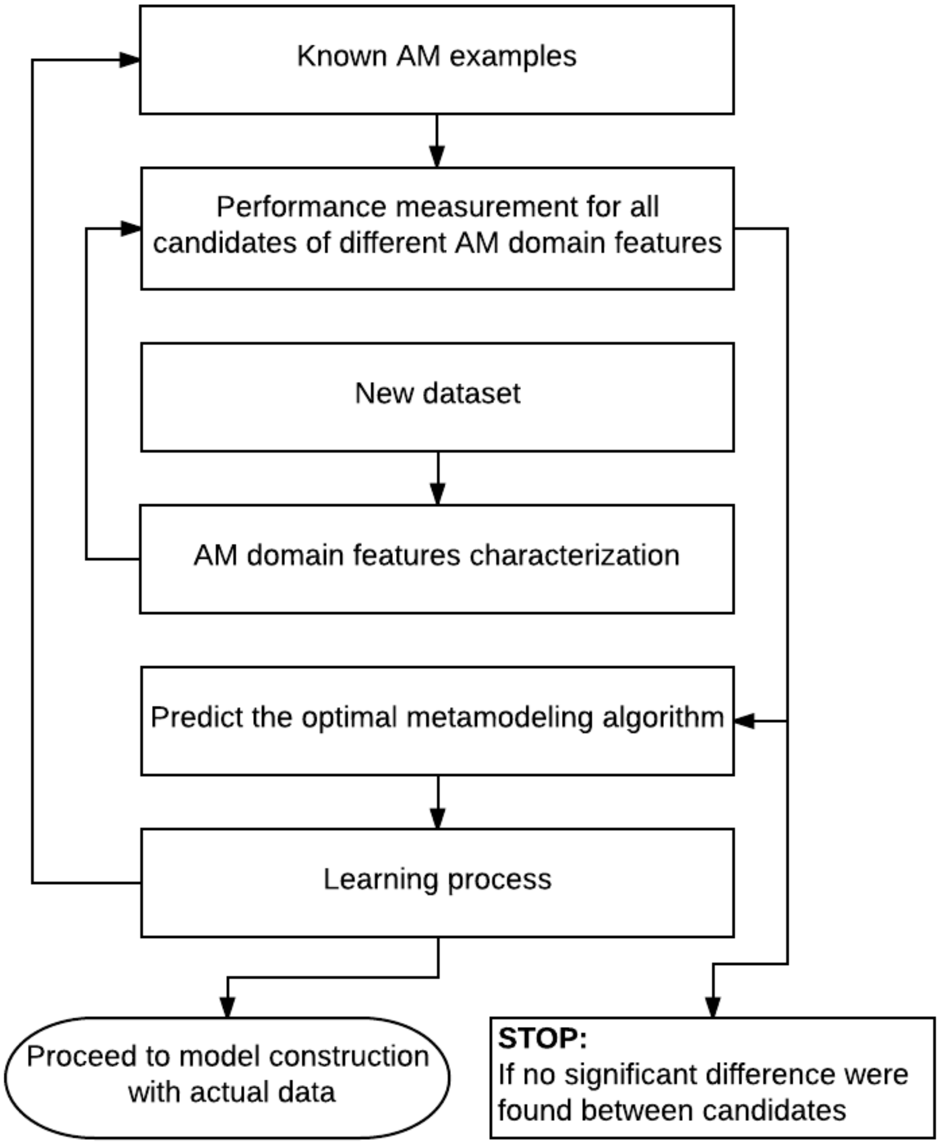

Known AM examples
Performance measurement for all candidates of different AM domain features
New dataset
AM domain features characterization
Predict the optimal metamodeling algorithm
Learning process
Proceed to model construction with actual data
STOP:
If no significant difference were found between candidates

Figure 5.6: General workflow of the proposed framework of AM domain-driven modeling selection method

5.4.1 AM Characterization

The AM characteristics mentioned at this stage are AM input/output parameters. At this step, all parameters are formed to input vector $\bar{X}$ of the selection framework:

$$\bar{X} = [x_1 \ \ x_2 \ \ x_3 \ldots x_n \ \ y_1 \ \ y_2 \ \ y_3 \ldots y_m]^T \tag{5.2}$$

where x and y are the inputs and outputs respectively and are equal to 1 or 0. 1 indicates that the problem includes the parameter and 0 indicates the parameter is not considered. These vectors are the inputs to the learning process. For problems that have exactly the same outputs, y can be ignored.

5.4.2 Performance measurement

Measuring model performance of known datasets is critical to improving model selection accuracy. Two criteria were employed in the case study in next subsection to evaluate the modeling performance by a set of candidate algorithms. For global measurement, normalized root mean square error (NRMSE) [83] is used. Maximum relative error magnitude (MREM), on the other hand, is used to evaluate the outstanding error of the models [127]. These were formulated as:

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \tilde{y_i})^2}{N}} \Bigg/ (y_{max} - y_{min}) \tag{5.3}$$

$$MREM = max\left(\frac{|y_i - \tilde{y_i}|}{y_i}\right) \quad (y_i \neq 0) \tag{5.4}$$

where $y_i(\neq 0)$ is the actual observed value, $\tilde{y_i}$ is the estimated value from the metamodel, y_max and y_min are the maximum and minimum actual observation, and N is the total amount of validation samples. With NRMSE and MREM, the framework can make its recommendation based on both global and local performances of the datasets by assigning appropriate weights to each criteria. Note, if y_i is very small, MREM could represent noise and be very large. This is a potential limitation of MREM. In this study, y_i is always an actual AM output that not close to zero. However, if the small y_i number is appeared, it should be removed to avoid noisy verification result. At this stage all case studies consider the NRMSE and MREM criteria equally to not bias either way. However, in some cases, these two objectives could conflict. Under these circumstances, a user could deploy a weighted multi-criteria decision making formulation.

5.4.3 Prediction Process

The prediction process of the proposed recommendation framework could be completed by either model-based or instance-based methods. Model-based methods build predictive models to determine the optimal modeling algorithm. The predictive selection is based on an input vector $\bar{X}$ (model variables) and the resulting modeling performance (model outcome). Once the model is built, the new instance with PBF-related information in $\bar{X}_{new}$ would then import to the model and calculate the predictive result. The model-based method is similar to what is discussed in previous subsection. For example, the vector $\bar{X}$ is the input variable set of the recommended predictive model. The values of NRMSE and MREM then become the predictive results. Once the recommended model is built based on existing instances, the model can predict the NRMSE and MREM of candidate algorithms for a new problem. The user can decide which algorithm would be employed according to these indicators of model performance.

An instance-based method by comparison solves the problem based on existing examples. It assumes an algorithm has similar performance on similar problems, where the similarity is measured by Euclidean distance between instance input and output vectors. The k-nearest neighbor (k-NN) ranking approach as was employed in this study. The k-NN approach ranks the nearby k nearest examples for their similarity. However, the simplest case of k-NN is the closest neighbor example based on the comparison of the Euclidean distance of all examples, which also called as 1-NN. The formulation of 1-NN is:

$$dist(i, j) = \sqrt{(a_i - a_j)^2}, \quad j = 1, 2, 3, \ldots, m \tag{5.5}$$

Where a_i represents the new instance, a_j represents the existing examples, and m is the total number of examples. In the case of the metamodeling algorithm, a set of datasets composed of input and output parameters would serve as the existing points, each of which has been characterized by a set of metamodels. By comparing the input parameters (a_j) to those used in the new dataset, the user can then determine how similar the data is to a known dataset. At this point, users would simply compare the performance of various modeling algorithms in the existing examples. Thus, a likely best predictive modeling algorithm can be chosen without costly characterization of information and data features of the dataset. This saves the cost of testing all candidate algorithms individually, allowing the user to directly proceed to model construction and parameters optimization. A demonstrative example in the following section shows how the method works.

5.5 Demonstrative Case Study

A simple example was constructed from existing AM datasets (2 for constructing knowledge and 1 for verification) to illustrate the proposed AM domain-driven framework. Tang et al (2003)'s and Morgan et al (2004)'s metal PBF datasets were used to construct the knowledge base [116, 120]. The dataset from Chatterjee et al (2003)'s was selected to verify the selection accuracy [52]. The three datasets have the same output parameter, relative density. Because of this, the $\bar{Y}$ output vector is omitted, and the results obtained may potentially be more accurate since the knowledge is constructed from somewhat similar examples. The 1-NN method was used to predict the optimal modeling algorithm. The predicted algorithm was then compared to actual modeling results with all candidate algorithms to assess the predictive solution [52].

5.5.1 Knowledge construction

The knowledge used for model predictions was composed of a small dataset with 15 samples [116] and a large dataset with 105 samples [120]. They were selected to build the knowledge model because of their similarities: 1) both are metal PBF process; 2) both use similar experimental conditions; 3) they have the same output as the new dataset (relative density). The differences between them also provide opportunities for future model selection for new instances, namely: 1) they use different input parameters; 2) they use a different DOE strategy and 3) both have different variables that are not considered in DOE such as materials and specific machines. Thus, the knowledge base of these datasets is reasonable and has useful variation.

For initial construction we consider 6 independent input variables, though neither dataset can cover the parameter of layer thickness. The two datasets overlap in individual parameters. The matrix of the inputs is shown in Table 5.1. Note, the order of input parameters that are in the table and elements in the vector are constantly fixed for a future prediction process.

Table 5.1: The inputs matrix of given datasets

	Laser power	**Scan speed**	**Powder density**	**Layer thickness**	**Pulse frequency**	**Hatch spacing**
Tang (2003)	1	1	1	0	0	1
Morgan (2004)	0	1	0	0	1	1

Writing these inputs as input vecors:

$$\bar{X}_{tang} = [1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1]^T \tag{5.6}$$

$$\bar{X}_{morgan} = [0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1]^T \tag{5.7}$$

The output vector $\bar{Y}$ is omitted in this case study since both knowledge and verification datasets have the same target output–relative density.

Three algorithms were used to characterize both input datasets. A PR model was built using the pure quadratic regression method. The Kriging model is built by ordinary Kriging method and the Gaussian correlation function with maximum likelihood approach. The ANN model is defined with 10 hidden layers. It should be noted that the candidate algorithms used in this case study are not meant to be exhaustive, but rather to represent a set of common modeling approaches. To calculate the performance, each original dataset is divided into training and testing sets with fixed ratio 80% and 20% using the Latin Hypercube based Minimum Euclidean Distance method [127]. For Tang et al (2003)'s dataset, PR works the best from three candidates as both NRSME (0.1580) and MREM (0.0220) are the lowest (Table 5.2). However, in the second dataset the Kriging model tested was found to be the best possible choice among the candidate algorithms. Thus, at a system level the knowledge base indicates: 1) while $\bar{X}_{tang} = [1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1]^T$, recommend model=PR; 2) while $\bar{X}_{morgan} = [0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1]^T$, recommend model=Kriging. The future model selection process is built based on these rules.

5.5.2 Modeling Algorithm Recommendation

The verification dataset consisted of 13 samples manufactured using the metal PBF process [52]. Compared to the datasets in the knowledge model, the experiment used carbon steel powder instead of stainless steel or a copper alloy, and has the

Table 5.2: Model performance of both datasets for three candidates

	NRSME			**MREM**		
	PR	**Kriging**	**ANN**	**PR**	**Kriging**	**ANN**
Tang (2003)	0.1580	0.1757	0.5603	0.0220	0.0230	0.1642
Morgan (2004)	0.2018	0.1332	0.3917	0.1055	0.0669	0.1866

smallest sample size (13) and number of input variables (2). The two input variables were layer thickness and hatch spacing, resulting in an input vector of:

$$\bar{X}_{new} = [0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1]^T \tag{5.8}$$

Based on 1-NN approach, the Euclidean distance between the new and former datasets are:

$$dist(new, tang) = 2 \tag{5.9}$$

$$dist(new, morgan) = 1.732 \tag{5.10}$$

The knowledge model at this stage is likely insufficient due to a very limited number of example instances. Though it has these defects, the distance results show that the dataset is closer to Morgan's data than Tang's. Thus, the recommended algorithm would be a Kriging model. Once confirmed, the result can be used to update the current knowledge model with a new instance – while $\bar{X}_{tang} = [1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1]^T$, and Y is relative density, the optimal candidate model is Kriging. Once this is done, the updated knowledge model was updated and can cover the aspect of layer thickness.

For verification, the performance of each candidate model with the new dataset is shown in Table 5.3. All models were constructed using the same methods and model parameters as in the knowledge model datasets. Based on the performance measurement, Kriging model shows small advantages in both NRSME and MREM compared to the PR and ANN models. Thus, the result is consistent with the solution predicted using the framework.

Table 5.3: Model performance for the new dataset

	NRSME			MREM		
	PR	**Kriging**	**ANN**	**PR**	**Kriging**	**ANN**
Chatterjee (2003)	0.3374	0.3186	1.0447	0.0282	0.0257	0.0681

5.6 Discussion

The AM domain-driven metamodeling recommendation framework has the potential to provide an efficient and reliable way to predict the optimal metamodel for a new problem. It is efficient as it can avoid exhaustively testing all possible candidate algorithms once a sufficient knowledge model is constructed. Moreover, the solution can help to direct future model construction when considering data-features that might allow the user to hone in from a broad class of algorithms to a specific one. The general framework was established based on the hypothesis that certain combinations of input/output parameters have consistent behavior. The predictive solution could be made more reliable if it were derived from a larger set of consolidated knowledge. A simple demonstrative case study that included three distinct metal PBF datasets shows the algorithm prediction process.

Though the proposed framework shows a multiple of advantages in AM metamodeling problems, the details of the method need further improvement. The current set of candidate algorithms is limited, including only PR, Kriging, and ANN. While suitable for demonstration, this limited size of candidates potentially restrains higher model performance of new datasets. Furthermore, each model only has the basic modeling configuration without the ability for user modification. It may cause false results due to incomplete consideration of modeling options. For example, the finding that the ordinary Kriging model works better than a pure quadratic regression model does not mean that it also works better than a higher order PR model. Without consideration of the range of available algorithm types, the current framework may mislead the user. To overcome these disadvantages, more detailed metamodeling techniques should be added to the current framework. This work is being undertaken.

Beyond adding more broad classes of algorithms, subclasses of algorithms also need to be considered for a more robust solution. For example, consideration of different Kriging methods might enrich the study. Simple Kriging, stochastic Kriging, and dynamic Kriging may further define the Kriging class in the set of candidates. In addition to adding more algorithms, it may be useful to bring modeling guidelines into the framework. For example, such guidelines might indicate that ANN may not be well suited for use with small datasets. Such considerations may improve the predictive accuracy of the framework for a larger breadth of datasets.

For the specific case of AM and the PBF process, algorithms for the AM characterization process also need further improvement. There are more than 50 independent variables in metal PBF process [48, 113]. This study has included less than 1/3 of them. Another disadvantage is that the framework can only count categorical input/output vectors, rather than considering broad classes of inputs and the relative similarity between different types of variables. For example, in the review of the literature, variables relating to energy density were found to behave very similarly within a range of outputs. This is knowledge that might improve the model selection process. If included in the knowledge model, the system could possess greater insight when calculating the distance between instances. More robust parameter classification may thus be needed for more accurate prediction. Similarly, as research continues, the vectors can be further detailed and classified in multiple levels based on process knowledge and empirical data. For example, materials could be classified as 0 (single component), 0.5 (multiple components without steel), or 1 (multiple components with steel), or using some other scheme to provide greater insight into problem similarity. However, development of reasonable methods requires a more comprehensive understanding of AM processes.

All of this suggests the need for a hybrid approach that utilizes a combination of process-specific knowledge and experience, algorithmic knowledge and dataset-specific considerations. The process knowledge might consist of empirical input/output relations as in this paper, utilize knowledge of problem physics to assess the similarity of datasets and suggest several candidate classes of algorithm. Algorithmic knowledge might consist of a well defined model of broad algorithm classes and subclasses, and defined model and data attributes that affect their performance. Simple data features such as sample size and the utilized DOE methods could then be used.

The most important challenge in this work currently is a lack of data to construct a more reliable knowledge model. In the case study, the naive knowledge model consists of only two instances. To improve the model, more AM knowledge is needed. In the current knowledge model, only empirical datasets can be used to build the knowledge model. Simulation models might be used to enhance the knowledge model. For example, Ma et al (2015)'s FEA model has 10 independent AM input variables, which may allow the framework to include a broader range of problem physics [76]. Formal information models may also contribute to better knowledge construction. More candidate metamodeling algorithms should be considered into such information models to not limit the overall performance. Recent development of an AM ontology might provide the basis for more effective utilization of process specific knowledge. If this information can be utilized in a future version of the proposed framework, it can potentially boost its predictive ability and accuracy.

CHAPTER 6
MAXIMUM PREDICTIVE ERROR UPDATING METHOD

6.1 Overview

In Chapter 5 we have developed the method to optimally select the metamodeling algorithm based on existing knowledge. Once the algorithm is selected, the next step is to maximize the predictive accuracy of the metamodel. Building metamodel using historical data is challenging since the data points may not what exactly wanted for the new problem. The desired data point may not exist in given dataset. Or there may be redundant data points that would not contribute to the new problem. Though one could generate additional data points, this requires that the new experiments can fully replicate the former experimental conditions. This can be extremely difficult if the data is adopted from historical experiments. If the conditions are not match, additional data may increase the uncertainties.

To address this challenge, this chapter analyzes and prescribes metamodeling techniques to select optimal sample points, construct and update metamodels, and test them for specific and isolated physical phenomena. Case studies of two different laser welding experiments are presented to illustrate and validate this method.

6.2 Background

AM processes are more complex, variable, and difficult to understand than subtractive manufacturing [48, 128]. Typical AM processes implement material patterning, energy patterning, new layer creation, and support from previous layers [129] to realize shape, material, and hierarchical complexities [130].

Material properties of AM-produced parts often depend upon the process parameters. For example, platform temperature, building direction, and post heat treatment influence the part microstructure that determines fatigue properties of selective laser melting parts [131]. Further, variations in layer thickness and hatching distance settings have affected material porosity along with hardness and density [52].

Various models have been developed in recent years to describe complex AM process-structure-property relationships. In spite of advances in model accuracy, the enormous computational cost of complex, high-fidelity physics-based simulations of AM makes these models impractical to adopt in industry [132, 108]. A more preferable strategy is to utilize surrogates, or metamodels, as they provide a "model of the model" to replace the expensive simulation model in design and optimization processes [80]. Metamodeling has been used successfully as an alternative to computationally expensive simulations in aerospace and other advanced manufacturing domains. [132, 133].

Currently, varieties of metamodeling techniques are applied in engineering design. Several comparative studies present the performance of these various techniques under different modeling criteria [132, 134]. Generally, different modeling methods show both advantages and disadvantages for different types of problems. These disadvantages include orders of nonlinearity and problem scales [132]. To simplify explanation in this chapter, we mainly focus on the PR and Kriging Method for metamodel construction to illustrate selection of the most applicable metamodeling techniques for these specific cases.

In spite of the benefits envisioned through the use of such metamodeling techniques, very little research has been done in this area. Some notable exceptions include a polynomial regression model of density, hardness, and porosity of a carbon steel selective laser sintering process [52], porosity predictions in selective laser melting [135] and an energy density model of CoCrMo powder material [123]. These approaches are limited to experimental designs for a specific portion of an AM process. There is a need for a complete AM metamodeling methodology to construct and integrate local metamodels [22] for robust prediction of AM process results. Challenges for the AM situation include cost of experimentation [136], accuracy of simulation capabilities [137], and complex interactions of different physical phenomena during the AM process [138].

This study aims to investigate metamodeling as a means to generate accurate predictive models compatible with a composable multilevel structure, defined as made up of highly reusable models that can be used together and mirror the general AM process model [22]. Such a metamodeling methodology will be able to address the challenges in AM processes such as high system complexity, uncertainty, and limitations of legacy data conducted by design of experiments (DOE) that designers may need to rely on due to the expense of producing experimental sample parts [22, 136, 139].

Section 6.3 introduces methodical approaches to construct and test individual metamodels. A pair of case studies in Section 6.4 illustrates the potential effectiveness of these approaches. Section 6.5 discusses this work and potential future work.

6.3 Development of Model Updating Method for AM

This section introduces a predictive metamodeling approach to address some of the unique challenges particular to metamodel construction to represent the various sub-processes in AM. The following subsection explains the rationale of this methodical approach based on the challenges identified previously in this paper related to predictive metamodeling for AM processes.

6.3.1 Minimum Euclidean Distance (MED) Method

Eddy et al's prior work addressed a similar situation of metamodel construction in non-ideal data locations for design space filling [140]. This work identified Latin Hypercube Design (LHD) as a potential approach for the reasons given in the last subsection. Given the inability to choose the points at the exact locations identified by an LHD sample set generation, this work proposed a method to find the minimum Euclidean distance between each identified data location and the data point identified in the data set [140]. The procedural steps begin with the generation of the desired amount of LHD points from a given data set. Next the Euclidean distance is calculated between each DOE data point to the generated LHD points by the Maximin method. Those points closest to the desired LHD points are selected for constructing the metamodel. Since the selected DOE points depend upon the LHD points, selection of the initial points from these LHD results is critical to improvement of metamodel construction. Such methods as Maximin LHD [40], orthogonal array-based LHD [141], and optimal Audze-Eglais uniform LHD [142] can generate optimal LHD sampling points.

6.3.2 Maximum Predictive Error Updating (MPEU) Method

Shao and Krishnamurty developed a surrogate model based design optimization (SMBDO) method to sequentially update a surrogate model by capturing the critical features of an unknown system in a simulation-based experiment [143]. Similary, a comprehensive adaptive sampling methodology is presented in Sandia's Dakota framework [144] to enable selection of successive sample points based on the maximum distance from existing points or the uncertainty of model prediction. Based on these methods, the initial LHD sample points are used to construct the initial surrogate model. During each updating step, potential optimal locations predicted by the current model are then validated. Those points that exhibit high predictive error are then added into the current model iteratively until the desired model accuracy is obtained. However, the clustering based multilocation search procedure of SMBDO and the adaptive sampling method both rely upon an ample supply of data points from efficient computer simulations, which is simply not realistic in this case. Thus, there is a need to develop a model updating method for the limited data sizes inherent with using historical data for metamodel construction.

To address these challenges and limitations, we introduce the Maximum Predictive Error Updating (MPEU) method to gradually improve model accuracy. Figure 6.1 outlines the general framework of the MPEU method using the MED method to select the most appropriate sampling points from original DOE data. The method begins by generating LHD points using the maximin method in the design space given by the data. The Kriging method with a Gaussian correlation function was employed to build the surrogate models. Then, the points are selected to construct the initial metamodel by using the MED method. A validation procedure next determines whether the current surrogate model needs further improvement. MREM(5.4 and average relative error magnitude (AREM) are used to test the metamodel for predictability at each iterative stage of model updating:

$$AREM = \frac{1}{m}\left(\frac{\sum_{i=1}^{m} |y_i - \hat{y}_i|}{y_i}\right) \quad (y_i \neq 0) \tag{6.1}$$

where y_i is the observed value from given data, $\hat{y}$ is the value predicted by the metamodel of the DOE points that were not selected to construct the metamodel, and m is the number of data points.

If either error calculation exceeds a preset threshold for robustness, the point presenting the largest predictive error is added into the initial sample pool. A new model is created based on the new sample set. The model creating-validating procedure will iteratively proceed until both MREM and AREM satisfy the preset threshold value for robustness. This is the MPEU method that sequentially infills the sample set by updating the model to improve the resulting metamodel construction.

Verification and validation techniques must test the metamodel at each stage [78]. Verification tests the internal consistency of constructed metamodels and validation tests reliability with external data [44]. To validate the newly built metamodel, a model validation criterion is established based on the prediction accuracies [78] of all non-selected DOE points. If the MREM and AREM are both greater than a specified preset threshold value for robustness, the data point with the lowest prediction accuracy (or highest MREM value) would be added into the current sample pool and the metamodel is updated accordingly. Subsequently, the newly built metamodel will be validated again with the same process iteratively until convergence to within the threshold MREM and AREM values. Thus, effective model construction can be achieved efficiently by combining predictive metamodel construction simultaneously with validation to robustness requirements.

The preset threshold values of MREM and AREM are based on design requirements such as penetration depth and melt pool width. Both average and maximum error are involved in the validation process since they represent general and distinguished model performance. A designer would need to decide on what model accuracy and predictability are necessary or acceptable before model construction [140]. An unnecessarily low threshold value may significantly increase the computational cost. Conversely, an excessively high threshold value may reduce model predictability and utility. The following section demonstrates the potential use of these proposed space filling and sequential infilling techniques in a pair of case study examples.

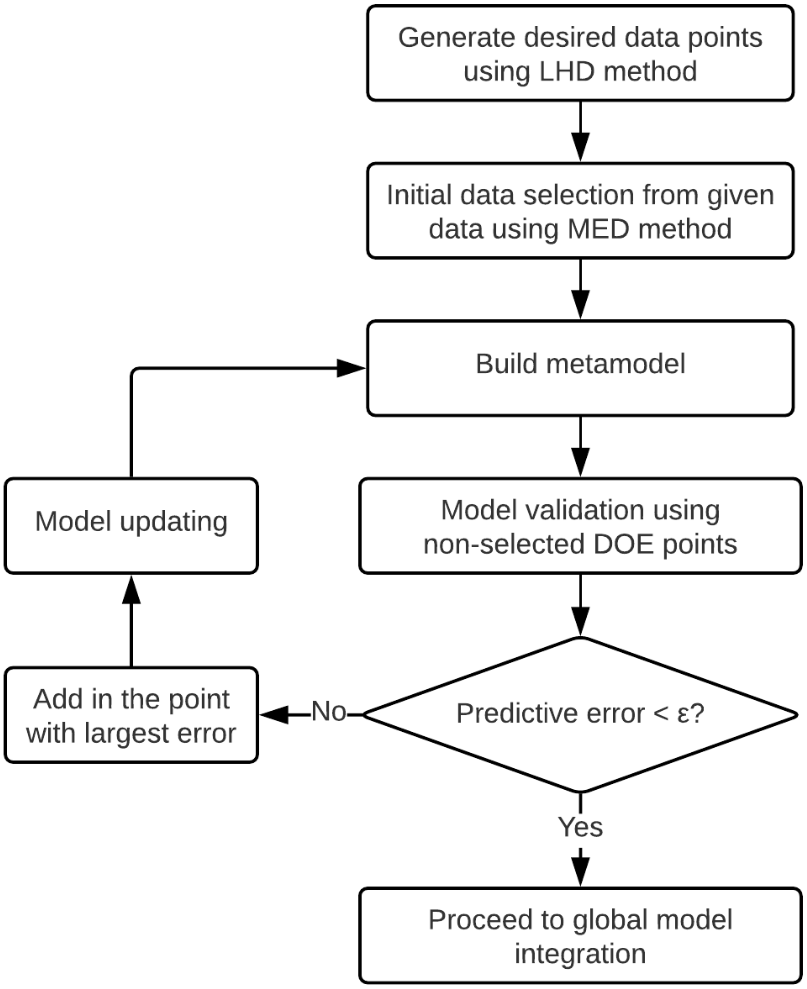

Generate desired data points using LHD method
Initial data selection from given data using MED method
Build metamodel
Model validation using non-selected DOE points
Model updating
Add in the point with largest error
No
Predictive error < ε?
Yes
Proceed to global model integration

Figure 6.1: Maximum Predictive Error Updating (MPEU) Method

6.4 MPEU Method in Advanced Manufacturing

The laser welding process is used in the case studies reported in this section due to its similarities to directed energy deposition processes. In both applications, a heat source fuses metal as it is being deposited. The processes share similar process parameters and their quality is determined by similar metrics (dimensional accuracy, surface finish, residual stresses and mechanical properties, all of which can be traced back to the geometry of the melt pool). With that said, data is more readily available for laser welding, making the process a good candidate for demonstrating proof-of-concept.

The following two case studies illustrate the potential applicability of the proposed MPEU method for different DOE data. Both cases focus on the same response of the penetration depth (P). The cases have similar experimental methods but different DOE strategies. These two simple and somewhat similar experiments help to illustrate various results that can be expected from different data sets. This section shows the potential to deploy methods to construct and test various individual AM metamodels by use of the method introduced in the prior subsection.

6.4.1 Full Factorial DOE Data Set

In the first case by Kahn, et al. [145], laser power (LP), welding speed (WS), and fiber diameter (FD) are the input variables. Among those three variables, LP and WS ranged from $800W$ to $1100W$ and $4.5m/min$ to $7.5m/min$ by three linear levels, with midpoint 950 and 6.0 respectively. The third variable of FD has only two levels at values of $300m$ and $400m$ FD [145]. The full factorial DOE consists of eighteen total data points for penetration depth, measured in micrometers after a standard washing procedure and with no special heating treatment.

The first step is generating an LHD sample set in the design space. In this case the LHD set consists of five points in order to give the initial model enough options for future updating. Fewer start points may not adequately cover the design space. Using the MED method described in the previous section, the Euclidean distance between each LHD point and DOE point are calculated. Table 6.1 lists the initial data points selected by the MED method. The first column represents the standard order number of each point in the original DOE. The initial metamodel would be constructed from these five points.

Table 6.1: Initial data points generated by MED method

	Input variables			**Observed value**
Data point number	**LP(W)**	**WS** (m/min)	**FD** (μm)	**P** (μm)
1	800	4.5	300	960
5	950	6	300	950
6	1100	6	300	1180
14	950	6	400	727
17	950	7.5	400	580

From the collected data, the initial metamodel is built using a standard kriging method. Kriging has built-in verification of internal consistency to prevent the error that can occur when RSM is used. The remaining thirteen data points next validate the metamodel by calculation of MREM and AREM as explained in the prior section. Model updating is next done iteratively by applying the MPEU method, as described in the prior section, to the preset thresholds for robustness of $\epsilon_{MREM} \leq 10\%$, and $\epsilon_{AREM} \leq 5\%$ in this case.

Table 6.2 lists the results for this example of the first four iterations of the MPEU method. Note that each sequential iteration represents the validation results calculated by the current updated metamodel. Only the points showing the most significant error are included in this table. At each iteration, the point with the

Table 6.2: MREM and AREM at each stage

Iter	**Data point**	**Input variables**			**Observation**	**Predictive value**	**MREM**	**AREM**
		LP	**WS**	**FD**	**P**			
Stage 1	3	1100	4.5	300	1610	1108	31.12%	28.37%
	7	800	7.5	300	560	891	59.18%*	
	12	1100	4.5	400	1307	875	33.02%	
	13	800	6.0	400	577	818	41.86%	
	16	800	7.5	400	492	756	54.43%	
Stage 2	3	1100	4.5	300	1610	1339	16.79%	11.10%
	9	1100	7.5	300	880	1019	15.76%	
	11	9.5	4.5	400	1043	899	13.82%	
	12	1100	4.5	400	1307	1094	16.31%	
	16	800	7.5	400	492	385	21.70%*	
Stage 3	2	950	4.5	300	1290	1107	14.20%	14.26%
	3	1100	4.5	300	1610	1244	22.71%	
	11	9.5	4.5	400	1043	845	18.93%	
	12	1100	4.5	400	1307	882	35.53%*	
	15	1100	6.0	400	920	806	12.37%	
Stage 4	2	950	4.5	300	1290	1241	3.81%	3.71%
	3	1100	4.5	300	1610	1471	8.57%*	
	8	950	7.5	300	730	702	3.87%	
	13	800	6	400	577	539	6.65%*	
	15	1100	6.0	400	920	963	4.62%	

greatest MREM is marked with star. At stage 1, the point at standard order 7 is selected by adding it into the initial sample pool since it shows the highest MREM (59.18%). As a result, after the third iteration both the MREM and the AREM values satisfy the preset threshold value. According to the MPEU method, the updating process converged to construct the final metamodel with eight DOE data points at 8.57% MREM and 3.35% AREM. Figure 6.2 shows the error values at each iteration.

Note that the MREM and AREM values did not always decrease monotonically prior to the final stage, as one would expect in the early stages in any numerical iterative approach, but shows monotonicity and convergence towards the end. Similar trends were observed in the application of SMBDO to several classical simulation-based model updating case studies [145].

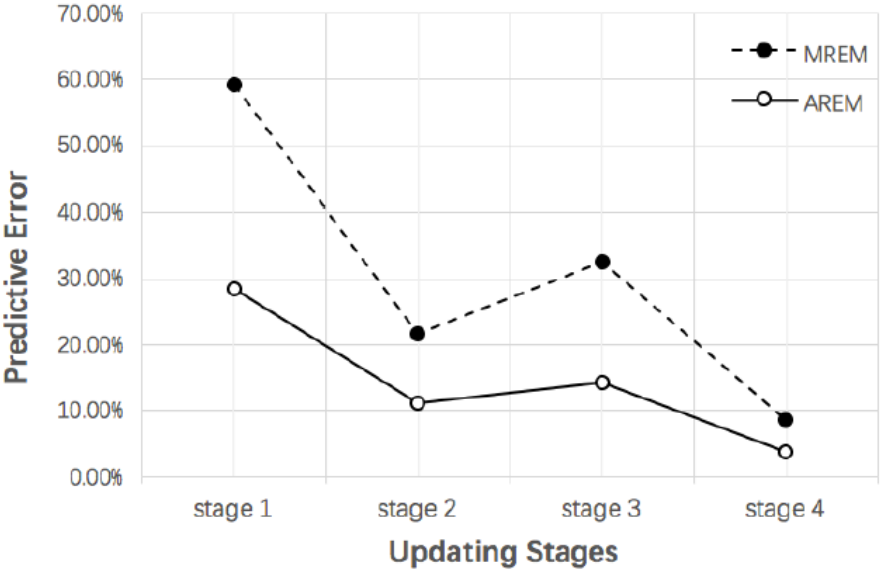


Figure 6.2: Error at iterations

In this first case study a two level, three factor DOE strategy becomes the only choice if one prefers to create the model by classical DOE techniques without collecting new experimental data. Beyond the proposed MPEU method and a two level DOE method, another compatible sampling method is a random search method. However, this method is not recommended here due to its uncontrolled behavior. The comparison results of AREM and MREM between random search method, two level full factorial DEO method, and MPEU method with different threshold values are listed in Table 6.3.

As shown in the table, both AREM and MREM of the model built by the MPEU method are significantly lower than the random search and DOE methods when the sample size is the same. When gradually reducing the threshold values of

Table 6.3: AREM and MREM results of random search method, two levels full factorial DOE method, and MPEU method

	Random	**DOE**	**MPEU**		
Sample size	n=8	n=8	n=8	n=10	n=12
AREM	22.76%	7.76%	3.35%	2.44%	1.94%
MREM	59.16%	12.99%	8.57%	4.74%	5.18%

AREM and MREM, the MPEU method typically incorporates a few more points to improve the model accuracy to the new convergence requirements.

Table 6.4: Comparison of different point selection strategies

	Single stage sampling	**MPEU method**	**Improvement**
Sample size	n=8	n=8	
AREM	7.60%	3.35%	55%
MREM	12.66%	8.57%	32%

With the MPEU method, the model is iteratively improved by updating sample points. However, only the initial sample points can evenly distribute across the given design space by use of the MED method. Newly updated points are selected based on the validation results from previous iterations without considering their location in a design space. As shown in Table 6.4 the metamodel that is constructed by sequential infilling reduces prediction errors at comparable sample sizes. Thus, despite the possibility of the initial five data points not adequately filling the design space, the MPEU method shows potential to generate a more accurate model through the updating strategy. The following subsection examines the results of applying this same method to a situation that provides fewer data points in a data set.

6.4.2 Fractional Factorial DOE

The second case study of laser welding DOE data is based on a three factor, three level Box-Behnken design with full replication [146]. "Beam angle" (BA) in this experiment replaced the input of "fiber diameter" from the first case study. The experimental design generated fifteen data points. A mean value of the data set's three replicate points reduces the size of the data set from fifteen to thirteen.

Table 5 lists the MREM and AREM values at each stage for those points having significant predictive error. As shown, the MREM started with five sample points from 82.00% and gradually decreased to 4.80% after five updates, or six stages. The error at the start could have a significant effect on the number of iterations required. It is notable that the error at the first stage is 39% higher than the amount shown for the first stage in Table 6.2 for the first case study. It is also notable that this second case study is covering more levels with less data than the first case study.

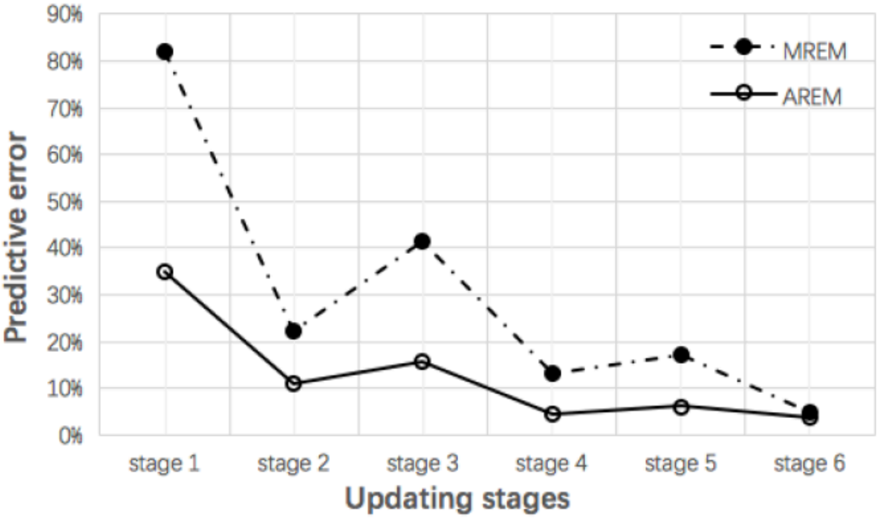


Figure 6.3: Error at iterations

After applying the MPEU method, accuracies of the final model satisfied the threshold values ($\epsilon_{MREM} \leq 10\%$) and $\epsilon_{AREM} \leq 5\%$ The error convergence progression is shown in Figure 6.3. Both MREM and AREM increased slightly during the middle stages as new points were added into the previous sample pool.

Table 6.5: MREM and AREM at each stage

Iter	Std. order	MREM	AREM
Stage 1	3	45.89%	34.93%
	4	82.00%*	
	6	52.50%	
	8	54.79%	
Stage 2	1	15.46%	11.09%
	3	22.16%*	
	8	21.71%	
	13	4.79%	
Stage 3	3	41.51%*	15.71%
	6	33.93%	
	8	0.85%	
	13	16.64%	
Stage 4	3	4.48%	4.60%
	8	1.30%	
	9	2.71%	
	13	13.32%*	
Stage 5	3	17.07%*	6.16%
	6	3.69%	
	8	3.22%	
	13	0.65%	
Stage 6	9	4.80%*	3.72%

6.5 Discussion

The objective of this chapter was to explore a metamodeling methodology tailored for AM and adaptable to different types of empirical data. To address the challenges identified, this work introduces an MED method to select usable sample points from different types of given data and an MPEU method with an updating procedure to create predictive metamodels to predetermined robustness requirements from limited data sets. The proposed MED method can select usable initial sample points from various types of DOE data since its foundation is based on the LHD sampling method, which is adaptable for most any design space. Though the generated LHD sample locations may not be occupied by given DOE data, the MED method can improve selection of more appropriate existing points over other methods.

The MPEU method allows model developers to balance the tradeoff between model accuracy and computational cost by adjusting the threshold values of MREM and AREM to achieve specified levels of robustness. As shown in Table 6.3, with the same number of sample points, the MPEU method, which also utilizes the MED method, provides a more accurate model than the random search and eight DOE data points for the example that was tested. The MPEU method also provides an option if one intends to improve the model at the expense of slightly higher computational costs. In the first case study, two added new points can significantly reduce the MREM from 8.57% to 4.74%. Furthermore, the updating strategy of the proposed method can contribute more to capture the critical features of an unknown system than simply picking up points from the given data set. As shown in Table 6.4, MPEU method significantly reduces both MREM and AREM. In other words, the proposed method focuses more on capturing the critical system features rather than the point locations.

Despite the advantages in model construction with the MED and MPEU methods, there are some limitations. Such disadvantages can potentially limit the application of proposed methods. For example, since the MED method selects the initial points through randomly generated LHD samples, each time the MPEU method may produce different models to the same convergence criteria. It cannot guarantee that the generated LHD sampling set is optimal in a given DOE design space. Rather, the updating procedure depends highly upon the initial MED selected points. Without a confirmed starting point, the overall performance of the MPEU method may decrease. For example, the final model required ten points for construction but left only three points for validation. Thus, another limitation relates to model validation. Unlike metamodels generated by computer simulations, historical DOE data is often not reproducible. One can only rely on the existing data since it is impossible to gather additional information. In this second case study, the start point accuracy and resulting number of points remaining to validate the model were not as acceptable as found in the first case study. This second case study also had less data than the first case study. While not conclusive, this supports the assertion that the amount of data or information can have a significant effect on the results of using metamodeling methods. Methods such as Grey System Theory that work with little data or information may be introduced along with this current proposed method [147]. Nonetheless, future work could potentially improve the MPEU method by adding a check and adjustment process based on the error at the first stage.

Two laser welding case studies in the prior subsection show that the proposed MPEU method is compatible with different DOE data sets in these cases. The two data sets have similar experimental conditions such as the same laser source, common input variables of laser power and welding speed, the same response of penetration depth. However, one must use these two metamodels separately due to their different ranges of data locations in the design space. To overcome such a shortcoming or data limitation, a future goal is to build towards a global metamodel by combing two local data sets. Such a development may more efficiently utilize different historical data sources to know more about a process and also raises the issue of uncertainty between data sets with different sources.

The MPEU method lays the foundation for a predictive metamodeling methodology to use in AM. Next, we would investigate development of a hybrid metamodeling method through the application of clustering techniques [148] and multisurrogate approximation (MSA) methods [149] to build the global model by combining data sets with different input variables, process conditions, or material parameters.

CHAPTER 7

GREY-BOX METAMODELING FOR PREDICTIVE ANALYTICS IN AM AND SMS

7.1 Overview

The MPEU method improves the predictive accuracy of the metamodel for historical data. A critical baseline of the MPEU method is the data points are from the same experiment. In some cases, however, we may have to combine the datasets from multiple experiments. Traditional model integration methods require the candidate datasets have same experimental conditions. Whereas, historical data usually does not have this advantage if they are from different resources. This condition increases the predictive error of the integrated metamodel. Therefore, to address this issue, we need to detect and interpret the differences between datasets.

This chapter develops a two-stage grey-box modeling approach that combines manufacturing knowledge-based (white-box) models with statistical (black-box) metamodels to improve model reusability and predictability. A white-box model can use various types of existing knowledge such as physical theory, high fidelity simulation or empirical data to build the foundation of the general model. The residual between a white-box prediction and empirical data can be represented with a black-box model. The combination of the white-box and black-box models provides the parallel hybrid structure of a grey-box. For any new point prediction, the estimated residual from the black-box is combined with white-box knowledge to produce the final grey-box solution. This approach was developed for use with manufacturing processes, and applied to a powder bed fusion additive manufacturing process. It can be applied in other common modeling scenarios. Two illustrative case studies are brought into the work to test this grey-box modeling approach; first for pure mathematical rigor and second for manufacturing specifically. The results of the case studies suggest that the use of grey-box models can lower predictive errors. Moreover, the resulting black-box model that represents any residual is a usable, accurate metamodel.

7.2 Background

Smart manufacturing is becoming increasing possible as access to technology improves. Industry is, and will continue to be, increasingly reliant on data and predictive analytics to improve overall process efficiencies [8]. With this trend, industry is now collecting data at never before seen rates in hopes of gaining competitive advantages related to their products and processes. Often data are collected without regard for their interrelation, and it is not readily apparent how the collected information can be used to improve system efficiencies. To address this issue, we investigate a novel metamodeling technique based on the context from which the data was acquired and the domain in which it is relevant.

White-box modeling methods use knowledge such as rules and theories, to formulate models such as those that represent physical phenomena. Such classical white box modeling methods have been used for thousands of years. Newton's Law or Euler–Bernoulli bending theory [150] is a classic example of a traditional physical model. Such models usually require comprehensive knowledge of the target system and are usually represented by parametric formulation. For instance, in manufacturing, physics-based models are often derived from theoretical analysis that mostly focuses on individual sub-processes with idealized assumptions. In reality, the actual multi-physical system of manufacturing may involve numerous interactions among these sub-processes. Such complexity can be difficult to fully understand. For example, in AM, the isotherm migration method develops a thermal model to calculate the temperature on a powder surface being heated by a laser beam modeled as a point source [151]. However, the complex inter-relationship between parameters of Powder Bed Fusion (PBF) processes renders these theoretical analyses insufficient for the needs of many practical applications [113]. On the other hand, metamodel is a typical representation of black-box modeling methods.

Either white-box or black-box approach has unique advantages. However, such metamodels built by pure statistical approaches usually lack information about the model's physical meaning and assumptions due to a large degree of data-dependency. Moreover, modeling inaccuracies might accumulate during the model construction process due to the lack of physical knowledge about the critical features of the represented system [78]. Thus, both white and black box approaches alone have accuracy limitations due to different reasons.

Due to these intrinsic limitations in both approaches, a technique that can harness the advantages of both white and black box models while reducing their disadvantages is desirable for complex problems with understood subdomains. The modeling approach known as grey-box, or hybrid modeling, was invented to combine the benefits of domain knowledge and empirical information [152]. The models generated by this approach can obey general physical rules (white box) while optimizing the parameters from actual experimental data (black box). Many of the newer and less established white box manufacturing physics-based models and numerical simulations may be founded on incomplete and/or inaccurate knowledge and idealized assumptions. For example, the previously mentioned isotherm migration model in AM does not account for the influence of powder particle size, part geometry, and environmental conditions [151]. The calculated solutions from current AM physics-based models are usually limited in the scope of what they describe, diminishing their predictive capability. Though AM metamodels can potentially avoid these errors, they usually require a large number of expensive samples and may not be reusable. Many examples in smart manufacturing have similar modeling challenges. These barriers potentially limit the adaptability of metamodeling in any manufacturing domain. Thus, neither approach can optimally construct robust, usable manufacturing models alone.

This chapter aims to develop a grey-box modeling approach which combines the benefits of traditional serial methods (where black and white box knowledge is applied sequentially) and parallel methods (where knowledge from black and white boxes is composed before being applied). The result is a hybrid combination of knowledge of physical phenomena and statistical information. To address the challenge of combining knowledge of physical phenomena with statistical information, a two-stage approach is used. The first stage deploys a serial grey–box approach to build a statistical black-box model to estimate the errors caused by the inaccuracies in a white-box. The second stage uses the model from the first stage to estimate the basic solution and the residual solution. The final solution is a combination of these two.

Section 7.3 provides fundamental background knowledge relating to grey-box modeling techniques and introduces this general algorithm of a two-stage grey-box modeling approach for additive or smart manufacturing. Case studies using a general mathematical example and a representative metal PBF AM problem are presented in Section 7.4. Section 7.5 discusses the results and identifies future work for this study.

7.3 Grey-box Modeling Method

A grey-box model is a hybrid model that combines different types of models such as physics-based models, numerical simulation models and statistical models

[153]. The term “grey-box” stems from the mixture of white-box and black-box models. A conventional grey-box model uses a physical formulation to maintain the physical interpretation and uses data to estimate parameters [153]. In general, the basic structure of a grey-box model is inherited from knowledge and further improved by experimental data.

7.3.1 Overview of Grey-box Modeling Technique

Grey-box model development can be summarized into three steps: 1) construct the foundation for the system with a simplified knowledge model; 2) determine the physical parameters from the description of the system behavior; 3) identify the value of model parameters from actual data [83]. The relationship between these three types of model and knowledge sources is shown in Figure 7.1. The proposed grey-box metamodeling method was developed based upon this viewpoint.

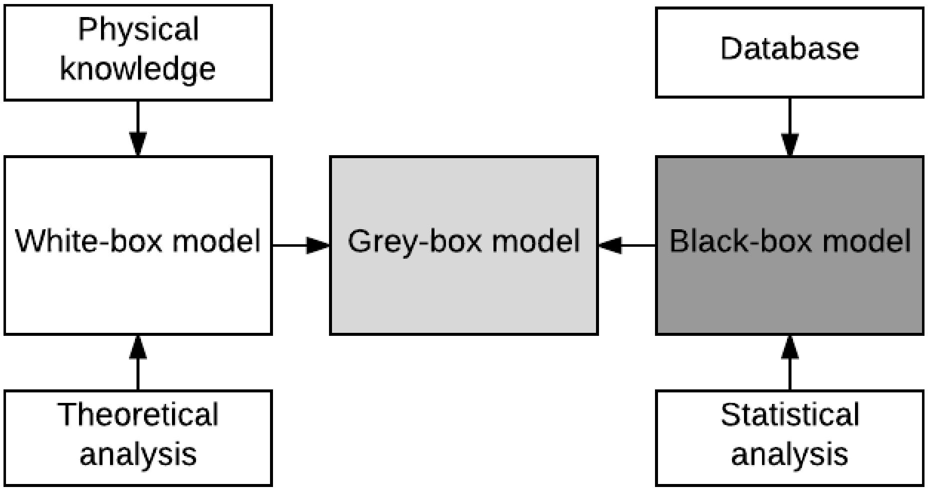


Figure 7.1: Relationships among physics-based white-box, statistics-based black-box, hybrid grey-box models and knowledge sources

Grey-box models can be generally classified into serial approach and parallel approaches [154, 155], which are shown in Figure 7.2. A serial approach aims to sequentially fill the gap between knowledge and experimental data. For example, the uncertainties raised from incomplete knowledge of a white-box model can be reduced by accompanying that model with actual data. A parallel approach, alternatively, aims to use both models together to estimate the correct results that would be difficult to approach by either a white or black-box model individually. A grey-box model with serial structure focuses on reducing the error between the prediction from physical model and actual result from experiments. For example, Duarte and coauthors developed a hybrid modeling approach that combined knowledge and mechanistic, rather than statistical models, to improve traditional model performance [156]. In this approach, the first model is built based on first-principles system behavior, and the second model estimates the residuals between real data and mechanistic predictions.

A traditional grey-box model with parallel structure uses data to estimate the correct values of model responses which are difficult to approach given incomplete knowledge of phenomena [157]. For example, Psichogios and coauthor's hybrid neural network model utilizes a partial first principles model [154]. This modeling approach combines available prior knowledge with an ANN model to derive an estimator of unmeasured process parameters. This hybrid structure can interpolate and extrapolate much more accurately than a standard "black-box" ANN with significantly fewer training sample points to accompany the knowledge model.

The next subsection will introduce the two-stage grey-box modeling approach developed for manufacturing problems in this study. To get the final prediction, the data serially flows into both types of grey-box models for the purpose of constructing the black-box model and estimating the residual. To demonstrate this approach, we chose a complex manufacturing process that we believe could particularly benefit. The AM-specific grey-box modeling approach is built upon both the Type II serial approach and parallel approach shown in Figure 7.2 based on current AM challenges.

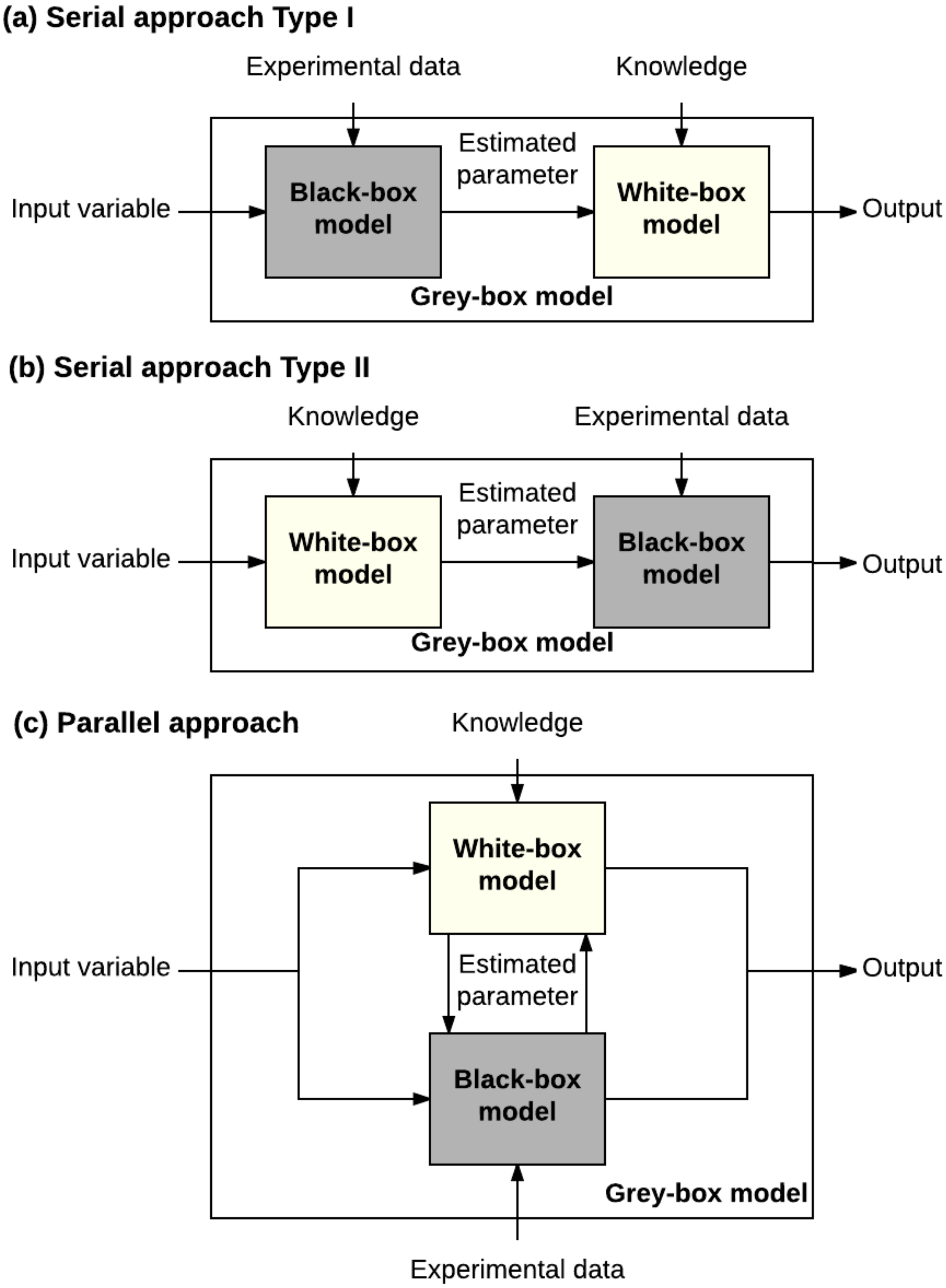


Figure 7.2: Basic grey-box modeling approaches

7.3.2 Two-stage Grey-box Modeling Approach

Many barriers and challenges, such as the large uncertainty of AM process results, have prevented its further adoption in industry [158]. The relationship between process parameters and mechanical properties are not fully understood for AM processes. For example, relative density, one of the major structural properties of the parts produced by metal PBF processes, depends upon multiple AM parameters such as laser power, scan speed, pulse frequency, and layer thickness [159]. Previous studies show that a typical metal PBF process consists of four general sub-systems classified by related physical phenomenon. Each sub-system can be further divided into multiple sub-processes [22]. A general AM process can involve more than fifty independent parameters [113]. For example, the melt pool sub-system is related to a number of factors that involve both thermal and fluid mechanics [28]. Though difficult, AM models built upon theoretical analysis, numerical simulation and statistical modeling have been developed for predictive purposes in recent years [127].

The general procedure to construct an AM grey-box model by using the approach introduced in this work is shown in Figure 7.3 and Figure 7.4. First, the method builds the white-box model from available prior knowledge. If knowledge was derived from theoretical analysis, the white-box model can be directly represented by a parametric formulation. Alternatively, a parametric model can be derived through an approximation of a physics-based model using a formulation such as FEA for computational simplification. However, if the knowledge is based on a complex numerical simulation that requires high computational cost, the white-box model can be redesigned to be represented by some simplified parametric function such as a PR response surface model to more rapidly estimate a white box model using fewer sample points of that expensive data.

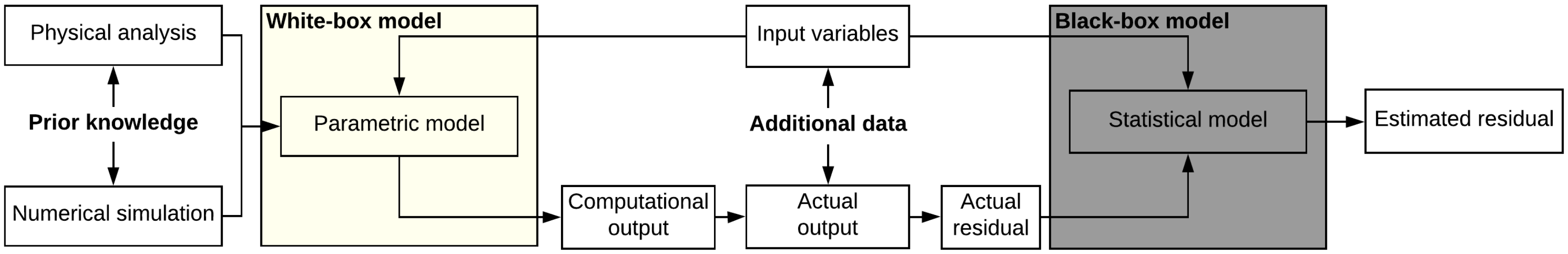


Figure 7.3: General workflow of the first stage

The sampling data to construct that PR model could be collected by a technique such as space filling sampling (SFS) or sequential infilling sampling (SIS) [148] from data generated by an adequate number of simulations. It can be expected that the solution from the white-box model would contain large errors due to limited knowledge.

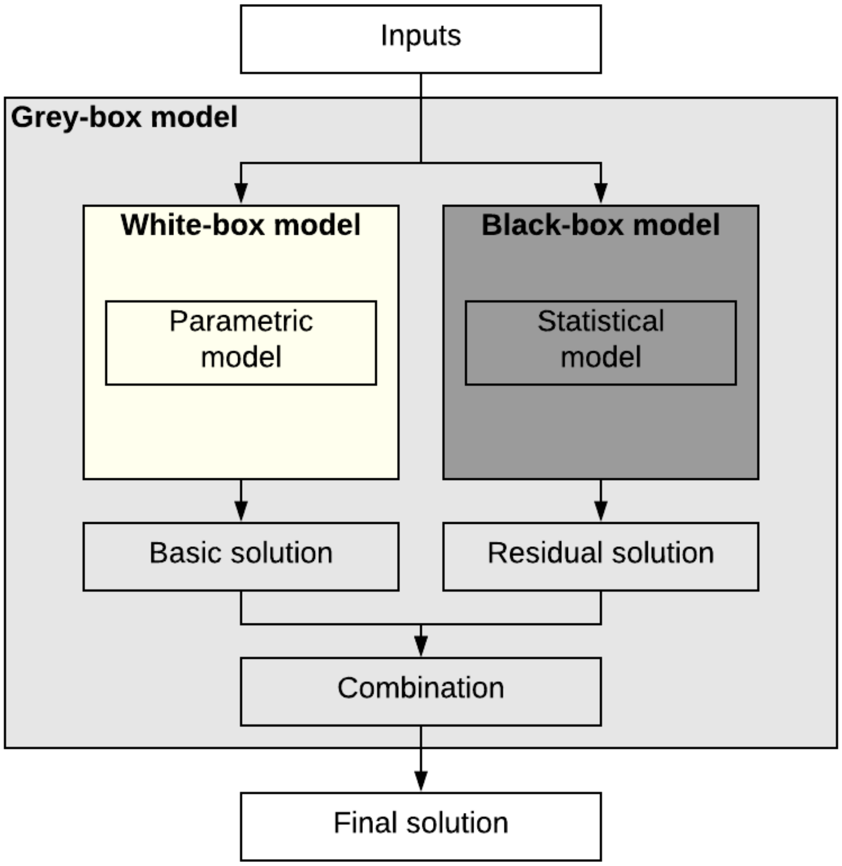


Figure 7.4: General workflow of the second stage

The next step is to build a black-box model from additional information. Potential sources of the additional information could be actual experimental data or a higher fidelity simulation used to generate the data. The black box model captures both the discrepancy between a lower fidelity FEA type of model and the real process as well as the discrepancy between a high fidelity model (FEA model) and the simplified model. The input values of this additional data are entered into the constructed white-box model to calculate the corresponding output responses. This computational output from the white-box model is next compared with the actual output values of the additional data to calculate each difference. This difference can be considered the white box model's actual residual at that data location as shown in Figure 7.3. Since each pair of the computational output and the actual output has the same input variables, the residual value directly represents the accumulated errors caused by incomplete and/or imperfect knowledge used to construct the white box model. The black-box model is used here to evaluate the relation between input variables and the estimated residual.

At the first stage, the serial grey-box structure is established based on the type II serial approach that is shown in Figure 7.2 b: the output from the white-box becomes an intermediate input to the black-box. This serial grey-box approach is used to build a black-box model to estimate the residual value that cannot be derived from the white-box alone. That residual is the difference between the responses predicted by the white and black boxes. The inputs to the black box are those used to generate the white-box responses. The black-box model uses the kriging method to model the relationship between input variable values and residual response values. The kriging method is applied since this interpolation approach helps to avoid any significant intrinsic error in the resulting model [92]. Once the black-box is created, it can compute the estimated residual for any new data point. The approach, illustrated in Figure 7.3, establishes the black-box model used to derive the grey-box model created in the subsequent steps shown in Figure 7.4.

Figure 7.4 depicts the second stage of the process, wherein the white-box and black-box of the residual built in the first stage are composed to a parallel structure. It is considered a parallel structure because the given values of input variables are entered into white-box and black-box models simultaneously. The white-box in Figure

7.4 is the same as the one in Figure 7.4.

However, at this grey-box modeling stage, the output from the white-box directly estimates the final solution in concert with the estimated residual from the black-box. For each new data point prediction, the output from the white-box is used as the basic solution. The residual solution from the black-box is the estimated residual for that same new data point. The final solution is the combination of basic solution with its estimated residual, or the results from both stages.

To illustrate the proposed grey-box modeling method, two case studies are presented in the next subsection. The first, a classical mathematical example of a mystery function [78, 102] demonstrates the process of constructing grey-box models from pre-existing knowledge that can be expressed numerically. The second example illustrates the use of this grey-box modeling technique to predict the relative density resulting from an AM process and represented by actual experimental data.

7.4 Application of Two-Stages Grey-box Metamodeling

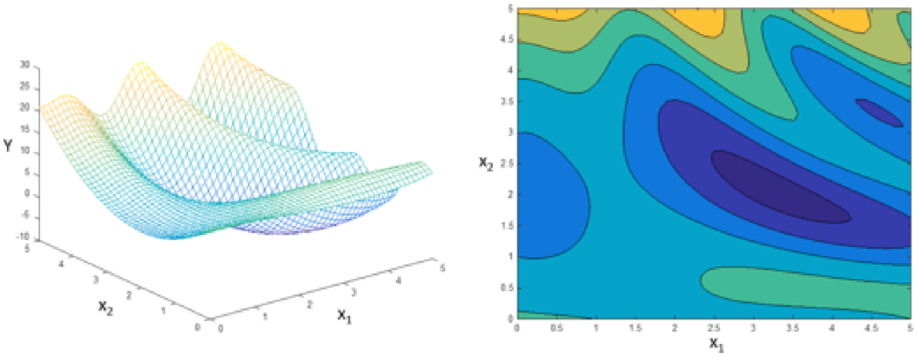
To illustrate the method, two case studies are presented in this section. The mystery function examples illustrate the process of grey-box model construction for different types of knowledge. A Metal PBF case study is used to test the method in actual situation. Similarly, MREM (5.4) and AREM (6.1) are used to represent the model predictability. To illustrate the method introduced in the prior section, two case studies are presented in this section. The mystery function (7.1) examples illustrate the process of grey-box model construction for different types of knowledge.

7.4.1 Case Study: Mystery Function Problem

A classical mystery function [78, 102] is brought into this study to mimic a complex unknown system. The function $f(x_1, x_2)$ that represents a nonlinear and complex system is used to generate experimental results used for model creation and assessment. The original equation of this mystery function is:

$$Y = f(x_1, x_2) = 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1) + 2(2 - x_2)^2 + 7sin(\frac{x_1}{2}) \tag{7.1}$$

Where x_1 and x_2 are two input variables and Y is the actual output. The true surface and contour plots of the original mystery function are shown in Figure 7.5. To illustrate the effectiveness of this method, the original equation is manipulated to illustrate a scenario similar to that of an inaccurate white-box model representing model construction with incomplete prior knowledge. In this situation, a parametric formulation is accessible before constructing the grey-box. However, another example simulates a situation where the parametric white-box model cannot be directly derived from current knowledge. In that case, the prior knowledge was delivered by running a hypothetical simulation-based model, i.e. the manipulated function $f_k(x_1, x_2)$. These two examples illustrate how to use this grey-box modeling approach for different types of problems.



(a) (b)

Figure 7.5: (a) True 3D surface plot and (b) contour plot of the original mystery function

Case study: Theoretical physics-based model

In this example, the available prior knowledge is assumed derived from theoretical analysis and represented by an inaccurate parametric formulation. In this paper, it is assumed that the white-box models are reasonable representations of the manufacturing phenomena being modeled, and no validation step was included in our approach. Thus, to mimic this condition, the original mystery function was modified to:

$$\begin{aligned} \bar{Y} = f_k(x_1, x_2) =& 2 + 0.001(x_1 - x_1^2)^2 + (1 - x_1) + 2(2 - x_2)^2 + \\ & 7sin(\frac{x_1}{2})sin(\frac{5x_1x_2}{10}) - 0.4x_1sin(2x_1)cos(x_2) \end{aligned} \tag{7.2}$$

Where subscript k indicates a function derived from knowledge. $\tilde{Y}$ is computational output from the white-box model $f_k(x_1, x_2)$.

Plots of the white-box model are shown Figure 7.6 (earlier stage of grey-box modeling). After the manipulation, the 3D surface maintains its general shape but several characteristics are changed, which can be observed in the figure. For example, the original local minima and maxima have shifted and the original sharp ridges became flatter. These changes result from the inaccurate white-box model. If we use the correct data from the original function to test current white-box model, the MREM and AREM are equal to 942.43 and 2.74, respectively. The large error indicates that the white-box model has very low fidelity and large predictive errors.

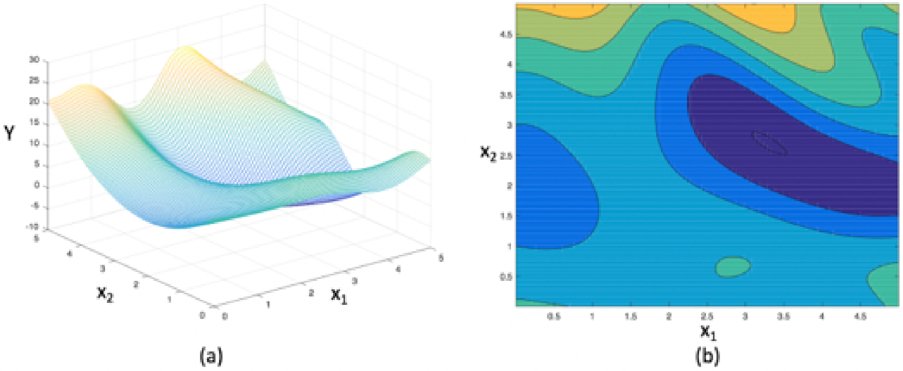


Figure 7.6: (a) 3D surface and (b) contour plots of manipulated mystery function

Though the white-box model has defects, it can contribute to a grey-box model. As mentioned in the last paragraph, the general information delivered from this white-box model is a reasonable representation of the level of knowledge to be expected from a white-box since the plots are generally similar to its original shape in that the local optima are still located close to their original positions. The next step is to add additional information to the initial, low fidelity prediction obtained from the white-box model. This high fidelity data is used alongside the low fidelity white-box prediction to build the black-box model. The additional information was generated from the original function using Latin Hypercube Sampling (LHS) [160] to generate 100 new data points. These additional data points represent an experimental or high fidelity model result as they were generated from the original function $f(x_1, x_2)$, which is defined as a high fidelity system without significant error.

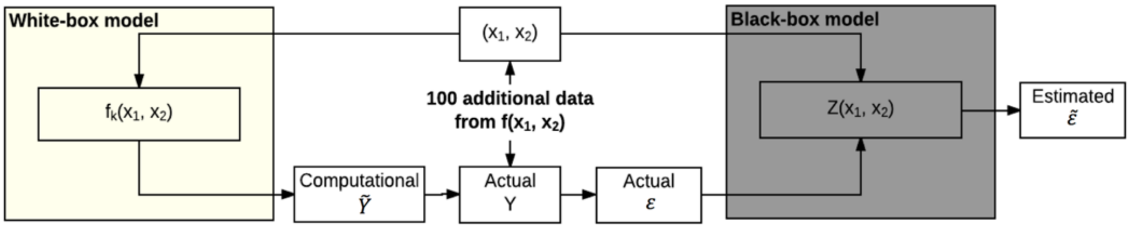


Figure 7.7: Black-box model construction to estimate residual

Table 7.1: Results at some sample data point locations

Input $(\mathbf{x_1}, \mathbf{x_2})$	Actual output $(\mathbf{y})$	Computational output $(\tilde{\mathbf{Y}})$	Actual residual (ϵ)
(3.73, 0.98)	6.1881	8.1416	-1.9535
(4.48, 4.03)	9.3751	11.9010	-2.5258
(0.23, 1.83)	3.0593	3.0066	0.0527
(4.33, 2.98)	5.0025	4.7074	0.2951

Figure 7.7 shows the process to construct the black-box model using the Type II serial approach (second stage of grey-box modeling). The input variables x_1 and

x_2 are first entered into the white-box model $f_k(x_1, x_2)$ and used to calculate the computational output $\tilde{Y}$. The residual ϵ is the difference between $\tilde{Y}$ and actual output Y. The input variables (x_1, x_2) and the residual ϵ are next used to construct the black-box model $Z(x_1, x_2)$ using the Ordinary Kriging method. The black box model is used to compute the estimated error $\tilde{\epsilon}$ in subsequent steps. Table 7.1 shows some examples from the 100 data points for illustration of the process shown in Figure 7.8. For example, one of the additional data points (3.73, 0.98) has an actual output 6.1881. This input when entered into the white-box model yields a prediction of 8.1416. The actual residual is next derived based on $\epsilon = Y - \tilde{Y}$, which is equal to -1.9535. Once the black-box model is built from the residual values, it can estimate the residual of any unknown point from its input variable values. This estimated residual represents an expected difference between the white-box prediction and an unknown actual output. As a result, the final grey-box solution $\tilde{Y}_{final}$ should at any point be equal to $\tilde{Y} - \tilde{\epsilon}$. This value combines the results from both stages, as shown in Figure 7.8. The white-box model in the parallel grey-box structure is the same one used in the prior serial approach. For any new point, the grey-box would combine basic solution $\tilde{Y}_{new}$ and the estimated residual $\tilde{\epsilon}_{new}$ to get the final solution at that point location.

One thousand randomly generated data points from the original mystery function were used to validate the resulting grey-box model. The grey-box model has reduced the initial white-box MREM from 942.24 to 2.7452 and AREM from 2.74 to 0.0359. The 3D surface and contour plots shown in Figure 7.9 are significantly improved and very close to the true plots of the original mystery function (Figure 7.5).

The grey-box constructed with 100 additional data points improved the initial white-box model globally. However, the MREM which represents the local error of the model remains higher than expected. It may be that the information provided by

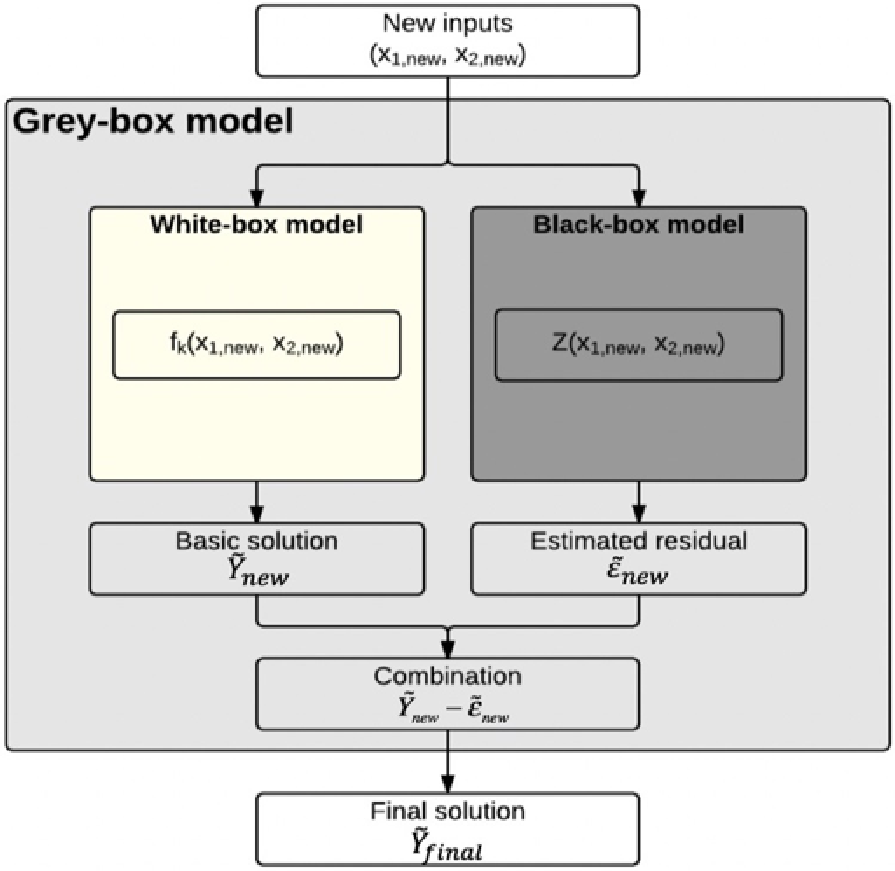

New inputs
($x_{1,new}$, $x_{2,new}$)
Grey-box model
White-box model
$f_k(x_{1,new}, x_{2,new})$
Black-box model
$Z(x_{1,new}, x_{2,new})$
Basic solution
$\tilde{Y}_{new}$
Estimated residual
$\tilde{\varepsilon}_{new}$
Combination
$\tilde{Y}_{new} - \tilde{\varepsilon}_{new}$
Final solution
$\tilde{Y}_{final}$

Figure 7.8: Grey-box model construction

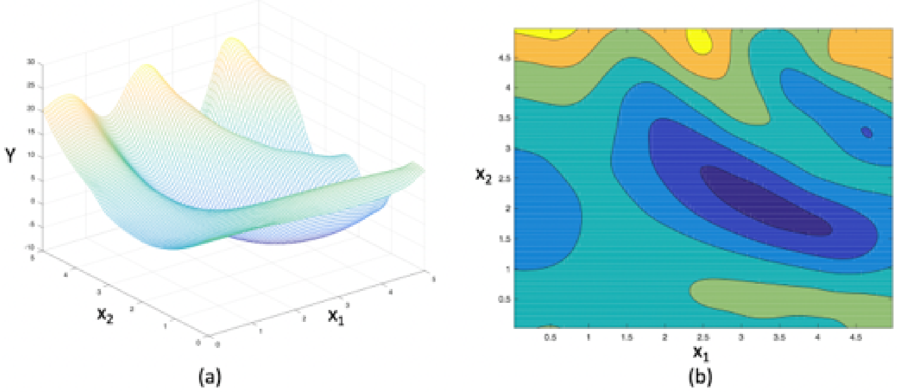


Figure 7.9: (a) 3D surface and (b) contour plots for grey-box model built based on 100 additional data points

the additional dataset is insufficient. To further test the proposed method, Table 7.2 shows the MREM and AREM of grey-box models that are constructed with different numbers of additional data points. The top number of zero data points is a case where no data is available and the results are derived from the white-box model only. As shown, the model performance can decrease exponentially with more points in this process. Convergence criteria can be established to determine the desired accuracy.

Table 7.2: Model performance of additional quantity of data

Number of additional data	**MREM**	**AREM**
0	942.4327	2.7451
100	2.7452	0.0359
200	0.9372	0.0030
500	0.0019	0.0001

Case Study: Simulation-based Knowledge

Many times, a theoretical physics-based parametric model is hard to access for complex problems. Simulation-based models have become more and more popular as basic reference points. Here, the initial knowledge-based parametric model fk is assumed to be no longer available. Instead, a hypothetical simulation model replaces the former parametric white-box model. As a result, the function $f_k(x_1, x_2)$ cannot be used to generate the data needed to directly construct a black-box model and a subsequent grey-box model. Thus, a simplified white-box model is necessary since it is costly to run a high fidelity simulation for each point. To address this issue, a PR model was built to represent the white-box model. The manipulated function in previous subsectio was assumed to be the simulation model. 1000 simulated data points were generated from function $f_k(x_1, x_2)$ and were used to create the PR model using LHS. The reason that the manipulated function f_k was employed instead of directly using the original mystery function is because the simulation-based model is also assumed to be low fidelity. The white-box model in PR form was generated as:

$$\begin{aligned}\tilde{Y}_{PR} = f_{PR}(x_1, x_2) = &-0.3048 - 2.753x_1 - 0.228x_2 + 3.543x_1^2 + 1.973x_1x_2 + 1.973x_1x_2 \\ &+ 8.582x_2 + 0.5552x_1^3 + 0.9604x_1^2x_x + 0.9191x_1x_2^2 + 0.6433x_2^3 \\ &- 1.103x_1^4 - 0.7201x_1^3x_2 + 0.3864x_1^2x_2^2 - 0.5393x_1x_2^3 - 1.435x_2^4\end{aligned} \tag{7.3}$$

The R^2 value of this PR model is 0.7296. Comparing the PR white-box model to the original function yields an MREM of 846.6876 and an AREM of 1.9458. This indicates that this white-box model has poor predictability. This finding is reflected visually in the 3D surface and contour plots shown in Figure 7.10. In this figure, the shape is completely different from the original model (Figure 7.5). The ridges on the original surface disappeared. Thus, it is necessary to use the additional data points to build the grey-box model.

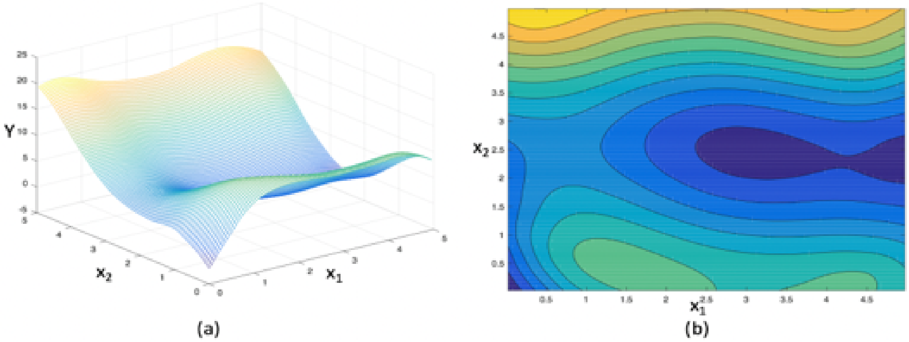


Figure 7.10: (a) 3D surface and (b) contour plots of the initial white-box model

The general updating process is similar to that shown in previous case study. The same 100 additional data points were used in this example compare the difference between to current case study. The Kriging black-box model was formulated by input variables and the corresponding residual values. The grey-box model was then developed by combining the PR and Kriging models by the process shown in Figure 7.8. The same validation process was executed to evaluate the model performance with the same validation dataset that was used previously. The final MREM and

AREM of this grey-box model are 3.4318 and 0.0506, which is slightly higher than using physics-based white-box model with same amount of additional data. The plots for this grey-box model are shown in Figure 7.11. The results from using additional data points are listed in Table 7.3.

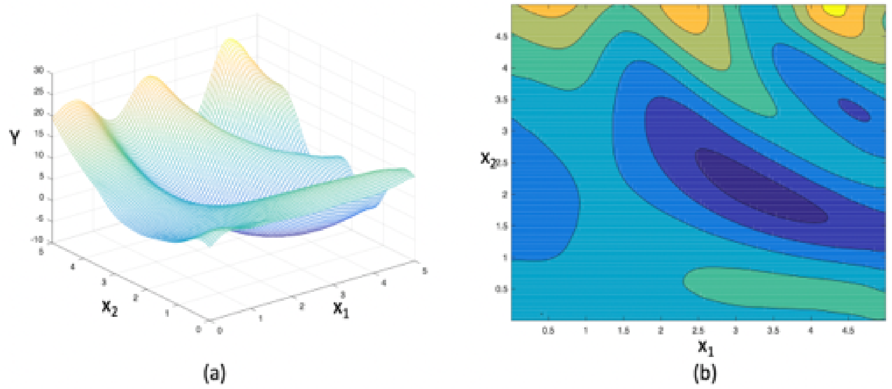


Figure 7.11: (a) 3D surface and (b) contour plots of the grey-box model built by simulation-based knowledge and additional actual data

Table 7.3: Model performance of additional data

Number of additional data	**MREM**	**AREM**
0	846.6878	1.9458
100	3.4318	0.0506
200	1.5988	0.0055
500	0.0052	0.0001

7.4.2 Case Study: Metal PBF Problem

This example uses the proposed method to build a grey-box model for a realistic PBF problem. Louvis and associates' experiments with a PBF process measured the relative density produced by different scan speed (v) and hatch spacing (d) for different aluminum alloy powders [118]. Relative density is the ratio of the actual part density to that of a completely filled solid with no porosity. The experimental results indicate higher relative density is generally produced by lower scan speed and closer hatch spacing. However, the relative density has unique behavior for specific powders and machines. For example, the density of AlSi12 and 6061 aluminum powder produced by the same parameters in different machines have different results [118], which indicates the a model built based on 6061's data may not be accurate for AlSi12. Instead of building an expensive new model, this study uses the findings from 6061's data to construct a grey-box model for AlSi12 for illustrative purposes.

In this case, there is no available physics-based knowledge to build the white-box model since the only available prior knowledge is from historical experiments. Therefore, the prior experimental knowledge of 6061 powder [118] was used to build a PR model to serve as the white-box model for illustrative purposes just as was done in previous section. The reported measurements from AlSi12 powder were used as the additional information. First, the 177 data points from the 6061 powder experiment were used to build the PR based white-box model. The resulting quadratic model was generated as:

$$\tilde{Y}_P R = f_{PR}(v,d) = 81.54 + 116.83v - 0.0127d - 0.079vd - 332.39v^2 + 7082600d^2 \quad (7.4)$$

The initial R^2 value of this model is 0.953. The set of 36 data points from the AlSi12 PBF experiment was divided into two sets. 80% (29 points) of the data was extracted from the initial dataset to use as additional information for grey-box construction. The remaining 20% (7 points) of the data set was used to validate the models. Table 7.4 lists the MREM and AREM for different types of models based on the data. The 7 validation data points from the AlSi12 experiment were entered into all three types of models to evaluate and compare the predictive accuracy. The pure white-box is the PR model built using the 6061 powder experiment. The pure black-box model represents the model built with the 29 AlSi12 data points with kriging method. The grey-box represents the model built with input from both experiments using the same method presented in the prior sections. As shown, even though the pure white-box model has low predictive error, the model can be further improved by the grey-box modeling approach. The MREM of the original model is reduced from 0.0375 to 0.0238, which is a 37% improvement. Compared to the pure black-box model, the MREM of the grey-box model reduced from 0.0485 to 0.0238, which is a 51% improvement after the completion of both modeling stages.

Table 7.4: The performance of different types of models

	MREM	**AREM**
Pure white-box	0.0375	0.0170
Pure black-box	0.0485	0.0169
Grey-box	0.0238	0.0134

7.5 Discussion

Effectively deploying predictive analytics in smart manufacturing is a challenge. This challenge is highlighted in AM, where current AM models often lack comprehensive information, and where information could be either knowledge-based or statistically generated. The lack of the former is typically the result of an incomplete understanding of the physical processes of AM. The lack of empirical data, on the other hand, may be caused by the difficulty of instrumenting AM processes, and more generally, the expense of producing AM parts. Even as more and more empirical data is available in the coming years, it is still difficult to duplicate all the conditions and the model predictability for all data sets. The experimental results can exhibit noticeable differences even where the experiments are operated in similar AM processes with comparable process parameters. It is also very difficult to build the connection between simulations and actual experimental data. These difficulties result from the uncertainties in and complexities of AM processes. The uncertainties significantly reduce the utility of AM predictive models since a well-validated model from one dataset may be difficult to apply to other experimental conditions.

The highlight of the two-stage grey-box modeling approach developed in this paper is that it can combine disparate knowledge and information together to produce an accurate hybrid model. To further extract the information from limited knowledge, this two-stage grey-box structure can functionally improve the predictive accuracy. However, a smaller sample size is expected in actual AM experiments. It is thus desirable to further reduce that sample size needed to achieve the higher model predictability.

The involved metamodeling algorithms were used as primary candidates to build the grey-box in this work. However, the general modeling process should have no bias to other black-box modeling techniques. Any suitable algorithm that can improve the model predictability might be introduced in future work

CHAPTER 8

DYNAMIC METAMODELING FOR PREDICTIVE ANALYTICS IN ADVANCED MANUFACTURING

8.1 Overview

Kriging as an important metamodeling method has been frequently deployed in this research. In Chapter 5, Kriging serves as a candidate method that would be used to build the optimal metamodel. The MPEU method from Chapter 6 is developed based on the characteristics of Kriging method. Chapter 7 builds the grey-box model by combining physics-based white-box model and Kriging based black-box model. It can work properly to predict the outcomes in complex and random processes of AM. However, for large and non-ideal data sets in SMS, the Kriging method may lose its predictability and efficiency.

To address these potential vulnerabilities, this chapter introduces a novel, dynamic metamodeling method that adapts Kriging covariance matrices to improve predictability in contextualized, non-ideal data sets. A key highlight of this approach is the optimal linking process, based on the location of prospective points, to alter the conventional stationary covariance matrices. This process reduces the size of resulting dynamic covariance matrices by retaining only the most critical elements necessary to maintain accuracy and reliability of new-point predictability. To further improve model fidelity, both the Gaussian parameters and design space attributes are optimized holistically within a problem space. Case studies with a representative test function show that the resulting Dynamic Variance-Covariance Matrix (DVCM) method is highly efficient without compromising accuracy [161]. A second case study representative of an advanced manufacturing setting demonstrates the applicability and advantages of the DVCM method, including significantly increased model robustness.

8.2 Background

In today's advanced manufacturing systems, model-based system engineering (MBSE) principles often guide how data are best used to evaluate, optimize, verify, and validate alternative system models [162, 163, 164]. Common MBSE techniques such as metamodeling have successfully been used in engineering for simplifying predictions of behavior for complex systems [78, 89, 90, 103]. In engineering design scenarios, representative data can be simulated in abundance. However, in process modeling of advanced manufacturing systems, in which direct data collection is preferred, accumulating data can be challenging and cost prohibitive [165, 166]. In such cases, manufacturers are often faced with the alternative of highly complex, high-fidelity physics-based simulations [108] that may not be representative of the process. This chapter introduces a scenario-agnostic dynamic metamodeling approach, in which the metamodel state is adapted and altered based on localized design space characteristics. This approach possesses unique capabilities that are able to handle the multifaceted data forms that are representative of advanced manufacturing systems.

Metamodeling approaches offer an excellent foundation to build predictive analytics, as they can establish robust models to realistically represent physical phenomena without intrinsic knowledge of the complex system [79, 80]. In predicting behaviors of advanced manufacturing processes, as is typical with metamodel applications, the trade-off between high fidelity and cost-effectiveness becomes a central issue in the model development process. Model fidelity, which represents the "degree of similarity" [167], can be influenced by multiple conditions, such as sampling scheme or data quality [78, 148]. A major challenge in the construction of metamodels of complex manufacturing processes is in the data collection process itself vis-à-vis how to know where to collect reliable data for model building of unknown systems prior to model construction [22]. Standard metamodeling methods, such as the Kriging technique [168, 86], are not well suited for advanced manufacturing systems, as the sampling points can greatly influence the resulting metamodels. To this end, we has introduced novel techniques to improve metamodeling of advanced manufacturing systems with methods that recursively update kriging metamodels [127]. We combine the benefits of kriging with knowledge-based models [169] and use process knowledge to select the best metamodeling technique [170].

An underpinning of our Kriging-based approach is that the effectiveness of any metamodeling techniques should be customized to meet the predictive analytics requirements of SMS. Accordingly, the preferred metamodeling approach is one that can (a) improve modeling efficiency without permanently eliminating sampling points from an existing data set; (b) improve the overall performance of a traditional kriging-based metamodeling process for large and nonideal data sets through modification of fundamental kriging covariance matrices; and (c) be equally applicable to construct different types of data-driven manufacturing predictive models. To meet these goals, this chapter introduces the dynamic variance-covariance matrix (DVCM) method, developed specifically to dynamically adjust the size of Kriging matrices derived from SMS. Section 8.3 introduces the DVCM method for dynamically reducing the size of the distance matrices and deriving DVCM parameters for optimization. To demonstrate the utility of the proposed method, the section "Demonstration of the DVCM Method" executes the DVCM method in two case studies illustrative of different aspects of advanced manufacturing systems. The effectiveness of the method and results are discussed and summarized in Section 8.4.2.

8.3 DVCM Method

From Section 3.2.2 the number of elements within matrix C from Equation 3.10 for a sampling size of n is n^2. It may potentially require $O(n3)$ computations to obtain the inversion of an $n \times n$ matrix [86], where O represents the Big O notation (computation time taken by an algorithm in computer science) [171]. In these circumstances, computation of inversion for a very large matrix becomes very time consuming even for modern computer systems, which indicates that the conventional Kriging method can lose efficiency when used for the very large data sets that can occur in manufacturing scenarios. Alternatively, reductions in data may permanently decrease the predictive accuracy. For SMS scenarios, a preferred method would account for the balance between predictive accuracy and computational cost.

Kriging matrices and vectors are constructed by relative distance and the predetermined spatial correlation function (Section 3.13. A value of 1 represents strongest correlation and 0 indicates no correlation. Figure 8.1 shows an example of a Gaussian covariogram in which the farthest distance between all given points is 6 and θ is equal to 1.4. For demonstrative purposes, all the numbers are randomly generated. As shown, the correlation diminishes rapidly to negligible correlation above radial distance values of 3. For a different Gaussian covariogram, the exact threshold value of the radius may vary because of the different distribution behavior. The size of the original distance matrix can consequently be reduced by eliminating data points that are outside of the determined radius. As the DVCM method is dynamic to maintain model integrity, any elimination is temporary and subject to reevaluation when other new points are estimated.

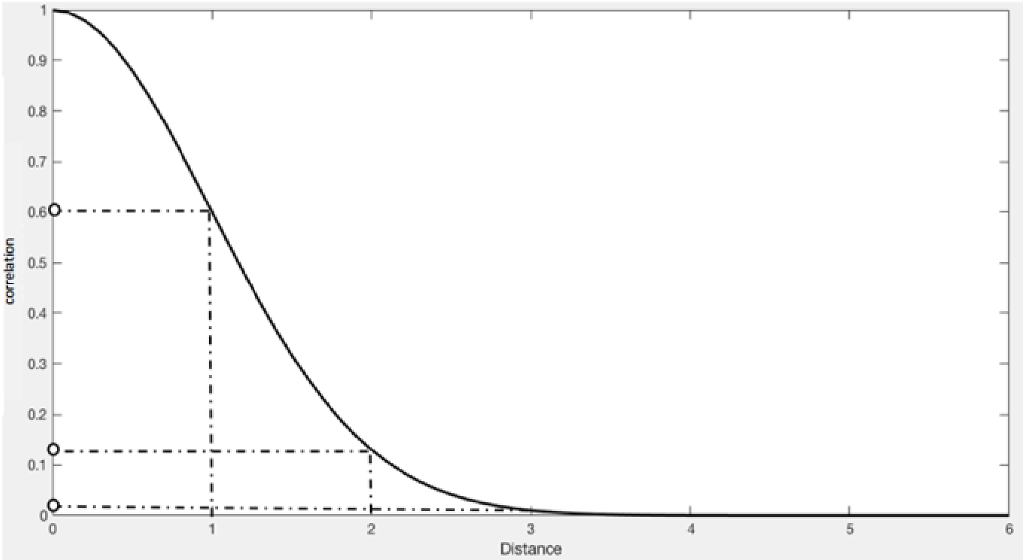


Figure 8.1: Gaussian covariogram for maximum distance of 6 and $\theta = 1.4$

Next, the original distance matrix D is transformed to the dynamic distance matrix $\dot{D}_k$, where k is the index of the estimating point. For single point estimation, k is equal to 1. Elements in $\dot{D}_k$ corresponding to non-selected points would be eliminated from the original matrix D. For example, if $d(i, j)$ is dropped due to being outside of the radius, the entire i_{th} row and j_{th} column would be erased from D. Thus, corresponding reductions in the $\dot{C}_k$ and $\dot{B}_k$ matrices that determine Λ reduce the computational cost of inversion from $O(n^3)$ to $O((n-r)^3)$, where r represents the number of eliminated elements. For example, an original 100 point problem has 10,000 elements in the matrix, but the reduced matrix only has 2,500 elements (75% reduction) if half the points are temporally eliminated. Moreover, the propagated numerical error of the matrix inversion can also be avoided by reduction.

The predictive accuracy and reduction of the DVCM method highly depend on the value of correlation parameter θ and optimal radius r. From Section 3.2.2, the optimal value of θ determines whether the covariogram can accurately represent the unknown system. Different methods have different advantages and disadvantages, such as the H-J method's [100] inability to provide the true optimum despite its efficiency or the L-M method's limitation to provide only the local optimum [101, 102]. In this chapter, GA method [103] is deployed to optimize design variables of θ and r. Parameter θ is the optimal correlation parameter that would be used for the kriging model, which should be larger than 0 to indicate that the correlation exists. The upper bound c_1 of θ is determined by the covariogram and the size of design space as determined by the user for a given problem. In this study, it was manually set to 10. Radius r specifies a distance from the estimating point in which data points are included in that estimation. The objective function of this optimization problem minimizes the average relative error magnitude (AREM), given in Equation 8.1. AREM is determined from an independent data set that contains m data points. Here, parameter y_i is the observed value at point i. r_{lb} is the lower bound of r that can guarantee that the DVCM is not empty, and r_{ub} is based on the maximum acceptable matrix size. The estimated response at point i is a function of θ, r, and data point values, x_i, as shown in Equation 8.1.

$$\begin{aligned} minimize: &\quad AREM(\theta, r) = \textstyle\sum_{i=1}^{m} \left|\frac{f_i(x_i,\theta,r)-y_i}{y_i}\right| \quad (y_i \neq 0) \\ subjectto: &\quad 0 \leq \theta \leq c_1 \\ &\quad r_{lb} \leq r \leq r_{ub} \end{aligned} \tag{8.1}$$

The θ and r values optimization procedure is introduced in Figure 8.2. The method first divides sampling points into training and testing sets. The training set constructs the metamodel that the testing set validates. Because the AREM and the MREM (Equation 5.4) are highly correlated, the latter would only be evaluated at the optimal AREM to improve computational efficiency.

The optimization process described in Figure 8.2 is iteratively updated by

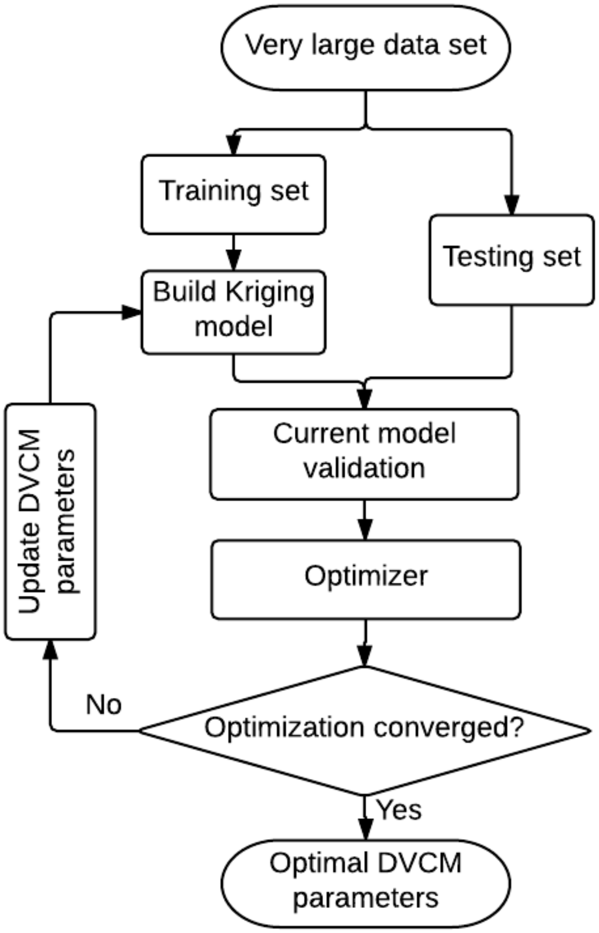


Figure 8.2: Optimization process for obtaining θ and r values

the DVCM process for point estimation (Figure 8.3). Based on the location of an estimated point $Z_{E,k}$, and the optimal parameter r derived from the previous optimization process, the original distance matrix D reduces to the DVCM distance matrix $\dot{D}_k$. The matrix $\dot{C}_k$ is next derived by the optimal θ parameter by the formulation in Section 3.2.2. The estimated value of each point $Z_{E,k}$ is computed from these temporary matrices. In next subsection, examples are demonstrated by a classical function [4, 33, 36] to illustrate how this method functions mathematically. The AM case study tests the method performance with large AM data sets generated by simulation.

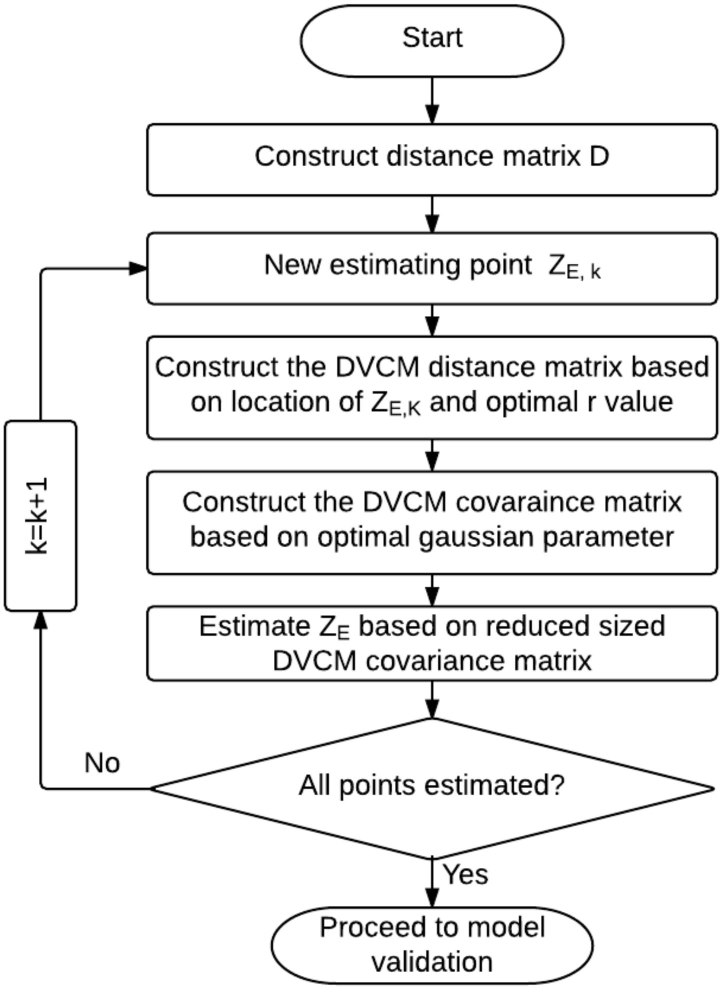

Start
Construct distance matrix D
New estimating point $Z_{E,k}$
Construct the DVCM distance matrix based on location of $Z_{E,K}$ and optimal r value
Construct the DVCM covaraince matrix based on optimal gaussian parameter
Estimate Z_E based on reduced sized DVCM covariance matrix
All points estimated?
No
k=k+1
Yes
Proceed to model validation

Figure 8.3: General DVCM algorithm for point estimation

8.4 Demonstration of DVCM Method

Two case studies will be presented. One validates the approach from a mathematical point of view and a second showcases its usefulness in advanced manufacturing. The first case study represents a complex nonlinear function. The second case study predicts an actual additive manufacturing process.

8.4.1 Case Study 1: Function to Represent Complex System

The first case study tests the development of the DVCM with both small and large data sets. Section 8.4.1.1 illustrates the process of building the DVCM from eight data points. Section 8.4.1.2 gives a comprehensive presentation of the DVCM method by comparing estimated results with and without using the DVCM method and with different DVCM parameters used for a larger data set. Mystery Function (Equation 7.1) is deployed in these examples.

8.4.1.1 Limited Data Scenario

In the first scenario, we simulate a small data set, perhaps the initial runs of a new production line. Until new data are generated that are representative of the observed process, we can only extrapolate known partial data sets. In this example, eight randomly generated data points (normal) are used to test the versatility of the dynamic matrix reduction process in this DVCM method, where Z_1 through Z_6 represent existing data points and $Z_{E,1}$ and $Z_{E,2}$ are the estimated points. This illustrative example focuses on the process of constructing DVCM matrices and vectors using the fundamentals described in Section 3.2.2, rather than metamodel construction. The locations of all the generated data are shown in Figure 8.4, with specific locations of the data listed in Table 8.1.

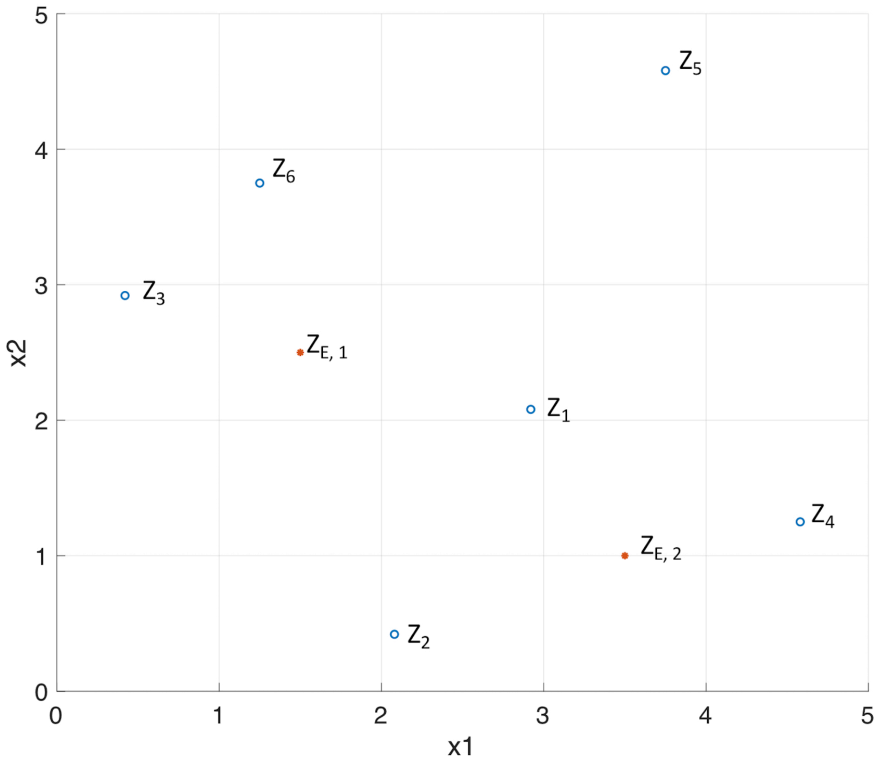


Figure 8.4: Location of existing data points (Z_1 to Z_6) and estimating points ($Z_{E,1}$ and $Z_{E,2}$)

Table 8.1: Sampling and estimating points locations

	Location	
NO.	$\mathbf{x_1}$	$\mathbf{x_2}$
$\mathbf{Z_1}$	2.92	2.08
$\mathbf{Z_2}$	2.08	0.42
$\mathbf{Z_3}$	0.42	2.92
$\mathbf{Z_4}$	4.58	1.25
$\mathbf{Z_5}$	3.75	4.58
$\mathbf{Z_6}$	1.25	3.75
$\mathbf{Z_{E,1}}$	1.50	2.50
$\mathbf{Z_{E,2}}$	3.50	1.00

The original distance matrix for the existing 6 data points is formulated as:

$$\mathbf{D} = \begin{array}{c} \\ Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \end{array} \begin{array}{c} \begin{array}{cccccc} Z_1 & Z_2 & Z_3 & Z_4 & Z_5 & Z_6 \end{array} \\ \left[\begin{array}{cccccc} 0.00 & 1.86 & 2.64 & 1.86 & 2.63 & 2.36 \\ 1.86 & 0.00 & 3.00 & 2.63 & 4.48 & 3.43 \\ 2.64 & 3.00 & 0.00 & 4.48 & 3.72 & 1.17 \\ 1.86 & 2.63 & 4.48 & 0.00 & 3.43 & 4.16 \\ 2.63 & 4.48 & 3.72 & 3.43 & 0.00 & 2.63 \\ 2.36 & 3.43 & 1.17 & 4.16 & 2.63 & 0.00 \end{array}\right] \end{array} \tag{8.2}$$

And distance vectors for $Z_{E,1}$ and $Z_{E,2}$ are formulated as:

$$\mathbf{A_{E,1}} = \begin{array}{c} \\ Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Z_6 \end{array} \begin{array}{c} Z_{E,1} \\ \left\{\begin{array}{c} 1.48 \\ 2.16 \\ 1.16 \\ 3.32 \\ 3.06 \\ 1.27 \end{array}\right\} \end{array} \tag{8.3}$$

$$\mathbf{A_{E,2}} = \begin{array}{c c} & Z_{E,2} \\ Z_1 & \left\{\begin{array}{c} 1.23 \\ 1.53 \\ 3.63 \\ 1.11 \\ 3.59 \\ 3.55 \end{array}\right\} \\ Z_2 & \\ Z_3 & \\ Z_4 & \\ Z_5 & \\ Z_6 & \end{array} \tag{8.4}$$

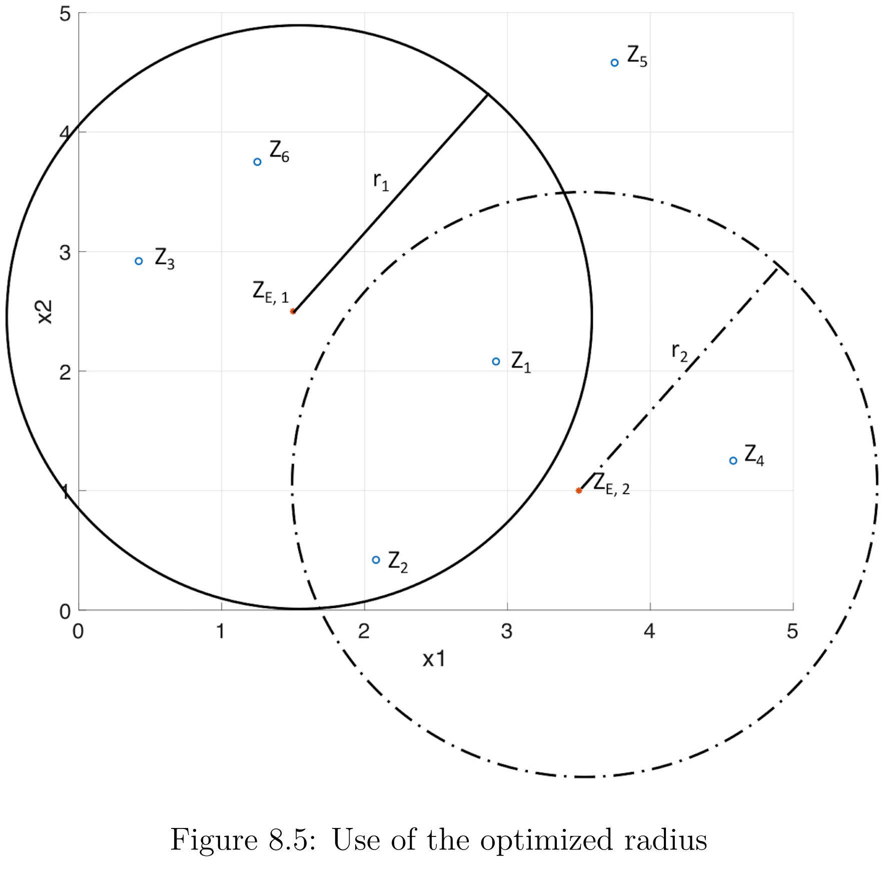


Figure 8.5: Use of the optimized radius

For illustrative purposes, we assume the optimal r value for this example is 2.5 (shown in Figure 8.5). The solid circle marks the selective region for $Z_{E,1}$ with radius

r_1 and the dashed circle marks the selective of region for $Z_{E,2}$ with radius r_2. Based on observation, two points, Z_4 and Z_5, become temporally eliminated for the DVCM distance matrix of $Z_{E,1}$. Similarly, three points, Z_3, Z_5 and Z_6, become temporally eliminated for the DVCM distance matrix of $Z_{E,2}$. The first step of the reduction process for constructing DVCMs is completed at this stage.

The DVCM to estimate the value at point $Z_{E,1}$ can now be reduced from the original distance matrix D to:

$$\tilde{\mathbf{D}}_{\mathbf{E,1}} = \begin{array}{cc} & \begin{array}{cccccc} Z_1 & Z_2 & Z_3 & \cancel{Z_4} & \cancel{Z_5} & Z_6 \end{array} \\ \begin{array}{c} Z_1 \\ Z_2 \\ Z_3 \\ \cancel{Z_4} \\ \cancel{Z_5} \\ Z_6 \end{array} & \begin{bmatrix} 0.00 & 1.86 & 2.64 & \cancel{1.86} & \cancel{2.63} & 2.36 \\ 1.86 & 0.00 & 3.00 & \cancel{2.63} & \cancel{4.48} & 3.43 \\ 2.64 & 3.00 & 0.00 & \cancel{4.48} & \cancel{3.72} & 1.17 \\ \cancel{1.86} & \cancel{2.63} & \cancel{4.48} & \cancel{0.00} & \cancel{3.43} & \cancel{4.16} \\ \cancel{2.63} & \cancel{4.48} & \cancel{3.72} & \cancel{3.43} & \cancel{0.00} & \cancel{2.63} \\ 2.36 & 3.43 & 1.17 & \cancel{4.16} & \cancel{2.63} & 0.00 \end{bmatrix} \end{array} = \begin{array}{cc} & \begin{array}{cccc} Z_1 & Z_2 & Z_3 & Z_6 \end{array} \\ \begin{array}{c} Z_1 \\ Z_2 \\ Z_3 \\ Z_6 \end{array} & \begin{bmatrix} 0.00 & 1.86 & 2.64 & 2.36 \\ 1.86 & 0.00 & 3.00 & 3.43 \\ 2.64 & 3.00 & 0.00 & 1.17 \\ 2.36 & 3.43 & 1.17 & 0.00 \end{bmatrix} \end{array} \tag{8.5}$$

Similarly, the DVCM vector for point $Z_{E,1}$ can be written as:

$$\tilde{\mathbf{A}}_{\mathbf{E},\mathbf{1}} = \begin{array}{cc} & Z_{E,1} \\ \begin{array}{c} Z_1 \\ Z_2 \\ Z_3 \\ \cancel{Z_4} \\ \cancel{Z_5} \\ Z_6 \end{array} & \left\{ \begin{array}{c} 1.48 \\ 2.16 \\ 1.16 \\ \cancel{3.32} \\ \cancel{3.06} \\ 1.27 \end{array} \right\} \end{array} = \begin{array}{cc} & Z_{E,1} \\ \begin{array}{c} Z_1 \\ Z_2 \\ Z_3 \\ Z_6 \end{array} & \left\{ \begin{array}{c} 1.48 \\ 2.16 \\ 1.16 \\ 1.27 \end{array} \right\} \end{array} \tag{8.6}$$

The DVCM and vector of point $Z_{E,2}$ become:

$$\tilde{\mathbf{D}}_{\mathbf{E},\mathbf{2}} = \begin{array}{cc} & \begin{array}{cccccc} Z_1 & Z_2 & \cancel{Z_3} & Z_4 & \cancel{Z_5} & \cancel{Z_6} \end{array} \\ \begin{array}{c} Z_1 \\ Z_2 \\ \cancel{Z_3} \\ Z_4 \\ \cancel{Z_5} \\ \cancel{Z_6} \end{array} & \left[\begin{array}{cccccc} 0.00 & 1.86 & \cancel{2.64} & 1.86 & \cancel{2.63} & \cancel{2.36} \\ 1.86 & 0.00 & \cancel{3.00} & 2.63 & \cancel{4.48} & \cancel{3.43} \\ \cancel{2.64} & \cancel{3.00} & \cancel{0.00} & \cancel{4.48} & \cancel{3.72} & \cancel{1.17} \\ 1.86 & 2.63 & \cancel{4.48} & 0.00 & \cancel{3.43} & \cancel{4.16} \\ \cancel{2.63} & \cancel{4.48} & \cancel{3.72} & \cancel{3.43} & \cancel{0.00} & \cancel{2.63} \\ \cancel{2.36} & \cancel{3.43} & \cancel{1.17} & \cancel{4.16} & \cancel{2.63} & \cancel{0.00} \end{array} \right] \end{array} = \begin{array}{cc} & \begin{array}{ccc} Z_1 & Z_2 & Z_4 \end{array} \\ \begin{array}{c} Z_1 \\ Z_2 \\ Z_4 \end{array} & \left[\begin{array}{ccc} 0.00 & 1.86 & 1.86 \\ 1.86 & 0.00 & 2.63 \\ 1.86 & 2.63 & 0.00 \end{array} \right] \end{array} \tag{8.7}$$

$$\tilde{\mathbf{A}}_{\mathbf{E,2}} = \begin{array}{cc} & Z_{E,2} \\ \begin{array}{c} Z_1 \\ Z_2 \\ \cancel{Z_3} \\ Z_4 \\ \cancel{Z_5} \\ \cancel{Z_6} \end{array} & \left\{ \begin{array}{c} 1.23 \\ 1.53 \\ \cancel{3.63} \\ 1.11 \\ \cancel{3.59} \\ \cancel{3.55} \end{array} \right\} \end{array} = \begin{array}{cc} & Z_{E,2} \\ \begin{array}{c} Z_1 \\ Z_2 \\ Z_4 \end{array} & \left\{ \begin{array}{c} 1.23 \\ 1.53 \\ 1.11 \end{array} \right\} \end{array} \qquad (8.8)$$

The original 6 × 6 distance matrix is now reduced to a 4 × 4 matrix for point $Z_{E,1}$ and a 3 × 3 matrix for point $Z_{E,2}$. Thus, the total number of elements in these DVCMs is reduced from the original 36 elements to 16 and 9, respectively. However, the complete data set is still available for more point estimations.

8.4.1.2 Large Data Set Scenario

Here we test the DVCM method against large data sets, which can be a challenge when building models of complex manufacturing systems. This second example, using the same nonlinear function, is designed to test the versatility of the DVCM method. A total of 100 points were generated from Latin Hypercube (LH) [141, 172]. The surface and contour plots of the exact solution of this function are shown in Figure 7.5, which also used in Equation 7.1. To best demonstrate the effects of both estimation parameters, this case study is executed in two sequential stages of the DVCM method.

In the first stage of the DVCM method (Figure 8.2), increments of θ values ranged from 3, the initial guess, to an optimal value of 1.0496, determined by the procedure. The values of 2, 1.5, and 1.25 included in this table were incrementally

Table 8.2: Estimation error as a function of θ

θ	**AREM**	**MREM**
3	35.45%	1287.63%
2	5.07%	141.76%
1.5	4.79%	112.70%
1.25	3.66%	59.55%
1.0496*	2.17%*	40.98%*

selected to illustrate the optimization process. These values were derived from a single variable GA optimization without varying parameter r. The results from testing estimates from 1000 additional random points (Table 8.2 and Figure 8.6) show that manipulation of parameter θ can positively affect the predictive accuracy of the metamodel, providing a more accurate spatial correlation for this unknown system. A "misread" initial correlation might fail to assign the appropriate weight factors to a given data point. When this happens, the estimated new points for Z_E are unlikely to approach true values accurately until θ is optimized.

In the second stage we optimize both θ and r using the entire process, with the goal of using all accessible data points for estimation without any reduction of computational efficiency. The Maximum Predictive Error Updating (MPEU) method [127] was applied to separate the 100 LH points into training and testing data sets. The distance matrix was reduced in size to only those data points within the neighborhood region determined by radius r, similar to the Section 8.4.1.1. Table 8.3 lists the returned error of the metamodels constructed with different DVCM parameters. All case studies were run on the same computer system to evaluate metamodeling efficiency.

We are able to demonstrate that the optimal design of both DVCM parameters will simultaneously improve predictive accuracy while reducing computational

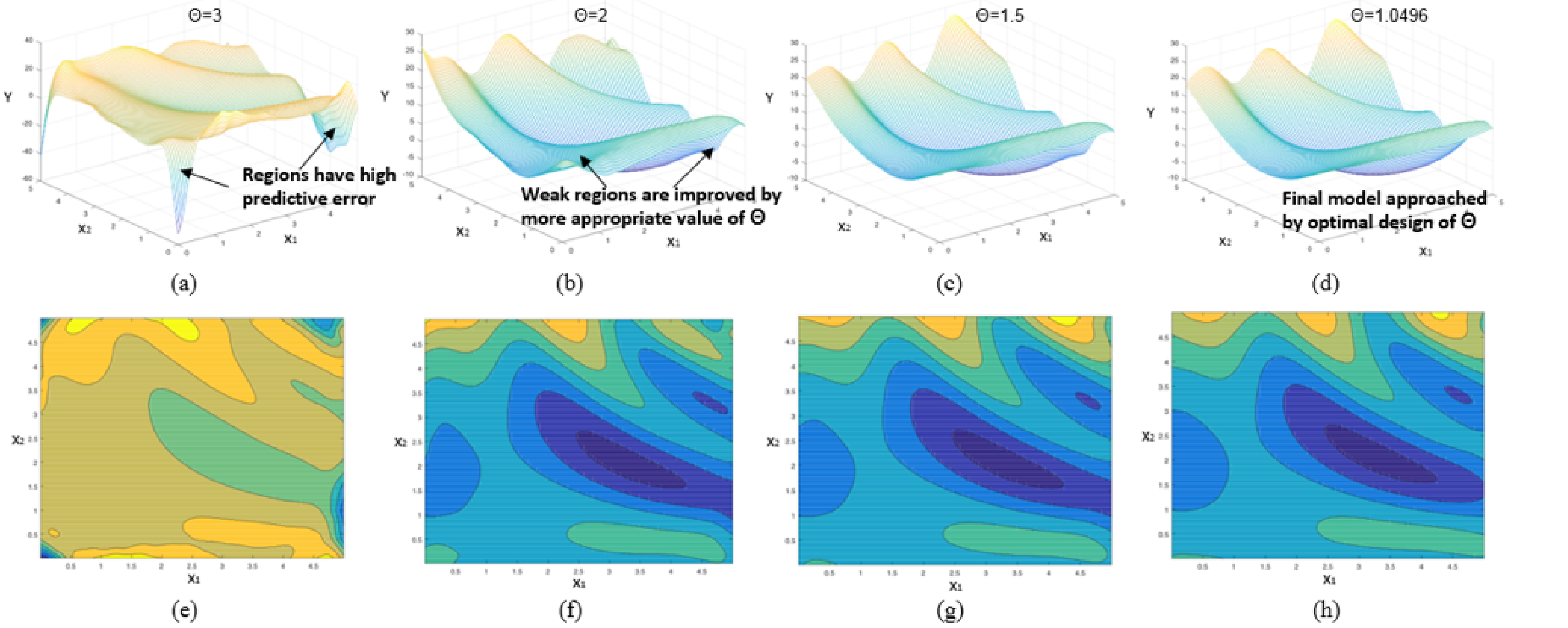


Figure 8.6: 3D surface and 2D contour plots of the metamodels built by θ values of: (a) and (e) $\theta = 3$; (b) and (f) $\theta = 2$; (c) and (g) $\theta = 1.5$; (d) and (h) the optimal design at $\theta = 1.0496$

Table 8.3: Metamodeling performance of DVCM

θ	**r**	**AREM**	**MREM**	Average points used	Total running time (s)
1.0496	**n/a**	2.17%	40.98%	100(100)	45.396
1.0496	**4.000**	2.05%	42.42%	85.06(100)	32.506
1.0496	**3.000**	2.04%	35.63%	62.38(100)	19.673
1.0760	**1.718**	1.48%	12.41%	26.70(100)	6.910

cost by limiting the size of original matrices. Furthermore, reducing the size of distance matrices does not increase the error for equivalent values of θ. However, the original optimal parameter θ shifted to a slightly higher value because of the stronger spatial correlation relationship between sampling points when using fewer points for prediction. The results in Table 8.3 were derived from two data sets. The first data set that includes the 100 LH points was divided into two subsets. One subset was used to build the Kriging model, and the other subset was used to find the optimal DVCM parameters. The second data set that includes the 100 random points was only used to calculate the AREM and MREM, which was not involved in the optimization work. An optimal DVCM radius was found at 1.7180, which is considerably smaller than the farthest pair of points in the design space (Euclidean distance equal to 7.071). Notably, the maximum error (MREM) column in Table 8.3 was reduced to less than one third the magnitude of that in Table 8.2. This reduction suggests significantly better predictability with the DVCM method. For comparison, the authors reproduced the same problem by use of the DACE kriging toolbox [35]. The MREM was found to be 62.37% with the same data set. The optimization of r using the DVCM method resulted in a speed increase of nearly six times faster than that found without distance matrix reduction. Moreover, Figure 8.7 nearly matches every detail in Figure 7.5, the true representation of this example function.

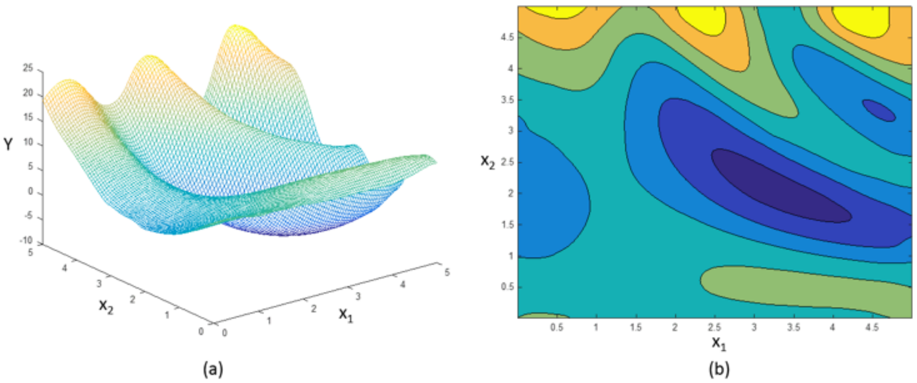


Figure 8.7: 3D surface and 2D contour plots of optimal DVCM metamodel

8.4.1.3 Case Study 2: Thermal Sub-process in Metal AM

This case study demonstrates the use of the DVCM method to build a metamodel for predicting AM melt pool size, part of a critical sub-process in metal AM systems. Laser PBF (L-PBF) AM can be divided into different sub-processes, such as a heating process, melting process, solidification process, etc. [22]. A predictive AM model may include multiple modeling techniques such as finite element models, physics-based models, and statistics-based models [76, 52, 151].

Metal L-PBF AM processes fabricate solid objects directly from three-dimensional (3D) model data by melting metal powders with a laser layer by layer [173]. Metal L-PBF is considered one of the most promising AM technologies because of its superior surface and geometric quality and its potential to work with a wide spectrum of materials. Contrary to other AM processes, L-PBF can produce almost fully dense products and reach mechanical properties that sometimes surpass those of components produced by traditional manufacturing methods.

Modeling continues to play a large part in maturing metal laser PBF technologies. A large number of processing parameters (e.g., material, laser power, scanning speed, layer thickness, scanning pattern) influence various process qualities, such as the prevention of under- and over-melting defects, dimensional accuracy and surface finish, microstructure and mechanical properties, and the distribution of residual stresses in the product. Various process signatures have been proposed to study the relationships between observations and process qualities in L-PBF processes [74].

Here, we choose to investigate melt pool width as a process signature because of the direct relationship it creates between geometry and solidification, which ultimately determines dimensional accuracy, microstructure, and mechanical properties (e.g., tensile strength). Larger melt pools may result in rough surface finish and poor dimensional accuracy for small features, whereas smaller melt pools may cause voids, low density, and low tensile strength. Melt pool width can be observed during the process with noninvasive cameras and after the process [174].

Both experimental and computational studies have sought to determine the influence of process parameters on melt pool geometry in L-PBF. Melt pool simulations continue to pose a challenge, as the physics are difficult to model, and empirical data are limited because of the high cost of running experimental studies. As an alternative, low-cost simulations offer the ability to quickly explore the large data sets associated with the high-dimensional spaces required for process optimization. Given the high complexity and difficult-to-define behavior of AM systems, data reduction before model validation can become extremely difficult. Additionally, constructing a conventional kriging metamodel over the substantial design space may also be very difficult because of the extremely high computational cost.

In this study, we use a heat transfer simulation to investigate the influence of laser power (P), scan speed (v), absorption coefficient (A), and thermal diffusivity (α) in melt pool width. This simulation accounts for heat transfer on a single scan track, neglecting track-to-track and layer-to-layer interactions. The model is based on the isotherm migration method for laser cladding [151], and adjusted for laser PBF [174]. In this case study, 1050 melt pool width data points were generated to illustrate the utility of the DVCM method. For consideration, the deployment of a conventional kriging method would result in a distance matrix that contains more than one million elements (10502 elements for this square matrix). The generation of the covariance matrix and its inversion is unrealistic and inefficient. However, these challenges highlight the efficacy of proposed DVCM method for its efficiency and accuracy.

Table 8.4 summarizes the experimental setup of this case study. Laser power, scan speed, heat absorption coefficient, and thermal diffusivity are the four input variables for simulating the melt pool width. A total of 1050 data points were generated to fill the design space of a full factorial design of experiment (DOE). The simulation results of melt pool width ranged from 87.57 to 244.80 μm. Here we adopt reduced order simulations without further corroboration to mimic a manufacturing scenario of non-ideal data.

Table 8.4: Experimental design of AM case study

Process factors	**Symbols**	**Levels**	**Range of value**
Laser power (w)	P	7	180 - 210
Scan speed (**m/s**)	v	5	0.75 - 0.85
Heat absorption coefficient ($\mathbf{W/(m^2K)}$)	A	5	0.5 - 0.7
Thermal diffusivity ($\mathbf{m^2/s}$)	α	6	0.75 - 1.75

After normalizing all input variable values, the entire data set was divided into training and testing sets with 750 of the points used to construct the metamodel. By deployment of the Minimum Euclidean Distance (MED) method with LH sampling [127], 100 points were selected to optimize the DVCM parameters. The c_1 (upper bound of parameter θ) value was set at 5 based on the initial maximum likelihood analysis. The r_{lb} value was set to 0.3 to avoid potential empty DVCM matrices based on the calculation of the minimum Euclidean distance between points. The r_{ub} value was set to 0.6 to guarantee that the DVCM is restricted to 200 data points, or 40,000 elements. Table 8.5 shows both errors at the optimal solution are reduced to less than 4% of the initial errors. Furthermore, the number of points required to construct new DVCM estimations is reduced to less than half of the number of points initially required, with a corresponding element reduction of 76% due to the optimized radius parameter value.

Table 8.5: Results of DVCM parameter optimization

Steps	θ	**r**	**AREM**	**MREM**	**Average points used**
Initial	1.000	0.6000	13.00%	141.00%	109(750)
	0.6020	0.4847	17.00%	257.00%	51(750)
	0.9970	0.4504	6.60%	170.00%	43(750)
Intermediate	1.0600	0.4720	3.59%	67.00%	51(750)
	3.1980	0.4703	0.57%	7.01%	48(750)
	3.5120	0.4713	0.53%	5.71%	48(750)
Optimal	4.5405	0.4997	0.44%	2.85%	53(750)

8.4.2 Discussion

The case studies in the section "Demonstration of DVCM Method" demonstrate that optimization of the SCF's correlation parameter θ can significantly reduce average predictive error (AREM) in the DVCM method. Although significant improvements can be made in the MREM, which typically represents the outstanding predictive error, it may not be amenable to further improvement if the correlation parameter is already optimal. In such cases, the newly introduced DVCM parameter can work jointly with the optimal for further improvement. The results indicate that, for specific scenarios, the DVCM approach has the ability to use fewer data to produce more accurate predictions.

This observation seems to contradict common intuition that more data can uncover hidden information about an unknown system to improve predictive analytics. Though often correct, scenarios exist in which potential weaknesses are shown when applied to large data sets, such as the findings from this AM study in Section 8.4.1.3. These findings indicated that estimation done locally by this DVCM approach may result in better performance, given that the local data are of the right type. A possible explanation for this phenomenon can be partially seen in the fundamental structure of the Kriging estimation algorithm. The covariance matrix that follows from an initial distance matrix has a diagonal structure. Elements contained in the covariance matrix usually formulate from 0 to 1 based on the determined covariogram. Conventional Kriging methods tend to construct the matrices with all given data, regardless of the distance between the data points and the estimating point. For a large data set, a vast number of data points are widely distributed in the entire design space, meaning a number of them may not be strongly correlated according to their remote distance. Thus, it follows that many of the elements contained in a resulting distance matrix are close to a 0 correlation. This DVCM approach can fundamentally avoid the aforementioned disadvantage by constructing a distance matrix locally and dynamically. Therefore, the outstanding estimated error of MREM can be significantly reduced with this approach.

Some issues were observed in the detailed inspections of certain DVCM distance matrices. Sets of points located close to the edge or corner of the design space or in low population areas may have an unreasonably small DVCM distance matrix. In the AM case study, for example, the smallest matrix contains only 15 data points, which is considerably inadequate given the average of 53 points. As differently sized matrices had the same correlation parameter applied to them, this could potentially produce inaccurate estimating results. One way to address this challenge is to allow the radius to be flexible with regard to its location and neighborhood population. With such a modification, all DVCM matrices could hypothetically be established without bias to a predetermined radius. With an equally sized matrix for every estimating point, the outstanding MREM might be further reduced to improve metamodels. In this study, AREM and MREM are employed as the validation criteria. Because we are using simulation data, we can manually remove any data points with response values nearly equal to 0 and replace these points by new simulations to avoid issues of division by 0. However, in some cases, removal of data points from an existing data set could reduce predictive accuracy. Root mean square error measurement could potentially be used to avoid this issue.

CHAPTER 9

A SUPER-METAMODELING FRAMEWORK TO OPTIMIZE SYSTEM PREDICTABILITY

9.1 Overview

Chapter 5 through Chapter 8 developed four different methods that can work together to produce accurate metamodel. They are designed for specific modeling conditions. The MPEU method is designed for highly nonlinear system with small dataset. Dynamic Kriging method, however, aims to improve the accuracy and efficiency for very large dataset. Grey-box metamodeling method aims to combine multiple historical datasets. It requires the user to determine which metamodeling method would be employed through carefully analysis of the given problem. This is a time consuming process and the predictive accuracy would highly depend on the user's personal decision. If there is one method that can simplify this process and optimize the system predictability, creating a metamodel can be easier.

From previous chapters, we already know that metamodels can robustly predict manufacturing process and engineering systems design results. Various techniques, such as Kriging, PR, ANN and others, are each best suited for different scenarios that can range across a design space. Thus, methods are needed to identify the most appropriate metamodel or model composite for a given problem. To account for pros and cons of different metamodeling techniques for a wide diversity of data sets, in this chapter we introduce a super-metamodel optimization framework

(SMOF) to improve overall prediction accuracy by integrating different metamodeling techniques without a need for additional data. The SMOF defines an iterative process first to construct multiple metamodels using different methods and then aggregate them into a weighted composite and finally optimize the super-metamodel through advanced sampling. The optimized super-metamodel can reduce an overall prediction error and sustains the performance regardless of dataset variation. To verify the method, we apply it to 24 test problems representing various scenarios. A case study conducted with AM process data shows the method is effectiveness in practice.

9.2 Background

Many approaches have addressed the metamodel technique selection problem from the sampling perspective [175]. Chapter 5 also develops a selection method use domain knowledge. However, generic methods are still needed to address other factors that affect the selection of the best metamodeling technique.

Brute force search, also known as "generate-and-test", is the most general problem solving method that consists of systematically enumerating all possible candidate metamodeling techniques and selecting the most appropriate one based on a set of criteria [126]. This "generate-and-test" method is robust but lacks efficiency when the size of the candidate space is big. Modified methods were developed to improve the efficiency of exhaustive search using data-driven approaches [112, 83, 170]. Some research focuses on characterizing different metamodeling techniques to provide a metamodeling selection strategy. For example, Jin and coauthors suggest that polynomial regression should be implemented first to see if a reasonable fit can be obtained when constructing metamodels [132].

A selected technique is typically used for subsequent updates and predictions. However, data sampling can have a significant impact on the performance of a metamodeling technique [132]. Changing or adding data points could also affect the accuracy of a metamodel [148, 127]. There is no guarantee that the technique with the lowest prediction error for one region in a design space will also have the lowest error for another region in that same space [176].

Integrated metamodeling approaches compose two or more techniques to address the issue of metamodel performance variation. For example, Turner (2005) introduced the Non-Uniform Rational B-splines (NURBs) method to model the hyperdimensional design spaces of products [177]. More conventionally, a method such as Universal Kriging combines PR and the ordinary Kriging method to improve modeling accuracy [91]. Grey-box metamodeling offers another approach to combine models of different fidelity levels [83, 169]. Both universal Kriging and grey-box modeling techniques require two datasets to construct a metamodel. One is a high fidelity small dataset and the other a lower fidelity large dataset. However, it is not always possible in practice to generate additional customized data. Furthermore, only certain types of techniques can be integrated using the existing approaches. A more general method is thus needed that can combine any assortment of candidate metamodeling techniques to generate a composite that is best suited for a given problem globally. Such an approach should be broadly applicable and eliminate the need for case-by-case exploration for the best technique for every new data set.

This chapter develops a metamodel integration approach called the super-metamodel optimization framework (SMOF). SMOF integrates PR, Kriging, and ANN metamodels to improve accuracy over the individual metamodels. SMOF iteratively approaches the optimal combination of the techniques for a given dataset.

9.3 Super-Metamodel Optimization Framework (SMOF)

To account for pros and cons of different metamodeling techniques for a wide diversity of data sets, this section introduces the super-metamodel optimization framework (SMOF). Figure 9.1 shows the schematics of a SMOF model that is a composition of weighted individual models. The architecture of SMOF is typically a three nodes neural network in which the nodes are individual metamodels.

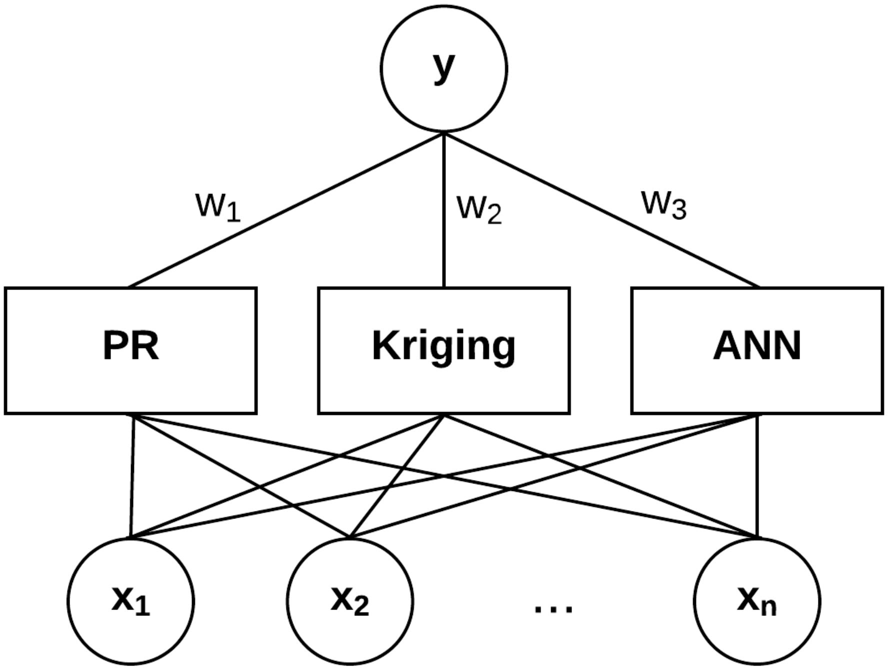


Figure 9.1: SMOF general model

The following equations provide a general formulation of the SMOF model:

$$\tilde{f}(x) = \sum_{i=1}^{3} w_i \tilde{f}_i(x) \quad \big(\sum_{i=1}^{3} w_i = 1\big) \tag{9.1}$$

where w_i is the weight factor of the i_{th} individual metamodel. Once the individual models are constructed, additional data can be used to find the optimal weight factors in the following procedure. The integrated SMOF model aims to minimize the sum of the prediction errors at these additional points, which is a function of the weight factors as expressed in the following equation.

$$Minimize : \sum_{j=1}^{n} Error^j(w_1, w_2, w_3) \tag{9.2}$$

where, $Error^j$ represents the relative prediction error of the weighted composed metamodel at $Point^j$. The absolute error of the composite model defined in Equation 9.1 can be calculated using Equation 6.1 for each data point. Equation 9.2 uniquely defines an optimization problem considering the performance of each individual metamodel at every data point.

The resulting formulation of a relative $Error^j$ thereby follows in Equation 9.3.

$$\begin{aligned}
\mathbf{Point^1}: \quad \mathbf{Error^1} \quad &= \left|\frac{w_1\tilde{y}_1^1 + w_2\tilde{y}_2^1 + w_3\tilde{y}_3^1 - y^1}{y^1}\right| \\
\mathbf{Point^2}: \quad \mathbf{Error^2} \quad &= \left|\frac{w_1\tilde{y}_1^2 + w_2\tilde{y}_2^2 + w_3\tilde{y}_3^2 - y^2}{y^2}\right| \\
\mathbf{Point^3}: \quad \mathbf{Error^3} \quad &= \left|\frac{w_1\tilde{y}_1^3 + w_2\tilde{y}_2^3 + w_3\tilde{y}_3^3 - y^3}{y^3}\right| \\
\vdots \qquad\quad \vdots \qquad\quad & \qquad\qquad \vdots \\
\mathbf{Point^j}: \quad \mathbf{Error^j} \quad &= \left|\frac{w_1\tilde{y}_1^j + w_2\tilde{y}_2^j + w_3\tilde{y}_3^j - y^j}{y^j}\right| \\
\vdots \qquad\quad \vdots \qquad\quad & \qquad\qquad \vdots \\
\mathbf{Point^n}: \quad \mathbf{Error^n} \quad &= \left|\frac{w_1\tilde{y}_1^n + w_2\tilde{y}_2^n + w_3\tilde{y}_3^n - y^n}{y^n}\right|
\end{aligned} \tag{9.3}$$

Where, y^j represents the observation value at the j_{th} point. $Error^j$ is calculated based on the weighted average of the predicted values from the individual models in Equation 9.1 and the observation value. After a transformation, the j^{th} point error becomes:

$$\left|\frac{w_1\tilde{y}_1^j + w_2\tilde{y}_2^j + w_3\tilde{y}_3^j - y^j}{y^j}\right| = \left|\frac{\tilde{y}_1^j}{y^j}w_1 + \frac{\tilde{y}_2^j}{y^j}w_2 + \frac{\tilde{y}_3^j}{y^j}w_3 - 1\right| \tag{9.4}$$

Thus, the matrix formation of Equation 9.3 is:

$$\{Error\} = \begin{bmatrix} \frac{\tilde{y}_1^1}{y^1} & \frac{\tilde{y}_2^1}{y^1} & \frac{\tilde{y}_3^1}{y^1} \\ \frac{\tilde{y}_1^2}{y^2} & \frac{\tilde{y}_2^2}{y^2} & \frac{\tilde{y}_3^2}{y^2} \\ \frac{\tilde{y}_1^3}{y^3} & \frac{\tilde{y}_2^3}{y^3} & \frac{\tilde{y}_3^3}{y^3} \\ \vdots & \vdots & \vdots \\ \frac{\tilde{y}_1^j}{y^j} & \frac{\tilde{y}_2^j}{y^j} & \frac{\tilde{y}_3^j}{y^j} \\ \vdots & \vdots & \vdots \\ \frac{\tilde{y}_1^n}{y^n} & \frac{\tilde{y}_2^n}{y^n} & \frac{\tilde{y}_3^n}{y^n} \end{bmatrix} \cdot \begin{Bmatrix} w_1 \\ \\ w_2 \\ \\ w_3 \end{Bmatrix} - \begin{Bmatrix} 1 \\ 1 \\ \\ \vdots \\ 1 \end{Bmatrix} \tag{9.5}$$

By applying linear algebra operations, a general optimization problem is formulated to minimize the sum of all prediction errors at all data points by choosing a set of weight factors. The weight factors are the independent variables to solve for in this optimization problem.

$$\begin{aligned} \textbf{Minimize}: \quad & \sum_{j=1}^{n}\sum_{i=1}^{3}\left|w_i\frac{\tilde{y}_i^j}{y^j}-1\right| \\ \textbf{Subject to}: \quad & \sum_{i=1}^{3} w_i = 1 \\ & w_i \geq 0 \end{aligned} \tag{9.6}$$

Figure 9.2 introduces a SMOF-based work flow that targets to compose multiple metamodels into a super-metamodel to minimize the total prediction error as formulated in Equation 9.6. The preceding step in Figure 9.2 of "build individual models" is based on established principles to generate and test individual metamodels using a given dataset [126]. If additional data is available, a further step follows to derive a global optimal super-metamodel by composing the individual metamodels through a weight optimization procedure.

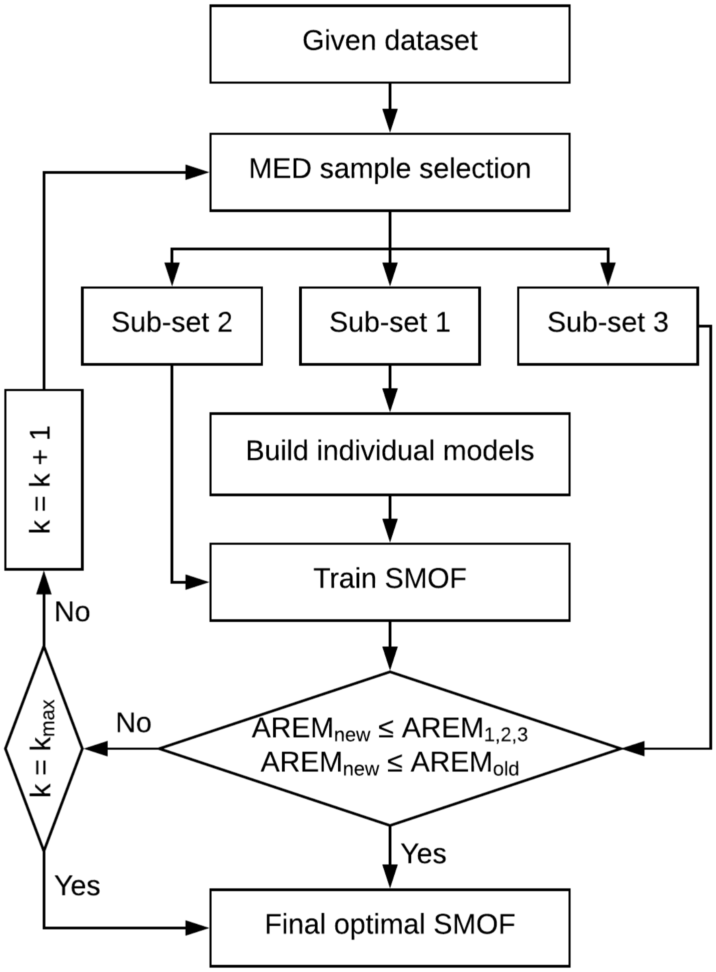


Figure 9.2: General procedure to build a SMOF metamodel

This SMOF-based approach also works when additional data is not available, by dividing the initial data set into three sub-sets: a set for metamodel construction, another set for super metamodel optimization, and the final set for metamodel validation. Since the optimal dataset for the original metamodel construction is unknown, it is necessary to iteratively segregate the data set until an acceptable AREM is achieved. As shown in Figure 9.2, the first step divides the given dataset into three sample sub-sets using the Minimum Euclidean Distance (MED) method [127]. Sub-set 1 is used to build individual metamodels. Sub-set 2 is used to find the optimal weight factors for the current super-metamodel. Sub-set 3 is used to verify whether the newly generated super-metamodel is better than the previous best super-metamodel and all the individual models.

For our case study, the percentages of the 3 sample sub-sets are set to 60%, 20% and 20%, respectively. The final optimal super-metamodel will be obtained when the error criteria are met or the total number of iterations reaches its preset maximal, k_{max}. The final super-metamodel is composed as an optimally weighted sum of the last set of individual metamodels, as shown in Equation 9.1.

It is true that a cross-validation technique could be used to verify the metamodels. However, there is no guarantee that the metamodel with the lowest error from the training set will also have the lowest test error [176]. Thus, the SMOF procedure dedicates a separate data set of sub-set 3 to verify whether or not the AREM criteria shown in Figure 9.2 are met.

9.4 Test of SMOF Effectiveness

This section presents two illustrative case studies to demonstrate the effectiveness of the SMOF method. The first case study applies SMOF to 3 different benchmark functions that have low, medium and high degrees of nonlinearity, respectively. Each function is further configured with various sample sizes and dimensionalities. The second case study tests the SMOF method using real manufacturing data. A Matlab build environment was deployed to execute the SMOF method and generate the results for these case studies. The ooDACE toolbox created the Kriging metamodels within Matlab [178].

9.4.1 Case Study 1: Benchmark Functions

To test the effectiveness of the SMOF method, 24 tests were designed to observe the hypothesized variations in recommended metamodeling techniques posed by differences in linearity, dimensionality, and sample size. Three benchmark functions with suggested different degrees of nonlinearity were deployed to generate the test datasets [179]. The Axis Parallel Hyper-Ellipsoid (APHE) function given below (continuous, convex and unimodal) generates data with a low-order of nonlinearity:

$$f(x) = \sum_{i=1}^{n}(i \cdot x_i^2) \quad (-5.12 \leq x_i \leq 5.12) \tag{9.7}$$

The Rastrigin function (with frequent and regularly distributed local minima and multimodal) below generates data of medium-order nonlinearity:

$$f(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)] \quad (-5.12 \leq x_i \leq 5.12) \tag{9.8}$$

The Ackley function (with frequent local minima and highly multimodal) below generates high-order nonlinear data:

$$\begin{aligned} f(x) = \quad & -a \cdot \exp\left(- b \cdot \sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^{n} x_i^2} \right) \\ & -32.768 \leq x_i \leq 32.768 \\ & a = 20, \;\; b = 0.2, \;\; c = 2\pi \end{aligned} \tag{9.9}$$

The dimensionality variable was grouped into three categories: small scale (number of variables = 2 or 3), medium scale (number of variables = 5), and large scale (number of variables = 8).

The LHS method [160] was used to generate the data for all of the problem configurations. To investigate the SMOF effectiveness for different sample sizes, two

Table 9.1: Design of experiment for case study 1

Problem No.	**Nonlinearity**	**Sample Size**	**Dimensionality (# of variables)**
1	Low	Small (20)	Small (n=2)
2	Low	Small (30)	Small (n=3)
3	Low	Small (50)	Medium (n=5)
4	Low	Small (80)	Large (n=8)
5	Low	Large (100)	Small (n=2)
6	Low	Large (300)	Small (n=3)
7	Low	Large (500)	Medium (n=5)
8	Low	Large (800)	Large (n=8)
9	Medium	Small (20)	Small (n=2)
10	Medium	Small (30)	Small (n=3)
11	Medium	Small (50)	Medium (n=5)
12	Medium	Small (80)	Large (n=8)
13	Medium	Large (100)	Small (n=2)
14	Medium	Large (300)	Small (n=3)
15	Medium	Large (500)	Medium (n=5)
16	Medium	Large (800)	Large (n=8)
17	High	Small (20)	Small (n=2)
18	High	Small (30)	Small (n=3)
19	High	Small (50)	Medium (n=5)
20	High	Small (80)	Large (n=8)
21	High	Large (100)	Small (n=2)
22	High	Large (300)	Small (n=3)
23	High	Large (500)	Medium (n=5)
24	High	Large (800)	Large (n=8)

sampling scenarios were considered: small datasets with 10n samples (n represents the number of variables) and large datasets with 50n samples. Table 9.1 details the experimental design of the resulting 24 trials.

Table 9.2 summarizes the results of running these trials. The iteration would continue if current AREM is larger than previous. The initial AREM was set to a very large value to enable iterations. The maximum number of iterations was set to 100 for all trails. For the more linear data generated by trials 1 through 8, the PR technique perfectly dominates the entire design space with prediction errors close to 0. Accordingly, the super-metamodel generates exactly the same AREM as the PR metamodel does, consistent with the optimal weights of [1, 0, 0]. For trials 9 to 16, the Kriging metamodel performed better than the other two techniques. Therefore, it weighs heavily on the super-metamodel and has comparable AREMs.

The highly nonlinear trials of 17 through 24 reveal more about the effects of sample size and dimensionality on the SMOF metamodel. The AREMs of the individual metamodels and the SMOF super-metamodels are shown in Figure 9.3. For all of the 8 trials, none of the individual metamodels is always superior to the other two. Further, the SMOF super-metamodel outperforms all of the three individual metamodels. Table 9.2 shows that even the least accurate individual metamodel has a nonzero weight factor in the super-metamodel composition. This indicates that every individual metamodel contributes to the super-metamodel accuracy for this type of function with higher-order nonlinearity. Thus, Table 9.2 suggests that the SMOF is most beneficial for highly nonlinear problems. These results in Table 9.2 also indicate that use of the SMOF may reveal insights about the order of linearity of a given data set.

The test result in Table 9.2 was calculated from sub-set 3, which was segregated from the original given data. In this case, additional data points can be generated to further verify the effectiveness of the SMOF method. For the following study,

Table 9.2: Test results of individual and SMOF metamodels. For each test problem, the AREM of individual models and SMOF models are presented to compare the performance. The optimal weight factors are presented to indicate the dominant individual model

No.	**PR**	**Kriging**	**ANN**	**SMOF**	**Weight**
1	0.0000	0.0204	2.0429	0.0000	[1.0000, 0.0000, 0.0000]
2	0.0000	0.0600	0.3921	0.0000	[1.0000, 0.0000, 0.0000]
3	0.0000	0.2535	0.4661	0.0000	[1.0000, 0.0000, 0.0000]
4	0.0000	0.1914	0.3188	0.0000	[1.0000, 0.0000, 0.0000]
5	0.0000	0.0001	0.0698	0.0000	[1.0000, 0.0000, 0.0000]
6	0.0000	0.0001	0.0197	0.0000	[1.0000, 0.0000, 0.0000]
7	0.0000	0.0022	0.0629	0.0000	[1.0000, 0.0000, 0.0000]
8	0.0000	0.0181	0.0634	0.0000	[1.0000, 0.0000, 0.0000]
9	1.8630	1.1284	1.8787	1.1274	[0.0000, 0.9990, 0.0010]
10	0.5051	0.2699	0.5266	0.2298	[0.2143, 0.7796, 0.0061]
11	1.1268	0.1496	0.9306	0.1483	[0.1478, 0.8511, 0.0011]
12	1.3834	0.1727	0.2746	0.1712	[0.0001, 0.9771, 0.0228]
13	0.5209	0.0082	0.2540	0.0081	[0.0001, 0.9961, 0.0038]
14	0.5545	0.3425	0.4110	0.3218	[0.0001, 0.9163, 0.0836]
15	0.3489	0.2164	0.4519	0.2105	[0.0867, 0.9115, 0.0018]
16	0.2649	0.2363	0.2523	0.2348	[0.0001, 0.7392, 0.2607]
17	0.1047	0.0714	0.1727	0.0564	[0.3402, 0.6570, 0.0028]
18	0.1251	0.1485	0.2554	0.1160	[0.5732, 0.1450, 0.2818]
19	0.0794	0.0836	0.1233	0.0255	[0.5068, 0.3726, 0.1206]
20	0.0548	0.1025	0.1657	0.0357	[0.5904, 0.2778, 0.1318]
21	0.0715	0.0943	0.1255	0.0608	[0.4986, 0.1013, 0.4001]
22	0.0459	0.0857	0.0825	0.0380	[0.5948, 0.1549, 0.2503]
23	0.0377	0.0490	0.0468	0.0287	[0.6739, 0.2075, 0.1186]
24	0.0258	0.0364	0.0436	0.0238	[0.6287, 0.3384, 0.0329]

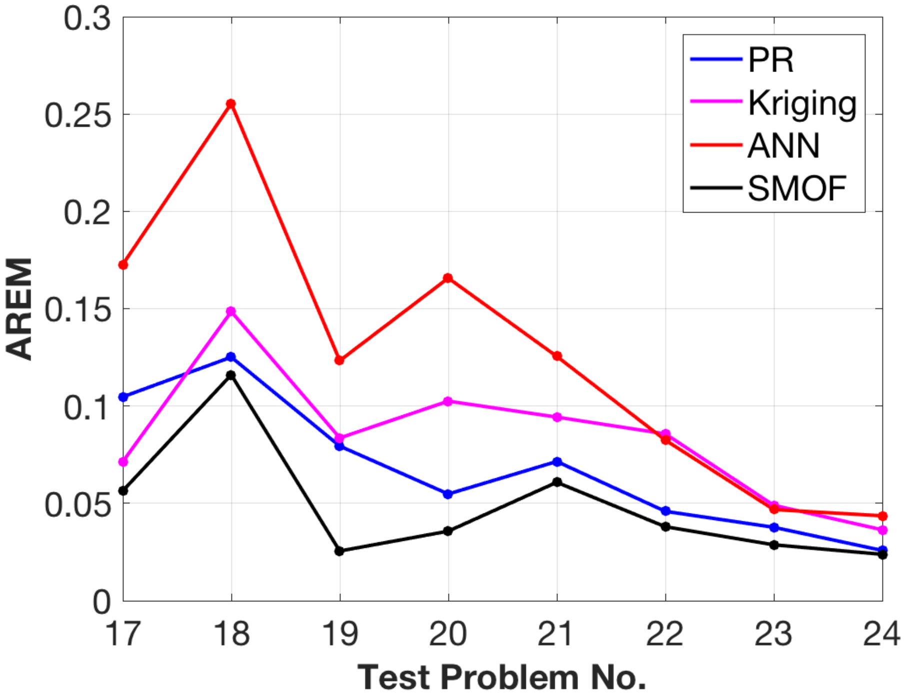


Figure 9.3: AREM of PR, Kriging, ANN, and SMOF metamodels for test problems 17 to 24 with high-order nonlinearity

additional 250n data points (n = number of independent variables) were generated using the Monte Carlo method.

The results for these trials are given in Table 9.3. For a consistent comparison, none of the existing metamodels were updated using any of this additional data. Therefore, the weight factors for all of the trials remain the same as shown in Table 9.2. The formerly developed SMOF super-metamodels still consistently outperform all the individual metamodels as shown in Figure 9.4, even though for some cases, for example in trial No. 20, the performance ranks of PR and Kriging have switched.

9.4.2 Case Study 2: A Manufacturing Application

This section illustrates a test of the SMOF effectiveness with real data from an additive manufacturing process. This example utilizes an experimental dataset of a Direct Metal Laser Re-Melting (DMLRM) additive manufacturing process [120].

Table 9.3: AREM results from additional data

No.	PR	Kriging	ANN	SMOF
1	0.0000	0.0115	13.4318	0.0000
2	0.0000	0.3016	0.9600	0.0000
3	0.0000	0.4044	0.6045	0.0000
4	0.0000	0.2553	0.3807	0.0000
5	0.0000	0.0000	0.0831	0.0000
6	0.0000	0.0000	0.0342	0.0000
7	0.0000	0.0011	0.1577	0.0000
8	0.0000	0.0147	0.2042	0.0000
9	1.8010	1.3374	1.5044	1.3359
10	0.6435	0.5986	0.6545	0.5866
11	0.4357	0.3453	0.4740	0.3439
12	2.0602	0.2498	0.3157	0.2420
13	1.4063	0.0147	0.1255	0.0146
14	0.5357	0.2516	0.3942	0.2453
15	0.3720	0.2973	0.3808	0.2859
16	0.2337	0.2197	0.2603	0.2087
17	0.1214	0.0989	0.2251	0.0896
18	0.1024	0.1396	0.2220	0.0959
19	0.0974	0.1130	0.1274	0.0709
20	0.1281	0.0904	0.1282	0.0894
21	0.1089	0.1056	0.1126	0.0950
22	0.0679	0.0643	0.0592	0.0551
23	0.0422	0.0478	0.0583	0.0367
24	0.0288	0.0392	0.0456	0.0275

DMLRM is a process variant of selective laser sintering (SLS). The experiment studied the effects of scanning speed, scanning spacing and laser pulse frequency on the

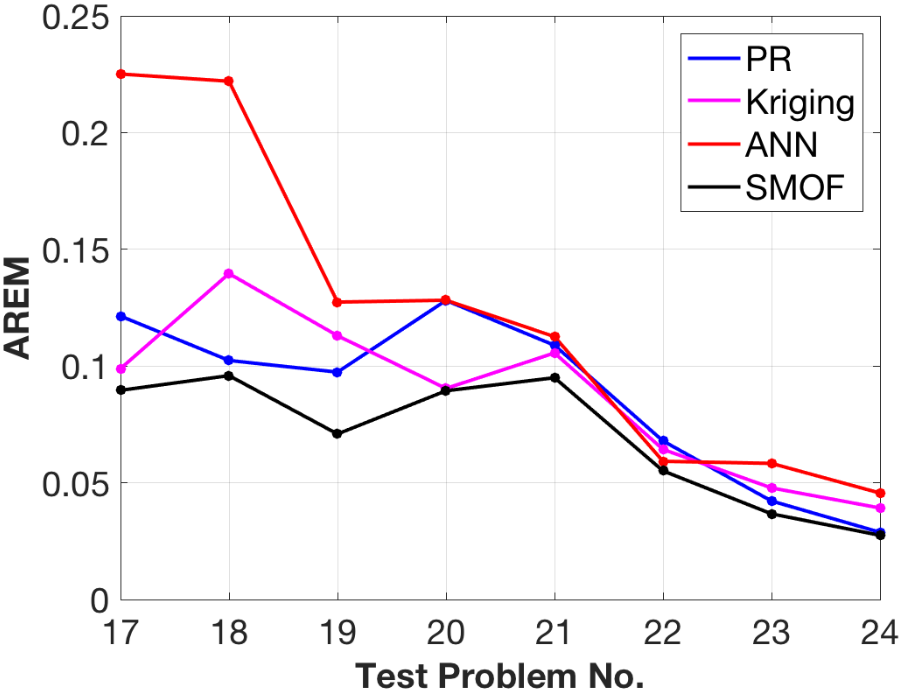


Figure 9.4: AREM of PR, Kriging, ANN, and SMOF models for test problems 17 to 24 using additional data

relative density of the parts produced by this SLS process. The experimental design was based on a fractional factorial DOE with 105 total trials for the three input variables with values sets as shown in Table 9.4. The laser power remained fixed at 80 W for all the experiments.

Table 9.4: Variable values in experiment

Variable	**Values**
Scan speed (**mms**$^{-1}$)	50, 100, 200, 300, 400, 500
Scan spacing (μ**m**)	25, 50, 75
Pulse frequency (**kHz**)	0, 10, 20, 30, 40, 50, 60

To implement SMOF, the data was first divided into 3 sub-sets using MED sampling method at every iteration, as shown in Figure 9.2. 63 data points (60%) were included in sub-set 1 to build individual metamodels. Sub-sets 2 and 3 each include

Table 9.5: AREM for individual and SMOF metamodels, and corresponding weight factors

	PR	**Kriging**	**ANN**	**SMOF**
Optimal Weight	0.151	0.571	0.277	N/A
Final AREM	9.50%	6.72%	9.42%	5.47%

21 different data points (20%). At the end of a total of 100 iterations, the weight factor vector is optimized to $w = [0.151, 0.571, 0.277]$. Table 9.5 shows the superiority of the SMOF super-metamodel compared to the three individual metamodels in this case. These results also verify that the individual metamodel with the smallest error always has the largest weight value. Based on the discussions of the prior subsection, the results in Table 9.5 also indicate that this problem likely has a high level of nonlinearity.

9.5 Discussion and Summary

The main objective of this work is to compose super-metamodels from multiple metamodeling techniques for a better global accuracy without the need to generate additional data, which can be expensive. The idea of the SMOF introduces several salient features. First, a matrix of all data points and metamodeling techniques accounts for each corresponding error value. Second, a composite weighted formulation aggregates predicted values from various metamodeling techniques. Third, iterative sampling of the data optimizes error as a function of the weights' vector to find the optimal weighted composite for a given problem.

These innovations lead to several main benefits. First, the advantages and disadvantages of different techniques for various conditions of different types of problems become irrelevant. Thus, superior predictive accuracy can be assured regardless of which technique is best for a given problem. Second, use of the SMOF provides some indication about the degree of linearity of the problem. Third, it becomes unnecessary to generate expensive additional data to overcome any inaccuracies related to technique selection. These benefits should help to address uncertainty about which metamodeling technique to use for a given problem, which should result in more consistent predictive accuracy.

All the case studies in the previous section corroborate the hypothesis that the proposed SMOF approach of an iteratively optimized and weighted composite of individual techniques can significantly and consistently improve prediction accuracy over individual techniques regardless of the sample size and the dimensionality of the data. The first case study also strongly suggests that significance of these advantages can increase with the degree of nonlinearity of a dataset. However, even for relatively linear conditions, SMOF can simultaneously verify the degree of linearity and reveal which metamodeling technique is best to use for that given dataset. Although only the PR, Kriging and ANN techniques were used in this study, the overall approach should be extendable to a mixture of other metamodeling techniques.

There are several notable limitations of the SMOF method to address. No more than 800 data points could verify the method for the data sets used in this work. Future work could examine scenarios of more complicated conditions such as large data sets of more than 10,000 data points. These larger datasets could run with various resolutions, and with various weighting schemes of the points in the design space to test more comprehensively.

The iterative process does require more computation than the established exhaustive "generate-and-test" approach [126], which is identical to the first iteration of SMOF execution without any optimization. Thus, users should be aware of this inherent tradeoff between predictive accuracy and computational cost, especially for large datasets on a case-by-case basis. Computations of the SMOF approach could be reduced by partially updating the sample sets for each iteration. The second limitation concerns the opportunity to improve the predictive accuracy of the individual metamodels. The current data sub-set segregation by the MED method runs the same sampling process at each iteration. However, it does not necessarily optimize all the metamodels. A second sampling stage of sequential infilling technique could be introduced to make improvements. Thus, a research opportunity exists for novel sampling techniques that could further improve both predictive accuracy and computational efficiency.

CHAPTER 10
METAMODELING METHODS INTEGRATION

The previous five chapters focused on the development of metamodeling methods that aim to solve various AM and SMS problems. Each developed method can execute independently under different conditions. Under complex conditions, however, these methods could potentially be integrated together to improve overall results. Functions such as modeling technique selection, sampling method, and data combination approaches could be procedurally organized and semi-automated with a user interface to further improve metamodels. The metamodel constructed in this way could fully represent the overall work flow presented in Figure 4.1. This chapter introduces the integrated metamodeling process along with a blueprint for associated tool package development.

The methods described in previous chapters provide the foundations upon which Unified Meta-Analytics (UMA) was derived. The UMA results in a MATLAB toolbox that operates the underlying algorithms that support and connect all of these functions. This results in an easy to use toolbox to build predictive metamodels under different conditions.

This chapter describes the architecture of UMA. Here, we introduce the data structures employed, commands, user interface, operating instructions, and case studies to demonstrate deployment in a MATLAB environment. The demonstration aims to show ease of use, speed and accuracy without requirements of strong programming skills or in depth understanding of metamodeling techniques.

10.1 Unified Meta-Analytics (UMA) Matlab Tool Package

10.1.1 General Work Flow

Figure 10.1 shows the overall integration for the UMA Toolbox. The data must first be properly prepared in a specific structure. The required form of this data structure is introduced in the following section.

The model selection process can be executed in different ways: 1) manual selection; 2) performance comparison; 3) domain-drive approach. Manual selection is the fastest way, but requires more understanding of the data and the model. Otherwise, performance comparison method is recommended to exhaustively compare each metamodeling technique's validation result to select the most appropriate one. The algorithm and codes of the domain-driven method is available in the tool package. However, a library of AM instances will be necessary to enable use of that technique.

The tool provides two ways to create the metamodel, the leave-one-out (LOO) and MPEU methods. If additional data is available after creation of the basic model, the process can proceed to the second stage. If no additional data is available, the current model would be saved as the final model. During this process, PR, Kriging, and ANN would serve as candidate algorithms for high fidelity data. However, Kriging is removed if the data quality is relatively low. When additional data is available, the data quality drives a choice between options of the MPEU or grey-box methods. If the initial dataset has lower quality than the additional dataset, the grey-box method would be selected. The data quality is determined according to such conditions as standard deviation of the measurement and the experimental conditions. DVCM method is employed for scenarios of highly a nonlinear system with a large amount of data. The final metamodel technique can be selected by the user.

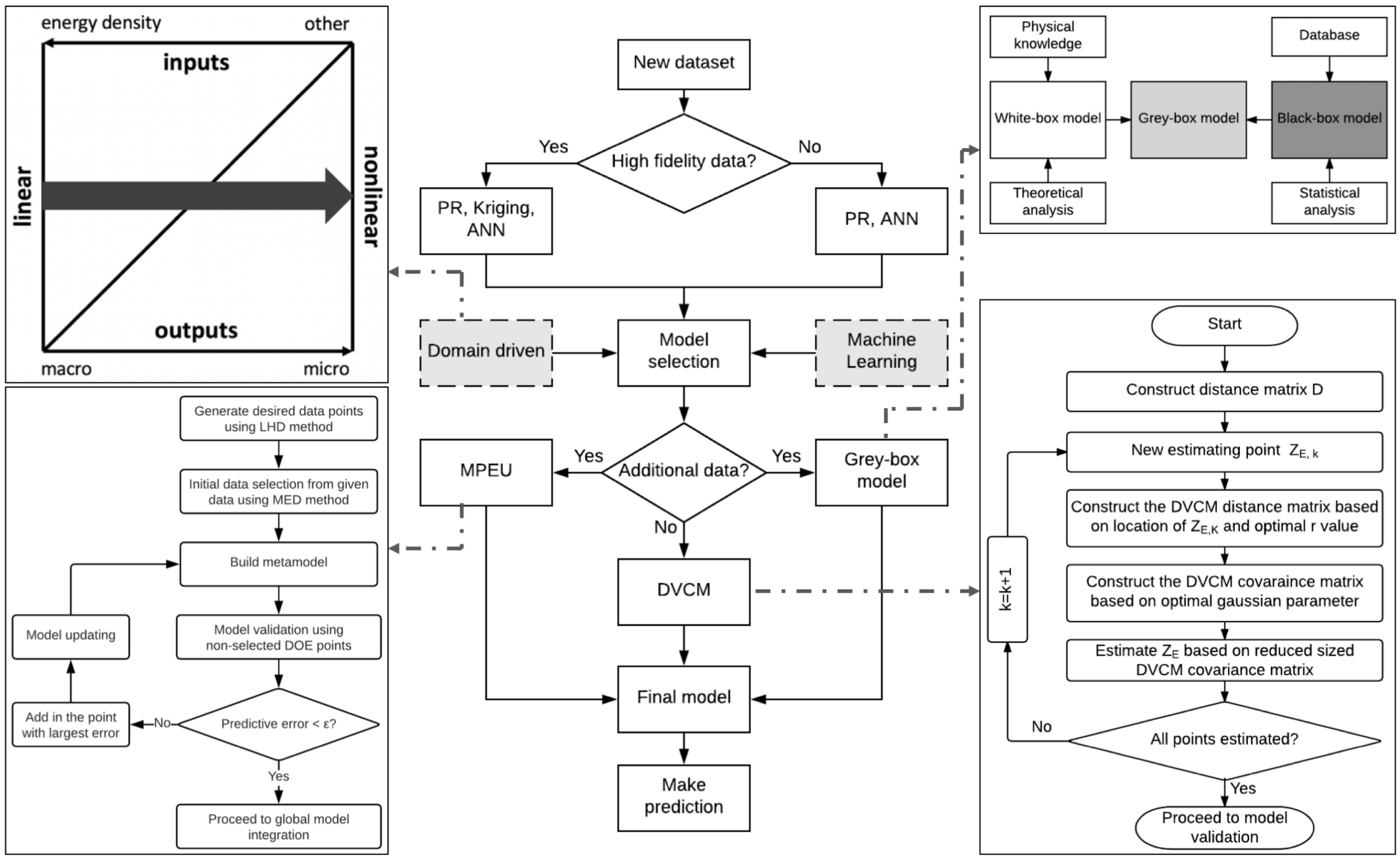

energy density
other
inputs
linear
nonlinear
outputs
macro
micro
New dataset
High fidelity data?
Yes
No
PR, Kriging, ANN
PR, ANN
Domain driven
Model selection
Machine Learning
MPEU
Additional data?
Yes
Yes
Grey-box model
No
DVCM
Final model
Make prediction
Physical knowledge
Database
White-box model
Grey-box model
Black-box model
Theoretical analysis
Statistical analysis
Generate desired data points using LHD method
Initial data selection from given data using MED method
Build metamodel
Model updating
Model validation using non-selected DOE points
Add in the point with largest error
No
Predictive error < ε?
Yes
Proceed to global model integration
Start
Construct distance matrix D
New estimating point $Z_{E,k}$
Construct the DVCM distance matrix based on location of $Z_{E,K}$ and optimal r value
k=k+1
Construct the DVCM covaraince matrix based on optimal gaussian parameter
Estimate Z_E based on reduced sized DVCM covariance matrix
No
All points estimated?
Yes
Proceed to model validation

Figure 10.1: Work flow of UMA

10.1.2 Data Structures

The data structures used throughout the UMA process must match a specific form of vectors and matrices. Here, spreadsheet (in .csv file format) columns correspond to different variables. Input variables are organized from the first column to the penultimate column. The output variable is located at the last column as shown in Table 10.1. The i represents the i^{th} input variable. The j represents the j^{th} data point. The top row has the literal name of each variable. Each execution of UMA predicts a single output.

Table 10.1: Data Structure of UMA tool package

x_1	x_2	x_3	...	x_i	y
x_1^1					y^1
x_1^2					y^2
x_1^3					y^3
$\vdots$					$\vdots$
x_1^j	x_2^j	x_3^j	x_4^j	x_i^j	y^j

10.1.3 Built-in Functions of UMA Tool Package

The program structure of the overall toolbox is composed of customized functions that can be called into other MATLAB programs. Table 10.2 lists the coded name and purpose of each function, and is followed by in depth description of each function. The functions are presented in Matlab command line.

This `dataPreProcess` function requests the input and output stored in different workspace variables. The function reads the .csv file described in the prior section and returns that data to a matrix of input variables of any dimension and a vector of the output variable.

Table 10.2: Functions for UMA

Functions Name	**Purpose**
[dataInput, dataOutput] = dataPreProcess(data)	Read given data file and return the input matrix and output vector
[trainMED, verMED] = selectMED(data, percentMED)	Divide the data into training and testing sets uses MED method with preferred ratio
[mdl_quad, quadAREM, quadNRMSE] = fit_quad(trainData, verData)	Build the PR model and return its AREM and NRMSE
[mdl_k, kAREM, kNRMSE] fit_k(trainData, verData)	Build the Kriging model and return its AREM and NRMSE
[mdl_ann, annAREM, annNRMSE] fit_ann(trainData, verData)	Build the ANN model and return its AREM and NRMSE
yp = fitData(input, type, mdl)	Make prediction uses selected model
[mdlGrey, greyAREM, greyNRMSE] greyBox(mdlInt, type, dataAdd)	Build the grey-box model and return its AREM and NRMSE by the additional dataset
yp = fitGrey(input, mdlGrey)	Make prediction uses grey-box model
[trainData, verData] = LooCrosVal(data, ith)	Return training and testing sets uses the LOO method
[mdlUpdate, updateAREM, updateNRMSE] =MPEU(dataInt, mdlInt, type, dataAdd)	Uses MPEU method to update the initial model and return updated AREM and NRMSE
[NRMSE, AREM] = modelError(outP, out)	Evaluate AREM and NRMSE

The function of selectMED executes the MED method developed in Chapter 6. It returns the training set and the testing set divided by MED method. The user defines the training-testing ratio after select the MPEU method. It is compatible with wide ranges of dimensions and data sizes.

The function `fit_quad` builds a quadratic PR metamodel. If there are not enough data points to fit quadratic parameters, the option of fitlm changes the metamodel construction technique to first order linear regression. This function returns the AREM and NRMSE of the PR model. The parameters of the PR model are saved in the first element of the returned vector. The input variables of this function are the training set and the testing set, which can be derived from the function selectMED or LooCrosVal.

The function fit_k is similar to fit_quad in regard to input variables and components returned. The difference is that this function returns a Kriging metamodel created by the ooDACE toolbox [178].

The function fit_ann builds an ANN model that uses a backpropagation method. To ensure compatibility, these codes build a single layer ANN metamodel. Backpropagation is used to optimize the ANN weight factors. Default values are: 1) total number of hidden neurons is set to 10; 2) the threshold value of convergence is set to 0.01; 3) default step length is set to $\frac{5}{B+m}$, where B is a constant equal to 1000 and m is the step number. The user can adjust these numbers for different problems; 4) each ANN model has five attempts to try different initial weight factors where only the best model is returned. The user can always adjust these parameters to approach better results for their model by modifying this fit_ann function. This function is also compatible with any dimension of data.

The UMA can predict from different models, algorithms, and input variables. However, for different conditions, this toolbox needs to call different functions and request different data structures. To simplify this process, the fitData function allows use of the same command to predict from different metamodels. When calling this function, the user can import the input variables and the name of the selected metamodel. UMA would automatically construct the required data structure and choose the correct MATLAB command. It returns the value of a predicted data point.

For example, to predict a data point x by a Kriging model named `mdl_k`, directly use the command:

```
y = fitData(x, "Kriging", mdl_k)
```

If using other models, simply change the last two input variables of `fitData` by proper names.

The function `greyBox` builds a grey-box metamodel using the method developed in Chapter 7. The user needs to type in the name of the initial model and the additional dataset. The initial model is considered a low fidelity model for this grey-box approach. The additional data is assumed to have higher fidelity.

Similar to `fitData`, the function of `fitGrey` predicts from given input variables that use the grey-box model.

The function `LooCrosVal` extracts the i_{th} data point from a given dataset. It returns the training set without the i_{th} data point and testing data that is the i_{th} data point.

The function `MPEU` uses the MPEU method introduced in Chapter 6 to sequentially update the model. To call this function, the user needs to import the initial dataset, initial model, and the additional data. It is originally developed for a problem with two datasets. However, if run with the function of `selectMED`, it can be applied to a single dataset problem by manually dividing into two datasets.

The function `modelError` is a simple way to evaluate the AREM and MREM simultaneously by one line command. The inputs for this function are the predicted value and the observed value. When combined with the function of `LooCrosVal`, it can derive the result of Leave-one-out Cross Validation (LOOCV).

10.2 Build AM Metamodel Through UMA

This section presents a pair of examples to show how to use UMA to build AM metamodels under different conditions. Here, datasets were collected from two independent experiments conducted for a laser melting PBF metal AM process [180]. Laser power (LP) and scanning speed (SS) are the input variables, and melt-pool width is the output variable of the metamodel for both data sets. Both experiments used the fractional factorial DOE method, with the laser power ranging from $100W$ to $250W$ and the scan speed ranging from 200 μms^{-1} to 1400 μms^{-1}. All 26 data points are listed in Table 10.3. The melt-pool width was measured multiple times at different locations on each scan trace. The results listed in the table are the mean values. The first experiment melted the powder directly on a bare build plate. This set of data was used to build an initial metamodel. The second experiment was conducted on a powder bed and the data generated was considered a new data set.

10.2.1 Test Scenarios

Two test scenarios were designed to test UMA: 1) build a metamodel for bare plate data; 2) build a grey-box model for the on-powder experiment using both datasets. For the first scenario, the model was created by the LOO method and the MPEU method, respectively. The second scenario aimed to predict the melt-pool width for on-powder condition. It assumed that the bare plate data has low fidelity since the experimental conditions are inconsistent to what is predicted. The powder data is treated as high fidelity data. The grey-box approach is employed to build this metamodel.

Table 10.3: Results of laser melting experiments

LP(W)	**SS(μms^{-1})**	**Meltpool width(μm)**	
		Bare build plate	**On powder**
100	200	134.57	127.77
100	400	114.75	112.43
100	800	80.52	97.98
100	600	87.58	86.64
100	1000	75.35	64.43
150	200	181.44	162.65
150	400	126.50	149.24
150	600	124.70	129.07
150	800	106.39	119.95
150	1000	103.50	101.26
150	1200	99.28	97.98
150	1400	99.40	95.95
195	200	235.94	225.16
195	400	178.07	150.01
195	600	150.52	153.05
195	800	129.57	7 151.04
195	1000	122.86	119.19
195	1200	115.38	125.60
195	1400	112.40	114.83
250	200	247.39	253.57
250	400	227.55	253.57
250	600	159.31	150.13
250	800	160.85	175.71
250	1000	141.34	141.05
250	1200	134.58	137.31
250	1400	126.69	124.42

10.2.2 Results from UMA

This section presents the modeling process, user interface, and results of using UMA for solving an AM problem. Screen shots and comments are presented to describe the modeling process.

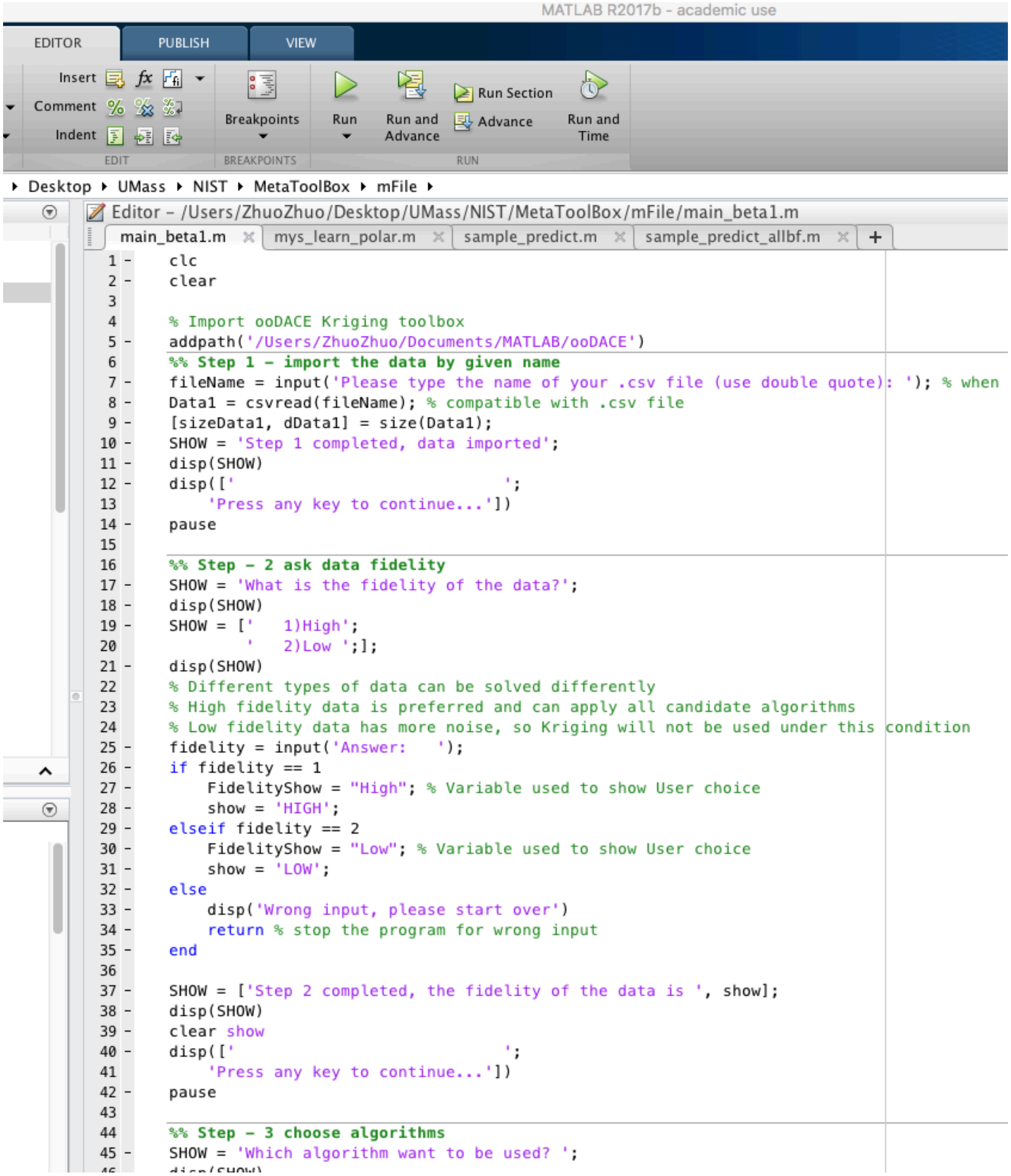
10.2.2.1 Bare Build Plate

The preliminary step is to import the UMA tool package into a MATLAB environment. The user can see the codes after calling the main function in the Editor (Figure 10.2). Note that the data files need to be placed in the same folder of the package. The user also needs to modify the codes on line 5 to specify the correct path of the ooDACE Kriging toolbox.

After the tool package is run, as shown in Figure 10.3, the command window first asks the user to provide the name of the data file. The file name needs to be typed in between the double quotes to claim the string data format. Once the data file is successfully imported, the command window asks the user to press any key to proceed to the next step (Figure 10.4).

The next step (Figure 10.5) asks the user to specify the fidelity of the given data. In this example, since the physical experiment is carefully operated and the standard deviation is less than 5%, the fidelity level of ”High” is selected. In this version of UMA, the tool uses a manual algorithm selection method since there are not yet enough AM examples in the current library to deploy the domain-driven approach. Candidate algorithms are PR, ANN, and Kriging. The user can decide which metamodeling technique to deploy after comparing the performance of each metamodel. In this example, LOO method is selected to validate the metamodel.

The tool builds the metamodel with selected algorithms and presents validation results in the command window. Figure 10.6 shows the result based on LOOCV method. Only one metamodel can be selected for future use. The decision can be

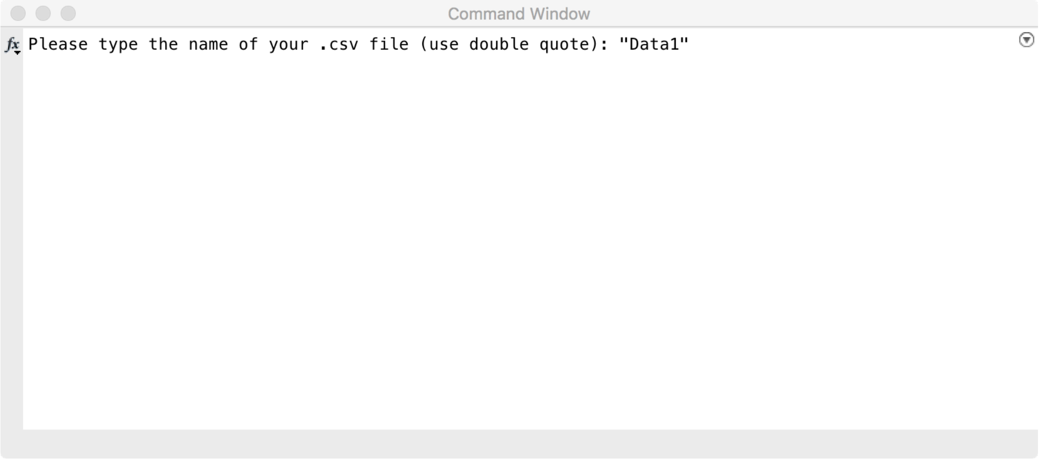


```
clc
clear

% Import ooDACE Kriging toolbox
addpath('/Users/ZhuoZhuo/Documents/MATLAB/ooDACE')
%% Step 1 - import the data by given name
fileName = input('Please type the name of your .csv file (use double quote): '); % when
Data1 = csvread(fileName); % compatible with .csv file
[sizeData1, dData1] = size(Data1);
SHOW = 'Step 1 completed, data imported';
disp(SHOW)
disp(['                             ';
    'Press any key to continue...'])
pause

%% Step - 2 ask data fidelity
SHOW = 'What is the fidelity of the data?';
disp(SHOW)
SHOW = ['   1)High';
        '   2)Low ';];
disp(SHOW)
% Different types of data can be solved differently
% High fidelity data is preferred and can apply all candidate algorithms
% Low fidelity data has more noise, so Kriging will not be used under this condition
fidelity = input('Answer:   ');
if fidelity == 1
    FidelityShow = "High"; % Variable used to show User choice
    show = 'HIGH';
elseif fidelity == 2
    FidelityShow = "Low"; % Variable used to show User choice
    show = 'LOW';
else
    disp('Wrong input, please start over')
    return % stop the program for wrong input
end

SHOW = ['Step 2 completed, the fidelity of the data is ', show];
disp(SHOW)
clear show
disp(['                             ';
    'Press any key to continue...'])
pause

%% Step - 3 choose algorithms
SHOW = 'Which algorithm want to be used? ';
```

Figure 10.2: Main function of UMA

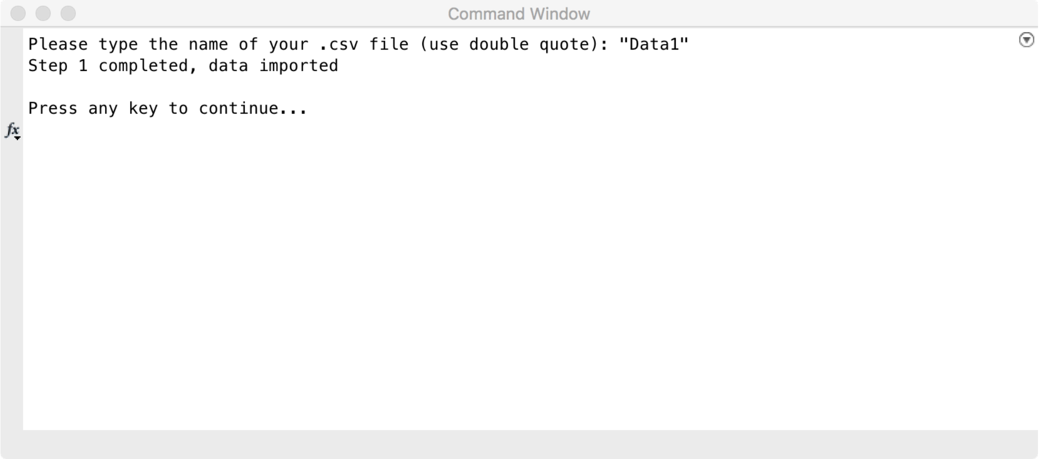
made based on the preferred AREM and NRMSE result.

The user next types in the name of the algorithm and the corresponding number of that metamodel. In this example, ”Kriging” and ”3” were entered to select the ”3-Kriging” model as it presents the lowest NRMSE and relatively lower AREM.



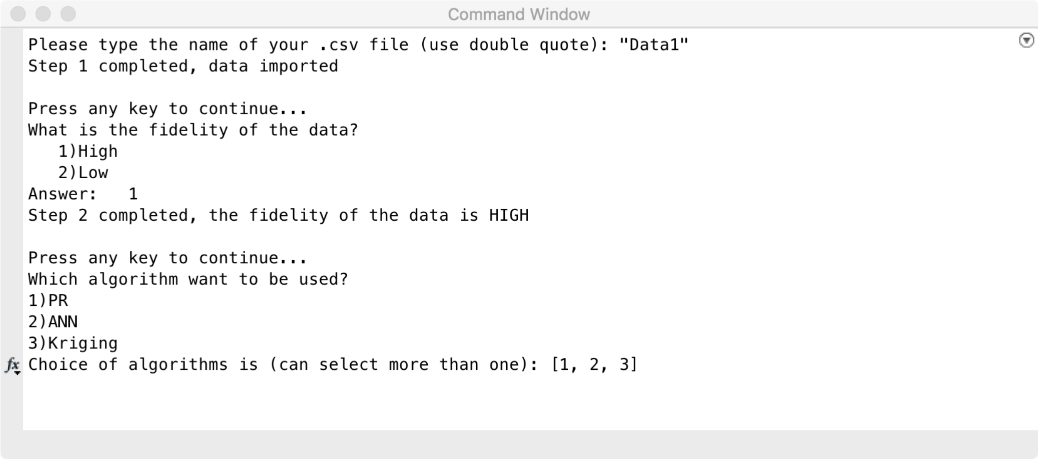
```
Please type the name of your .csv file (use double quote): "Data1"
```

Figure 10.3: Import the AM data



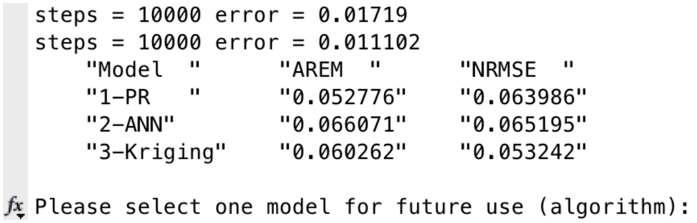
```
Please type the name of your .csv file (use double quote): "Data1"
Step 1 completed, data imported

Press any key to continue...
```

Figure 10.4: Proceed to the next step once data file is successfully imported



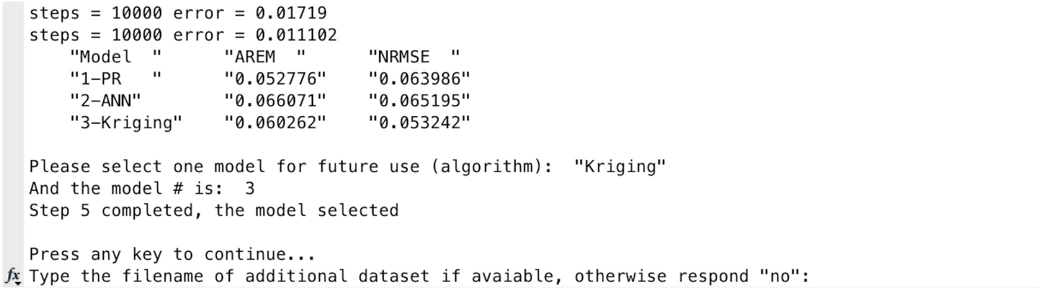
```
Please type the name of your .csv file (use double quote): "Data1"
Step 1 completed, data imported

Press any key to continue...
What is the fidelity of the data?
   1)High
   2)Low
Answer:   1
Step 2 completed, the fidelity of the data is HIGH

Press any key to continue...
Which algorithm want to be used?
1)PR
2)ANN
3)Kriging
Choice of algorithms is (can select more than one): [1, 2, 3]
```

Figure 10.5: Select data quality and the candidate algorithms



```
steps = 10000 error = 0.01719
steps = 10000 error = 0.011102
    "Model  "      "AREM  "      "NRMSE  "
    "1-PR   "      "0.052776"    "0.063986"
    "2-ANN"        "0.066071"    "0.065195"
    "3-Kriging"    "0.060262"    "0.053242"

Please select one model for future use (algorithm):
```

Figure 10.6: Results of the metamodels built by all candidate algorithms



```
steps = 10000 error = 0.01719
steps = 10000 error = 0.011102
    "Model  "      "AREM  "      "NRMSE  "
    "1-PR   "      "0.052776"    "0.063986"
    "2-ANN"        "0.066071"    "0.065195"
    "3-Kriging"    "0.060262"    "0.053242"

Please select one model for future use (algorithm):  "Kriging"
And the model # is:  3
Step 5 completed, the model selected

Press any key to continue...
Type the filename of additional dataset if avaiable, otherwise respond "no":
```

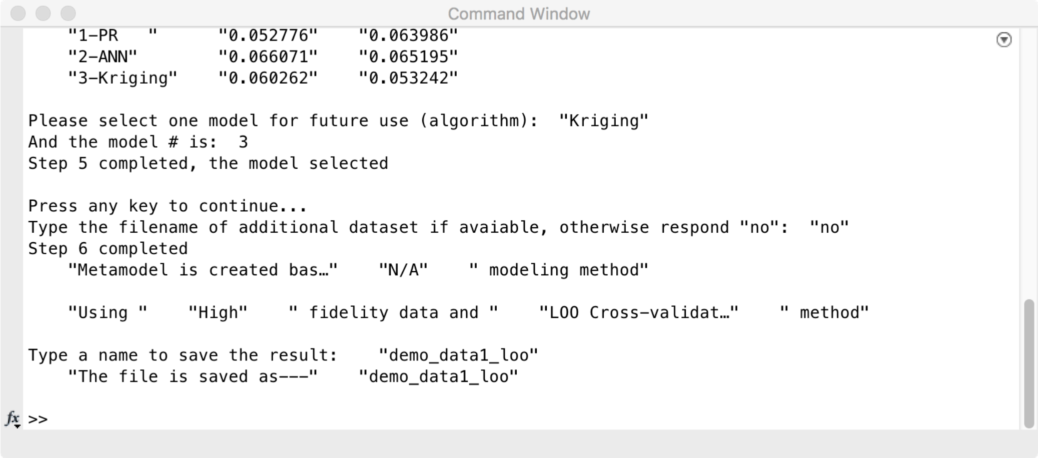
Figure 10.7: Select the model for future use

In this example, we do not have additional data to improve the current metamodel. The command window would skip the following steps and shows the result. If needed, the user can type a name to save current model into a .m file. Here, the final model is saved into a filename of `"demo_data1_loo"`. The following command line shows the data structure of the saved model.

```
mdlInfo = {fidelity, validMethod, algmOpt, methodAdd, mdlFinal,
                          finalAREM, finalNRMSE}
```

The element of `fidelity` stores a data fidelity value of high or low. Element `validMethod` stores the validation method (LOOCV or MPEU). Element `algmOpt` stores the metamodeling algorithm used. Element `methodAdd` stores the sampling method used to infill data. Element `mdlFinal` stores the model parameters. Elements `finalAREM` and `finalNRMSE` store the model performance results. After the file name is entered, MetaAnalytica displays a summary of the final model, as shown in Figure 10.8.

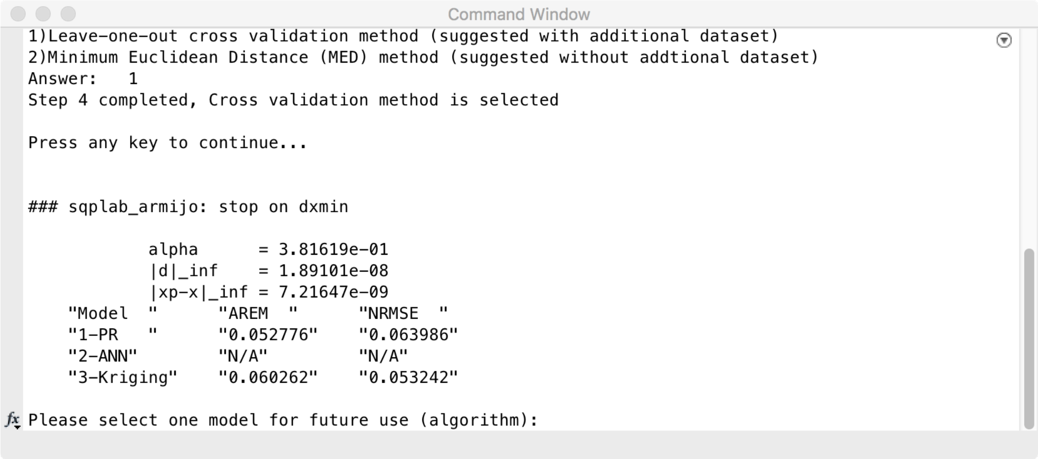
The user can omit some candidate algorithms to accelerate the modeling time.



```
    "1-PR   "      "0.052776"    "0.063986"
    "2-ANN"        "0.066071"    "0.065195"
    "3-Kriging"    "0.060262"    "0.053242"

Please select one model for future use (algorithm):  "Kriging"
And the model # is:  3
Step 5 completed, the model selected

Press any key to continue...
Type the filename of additional dataset if avaiable, otherwise respond "no":  "no"
Step 6 completed
    "Metamodel is created bas…"    "N/A"    " modeling method"

    "Using "    "High"    " fidelity data and "    "LOO Cross-validat…"    " method"

Type a name to save the result:    "demo_data1_loo"
    "The file is saved as---"    "demo_data1_loo"

fx >>
```

Figure 10.8: Result of built metamodel is presented in the command window

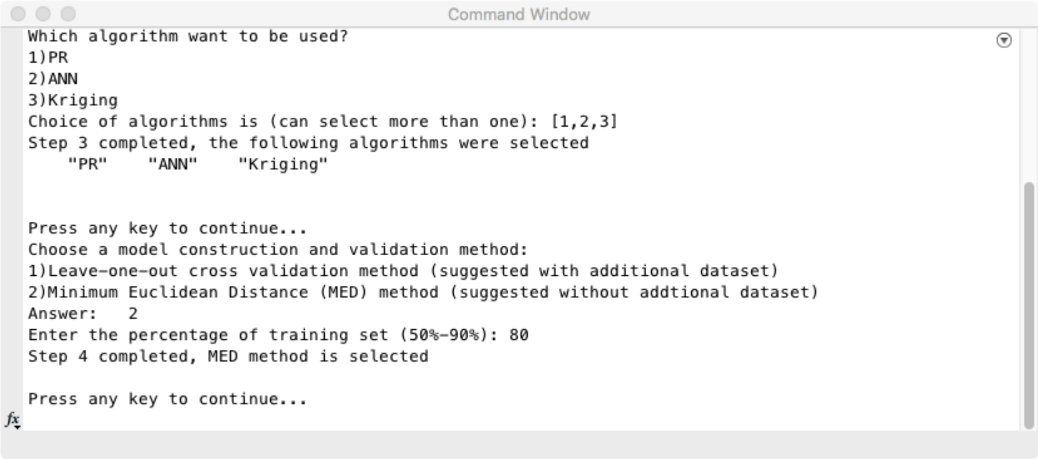
For example, as shown in Figure 10.9, the result from the ANN model is presented as ”N/A” since it was not selected in the previous step.



```
1)Leave-one-out cross validation method (suggested with additional dataset)
2)Minimum Euclidean Distance (MED) method (suggested without addtional dataset)
Answer:   1
Step 4 completed, Cross validation method is selected

Press any key to continue...


### sqplab_armijo: stop on dxmin

            alpha      = 3.81619e-01
            |d|_inf    = 1.89101e-08
            |xp-x|_inf = 7.21647e-09
    "Model  "      "AREM  "      "NRMSE  "
    "1-PR   "      "0.052776"    "0.063986"
    "2-ANN"        "N/A"         "N/A"
    "3-Kriging"    "0.060262"    "0.053242"

fx Please select one model for future use (algorithm):
```

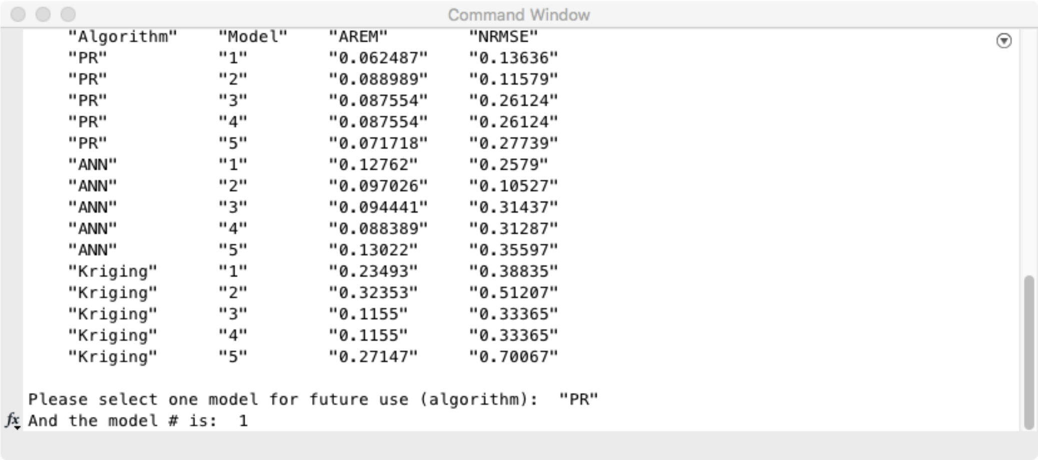
Figure 10.9: When PR and Kriging models are selected

Figure 10.10 and Figure 10.11 show the entries in a command window when the MPEU method was selected. To complete this process, the user needs to input the percentage of the training dataset. Due to the randomness of the MPEU method, MetaAnalytica would attempt several times to evaluate the results from different sample sets. A user can adjust this number in the program if more attempts are needed. The command window would ask which metamodel to select. As shown in Figure 10.11, ”PR” and the number ”1” are entered to select the first PR model.



```
Which algorithm want to be used?
1)PR
2)ANN
3)Kriging
Choice of algorithms is (can select more than one): [1,2,3]
Step 3 completed, the following algorithms were selected
    "PR"    "ANN"    "Kriging"


Press any key to continue...
Choose a model construction and validation method:
1)Leave-one-out cross validation method (suggested with additional dataset)
2)Minimum Euclidean Distance (MED) method (suggested without addtional dataset)
Answer:   2
Enter the percentage of training set (50%-90%): 80
Step 4 completed, MED method is selected

Press any key to continue...
```

Figure 10.10: When MPEU method is selected



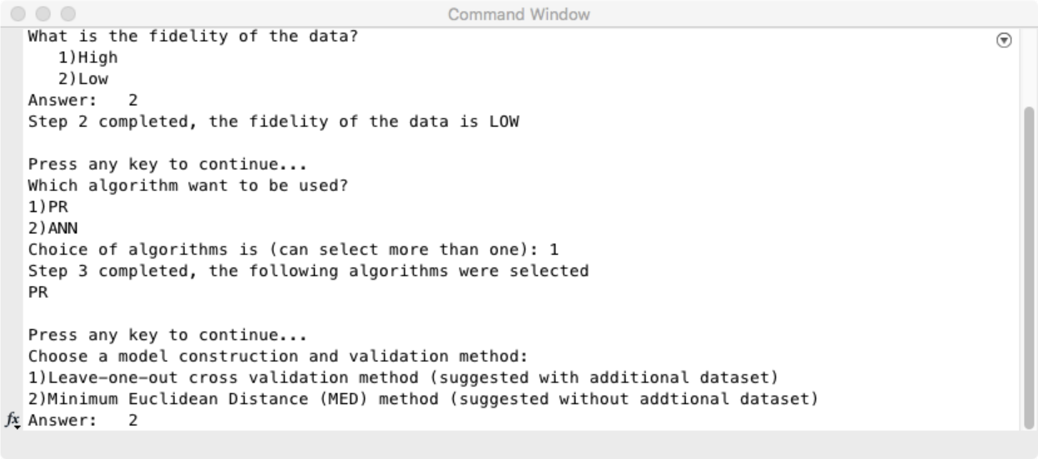
```
    "Algorithm"    "Model"    "AREM"        "NRMSE"
    "PR"           "1"        "0.062487"    "0.13636"
    "PR"           "2"        "0.088989"    "0.11579"
    "PR"           "3"        "0.087554"    "0.26124"
    "PR"           "4"        "0.087554"    "0.26124"
    "PR"           "5"        "0.071718"    "0.27739"
    "ANN"          "1"        "0.12762"     "0.2579"
    "ANN"          "2"        "0.097026"    "0.10527"
    "ANN"          "3"        "0.094441"    "0.31437"
    "ANN"          "4"        "0.088389"    "0.31287"
    "ANN"          "5"        "0.13022"     "0.35597"
    "Kriging"      "1"        "0.23493"     "0.38835"
    "Kriging"      "2"        "0.32353"     "0.51207"
    "Kriging"      "3"        "0.1155"      "0.33365"
    "Kriging"      "4"        "0.1155"      "0.33365"
    "Kriging"      "5"        "0.27147"     "0.70067"

Please select one model for future use (algorithm):  "PR"
And the model # is:  1
```

Figure 10.11: Results of MPEU method

10.2.2.2 On Powder

The second case study uses the two datasets from a bare plate and on a powder bed to build a grey-box model. Figure 10.12 and Figure 10.13 show the similar process to that of the previous case study. The dataset of bare plate experiment is imported first as low fidelity data and the dataset of powder bed experiment is imported late as high fidelity data. In this example, we select the MPEU method with 80% training data to build the initial model based on the bare plate data. PR is the only selected algorithm which produces 5 different models.

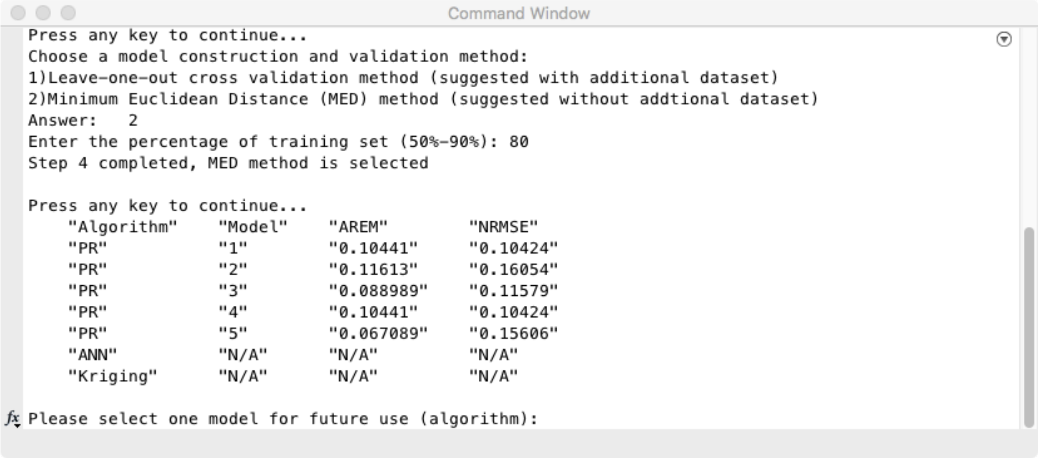
When the initial model is created, UMA asks the user to import the additional



```
What is the fidelity of the data?
   1)High
   2)Low
Answer:   2
Step 2 completed, the fidelity of the data is LOW

Press any key to continue...
Which algorithm want to be used?
1)PR
2)ANN
Choice of algorithms is (can select more than one): 1
Step 3 completed, the following algorithms were selected
PR

Press any key to continue...
Choose a model construction and validation method:
1)Leave-one-out cross validation method (suggested with additional dataset)
2)Minimum Euclidean Distance (MED) method (suggested without addtional dataset)
Answer:   2
```

Figure 10.12: Result of initial grey-box metamodel

dataset. As shown in Figure 10.14, the AREM is 0.078 and the NRMSE is 0.079 from the initial model to predict the melt pool width from the on-powder experiment. The model ”grey-box 3” can reduce the AREM to 0.073 and NRMSE to 0.045. Thus, this model is selected as the final model and can be saved.

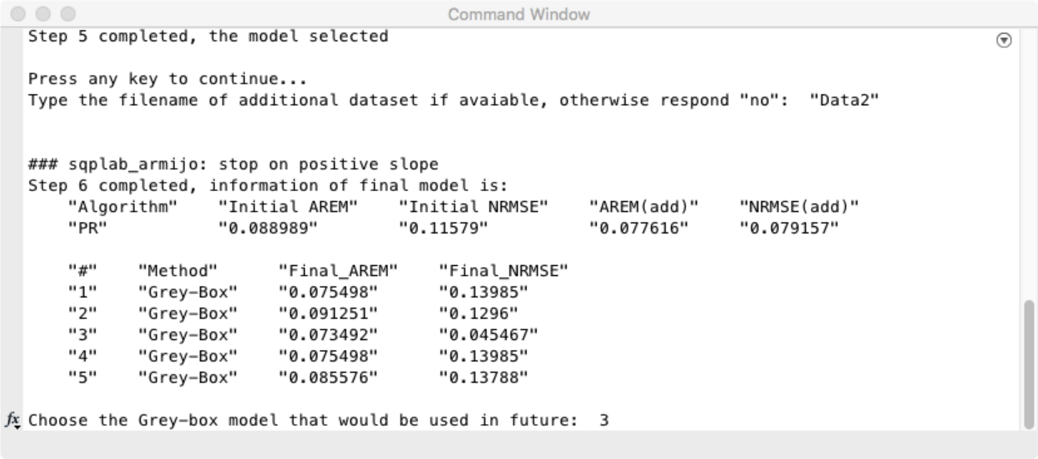


```
Press any key to continue...
Choose a model construction and validation method:
1)Leave-one-out cross validation method (suggested with additional dataset)
2)Minimum Euclidean Distance (MED) method (suggested without addtional dataset)
Answer:   2
Enter the percentage of training set (50%-90%): 80
Step 4 completed, MED method is selected

Press any key to continue...
    "Algorithm"    "Model"    "AREM"        "NRMSE"
    "PR"           "1"        "0.10441"     "0.10424"
    "PR"           "2"        "0.11613"     "0.16054"
    "PR"           "3"        "0.088989"    "0.11579"
    "PR"           "4"        "0.10441"     "0.10424"
    "PR"           "5"        "0.067089"    "0.15606"
    "ANN"          "N/A"      "N/A"         "N/A"
    "Kriging"      "N/A"      "N/A"         "N/A"

Please select one model for future use (algorithm):
```

Figure 10.13: Result of final grey-box model

10.3 Summary

This chapter introduced the UMA tool package and tested it with two AM datasets. One dataset is tested to build individual metamodel using different ap-



```
Step 5 completed, the model selected

Press any key to continue...
Type the filename of additional dataset if avaiable, otherwise respond "no":  "Data2"


### sqplab_armijo: stop on positive slope
Step 6 completed, information of final model is:
    "Algorithm"    "Initial AREM"    "Initial NRMSE"    "AREM(add)"    "NRMSE(add)"
    "PR"           "0.088989"        "0.11579"          "0.077616"     "0.079157"

    "#"    "Method"      "Final_AREM"    "Final_NRMSE"
    "1"    "Grey-Box"    "0.075498"      "0.13985"
    "2"    "Grey-Box"    "0.091251"      "0.1296"
    "3"    "Grey-Box"    "0.073492"      "0.045467"
    "4"    "Grey-Box"    "0.075498"      "0.13985"
    "5"    "Grey-Box"    "0.085576"      "0.13788"

fx Choose the Grey-box model that would be used in future:  3
```

Figure 10.14: Result of built metamodel is presented in the command window

proaches. The second dataset is used to build the grey-box metamodel. The functions of UMA work properly for different conditions. The tool allows the user provides minimum input to build the metamodels with high predictive accuracy. For different conditions, the tool provides instructions in the command window to guide the user. However, this tool was only tested on a few datasets. In the future, more case studies are request to fully test the UMA.

CHAPTER 11
CONCLUSION AND FUTURE WORK

11.1 Conclusion

The overall goal of this dissertation is to address the challenges of building metamodel in AM and SMS using historical datasets. The first challenge is that the sampling methods designed for different problems are not compatible with the new design conditions. There may be not enough samples, missing data, and too large raw dataset. This work addressed this challenge by the unique contribution of the developed metamodeling methods such as the MPEU and DVCM approaches. Salient features of these approaches include a optimal post-sampling strategy that able to improve the predictive accuracy and modeling efficiency simultaneously without adding more data.

Second, building one overall metamodel by combining multiple historical datasets is difficult since it is nearly impossible to integrate the data with different experimental conditions. This challenge is addressed by the two stages grey-box modeling method. This grey-box approach efficiently utilized both low fidelity and high fidelity datasets to build a more accurate metamodel. It also improves the collaboration between physics-based knowledge and statistics-based information. Results showed that significantly reduced the amount of samples that needed to build accurate predictive metamodel.

Third, algorithm selection is very important to the metamodel as it correlates to the sampling method and the predictive accuracy. The challenge is that the selection process can be extremely difficult in the fact of having limited information. To address this challenge, we developed the domain-driven approach and the super-metamodeling method. Domain-driven approach uses prior physics-based knowledge to accurately predict the metamodeling algorithm without investigating the statistical data. A unique feature of this approach is that the metamodel is no longer limited by the sampling method. The most appropriate metamodel can be derived before the sampling stage. The super-metamodeling method mainly addressed this challenge by systematically integrate different metamodeling algorithms to utilize the benefits of each of them. The integrated metamodel created by the super-metamodeling approach showed the highest predictive accuracy than all individual metamodels. But this was only tested on a few datasets.

The development of these methods for AM processes and SMS revealed a number of important outcomes. The domain-driven approach is able to use AM domain knowledge to guide the construction of statistics-based models. It provides a method to investigate physics-based correlations between AM parameters. Through these correlations, the metamodeling algorithm can be selected without actual data. This approach can significantly improve the predictive analytics in AM because metal AM parts and test samples are very expensive. The MPEU method provides a unique sampling method for inflexible historical datasets. It builds upon the LHS and Kriging methods to build optimal models from historical data. The grey-box modeling approach introduced a way to use less expensive high fidelity data and more inexpensive low fidelity data to improve the predictability. The low fidelity data can be easily derived from physics-based models without additional cost. This approach requests small amounts of high fidelity data from actual experiments to build the grey-box metamodel. This method provides another solution to predict more about AM processes using more AM physics-based knowledge. The DVCM approach modified the traditional Kriging method to improve its performance in the SMS domain. The prediction derived from this DVCM approach is dynamically determined by the nearby data points and optimal correlation factors. It significantly reduced the computational cost of traditional Kriging method for large data scenarios. The super-metamodeling method is a robust method compatible with different modeling conditions. It simplifies the modeling procedure by eliminating the technique selection challenge. The super-metamodel is able to automatically find the optimal weight factors for each integrated algorithm. The metamodel, thus, can utilize the benefits from all candidate algorithms and can automatically adapt to different data.

Another outcome of this dissertation is the UMA integrated process introduced in Chapter 10. An easy to use, interactive execution was demonstrated to show fast and accurate design prediction. It integrates and semi-automates the functions of methods introduced in this dissertation. It provides a user interface with instructions to enable building usable predictive metamodels. Furthermore, the functions developed for this toolbox can be extracted and used independently in other MATLAB programs.

11.2 Future Work

Future work could advance and build upon this work in several critical ways. Chapter 5 revealed the utilization of physics-based knowledge in the predictive metamodeling area. This provides a non-empirical way for the earlier stage of metamodel development. The result of the domain-driven approach can potentially guide the sampling strategy of metamodel development in later stages. This notion raises the hypotheses that prior information such as physics-based knowledge and/or historical data could possibly be used to guide future new metamodel development. It may be possible to use current knowledge and information to predict the sampling strategy such as an optimal DOE method and sample size for future metamodels. For example, with the knowledge from a metamodel in a current design space, it should be possible to estimate the conditions in an interpolated or extrapolated design space. Such meta-features as Kurtosis and nonlinearity could be used to make predictions. Future work could focus on ways to use meta-features and physics-based knowledge to guide DOEs for the same or similar problems.

In Chapter 7, the grey-box modeling approach combined the physics-based white-box model and statistics-based black-box models to make more accurate predictions. However, the prerequisite is that the dataset must have same input and output variables and under the same physical environment. This limits the use of the grey-box approach. As introduced in Chapter 2, AM processes compose multiple sub-models in different domains. This approach is not compatible with the datasets and models in different domains though they may be partially correlated. Thus, future work could investigate ways to compose metamodels in different domains. In AM, for example, the research can focus on how to compose the metamodels from a heat source domain and a laser melting domain. The key challenges are: 1) it is difficult to compose the model with a non-overlapped design space; 2) sub-models have different input variables; 3) datasets are collected from unique physical environments and different experimental conditions. The goal of this future work is to build an overall AM metamodel by multiple sub-models. This can potentially improve the sampling efficiency, predict more of the overall system and provide a better understanding of AM processes.

APPENDIX

DATA TABLE FOR DVCM CASE STUDY

Table A1: The 100 LH data points used in Section 8.4.1.2

$\mathbf{x_1}$	$\mathbf{x_2}$	Y
3.725	0.975	6.1881
4.475	4.025	9.3751
0.225	1.825	3.0593
4.325	2.975	5.0025
2.525	1.025	9.0019
0.475	3.275	7.2363
1.325	1.625	6.2250
2.625	3.525	6.5130
4.925	3.625	6.7891
4.975	3.775	10.5713
0.725	2.525	5.2057
4.575	0.425	11.2235
4.375	1.325	-2.8137
2.575	4.675	20.5493
3.52	0.925	7.7824
0.925	4.275	13.5660
2.025	3.025	-2.2909
4.875	4.975	15.0679
3.375	0.325	10.7076

4.825	2.475	6.1129
2.775	0.575	10.7148
0.575	4.925	21.3560
3.775	0.025	8.5628
4.775	3.875	10.3089
3.425	4.175	5.9007
4.175	4.425	14.5198
3.075	0.475	10.9537
1.725	2.175	3.9716
2.875	15.575	0.5819
1.225	4.125	9.2562
3.225	2.375	-5.0001
0.325	4.625	17.4403
1.075	1.125	6.1407
0.075	1.675	3.1593
0.125	0.875	5.4398
3.825	0.825	8.4012
3.625	4.475	6.1703
1.575	3.225	2.4393
4.675	3.125	0.1852
0.025	0.775	5.9774
2.375	4.375	17.4413
1.875	3.325	-0.6405
3.575	1.925	-6.5150
4.075	2.925	7.7235
0.825	2.25	4.9689
1.625	0.275	8.9005
1.675	1.425	7.1738
2.475	0.675	10.2586

3.125	1.225	4.6367
3.325	1.475	-1.1792
1.375	2.325	5.3291
2.825	0.175	9.4467
3.275	4.575	7.4369
1.425	4.825	12.9869
0.375	2.775	4.6954
2.325	2.725	-4.3577
4.725	1.875	1.0031
1.925	4.525	12.7971
2.675	2.075	-4.0744
0.625	0.625	6.7380
3.875	3.975	1.7810
1.125	2.125	5.6199
2.075	1.525	6.2456
3.175	2.275	-6.1363
3.975	0.525	11.1370
4.425	0.725	8.5014
2.975	4.774	12.3105
0.975	4.875	17.9512
0.875	3.175	7.6472
0.775	0.225	8.8486
4.125	1.375	-3.2303
1.025	1.975	5.3683
4.275	4.325	13.6661
0.525	0.075	9.9369
4.225	0.125	9.8300
1.525	1.075	7.6012
1.175	3.475	7.2616

1.475	3.675	4.2819
3.025	3.425	10.1729
2.175	2.625	-3.0054
1.775	3.075	0.1456
4.025	1.725	-5.6505
2.925	3.725	13.1237
2.275	1.275	7.5541
4.625	3.575	1.8257
1.275	1.175	6.6987
3.475	2.025	-6.4804
3.675	3.375	8.6387
1.975	3.925	4.0575
2.225	0.375	9.5921
0.675	4.725	19.0053
0.425	3.825	10.5770
2.425	2.675	-4.8536
3.925	2.425	3.1511
4.525	2.575	6.8271
0.275	1.775	3.1481
2.725	2.825	-3.4841
2.125	2.875	-3.0800
0.175	4.225	13.0291
1.825	4.075	4.9322

BIBLIOGRAPHY

[1] Serope Kalpakjian and Steven R Schmid. *Manufacturing engineering and technology.* Pearson Upper Saddle River, NJ, USA, 2014.

[2] Mario Hermann, Tobias Pentek, and Boris Otto. Design principles for industrie 4.0 scenarios. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 3928–3937. IEEE, 2016.

[3] Henning Kagermann, Johannes Helbig, Ariane Hellinger, and Wolfgang Wahlster. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group.* Forschungsunion, 2013.

[4] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & Information Systems Engineering*, 6(4):239–242, 2014.

[5] Kiwook Jung, KC Morris, Kevin W Lyons, Swee Leong, and Hyunbo Cho. Mapping strategic goals and operational performance metrics for smart manufacturing systems. *Procedia Computer Science*, 44:184–193, 2015.

[6] Yan Lu, Katherine C Morris, and Simon Frechette. Current standards landscape for smart manufacturing systems. *National Institute of Standards and Technology, NISTIR*, 8107, 2016.

[7] Zhuo Yang, Douglas Eddy, Sundar Krishnamurty, Ian Grosse, Peter Denno, Yan Lu, and Paul Witherell. Investigating grey-box modeling for predictive analytics in smart manufacturing. 2017.

[8] Yan Lu, Paul Witherell, Felipe Lopez, and Ibrahim Assouroko. Digital solutions for integrated and collaborative additive manufacturing. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V01BT02A033–V01BT02A033. American Society of Mechanical Engineers, 2016.

[9] Radhakisan Baheti and Helen Gill. Cyber-physical systems. *The impact of control technology*, 12:161–166, 2011.

[10] Edward A Lee. Cyber physical systems: Design challenges. In *Object oriented real-time distributed computing (isorc), 2008 11th ieee international symposium on*, pages 363–369. IEEE, 2008.

[11] Katsuhiko Ogata. *Discrete-time control systems*, volume 8. Prentice-Hall Englewood Cliffs, NJ, 1995.

[12] Roy R Craig Jr. A review of time-domain and frequency-domain component mode synthesis method. 1985.

[13] Karl J Åström. *Introduction to stochastic control theory*. Courier Corporation, 2012.

[14] Kang G Shin and Parameswaran Ramanathan. Real-time computing: A new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1):6–24, 1994.

[15] Craig Upson, TA Faulhaber, David Kamins, David Laidlaw, David Schlegel, Jefrey Vroom, Robert Gurwitz, and Andries Van Dam. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, 1989.

[16] Jay Lee, Edzel Lapira, Shanhu Yang, and Ann Kao. Predictive manufacturing system-trends of next-generation production systems. *IFAC Proceedings Volumes*, 46(7):150–156, 2013.

[17] Jay Lee, Edzel Lapira, Behrad Bagheri, and Hung-an Kao. Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters*, 1(1):38–41, 2013.

[18] Bruce H Krogh. Cyber physical systems: the need for new models and design paradigms. *Presentation Report*, 2008.

[19] Jay Lee, Behrad Bagheri, and Hung-An Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23, 2015.

[20] Ragunathan Raj Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-physical systems: the next computing revolution. In *Proceedings of the 47th Design Automation Conference*, pages 731–736. ACM, 2010.

[21] Kyoung-Dae Kim and Panganamala R Kumar. Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308, 2012.

[22] Paul Witherell, Shaw Feng, Timothy W Simpson, David B Saint John, Pan Michaleris, Zi-Kui Liu, Long-Qing Chen, and Rich Martukanitz. Toward metamodels for composable and reusable additive manufacturing process models. *Journal of Manufacturing Science and Engineering*, 136(6):061025, 2014.

[23] Kevin Ashton. That 'internet of things' thing. *RFiD Journal*, 22(7), 2011.

[24] Ciprian-Radu Rad, Olimpiu Hancu, Ioana-Alexandra Takacs, and Gheorghe Olteanu. Smart monitoring of potato crop: a cyber-physical system architecture model in the field of precision agriculture. *Agriculture and Agricultural Science Procedia*, 6:73–79, 2015.

[25] Eric Brown. Who needs the internet of things? *Linux. com. Retrieved*, 23, 2016.

[26] Daniel Giusto, Antonio Iera, Giacomo Morabito, and Luigi Atzori. *The internet of things: 20th Tyrrhenian workshop on digital communications.* Springer Science & Business Media, 2010.

[27] Mark Bartolomeo. Internet of things: Science fiction or business fact. *A Harvard Business Review Analytic Services Report, Tech. Rep*, 2014.

[28] Amy Nordrum. Popular internet of things forecast of 50 billion devices by 2020 is outdated. *IEEE Spectrum*, 18, 2016.

[29] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[30] Ovidiu Vermesan and Peter Friess. *Internet of things: converging technologies for smart environments and integrated ecosystems.* River Publishers, 2013.

[31] Angela H Eichelberger and Anne T McCartt. Toyota drivers' experiences with dynamic radar cruise control, pre-collision system, and lane-keeping assist. *Journal of safety research*, 56:67–73, 2016.

[32] AM Vilamovska, E Hattziandreu, R Schindler, C Van Oranje, H De Vries, and J Krapelse. Rfid application in healthcare–scoping and identifying areas for rfid deployment in healthcare delivery. *RAND Europe, February*, 2009.

[33] Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, and David Boyle. *From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence.* Academic Press, 2014.

[34] Guido Noto La Diega and Ian Walden. Contracting for the 'internet of things': Looking into the nest. 2016.

[35] Drew Hendricks. The trouble with the internet of things. *London Datastore. Greater London Authority. Retrieved*, 10, 2015.

[36] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[37] Philip Russom et al. Big data analytics. *TDWI best practices report, fourth quarter*, 19:40, 2011.

[38] Martin Hilbert and Priscila López. The world's technological capacity to store, communicate, and compute information. *science*, 332(6025):60–65, 2011.

[39] Danah Boyd and Kate Crawford. Six provocations for big data. In *A decade in internet time: Symposium on the dynamics of the internet and society*, volume 21. Oxford Internet Institute Oxford, 2011.

[40] Matthew A Waller and Stanley E Fawcett. Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management. *Journal of Business Logistics*, 34(2):77–84, 2013.

[41] Omer Tene and Jules Polonetsky. Big data for all: Privacy and user control in the age of analytics. *Nw. J. Tech. & Intell. Prop.*, 11:xxvii, 2012.

[42] Saint John Walker. Big data: A revolution that will transform how we live, work, and think, 2014.

[43] Lin Zhang, Yongliang Luo, Fei Tao, Bo Hu Li, Lei Ren, Xuesong Zhang, Hua Guo, Ying Cheng, Anrui Hu, and Yongkui Liu. Cloud manufacturing: a new manufacturing paradigm. *Enterprise Information Systems*, 8(2):167–187, 2014.

[44] Fei Tao, Lin Zhang, VC Venkatesh, Y Luo, and Ying Cheng. Cloud manufacturing: a computing and service-oriented manufacturing model. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 225(10):1969–1976, 2011.

[45] Bo-Hu Li, Lin Zhang, Shi-Long Wang, Fei Tao, JW Cao, XD Jiang, Xiao Song, and XD Chai. Cloud manufacturing: a new service-oriented networked manufacturing model. *Computer integrated manufacturing systems*, 16(1):1–7, 2010.

[46] Jane Bird. Exploring the 3d printing opportunity. *The Financial Times. Retrieved*, pages 08–30, 2012.

[47] Kaufui V Wong and Aldo Hernandez. A review of additive manufacturing. *ISRN Mechanical Engineering*, 2012, 2012.

[48] Ian Gibson, David W Rosen, Brent Stucker, et al. *Additive manufacturing technologies*, volume 238. Springer, 2010.

[49] Mohammad Vaezi, Hermann Seitz, and Shoufeng Yang. A review on 3d micro-additive manufacturing technologies. *The International Journal of Advanced Manufacturing Technology*, 67(5-8):1721–1754, 2013.

[50] Gustavo Tapia and Alaa Elwany. A review on process monitoring and control in metal-based additive manufacturing. *Journal of Manufacturing Science and Engineering*, 136(6):060801, 2014.

[51] LE Roscoe et al. Stereolithography interface specification. *America-3D Systems Inc*, page 27, 1988.

[52] AN Chatterjee, Sanjay Kumar, P Saha, PK Mishra, and A Roy Choudhury. An experimental design approach to selective laser sintering of low carbon steel. *Journal of Materials Processing Technology*, 136(1):151–157, 2003.

[53] Khuram Shahzad, Jan Deckers, Jean-Pierre Kruth, and Jef Vleugels. Additive manufacturing of alumina parts by indirect selective laser sintering and post processing. *Journal of Materials Processing Technology*, 213(9):1484–1494, 2013.

[54] Chee Kai Chua, Kah Fai Leong, and Chu Sing Lim. *Rapid prototyping: principles and applications*. World Scientific, 2010.

[55] Huy Nguyen and Michael Vai. Rapid prototyping technology. *Linc. Lab. J.*, 18:17–27, 2010.

[56] Ismail Durgun and Rukiye Ertan. Experimental investigation of fdm process for improvement of mechanical properties and production cost. *Rapid Prototyping Journal*, 20(3):228–235, 2014.

[57] GD Kim and YT Oh. A benchmark study on rapid prototyping processes and machines: quantitative comparisons of mechanical properties, accuracy, roughness, speed, and material cost. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 222(2):201–215, 2008.

[58] T Wohlers. Wohlers report 2004: Rapid prototyping, tooling and manufacturing state of the industry report. *Wohlers Ass., Oak Ridge Colorado, USA*, 2004.

[59] Ivan Vu, Lindsey Bass, Nicholas Meisel, Bruce Orler, Christopher B Williams, and David A Dillard. Characterization of mutli-material interfaces in polyjet additive manufacturing. In *Solid Freeform Fabrication Symposium*, pages 959–982, 2014.

[60] Rupinder Singh. Process capability study of polyjet printing for plastic components. *Journal of mechanical science and technology*, 25(4):1011–1015, 2011.

[61] M Sugavaneswaran and G Arumaikkannu. Modelling for randomly oriented multi material additive manufacturing component and its fabrication. *Materials & Design (1980-2015)*, 54:779–785, 2014.

[62] Stanisław Adamczak, Jerzy Bochnia, and Bożena Kaczmarska. An analysis of tensile test results to assess the innovation risk for an additive manufacturing technology. *Metrology and Measurement Systems*, 22(1):127–138, 2015.

[63] Emanuel Sachs, M Cima, P Williams, D Brancazio, and J Cornie. Three dimensional printing: rapid tooling and prototypes directly from a cad model. *Journal of engineering for industry*, 114(4):481–488, 1992.

[64] Simon Meteyer, Xin Xu, Nicolas Perry, and Yaoyao Fiona Zhao. Energy and material flow analysis of binder-jetting additive manufacturing processes. *Procedia CIRP*, 15:19–25, 2014.

[65] Christopher Bryant Williams. *Design and development of a layer-based additive manufacturing process for the realization of metal parts of designed mesostructure*. Georgia Institute of Technology, 2008.

[66] Scott M Thompson, Linkan Bian, Nima Shamsaei, and Aref Yadollahi. An overview of direct laser deposition for additive manufacturing; part i: Transport phenomena, modeling and diagnostics. *Additive Manufacturing*, 8:36–62, 2015.

[67] WE King, AT Anderson, RM Ferencz, NE Hodge, C Kamath, SA Khairallah, and AM Rubenchik. Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges. *Applied Physics Reviews*, 2(4):041304, 2015.

[68] I Yadroitsev, A Gusarov, I Yadroitsava, and I Smurov. Single track formation in selective laser melting of metal powders. *Journal of Materials Processing Technology*, 210(12):1624–1631, 2010.

[69] Nikolay K Tolochko, Yurii V Khlopkov, Sergei E Mozzharov, Michail B Ignatiev, Tahar Laoui, and Victor I Titov. Absorptance of powder materials suitable for laser sintering. *Rapid Prototyping Journal*, 6(3):155–161, 2000.

[70] AV Gusarov, Tahar Laoui, Ludo Froyen, and VI Titov. Contact thermal conductivity of a powder bed in selective laser sintering. *International Journal of Heat and Mass Transfer*, 46(6):1103–1109, 2003.

[71] M Shiomi, A Yoshidome, F Abe, and K Osakada. Finite element analysis of melting and solidifying processes in laser rapid prototyping of metallic powders. *International Journal of Machine Tools and Manufacture*, 39(2):237–252, 1999.

[72] Bin Xiao and Yuwen Zhang. Marangoni and buoyancy effects on direct metal laser sintering with a moving laser beam. *Numerical Heat Transfer, Part A: Applications*, 51(8):715–733, 2007.

[73] M Shiomi, K Osakada, K Nakamura, T Yamashita, and F Abe. Residual stress within metallic model made by selective laser melting process. *CIRP Annals-Manufacturing Technology*, 53(1):195–198, 2004.

[74] Mahesh Mani, Shaw Feng, Brandon Lane, Alkan Donmez, Shawn Moylan, and Ronnie Fesperman. *Measurement science needs for real-time control of additive manufacturing powder bed fusion processes*. US Department of Commerce, National Institute of Standards and Technology, 2015.

[75] Zhuo Yang, Jaclyn Hayes, Sundar Krishnamurty, and Ian R Grosse. 3d finite element modeling of pelvic organ prolapse. *Computer methods in biomechanics and biomedical engineering*, 19(16):1772–1784, 2016.

[76] Li Ma, Jeffrey Fong, Brandon Lane, Shawn Moylan, James Filliben, Alan Heckert, and Lyle Levine. Using design of experiments in finite element modeling to identify critical variables for laser powder bed fusion. In *International Solid Freeform Fabrication Symposium*, pages 219–228. Laboratory for Freeform Fabrication and the University of Texas Austin, TX, USA, 2015.

[77] W King, AT Anderson, RM Ferencz, NE Hodge, C Kamath, and SA Khairallah. Overview of modelling and simulation of metal powder bed fusion process at lawrence livermore national laboratory. *Materials Science and Technology*, 31(8):957–968, 2015.

[78] Tiefu Shao. *Toward a structured approach to simulation-based engineering design under uncertainty*. University of Massachusetts Amherst, 2007.

[79] GA Hazelrigg. On the role and use of mathematical models in engineering design. *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF MECHANICAL DESIGN*, 121:336–341, 1999.

[80] Jack PC Kleijnen. *Statistical tools for simulation practitioners*. Marcel Dekker, Inc., 1986.

[81] G Gary Wang and Songqing Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical design*, 129(4):370–380, 2007.

[82] Ruichen Jin, Xiaoping Du, and Wei Chen. The use of metamodeling techniques for optimization under uncertainty. *Structural and Multidisciplinary Optimization*, 25(2):99–116, 2003.

[83] Can Cui. *Building Energy Modeling: A Data-Driven Approach*. Arizona State University, 2016.

[84] George EP Box, Norman Richard Draper, et al. *Empirical model-building and response surfaces*, volume 424. Wiley New York, 1987.

[85] Stephen M Stigler. Gergonne's 1815 paper on the design and analysis of polynomial regression experiments. *Historia Mathematica*, 1(4):431–439, 1974.

[86] Noel Cressie and Gardar Johannesson. Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):209–226, 2008.

[87] JR Koehler and AB Owen. 9 computer experiments. *Handbook of statistics*, 13:261–308, 1996.

[88] Michel D Ingham, Robert D Rasmussen, Matthew B Bennett, and Alex C Moncada. Generating requirements for complex embedded systems using state analysis. *Acta Astronautica*, 58(12):648–661, 2006.

[89] Timothy W Simpson, Andrew J Booker, Dipankar Ghosh, Anthony A Giunta, Patrick N Koch, and R-J Yang. Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Structural and multidisciplinary optimization*, 27(5):302–313, 2004.

[90] Songqing Shan and G Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, 2010.

[91] Noel Cressie. *Statistics for spatial data.* John Wiley & Sons, 2015.

[92] Noel AC Cressie. Statistics for spatial data: Wiley series in probability and mathematical statistics. *Find this article online*, 1993.

[93] Jack PC Kleijnen. Kriging metamodeling in simulation: A review. *European journal of operational research*, 192(3):707–716, 2009.

[94] Ahmed A Eldeiry and Luis A Garcia. Comparison of ordinary kriging, regression kriging, and cokriging techniques to estimate soil salinity using landsat images. *Journal of Irrigation and Drainage Engineering*, 136(6):355–364, 2010.

[95] Zekai Sen. *Spatial modeling principles in earth sciences.* Springer, 2009.

[96] Isobel Clark. *Practical geostatistics*, volume 3. Applied Science Publishers London, 1979.

[97] Ricardo A Olea. *Optimum mapping techniques using regionalized variable theory.* Kansasgeological Survey, 1975.

[98] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.

[99] Jan Willem Van Groenigen. The influence of variogram parameters on optimal sampling schemes for mapping by kriging. *Geoderma*, 97(3):223–236, 2000.

[100] Søren Nymand Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard. Dace-a matlab kriging toolbox, version 2.0. Technical report, 2002.

[101] Liang Zhao, KK Choi, and Ikjin Lee. Metamodeling method using dynamic kriging for design optimization. *AIAA journal*, 49(9):2034–2046, 2011.

[102] Jay D Martin. Computational improvements to estimating kriging metamodel parameters. *Journal of Mechanical Design*, 131(8):084501, 2009.

[103] Alexander IJ Forrester and Andy J Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009.

[104] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[105] B Yegnanarayana. *Artificial neural networks.* PHI Learning Pvt. Ltd., 2009.

[106] JW Clark. Neural network modelling. *Physics in Medicine and Biology*, 36(10):1259, 1991.

[107] Felipe Lopez, Paul Witherell, and Brandon Lane. Identifying uncertainty in laser powder bed fusion models. In *Proceedings of the ASME 2016 Manufacturing Science and Engineering Conference*, 2016.

[108] Saad A Khairallah, Andrew T Anderson, Alexander Rubenchik, and Wayne 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.

[109] Ibiye Aseibichin Roberts. Investigation of residual stresses in the laser melting of metal powders in additive layer manufacturing. 2012.

[110] Douglas C Montgomery. *Design and analysis of experiments.* John Wiley & Sons, 2017.

[111] Anthony A Giunta, Steven F Wojtkiewicz, Michael S Eldred, et al. Overview of modern design of experiments methods for computational simulations. In *Proceedings of the 41st AIAA aerospace sciences meeting and exhibit, AIAA-2003-0649*, 2003.

[112] John R Rice. The algorithm selection problem. *Advances in computers*, 15:65–118, 1976.

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

[114] Haijun Gong, Khalid Rafi, Hengfeng Gu, Thomas Starr, and Brent Stucker. Analysis of defect generation in ti–6al–4v parts made using powder bed fusion additive manufacturing processes. *Additive Manufacturing*, 1:87–98, 2014.

[115] N Raghunath and Pulak M Pandey. Improving accuracy through shrinkage modelling by using taguchi method in selective laser sintering. *International journal of machine tools and manufacture*, 47(6):985–995, 2007.

[116] Y Tang, HT Loh, YS Wong, JYH Fuh, L Lu, and X Wang. Direct laser sintering of a copper-based alloy for creating three-dimensional metal parts. *Journal of Materials Processing Technology*, 140(1):368–372, 2003.

[117] Karolien Kempen, Evren Yasa, Lore Thijs, J-P Kruth, and Jan Van Humbeeck. Microstructure and mechanical properties of selective laser melted 18ni-300 steel. *Physics Procedia*, 12:255–263, 2011.

[118] Eleftherios Louvis, Peter Fox, and Christopher J Sutcliffe. Selective laser melting of aluminium components. *Journal of Materials Processing Technology*, 211(2):275–284, 2011.

[119] Jean-Pierre Kruth, X Wang, Tahar Laoui, and Ludo Froyen. Lasers and materials in selective laser sintering. *Assembly Automation*, 23(4):357–371, 2003.

[120] R Morgan, CJ Sutcliffe, and W O'neill. Density analysis of direct metal laser re-melted 316l stainless steel cubic primitives. *Journal of materials science*, 39(4):1195–1205, 2004.

[121] Joseph J Beaman, Joel W Barlow, David L Bourell, Richard H Crawford, Harris L Marcus, and Kevin P McAlea. Solid freeform fabrication: a new direction in manufacturing. *Kluwer Academic Publishers, Norwell, MA*, 2061:25–49, 1997.

[122] H Meier and Ch Haberland. Experimental studies on selective laser melting of metallic parts. *Materialwissenschaft und Werkstofftechnik*, 39(9):665–670, 2008.

[123] Joaquim Ciurana, Luis Hernandez, and Jordi 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.

[124] Zemin Wang, Kai Guan, Ming Gao, Xiangyou Li, Xiaofeng Chen, and Xiaoyan Zeng. The microstructure and mechanical properties of deposited-in718 by selective laser melting. *Journal of Alloys and Compounds*, 513:518–523, 2012.

[125] Kenneth C Mills. *Recommended values of thermophysical properties for selected commercial alloys*. Woodhead Publishing, 2002.

[126] Patrenahalli M. Narendra and Keinosuke Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, 9(C-26):917–922, 1977.

[127] Zhuo Yang, Douglas Eddy, Sundar Krishnamurty, Ian Grosse, Peter Denno, and Felipe Lopez. Investigating predictive metamodeling for additive manufacturing. In *Proceedings of the ASME 2016 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2016.

[128] Duck Bong Kim, Paul Witherell, Robert Lipman, and Shaw C Feng. Streamlining the additive manufacturing digital spectrum: A systems approach. *Additive manufacturing*, 5:20–30, 2015.

[129] Christopher B Williams, Farrokh Mistree, and David W Rosen. A functional classification framework for the conceptual design of additive manufacturing technologies. *Journal of Mechanical Design*, 133(12):121002, 2011.

[130] David W Rosen. Design for additive manufacturing: a method to explore unexplored regions of the design space. In *Eighteenth Annual Solid Freeform Fabrication Symposium*, pages 402–415, 2007.

[131] Erhard Brandl, Ulrike Heckenberger, Vitus Holzinger, and Damien Buchbinder. Additive manufactured alsi10mg samples using selective laser melting (slm): Microstructure, high cycle fatigue, and fracture behavior. *Materials & Design*, 34:159–169, 2012.

[132] Ruichen Jin, Wei Chen, and Timothy W Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and multidisciplinary optimization*, 23(1):1–13, 2001.

[133] Jack PC Kleijnen and Robert G Sargent. A methodology for fitting and validating metamodels in simulation1. *European Journal of Operational Research*, 120(1):14–29, 2000.

[134] Sriram Varadarajan, WEI CHEN∗, and Chester J Pelka. Robust concept exploration of propulsion systems with enhanced model approximation capabilities. *Engineering Optimization+ A35*, 32(3):309–334, 2000.

[135] G Tapia and AH Elwany. Prediction of porosity in slm parts using a mars statistical model and bayesian inference. In *Proceedings of the 2015 Annual International Solid Freeform Fabrication Symposium*, pages 1205–1219, 2015.

[136] Douglas S Thomas and Stanley W Gilbert. Costs and cost effectiveness of additive manufacturing. *NIST Special Publication*, 1176:12, 2014.

[137] J Ding, P Colegrove, J Mehnen, S Williams, F Wang, and P Sequeira Almeida. A computationally efficient finite element model of wire and arc additive manufacture. *The International Journal of Advanced Manufacturing Technology*, 70(1-4):227–236, 2014.

[138] John G Michopoulos, Samuel Lambrakos, and Athanasios Iliopoulos. Multiphysics challenges for controlling layered manufacturing processes targeting thermomechanical performance. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V01AT02A050–V01AT02A050. American Society of Mechanical Engineers, 2014.

[139] Maarten Van Elsen, Farid Al-Bender, and Jean-Pierre Kruth. Application of dimensional analysis to selective laser melting. *Rapid Prototyping Journal*, 14(1):15–22, 2008.

[140] Douglas C Eddy, Sundar Krishnamurty, Ian R Grosse, Jack C Wileden, and Kemper E Lewis. A predictive modelling-based material selection method for sustainable product design. *Journal of Engineering Design*, 26(10-12):365–390, 2015.

[141] Boxin Tang. Orthogonal array-based latin hypercubes. *Journal of the American statistical association*, 88(424):1392–1397, 1993.

[142] Stuart J Bates, Jonathan Sienz, and Dean S Langley. Formulation of the audze–eglais uniform latin hypercube design of experiments. *Advances in Engineering Software*, 34(8):493–506, 2003.

[143] Tiefu Shao and Sundar Krishnamurty. A preference-performance hybrid method for surrogate model updating in engineering design optimisation. *International Journal of Product Development*, 9(1-3):218–264, 2009.

[144] Brian M Adams, WJ Bohnhoff, KR Dalbey, JP Eddy, MS Eldred, DM Gay, K Haskell, Patricia D Hough, and LP Swiler. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.0 user's manual. *Sandia National Laboratories, Tech. Rep. SAND2010-2183*, 2009.

[145] MMA Khan, L Romoli, M Fiaschi, F Sarri, and G Dini. Experimental investigation on laser beam welding of martensitic stainless steels in a constrained overlap joint configuration. *Journal of Materials Processing Technology*, 210(10):1340–1353, 2010.

[146] KR Balasubramanian, N Siva Shanmugam, G Buvanashekaran, and K Sankaranarayanasamy. Numerical and experimental investigation of laser beam welding of aisi 304 stainless steel sheet. *Adv. Produc. Engineer. Manag*, 3(2):93–105, 2008.

[147] Julong Deng. Grey information space. *The Journal of Grey System*, 1(1):103–117, 1989.

[148] Tiefu Shao and Sundar Krishnamurty. A clustering-based surrogate model updating approach to simulation-based engineering design. *Journal of Mechanical Design*, 130(4):041101, 2008.

[149] Dong Zhao and Deyi Xue. A multi-surrogate approximation method for metamodeling. *Engineering with Computers*, 27(2):139–153, 2011.

[150] Olivier Andre Bauchau and James I Craig. *Structural analysis: with applications to aerospace structures*, volume 163. Springer Science & Business Media, 2009.

[151] Wim Devesse, Dieter De Baere, and Patrick Guillaume. The isotherm migration method in spherical coordinates with a moving heat source. *International Journal of Heat and Mass Transfer*, 75:726–735, 2014.

[152] Niels Rode Kristensen, Henrik Madsen, and Sten Bay Jørgensen. A method for systematic improvement of stochastic grey-box models. *Computers & chemical engineering*, 28(8):1431–1449, 2004.

[153] Torsten P Bohlin. *Practical grey-box process identification: theory and applications.* Springer Science & Business Media, 2006.

[154] Dimitris C Psichogios and Lyle H Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.

[155] Jörg Schubert, Rimvydas Simutis, Michael Dors, Ivo Havlík, and Andreas Lübbert. Hybrid modelling of yeast production processes–combination of a priori knowledge on different levels of sophistication. *Chemical engineering & technology*, 17(1):10–20, 1994.

[156] Belmiro PM Duarte and Pedro M Saraiva. Hybrid models combining mechanistic models with adaptive regression splines and local stepwise regression. *Industrial & engineering chemistry research*, 42(1):99–107, 2003.

[157] Jules Thibault, Vincent Van Breusegem, and Arlette Chéruy. On-line prediction of fermentation variables using neural networks. *Biotechnology and Bioengineering*, 36(10):1041–1048, 1990.

[158] David L Bourell, Joseph J Beaman, Ming C Leu, and David W Rosen. A brief history of additive manufacturing and the 2009 roadmap for additive manufacturing: looking back and looking ahead. *Proceedings of RapidTech*, pages 24–25, 2009.

[159] Lawrence E Murr, Sara M Gaytan, Diana A Ramirez, Edwin Martinez, Jennifer Hernandez, Krista N Amato, Patrick W Shindo, Francisco R Medina, and Ryan B Wicker. Metal fabrication by additive manufacturing using laser and electron beam melting technologies. *Journal of Materials Science & Technology*, 28(1):1–14, 2012.

[160] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[161] Zhuo Yang, Douglas Eddy, Sundar Krishnamurty, Ian Grosse, Peter Denno, Paul Witherell, and Felipe Lopez. Dynamic metamodeling for predictive analytics in advanced manufacturing. *Smart and Sustainable Manufacturing Systems*, 2(1), 2018.

[162] Loyd Baker, Paul Clemente, Bob Cohen, Larry Permenter, Byron Purves, and Pete Salmon. Foundational concepts for model driven system design. *INCOSE Model Driven System Design Interest Group*, 16:15–16, 2000.

[163] TO INCOSE. Systems engineering vision 2020 (incose-tp-2004-004-02). *INCOSE. Google Scholar*, 2007.

[164] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, 2014.

[165] Samuel H Huang, Peng Liu, Abhiram Mokasdar, and Liang Hou. Additive manufacturing and its societal impact: a literature review. *The International Journal of Advanced Manufacturing Technology*, 67(5-8):1191–1203, 2013.

[166] Benjamin Vayre, Frédéric Vignat, and François Villeneuve. Metallic additive manufacturing: state-of-the-art review and prospects. *Mechanics & Industry*, 13(2):89–96, 2012.

[167] Robert T Hays and Michael J Singer. *Simulation fidelity in training system design: Bridging the gap between reality and training.* Springer Science & Business Media, 2012.

[168] Timothy Simpson, Farrokh Mistree, John Korte, and Timothy Mauery. Comparison of response surface and kriging models for multidisciplinary design optimization. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page 4755, 1998.

[169] Zhuo Yang, Douglas Eddy, Sundar Krishnamurty, Ian Grosse, Peter Denno, Yan Lu, and Paul Witherell. Investigating grey-box modeling for predictive analytics in smart manufacturing. In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V02BT03A024–V02BT03A024. American Society of Mechanical Engineers, 2017.

[170] Zhuo Yang, Thomas Hagedorn, Douglas Eddy, Sundar Krishnamurty, Ian Grosse, Peter Denno, Yan Lu, and Paul Witherell. A domain-driven approach to metamodeling in additive manufacturing. In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V001T02A028–V001T02A028. American Society of Mechanical Engineers, 2017.

[171] Sanjoy Dasgupta, Christos H Papadimitriou, and Umesh Vazirani. *Algorithms.* McGraw-Hill, Inc., 2006.

[172] Bart GM Husslage, Gijs Rennen, Edwin R van Dam, and Dick den Hertog. Space-filling latin hypercube designs for computer experiments. *Optimization and Engineering*, 12(4):611–630, 2011.

[173] ASTM Committee F42 on Additive Manufacturing Technologies and Subcommittee F42. 91 on Terminology. *Standard Terminology for Additive Manufacturing Technologies.* ASTM International, 2012.

[174] Jason C Fox, Felipe F Lopez, Brandon M Lane, Ho Yeung, and Steven Grantham. On the requirements for model-based thermal control of melt pool geometry in laser powder bed fusion additive manufacturing. In *Proceedings of the 2016 Material Science & Technology Conference, Salt Lake City, UT*, 2016.

[175] Andrea Garbo and Brian German. Adaptive sampling with adaptive surrogate model selection for computer experiment applications. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 4430, 2017.

[176] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[177] Cameron John Turner. *HyPerModels: hyperdimensional performance models for engineering design*. PhD thesis, 2005.

[178] Ivo Couckuyt, Tom Dhaene, and Piet Demeester. oodace toolbox: a flexible object-oriented kriging implementation. *Journal of Machine Learning Research*, 15:3183–3186, 2014.

[179] Ke Tang, Xın Yáo, Ponnuthurai Nagaratnam Suganthan, Cara MacNish, Ying-Ping Chen, Chih-Ming Chen, and Zhenyu Yang. Benchmark functions for the cec'2008 special session and competition on large scale global optimization. *Nature Inspired Computation and Applications Laboratory, USTC, China*, 24, 2007.

[180] Yan Lu, Douglas Eddy, Zhuo Yang, and Sundar Krishnamurty. Self-improving additive manufacturing knowledge management. *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2018.