

TECHNOLOGY CHARACTERIZATION MODELS AND THEIR USE IN  
DESIGNING COMPLEX SYSTEMS

A Thesis

by

ROBERT REED PARKER

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

May 2011

Major Subject: Mechanical Engineering

Technology Characterization Models and their Use in Designing Complex Systems

Copyright 2011 Robert Reed Parker

TECHNOLOGY CHARACTERIZATION MODELS AND THEIR USE IN  
DESIGNING COMPLEX SYSTEMS

A Thesis

by

ROBERT REED PARKER

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Richard J. Malak Jr.
Committee Members,	Daniel McAdams
	Robert Balog
Head of Department,	Dennis O'Neal

May 2011

Major Subject: Mechanical Engineering

## ABSTRACT

Technology Characterization Models and Their Use in Designing Complex Systems.

(May 2011)

Robert Reed Parker, B.S., Iowa State University

Chair of Advisory Committee: Dr. Richard J. Malak Jr.

When systems designers are making decisions about which components or technologies to select for a design, they often use experience or intuition to select one technology over another. Additionally, developers of new technologies rarely provide more information about their inventions than discrete data points attained in testing, usually in a laboratory. This makes it difficult for system designers to select newer technologies in favor of proven ones. They lack the knowledge about these new technologies to consider them equally with existing technologies. Prior research suggests that set-based design representations can be useful for facilitating collaboration among engineers in a design project, both within and across organizational boundaries. However, existing set-based methods are limited in terms of how the sets are constructed and in terms of the representational capability of the sets. The goal of this research is to introduce and demonstrate new, more general set-based design methods that are effective for characterizing and comparing competing technologies in a utility-based decision framework. To demonstrate the new methods and compare their relative strengths and weaknesses, different technologies for a power plant condenser are compared. The capabilities of different condenser technologies are characterized in terms of sets defined over the space of common condenser attributes (cross sectional area, heat exchange effectiveness, pressure drop, etc.). It is shown that systems designers can use the resulting sets to explore the space of possible condenser designs quickly and effectively. It is expected that this technique will be a useful tool for system designers to evaluate new technologies and compare them to existing ones, while also encouraging the use of

new technologies by providing a more accurate representation of their capabilities. I compare four representational methods by measuring the solution accuracy (compared to a more comprehensive optimization procedure's solution), computation time, and scalability (how a model changes with different data sizes). My results demonstrate that a support vector domain description-based method provides the best combination of these traits for this example. When combined with recent research on reducing its computation time, this method becomes even more favorable.

## ACKNOWLEDGEMENTS

I would like to thank my committee chair and advisor, Dr. Rich Malak for his continual support and guidance throughout this research process and for his encouragement, advice, and knowledge. I would also like to thank my committee members, Dr. McAdams, and Dr. Balog, for their support of my research and for taking the time to participate in this process.

Thanks also go to my friends and colleagues at Texas A&M, specifically the great people in the design research areas under Dr. Lindsey, Dr. McAdams, and Dr. Malak. I have received great support from my fellow students in the design field including Shradda, Vimal, Anosh, Jon, Chuck, Erin, Carlos, Zohreh, and especially Edgar, who helped me with many Matlab coding nightmares.

I also wish to thank the professors at Texas A&M that taught me so many things. All the courses I took here will benefit me in the future and I appreciate their efforts to share their knowledge with me. I want to especially acknowledge Dr. Charles Culp whose efforts both inside and outside the classroom to help me in my career and research are second only to my advisor's.

The Department of Mechanical Engineering and TEES have both been instrumental in my success here by providing me with both the equipment for my research and the generous financial support. I appreciate their support of my research greatly.

Finally, thanks to my parents, in-laws, grandparents, aunts, and uncle for their encouragement, financial assistance, love, and visits to Texas to see me, and to my wife for her patience, love, support, encouragement, empathy, sacrifice, and for putting up with my complaints or monologues about my research these past 18 months.

## TABLE OF CONTENTS

	Page
ABSTRACT .....	iii
ACKNOWLEDGEMENTS .....	v
TABLE OF CONTENTS .....	vi
LIST OF FIGURES .....	viii
LIST OF TABLES .....	ix
1. INTRODUCTION .....	1
1.1 Problem background.....	1
1.2 Motivation from a systems design perspective .....	3
1.3 Motivation from an innovation perspective .....	5
1.4 Illustrative example .....	8
1.5 Abstracted models of technologies.....	11
1.6 Technology characterization models (TCMs).....	13
1.7 Summary of introduction.....	15
2. BACKGROUND ON MATHEMATICAL TOOLS USED IN TCMS .....	17
2.1 Parameterized Pareto dominance .....	17
2.2 Support vector domain description.....	18
2.3 Interpolation and Kriging .....	21
2.4 Linear regression .....	21
3. TCM DEVELOPMENT .....	23
3.1 Support vector domain description (SVDD) .....	24
3.2 Interpolation on efficient set.....	26
3.3 Parameterized Pareto dominance and SVDD .....	27
3.4 Linear regression on efficient set .....	28
4. EXAMPLE PROBLEM.....	29
4.1 System design problem.....	30
4.2 Heat exchanger technologies studied and design variables.....	34
4.3 Heat exchanger attributes .....	38

	Page
4.4 TCM generation and optimization.....	39
4.5 All-at-once optimization (AAO) .....	43
4.6 Results of example problem .....	44
4.7 Summary of example problem .....	47
5. COMPARISON STUDY RESULTS.....	48
5.1 Accuracy.....	48
5.2 Computation time .....	51
5.3 Scalability .....	51
5.4 Summary.....	52
6. CONCLUSIONS AND FUTURE WORK.....	58
REFERENCES.....	64
APPENDIX.....	69
VITA .....	82



## LIST OF FIGURES

	Page
Figure 1-1 System Design Definitions .....	2
Figure 1-2 Window Selection Problem Options.....	10
Figure 1-3 Example of Transition of Design Vars. To Attributes.....	14
Figure 1-4 Four Methods of TCM.....	15
Figure 2-1 Simplified Description of SVDD Fitting.....	20
Figure 3-1 Flowchart of TCM Generation and Optimization Process.....	23
Figure 4-1 Non-ideal Rankine Cycle.....	31
Figure 4-2 Systems Design Problem Layout.....	32
Figure 4-3 Individual Utility Functions.....	34
Figure 4-4 Heat Exchanger Technologies .....	36
Figure 5-1 Comparison Study Metrics .....	49
Figure 5-2 Distance Metric for Different Sample Sizes .....	49
Figure 5-3 Computation Time for Different Sample Sizes.....	50
Figure 5-4 Comparison Study Metrics for Sample Size 25.....	53
Figure 5-5 Comparison Study Metrics for Sample Size 50.....	53
Figure 5-6 Comparison Study Metrics for Sample Size 75.....	54
Figure 5-7 Distance Metric for Different Sample Sizes (Tech 2).....	54
Figure 5-8 Distance Metric for Different Sample Sizes (Tech 3).....	55
Figure 5-9 Computation Time for Different Sample Sizes (Tech 2).....	55
Figure 5-10 Computation Time for Different Sample Sizes (Tech 3).....	56
Figure 6-1 Future Applications of TCMs .....	58
Figure 6-2 Technology Development with TCMs.....	59
Figure 6-3 Lighting Technology Comparison .....	60
Figure 6-4 Hypothetical System Designer-Inventor Interaction.....	62

## LIST OF TABLES

	Page
Table 1-1	Definitions of Technology ..... 6
Table 3-1	Steps in Four TCM Methods..... 24
Table 4-1	Cycle Assumptions ..... 31
Table 4-2	Heat Exchanger Calculation Procedure ..... 37
Table 4-3	Design Generation Limits ..... 39
Table 4-4	Number of Support Vectors by TCM and Technology..... 40
Table 4-5	Number of Non-Dominated Points by Technology ..... 41
Table 4-6	AAO Constraints..... 44
Table 4-7	Example Problem Results ..... 46
Table 4-8	More Example Problem Results ..... 46
Table 5-1	Breakdown of Computation Time..... 50
Table A-1	Results of Optimization of Condenser Design..... 69
Table A-2	Utility Values of Optimization Results..... 72
Table A-3	Gaussian Width Parameters ( $q$ ) Used in This Research ..... 73
Table A-4	Number of Support Vectors by TCM, Technology, and Data Size ..... 73
Table A-5	100 Designs Used to Generate TCM Attribute Data ..... 74
Table A-6	Support Vectors Used for Full Data Set by Technology and TCM..... 76

# 1. INTRODUCTION

## 1.1 Problem Background

In complex engineered systems design, systems designers must select components to use in their systems. These components may have different analytical models, manufacturers and inventors. Further, systems designers may need to evaluate how these components interact with their systems in order to make decisions about which continuum of component technologies are best. In making these evaluations, designers may require models, performance data or other information from the component producers or, in the case of new products, the inventor. Therefore, the interaction between the systems designers and component inventors/creators is important for systems decision making. However, if inventors wish to keep their models and information private to protect proprietary interests, systems designers will not have access to the information they need to select components. Systems designers may also lack the domain knowledge to evaluate or derive component-level models on their own. This gap between systems designers and inventors makes complex systems design more difficult. It leaves systems designers with no direct mathematical way to compare competing component technologies to each other in a meaningful way in order to optimize the entire system. Closing this communication gap using abstracted technology models and evaluating the best way to characterize these models is the subject of this thesis. Others have looked at representations of these higher-level models, but none have examined the relative benefits and deficiencies of different representations. I will investigate four of these representations and compare their strengths and weaknesses in terms of accuracy, producing feasible solutions, and computational effort.

---

This thesis follows the style of Journal of Mechanical Design.

There are a variety of definitions for some of the key terms I use in this thesis, so I will provide my definitions for them here. In systems design, there is typically a hierarchy to the design problem. At the highest level, there is the complete *system*. This system may be made up of one or more *sub-systems*. Each *sub-system* is made up of *components*. These are the individual parts that make up the sub-system at the lowest level of the design. Components are described by *design variables*. These are characteristics of the component that can be directly controlled by designers during the design process. *Attributes*, on the other hand, are performance measures of components at the sub-system level. At the system level, the whole system's performance is measured by *system objectives*, which are the values the system designer wishes to maximize or minimize to improve system performance. By using *component-level behavioral models*, design variables are transformed into attributes. Similarly, *system-level behavioral models* transform attributes to system objective values. Figure 1-1 shows a more concrete example of these definitions in an automobile design problem.

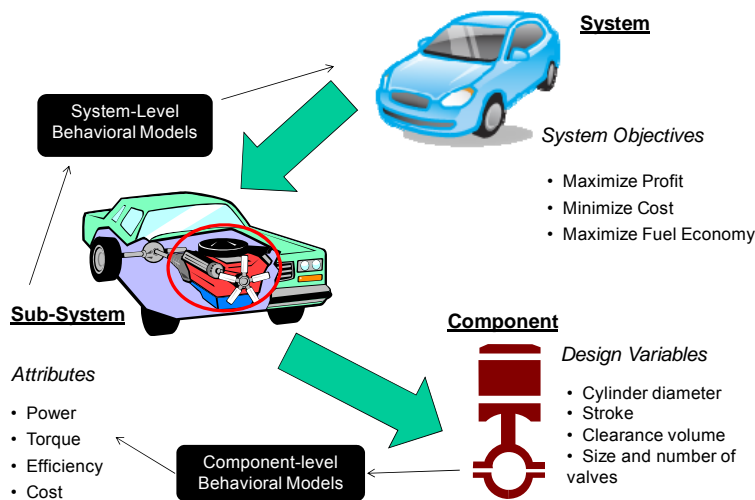


Figure 1-1 - System Design Definitions

In summary, the following list defines the problem I study in this thesis:

- Inventors do not want to give up detail information
- Systems designers lack the knowledge to create their own models of different component technologies
- Designers must make decisions about which technology to use
- Designers must be able to limit their choices in optimization (of their system objectives) to feasible designs
- Designers have limited time and resources and therefore may not be able to run separate component-level models for each technology and may not have the software or other tools to use these models

This thesis demonstrates and compares types of abstracted (attribute/sub-system level) models of component technologies that seek to address these issues.

## **1.2 Motivation from a Systems Design Perspective**

The design of a complex engineered system involves many designers with differing expertise and technical background. Some engineers may be experts in specific components or parts of the system, while others have expertise in systems integration and the system as a whole. The design team must make decisions about the entire system and they must deal with these differences in knowledge and expertise. In making these decisions, they may employ a variety of techniques. In custom-built component problems, the component designs available to the systems designer are not quantized into a table in a catalog, but rather are nearly infinite, limited only by the manufacturing processes used to create them. Further, these problems require decision-makers to have information from low-level designers to make high-level system decisions. The fact that the component can be custom-tailored to suit the systems designer's needs increases both the flexibility and complexity of the design problem.

Custom-designed component selection problems are common in the design of complex engineered systems. There are at least three main categories of components in systems design problems that fall into this type: new inventions, mass customization products, and components that are large in size, complexity or both. Inventors of new products may not have enough product performance data to publish significant catalog information for systems designers to use in decision making. Also, inventors may not yet fully understand the performance limits of their inventions. They may be in a state of manufacturing development where production processes are purely experimental and easily modified, allowing for a nearly continuous region of design possibilities. As discussed in Section 1.3, the lack of a good method for describing the capabilities of new products mathematically may hold back the adoption of a product. Consequently, inventors could benefit from using a model to describe their new product instead of a few data points from testing. They may also wish to keep the proprietary information about their invention and only provide higher-level information about the invention. This leads to a complex design problem where new representational methods are needed to capture the real designs available to the systems designer. Mass customization has been applied mainly to consumer products, but could be used to create a near-infinite array of complex engineered componentry [1]. There are many other examples of these problems in systems design, but they all share the same key trait: they involve customization of the component's design by the customer or systems designer. However, when the systems designer must choose a component to use in his or her system, what information is at his or her disposal? What form is that information in? What mathematical information is provided to the systems designer about each custom-made component so he or she can rationally choose one over another? And, how can component-level and system-level experts communicate component performance traits or metrics useful for system-level decision making? These questions will be addressed using abstracted models of technologies, as introduced in Sections 1.5 and 1.6.

### 1.3 Motivation from an Innovation Perspective

Innovation can be thought of as the process of bringing an invention from its beginnings to adoption by customers [2]. An important part of innovation, as it is defined here, is the communication of a new invention's capabilities from the inventor to its potential adopters. Additionally, Rogers, in his work on innovation, Diffusion of Innovations, lists communication as an important part of technology diffusion [3]. Thus, improvement in the methods of communicating an invention's performance to customers may reduce its innovation time, or the time it takes to reach adoption in the market. Put another way, by more richly communicating the possible performance of a product, customers may be more likely to adopt this new invention into their systems. Additionally, systems designers benefit from these richer descriptions of product performance because they allow for a potentially higher system performance by expanding the component-level performance options available to them and their system models and optimizers.

The word technology will be used throughout this thesis. Because it has a variety of definitions, I examined and compared several sources. The definitions I found are summarized in Table 1-1 below. Utilizing aspects of these definitions and keeping in mind the goal of generating a definition relating to the present research (i.e. not concerned with social or managerial aspects of technology), I composed the following definition of technology:

Technology: an artifact, process, or digital entity used to accomplish a task using specific technical processes, methods, or knowledge.

Under this definition, two different technologies are those that may accomplish the same task using different processes, methods or knowledge. Unrelated technologies are noticeably different: televisions and defibrillators, for example, but the relevant issue for this present work is the comparison of two or more similar technologies that are related

in end use and application, but not in working principles or behavior. For instance, gasoline and electric automobiles both have the same task: move people and cargo from place to place using energy stored on board. They achieve this task using very different methods and processes: liquid fuel and internal combustion versus batteries and electric motors. The engineering models needed to analyze these two technologies are very different and require separate design optimization loops to design. The present research seeks to create a way to demonstrate the performance capabilities of two or more competing technologies, such as these, that share a common task but require different engineering models at the subsystem level due to their inherent unique processes.

Table 1-1 - Definitions of Technology

Definition	Source
A design for instrumental action that reduces the uncertainty in the cause-effect relationships involved in achieving a desired outcome	Diffusion of Innovation by Everett Rogers [3]
Material Artifacts mediating task execution in the workplace	“The Duality of Technology: Rethinking the Concept of Technology in Organizations” Orlikowski, Wanda J. Organizational Science vol. 3 num 3 Aug 1992 pp 398-427 [4]
The practical application of knowledge especially in a particular area	Merriam Webster Online Dictionary [5]
A capability given by the practical application of knowledge	Merriam Webster Online Dictionary [5]
A manner of accomplishing a task especially using technical processes, methods or knowledge	Merriam Webster Online Dictionary [5]
Changing the natural world to satisfy our needs	ITEA/Gallup Poll Reveals What Americans Think About Technology: A Report of the Survey Conducted by the Gallup Organization for the International Technology Education Association. Rose, Lowell C. Dugger Jr. William E. ; The Technology Teacher, Vol. 61, 2002 [6]
An ordering of the world to make it available as a standing reserve poised for problem solving and, therefore, as the means to an end	The question of technology and other essays Heidegger, M. 1977 p. 19 Trans. W Lovett. New York Harper and Row [7]



Although decisions are currently made about whether to adopt new technologies or not, designers often use prior experience and engineering judgment with limited or discrete information only to make these decisions. This can often lead to lengthy and costly design iteration as new designs must be generated and evaluated and then thrown out for another design. Additionally, prior experience may be irrelevant or useless when evaluating new technologies because their operating principles may differ completely from existing technologies with which system designers are familiar. This also may lead to qualitative decisions based on biases from past experiences, reducing the full range of alternatives considered. Optimal or preferable designs may be completely missed by not using a formal method. Novel technologies, in particular, may be overlooked out of ignorance about their capabilities/operating principles or fear of the high risk nature of the new technology. This process is therefore wasteful and begs for a robust, useful solution.

This solution should produce a high-level model of the performance achievable by a new technology in the form of a design space of alternatives. This would be useful for top level decision makers making evaluations about low-level components that affect the performance of the system. For example, a lead architect may not feel confident evaluating or understanding the thermal properties of a vegetative roof on a building (conductivity, albedo, heat capacity, etc.), but would like to know what the range of energy savings percentages of total building energy use are possible for a vegetative roof design and the associated costs (e.g. this vegetative roof could save 20-30% of the building's energy compared to a standard roof design with cost tradeoffs, say 10 to 20 thousand dollars in extra capital costs). Additionally, this lead architect would like to be able to compare the performance of several different types of roof (vegetative, aluminum covered, asphalt, tar, etc.) against each other in the same design space. The solution to this problem would need to provide a direct method for preparing the information needed to make this multiple-technology type of decision. Also, inventors may not want to provide proprietary information to designers, so providing them with only system-

level performance feasibility spaces would be preferable and then, no unnecessary additional information is given to the designers that they do not need. This solution would also allow inventors to easily compare the performance range of their technology to existing technologies in a consistent space to determine if their technology is on the performance frontier and offers advantages over existing technologies. This solution would be most useful during conceptual design when designers are deciding which broad technology category (or product line/type) to investigate further and devote resources to that would still meet their design criteria and would offer better performance possibilities.

#### **1.4 Illustrative Example**

The following example illustrates this process. Suppose an inventor develops a new type of window. It is a photo-chromic window (changes its optical properties to react to changes in light levels). He wishes to market this window technology to architects in order to sell more windows. However, because the materials and design details of his window are proprietary and very valuable to him, he does not want to provide models of the window's behavior in terms of its design variables. His models predict the U-value, transmittance, and solar heat gain factor (SHGF) based on the window thickness, material properties, and assumed outdoor conditions. The predicted metrics above all relate to the heat transfer properties of the window and therefore, the energy lost or gained through the window. The inventor's model takes his design variables (window thickness, material properties, etc.) and predicts higher-level performance metrics (U-value, transmittance, SHGF).

In lieu of sending *this* model to an architect (to prevent divulging trade secrets or because the architect may not want to deal with these detailed models), the inventor could create a more abstracted model of the performance of his window technology. To do this, he could generate a number of window designs, run his model on these designs, take the set of performance metrics output by the model for each design and fit some

other abstracted model around it. This “meta-model” of the performance metric data derived from the original window designs could then be passed to an architect for systems design optimization.

Now suppose an architect is selecting a window technology to use in a building that minimizes energy consumption. He has two choices: the inventor’s photo-chromic window or an electro-chromic window (changes its optical properties to react to an applied voltage). I will assume that both technologies are completely customizable (size, shape, thickness, how it changes properties, etc.). Suppose he is interested in only some properties of the windows that affect the energy consumption namely, U-value, transmittance, and solar heat gain factor (SHGF). He must decide which process he should use to choose a technology and a specific design such that it optimizes his building design. He could handle this problem in at least two distinct ways:

- Option 1: Use or create low-level models of each technology that determine U-value, transmittance and SHGF (the window’s *attributes*) from design variables like window thickness and material properties (e.g. the inventor’s own model)
  - Use these to run energy simulations using an optimizer to find the best design for each technology by varying design variables
- Option 2: Use higher level performance metric (attribute) models from the inventor of each technology (models from data for U-value, transmittance, and SHGF)
  - Use this attribute model to generate system-level variables of interest (those needed for energy simulation)
  - Run energy simulations using this model and optimize to find the best attributes and best technology
  - Return to inventor and provide him with desired attributes from optimization so he can use his component-to-attribute model to determine design variables needed to achieve desired attributes

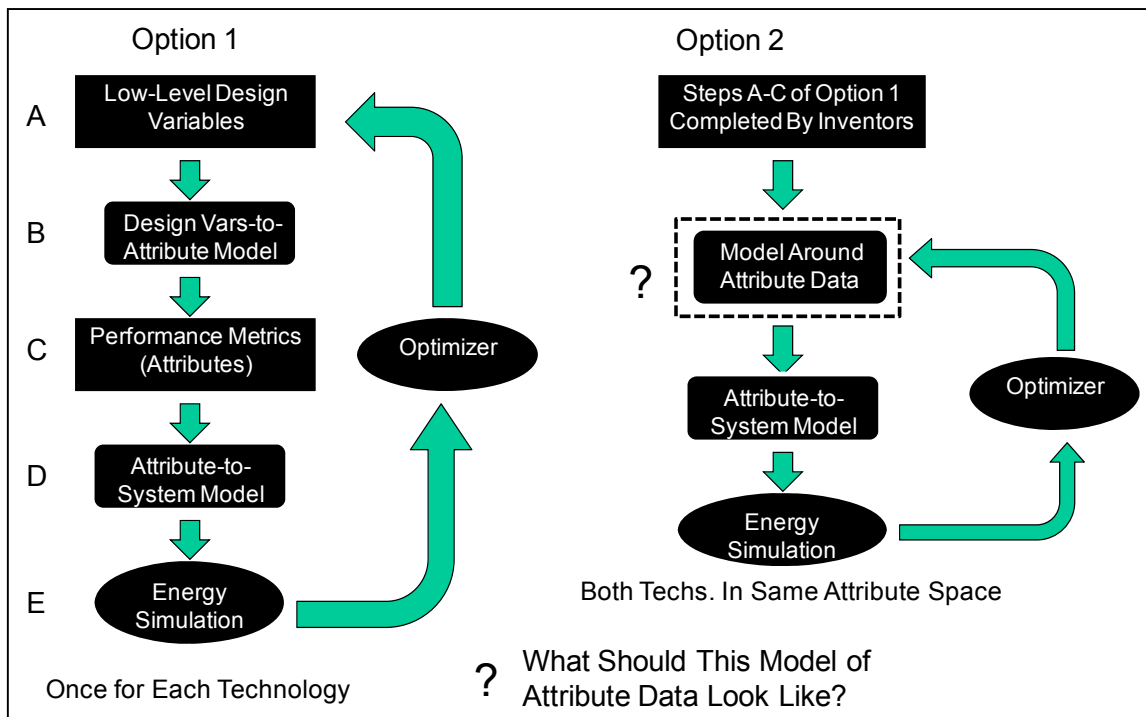


Figure 1-2 - Window Selection Problem Options

Figure 1-2 shows a summary of these two options. It should be noted that option 1 is not possible if, as I mentioned, the inventor does not share his model because system-level designers rarely have the domain knowledge to construct such models on their own. There are also computational drawbacks to option 1. Option 2 only needs to be run once because only one model is needed that handles both technologies. Option 1 must be run two times, once for each technology because each technology has different design variables and models. The attribute-to-system model is the same in both cases because both technologies relate to the system the same way at the attribute level. With more complex systems and more technology options, this advantage of option 2 becomes more significant. Also, option 1's design variables-to-attribute model may be complex and computationally intensive, slowing the optimization. The window inventor's role in this process is that of information provider. He or she would have to provide the low-level models (if the architect could not generate them himself) for option 1 or the attribute "meta-model" for option 2. This attribute "meta-model" should be constructed in such a

way that the attributes represented in the model correspond to designs that could actually be produced (feasible designs). Thus, it should indirectly contain feasibility information to constrain the optimization to attribute values (and their corresponding designs) that are actually attainable. In order to use option 2, the inventor would need to characterize the “model around attribute data”. This is a model of the abstracted capabilities (attributes) of the technologies and is the subject of the next sub-section.

### **1.5 Abstracted Models of Technologies**

Prior research has shown that abstracted models have value in decision-making. Ferguson *et al.* demonstrate the use of what they call “technical feasibility models” to map between the performance and design spaces and determine new automobile designs for a given set of performance specifications [8]. The technical feasibility models are based on solutions on the Pareto frontier in the attribute (performance) space. In this research, I utilize their ideas about using Pareto frontiers of attributes to constrain to feasible designs (the Pareto set is a subset of the feasible set) and their description of the process of taking attributes on the Pareto frontier and mapping them back into the design space. They do not, however, discuss how to model the Pareto frontier mathematically. Gurnani *et al.* continue this work and show how Pareto frontier models can be used as constraints in feasibility assessments [9]. They also add a simple quadratic regression model of the Pareto frontier to make it continuous. I use a similar regression model of the Pareto frontier in this thesis. However, they do not explore other ways of modeling the Pareto frontier or ways to deal with attributes that the designer does not yet have a clear preference for (e.g. want larger or smaller values).

Mattson and Messac explain how what they call “s-Pareto” frontiers can be used to perform concept selection in the performance space and they later add uncertainty and a visualization of the “goodness” of concepts to their method [10, 11]. Their s-Pareto frontiers are developed by finding the global Pareto frontier for multiple design concepts (instead of one, as in Ferguson and Gurnani above). Design concepts not along the s-

Pareto frontier are dominated and excluded from the decision-making process. I also use Pareto frontiers (though not s-Pareto frontiers) in this thesis to compare competing concepts (technologies) at the attribute level. They do not generate a model of the s-Pareto frontier, however.

Malak and Paredis show how abstracted models could be developed using a technique called “parameterized Pareto dominance” (an extension of Pareto dominance to include attributes for which a designer does not yet know his or her preference) and outline a general methodology for generating these abstracted, parameterized Pareto set models by composing representations together, including a method dealing with uncertainty [12-14]. They use parameterized Pareto dominance to develop “tradeoff models,” which model the Pareto frontier in the attribute space. Once again, the Pareto frontier is only a subset of the entire feasible set, so they are only modeling this portion of the feasible set. I use an interpolation model of the Pareto frontier (the “tradeoff model”) in this thesis just as they do in their research. Further, Sobek, Ward, and Liker demonstrate the usefulness of set-based design methods in systems design by describing how Toyota passes design feasibility information in sets (in the form of intervals), instead of discrete points [15]. They argue that the additional flexibility of sets of performance targets as opposed to single points reduces design cycle time and makes it easier for Toyota to communicate with suppliers. They only demonstrate feasible sets described by simple intervals on design variables or attributes, not more complex mathematical representations of the feasible domain that can be easily applied to optimization problems.

Representations of abstracted models vary and different representations may have unique benefits over others. I am not aware of any study that seeks to determine which representational method comes closest to the ideal of providing an accurate solution to the design problem while being computationally (and temporally) efficient. Additionally, I know of no study that examines how these methods scale with the

amount of available attribute data. The present work seeks to fill this gap in the research and determine if there is a superior method for dealing with custom-built-component systems design problems (at different data sizes) and assigns a name to all such methods for clarity and brevity. I compare four mathematical representational methods and find that one method stands out for its combination of accuracy and computation time.

### **1.6 Technology Characterization Models (TCMs)**

In this research I use the term Technology Characterization Model (TCM) to refer to a mathematical representation of the capabilities of a given technology (or product) in the technology's abstracted, attribute space. The abstraction and attribute parts of this definition are important because they allow the systems designer to focus only on those variables that relate component performance to system performance and ignore lower-level, complex, domain-specific variables or models that may be proprietary anyway. Abstraction of lower-level variables to attributes also potentially allows systems designers to compare competing technologies that may have different component models and low-level design variables in the same attribute space. Thus, the TCMs of the competing technologies would all be defined in the same space, allowing for easy comparison and analysis. The idea behind abstraction can be seen in Figure 1-3. Low-level design variables such as part dimensions are passed through a component model that calculates component metrics like torque, power, efficiency, internal stresses, etc. and these metrics of the component become its attributes. Thus, it is said that the component's design variables are *abstracted* to the attribute level.

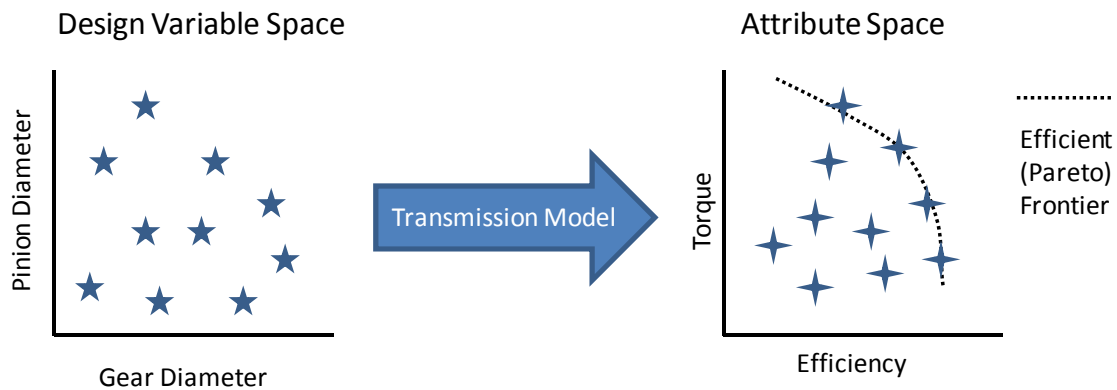


Figure 1-3 - Example of Transition of Design Vars. To Attributes

An example of a simple TCM is a mathematical model of a Pareto or efficient frontier for a set of design alternatives. In this case, the TCM represents the trade-offs in one design attribute to achieve better performance in another for those design alternatives that lie on this frontier. As I will show later, Pareto dominance analysis (the elimination of those alternatives not on the frontier and the formation of the frontier itself) plays an important role in two of the TCM methods I studied.

Four methods to characterize a technology's attributes (TCMs) will be discussed at length in this thesis: Feasible Set, Efficient Set Interpolation, Efficient Set Regression Model, and Feasible Set on Efficient Set (hereafter also referred to as: SVDD, Interpolation, Regression, and PPS+SVDD, respectively). A visual representation and comparison of typical examples of each of these methods is displayed in Figure 1-4. SVDD is a model of the entire feasible set, of which the Pareto set is a subset, PPS+SVDD is a model of the Pareto frontier only that does not rely on a predictive model, Interpolation is a model of the Pareto frontier that passes through each point on the frontier and is predictive between points, and Regression is a model of the Pareto frontier that does not necessarily pass through the points on the frontier and is predictive.



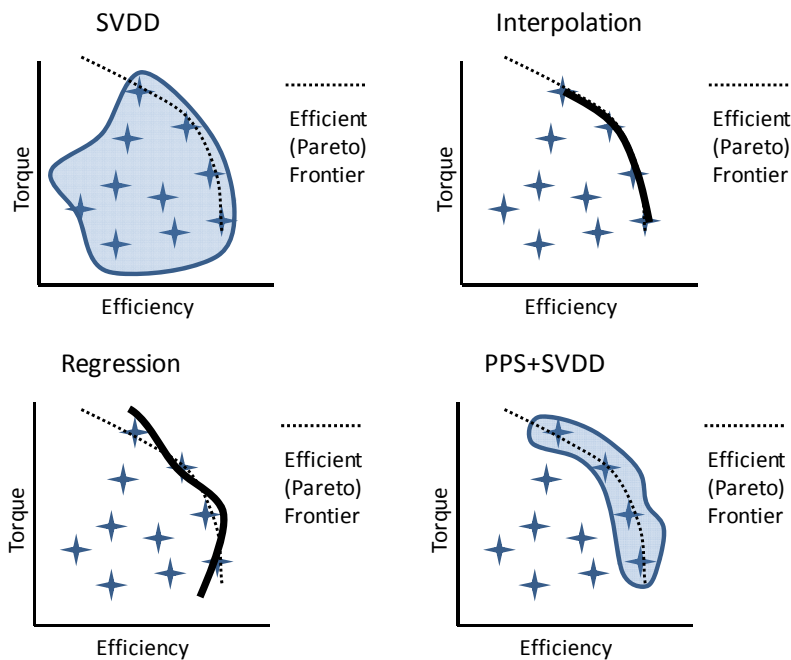


Figure 1-4 - Four Methods of TCM

## 1.7 Summary of Introduction

The preceding sub-sections show that TCMs can be useful in systems design and in assisting inventors in the innovation process. They utilize the benefits of abstraction to reduce the problem complexity, remove proprietary information and allow for different technologies to be aggregated into a single model. The remainder of this thesis is focused on answering some key questions related to TCMs:

1. What is the foundation of each of the four TCM methods?
2. How is each method used?
3. How can they be applied to systems design problems?
4. Are attribute solutions from optimization of a TCM feasible? (Do feasible design variables corresponding to these attributes exist?)
5. Which TCM type is the most accurate when its solution is compared to a trusted, well-defined method's solution? **RQ**

6. Which TCM type requires the least computation time? **RQ**
7. How do all four types scale with the size of the attribute data? **RQ**
8. Which TCM is the best overall? **RQ**

Questions 5, 6, 7, and 8 constitute my research questions for this thesis. The previous questions merely provide a background and support for answering the research questions. Question 1 is answered in Section 2, where the mathematical techniques used to derive the TCM methods are described in detail. Question 2 is answered in Section 3, where the process for producing each type of TCM is broken down to show how they can be practically used. Question 3's answer is the subject of Section 4, which details a systems design example problem where TCMs are used to find an optimal component design. In this case, a steam power plant designer is seeking a condenser technology to use in a Rankine cycle and uses each of the TCM methods to select the best technology and condenser design. This section also answers question 4 by displaying the results of the example problem in terms of design variables, showing that the optimal attribute values found from the TCM optimization can be feasibly achieved and there exists a set of design variables that can reach those attribute values within a reasonable amount of error. The research questions (5-8) are answered in Section 5, which describes the results of a comparison study on the condenser example problem. I run the example problem multiple times, varying the attribute data size (this attribute data is generated by low-level models and is used to construct the TCMs) to determine the scalability of the methods. This section also describes the accuracy and computation time of each method for the condenser problem. By combining all of the above information, I then make a conclusion about which TCM is best. Section 6 then describes my conclusions about TCMs and future work related to TCMs in the areas of technology comparison, technology development, innovation, and set-based design.

## 2. BACKGROUND ON MATHEMATICAL TOOLS USED IN TCMs

The four methods of TCM I consider involve many different modeling techniques and methods. They utilize parameterized Pareto dominance, support vector domain description, interpolation, and linear regression. These techniques have all been developed in prior work, to be detailed below.

### 2.1 Parameterized Pareto Dominance

Parameterized Pareto dominance is the elimination of designs from a set of designs such that the eliminated (or dominated) designs would never be preferred over the remaining (non-dominated) designs based on the preferences of the designer including considerations of design variables for which the designer does not yet know his or her preferences (termed “parameters”) [12, 16]. The complete set of non-dominated designs is called the parameterized Pareto set. This dominance criterion is an extension of classical Pareto dominance. Preference variables must have a preferred direction of improvement: smaller mass is preferred, smaller cost is preferred, higher efficiency is preferred, etc. Parameters are those variables or attributes, for which a designer does not currently have enough information to determine a preference. The inclusion of parameters in dominance analysis is important because systems designers often encounter variables related to components that they may not know enough about to have preferences for. In other words, some systems design decisions cannot be made early in the process and preferences may be unknown. When one’s view is at the system level, preferences for lower-level variables may be difficult to determine. The mathematical formulation of parameterized Pareto dominance is shown in definition 1, where  $P$  is the set of parameter attributes,  $D$  is the set of domination attributes,  $Z$  is the set of design alternatives,  $z'$  is one design alternative, and  $z''$  is another alternative. I will use this technique later to pare down the initial feasible set of designs prior to generating a model

around the non-dominated designs to speed up the model-fitting process and to remove unnecessary undesirable designs early in the process.

Definition 1:

An alternative having attributes  $z \in \mathbf{Z}$  is parametrically Pareto dominated by one with attributes  $z' \in \mathbf{Z}$  if  $z'_i = z_i \forall i \in P, z'_i \geq z_i \forall i \in D$  and  $z'_i > z_i \exists i \in D$ .

## 2.2 Support Vector Domain Description

Support vector domain description (SVDD) is a technique for determining a continuous boundary around data (classifying points as either in or out of the set) using a machine learning algorithm. This technique works for both convex and concave data sets. The original concept of using support vector machines for creating domain descriptions comes from Tax and Duin [17]. They developed the mathematics behind SVDD and demonstrated its use. Malak and Paredis furthered this work by demonstrating SVDD's use in engineering design for model input domain definition [18]. SVDD works by finding the smallest radius hypersphere that contains the input data in an  $n$ -dimensional feature space. The support vectors are those that form the boundary of the hypersphere. These support vectors are found by solving the equation below, also called the Wolfe dual problem, by finding the  $\beta_i$  that maximize the equation.

$$\begin{aligned} \max_{\beta_i} W &= \sum_i \beta_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_j \sum_i \beta_i \beta_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{s.t.} \quad &0 \leq \beta_i \leq C \quad \forall i \\ &1/N \leq C \leq 1, \end{aligned} \tag{1}$$

where  $\mathbf{x}_i$  is a point from the data set,  $C$  is a user-defined variable called the "exclusion constant", and  $N$  is the number of data points. This equation is only useful when a hypersphere is a good model for the data ( $\mathbf{x}$ ). Since this is rarely the case, the equation needs to be mapped into a higher-dimension feature space where a hypersphere is a good

fit for the given data. To do this, the dot products can be replaced by the dot products of non-linear functions  $\Phi(\mathbf{x}_i)$  which perform the desired mapping. With this change, the Wolfe dual equation becomes:

$$\begin{aligned}
 W &= \sum_i \beta_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) - \sum_j \sum_i \beta_i \beta_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\
 \text{s.t.} \quad &0 \leq \beta_i \leq C \quad \forall i \\
 &1/N \leq C \leq 1
 \end{aligned} \tag{2}$$

In this thesis, I replace the dot product of the non-linear transformations with a Gaussian kernel function,  $K_G(\mathbf{x}_i, \mathbf{x}_j)$ , by using a technique known as the “kernel trick” [19]. This allows one to perform the nonlinear transformation without an explicit description of the transformation or higher-dimensional space [17]. The Gaussian kernel function is:

$$K_G(\mathbf{x}_i, \mathbf{x}_j) = e^{-q \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \tag{3}$$

where  $q$  is the user-defined “width parameter” and affects the shape of the domain description by causing the domain to fit more tightly around the data at higher values. As  $q$  increases, the domain often forms “clusters” around smaller and smaller groups of data points, making the domain fit to the data more tightly, but dividing the domain into discrete sections. It is often desirable to prevent this “clustering” by limiting the value of  $q$  to one that fits the data loosely enough to fit all the data points into one single cluster. This is usually a relatively small value (0.5 – 4). The other user-defined variable,  $C$ , determines the domain’s sensitivity to excluding data points from the domain description, but in practice has little effect on the shape of the SVDD [18]. The kernel function replaces the non-linear mapping from the data space to a “feature space” in which the data fits inside a hypersphere. After applying this kernel trick to the previous Wolfe dual formulation, the final Wolfe dual equation to maximize is determined:

$$\begin{aligned}
 W &= \sum_i \beta_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_j \sum_i \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \\
 \text{s.t. } &0 \leq \beta_i \leq C \quad \forall i \\
 &1/N \leq C \leq 1
 \end{aligned} \tag{4}$$

Figure 2-1 shows an approximation of the SVDD generation process visually as a boundary is fit around a sample data set.

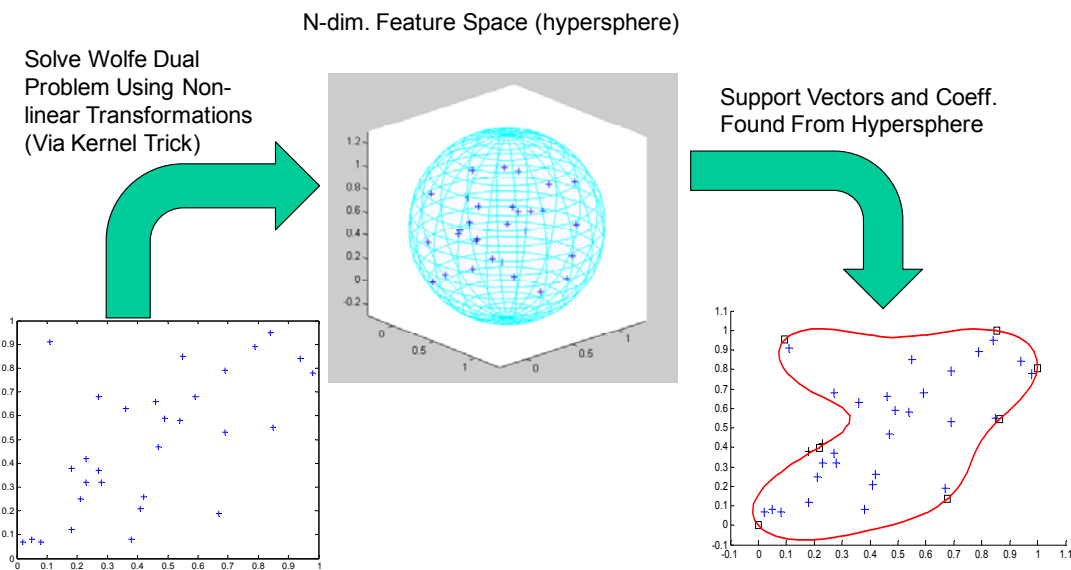


Figure 2-1 - Simplified Description of SVDD Fitting

The radius of the hypersphere is used to determine whether new points fall inside or outside the domain by comparing the distance from the new point to the hypersphere center with the radius. The equation for this calculation is below, where  $\mathbf{z}$  is the test point,  $\mathbf{x}_i$  are the support vectors, and  $R^2(\cdot)$  are the distances.

$$\begin{aligned}
R^2(\mathbf{x}) - R^2(\mathbf{z}) &= K(\mathbf{x}, \mathbf{x}) - K(\mathbf{z}, \mathbf{z}) + \\
2 \sum_i^m \beta_i (K(\mathbf{x}_i, \mathbf{z}) - K(\mathbf{x}_i, \mathbf{x})) &\geq 0, \quad m \leq n
\end{aligned} \tag{5}$$

### 2.3 Interpolation and Kriging

Interpolation is a curve-fitting method in which the model passes through all the data points. The model uses the relative location of the data points to each other in its fitting process. Interpolation assumes that the closer the input data points are to each other, the more positively correlated their outputs are. Kriging (a method of interpolation) can be accomplished in a variety of ways by using different algorithms [20]. A common Kriging approach is known as Ordinary Kriging. In this approach, the predicted value of a new unobserved input is a weighted linear combination of all the previously observed outputs. The following equations describe the Ordinary Kriging model:

$$\begin{aligned}
\hat{Y}(\mathbf{x}_{n+1}) &= \sum_{i=1}^n \lambda_i \cdot Y(\mathbf{x}_i) = \boldsymbol{\lambda}^T \cdot \mathbf{Y} \\
\text{with } \sum_{i=1}^n \lambda_i &= 1, \lambda_i = (\lambda_1, \dots, \lambda_n)^T \text{ and } \mathbf{Y} = (Y(x_1), \dots, Y(x_n))^T,
\end{aligned} \tag{6}$$

where  $\mathbf{x}_{n+1}$  denotes the unobserved input,  $\hat{Y}(\mathbf{x}_{n+1})$  denotes the predictor for the input,  $\mathbf{x}_i$  are the  $n$  previously observed outputs,  $\lambda_i$  are called Kriging weights and capital letters are random variables that are determined through the fitting process. For a more detailed description of this technique and how the weights and random variables are determined see [20]. I use DACE Kriging, a tool developed for Matlab by Lophaven *et. al* to generate Kriging interpolation models of data sets [21].

### 2.4 Linear Regression

Linear least-squares regression is a method for generating a mathematical model for a set of data that seeks to minimize the mean square error between the predicted values of the modeled functions (user-selected functions) and the input data points, while not necessarily passing through the data points like interpolation [22]. Regression fits are

designed to handle noisy data, resulting in the model not necessarily passing through the data points. The model generated by regression analysis is a linear combination of functions included in the fitting process by the user. The “goodness” of the regression fit depends primarily on the regression functions included in the regression model. For most problems, a simple linear regression is not sufficiently accurate. Additional terms such as quadratic terms, cross-terms for multi-variate problems, trigonometric functions, exponential functions, etc. are used to improve the fit by reducing error metrics such as mean-square-error or increasing correlation metrics such as r-squared. One key drawback of linear regression for modeling optimal or Pareto frontiers of data is that the linear regression model may not pass through the points on the frontier and may over- or underestimate the frontier boundary. This point is demonstrated in Figure 1-4 from Section 1. The Pareto frontier is plotted with an example of a linear regression fit on the same graph. The linear regression fit in the figure sometimes extends beyond the Pareto frontier into a region that is not possible with current technology for that particular product. Other times, it falls below the boundary, indicating that no better designs are possible when in fact, there are better designs to the right and above the linear regression model fit. This mischaracterization of the frontier could be detrimental in a systems design problem because it could falsely favor one design over another in an optimization, resulting in a sub-optimal (in the case of the regression model lying under the frontier) or infeasible (in the case of the model lying above the frontier) solution.



### 3. TCM DEVELOPMENT

The tools described in Section 2 can be combined in various ways to develop different TCM representation methods. Table 3-1 shows the steps I took to produce each of the four representation methods I study in this thesis: support vector domain description, parameterized Pareto dominance with interpolation modeling, parameterized Pareto dominance with support vector domain description, and parameterized Pareto dominance with linear regression modeling. Similarly, Figure 3-1 shows a flowchart demonstrating the path taken to produce each type of TCM. This flowchart emphasizes the differences (as noted by the decision nodes) and similarities of the four methods. These steps will be described in more detail below.

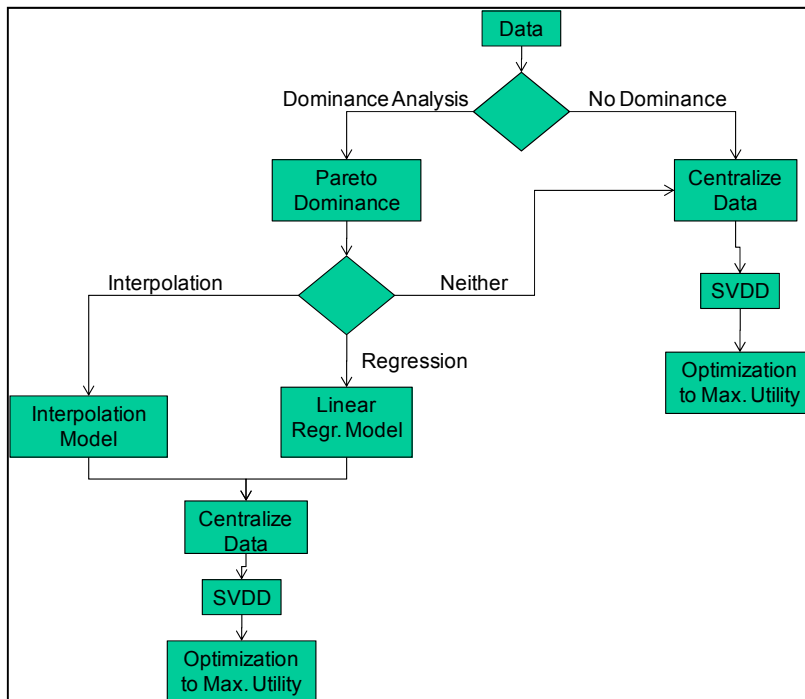


Figure 3-1 - Flowchart of TCM Generation and Optimization Process

Table 3-1 - Steps in Four TCM Methods

Linear Regression on Efficient Set	Interpolation (DACE Kriging) on Efficient Set	SVDD	Efficient set and SVDD
<ol style="list-style-type: none"> <li>1. Use dominance reasoning to reduce data</li> <li>2. Model efficient frontier using linear regression</li> <li>3. Centralize data</li> <li>4. Choose <math>q</math> and <math>C</math> values</li> <li>5. Compute SVDD of efficient set</li> <li>6. Run optimizer on efficient frontier to maximize objective function (constrained by SVDD)</li> </ol>	<ol style="list-style-type: none"> <li>1. Use dominance reasoning to reduce data</li> <li>2. Model efficient frontier using interpolator (DACE Kriging)</li> <li>3. Centralize data</li> <li>4. Choose <math>q</math> and <math>C</math> values</li> <li>5. Compute SVDD of efficient set</li> <li>6. Run optimizer on efficient frontier to maximize objective function (constrained by SVDD)</li> </ol>	<ol style="list-style-type: none"> <li>1. Centralize data</li> <li>2. Choose <math>q</math> and <math>C</math> values</li> <li>3. Compute SVDD of data</li> <li>4. Run optimizer on data to maximize objective function (constrained by SVDD)</li> </ol>	<ol style="list-style-type: none"> <li>1. Use dominance reasoning to reduce data</li> <li>2. Centralize data</li> <li>3. Choose <math>q</math> and <math>C</math> values</li> <li>4. Compute SVDD of efficient set</li> <li>5. Run optimizer on efficient set to maximize objective function (constrained by SVDD)</li> </ol>

### 3.1 Support Vector Domain Description (SVDD)

As shown in Table 3-1, the first step in this TCM method is centralization. Centralizing (scale all data to a -1 to 1 range) the data improves the support vector domain description model [18]. With the data centralized, I proceed to select the important SVDD parameters. Support Vector Domain Description uses a Gaussian width parameter,  $q$ , and exclusion constant,  $C$ , to determine the type of fit modeled. Previous work has shown that  $C$  has little effect on the SVDD, while  $q$  has a significant effect [18]. I select  $q$  values for each dataset by choosing the maximum  $q$  value such that the domain description consists of only one continuous cluster (there are no discontinuities in the domain). Increasing  $q$  beyond such a value produces a domain that is disjointed in more than one cluster, making searching the domain using gradient-based optimization methods more difficult. Others have investigated algorithms (numerical and evolutionary) or heuristics to tune  $q$ , but I chose to use my own algorithm to suit my optimization needs (force the model into one continuous domain for a good search

space) [23, 24]. My algorithm utilizes support vector clustering (SVC), which involves determining how many “clusters”, or disjoint groups of support vectors, a given SVDD contains [25]. I use a bisection algorithm to find the  $q$  value where the SVDD transitions from 1 cluster of support vectors to two. The algorithm fits the SVDD at an upper bound  $q$  value, a lower bound value, and a midpoint, then computes the number of clusters using SVC. If the number of clusters is greater than 1 at the midpoint, the algorithm searches between the lower bound and the midpoint. This continues until the midpoint and the lower or upper bound are within 0.001.

After determining  $q$ , I solve the Wolfe dual problem described in section 2.2. Solving this Wolfe dual problem can be computationally intensive and is highly sensitive to the number of data points being modeled, with computation time increasing super-linearly with the data size [18]. The results of the Wolfe dual computation are values for the support vectors (designated  $\mathbf{x}_{SV}$ ) and support vector coefficients ( $b_{SV}$ ). Using these values, I am able to constrain my gradient based optimizer (for system decision making) by limiting the search to values that fall within the hyperspheric SVDD domain. I do this by calculating a support vector radius,  $r_{SV}$  and hypersphere center,  $\mathbf{a}$ , using the  $\mathbf{x}_{SV}$  and  $b_{SV}$  values. Any design with attribute values that lie a distance  $r_S > r_{SV}$  from the hypersphere center are invalid and cannot be searched by the optimizer. The following set of equations describes the general form of a decision problem solution using a SVDD model:

$$\begin{aligned} \mathbf{z}^* &= \arg \max_{\mathbf{z}} u(\mathbf{z}) \\ s.t. \quad r_S(\mathbf{z}) &\leq r_{SV}, \end{aligned} \tag{7}$$

where  $\mathbf{z}$  is an attribute vector,  $u$  is an objective function,  $r_S(\mathbf{z})$  is the distance between the hypersphere center and the attribute vector,  $r_{SV}$  is the support vector radius, and  $\mathbf{z}^*$  is the solution. The SVDD acts as a constraint in the decision problem optimization to limit the optimizer to the feasible domain for the given data.

### 3.2 Interpolation on Efficient Set

The first step in this method involves parameterized Pareto dominance analysis. This technique, as described above, attempts to reduce the data set by removing designs that would never be rationally chosen by a designer due to the presence of a design that is superior. This step reduces the size of the data for subsequent modeling steps. This is especially important given the super-linear relationship between data size and support vector domain description fitting time mentioned earlier. The product of this step is known as the parameterized Pareto frontier.

The next step involves fitting a mathematical model to the efficient frontier data. My interpolation model uses the DACEfit toolbox Kriging model to develop a model of the frontier, as described in Section 2.3. I select the Kriging model parameters such as: the correlation function (Gaussian, linear, spherical, etc.), and the regression function (2<sup>nd</sup> order polynomial, 1<sup>st</sup> order, etc.) because initial testing showed these settings worked well. I use the Gaussian correlation function and a 2<sup>nd</sup> order polynomial regression function for all three data sets unless the 2<sup>nd</sup> order is a poor fit, in which case I utilize a 1<sup>st</sup> order function. The interpolation model predicts the value of one attribute given the values of the others. Thus, I have reduced the remaining non-predicted data's dimensionality by one, making the data size smaller for the SVDD computation.

Finally, with the interpolation model found, I centralize the non-predicted variable data, select SVDD parameters as before and compute the SVDD of this data on the efficient frontier. Once again, I use the SVDD to bound my optimization problem, but in this case, I bound only the non-predicted attributes and predict the value of the other during each step of the optimization by using my interpolation model. This constraint is necessary because certain combinations of inputs to the Kriging model will give invalid results. The decision problem solution formulation is slightly different for this TCM representation as shown in the following set of equations:

$$\begin{aligned} \mathbf{z}^* &= \arg \max_{\hat{z}} u([\hat{z} = f_K(\tilde{\mathbf{z}}, \tilde{\mathbf{z}})]) \\ \text{s.t. } & r_I(\tilde{\mathbf{z}}) \leq r_{SVI}, \end{aligned} \quad (8)$$

where  $\tilde{\mathbf{z}}$  is a vector of attributes not predicted by interpolation model,  $f_K$  is the Kriging model,  $\hat{z}$  is the predicted attribute,  $u$  is an objective function,  $r_I(\tilde{\mathbf{z}})$  is the distance between the hypersphere center and  $\tilde{\mathbf{z}}$  using the support vectors and their coefficients found from fitting an SVDD to the non-dominated, non-predicted attribute data,  $r_{SVI}$  is the support vector radius, and  $\mathbf{z}^*$  is the solution.

### 3.3 Parameterized Pareto Dominance and SVDD

This method combines the parameterized Pareto dominance of the interpolation on efficient set method and the simplicity of the SVDD method. I first use parameterized Pareto dominance to eliminate dominated designs (especially important because SVDD's computation time is so dependent on the size of the data) and then proceed with the previously defined steps for SVDD: centralize the remaining data (the efficient set), select values for  $q$  and  $C$ , and solve the Wolfe dual problem to determine the support vectors of the efficient set. Finally, I run my optimization with the SVDD serving to constrain my optimizer to a model of the efficient set. The decision problem solution formulation is very similar to that of the SVDD method, as shown by this set of equations:

$$\begin{aligned} \mathbf{z}^* &= \arg \max_{\mathbf{z}} u(\mathbf{z}) \\ \text{s.t. } & r_P(\mathbf{z}) \leq r_{SVP}, \end{aligned} \quad (9)$$

where  $\mathbf{z}$  is an attribute vector,  $u$  is an objective function,  $r_P(\mathbf{z})$  is the distance between the hypersphere center and the attribute vector,  $r_{SVP}$  is the support vector radius of a domain description fit to the parameterized Pareto set, and  $\mathbf{z}^*$  is the solution. The only difference is that the SVDD is fit to the efficient set and not all of the data (the “P” added to the subscripts indicates this change in the model). The rest of the problem is identical.

### 3.4 Linear Regression on Efficient Set

This method is nearly identical to the Interpolation on Efficient Set method. I first perform parameterized Pareto dominance, but then rather than generating an interpolation model of the efficient set, I develop a least-squares regression model fit to the efficient set. Because linear regression models are dependent on the suitability of the data model selected during fitting, I use Matlab's stepwise fit function to select which terms of a full quadratic function with cross-terms have a significant effect on the regression model. I then use these terms to fit the regression model. The remaining steps parallel those of the Interpolation method: centralize the non-predicted attribute data, select  $q$  and  $C$  values, compute the SVDD of this data, and optimize using the SVDD of the non-predicted data as a non-linear constraint. A formalized set of equations for using this TCM representation in a decision problem are shown below:

$$\begin{aligned} \mathbf{z}^* &= \arg \max_{\tilde{\mathbf{z}}} u([\hat{z} = f_R(\tilde{\mathbf{z}}, \tilde{\mathbf{z}})]) \\ \text{s.t. } r_{LR}(\tilde{\mathbf{z}}) &\leq r_{SVLR}, \end{aligned} \quad (10)$$

where  $\tilde{\mathbf{z}}$  is a vector of attributes not predicted by regression model,  $f_R$  is the regression model,  $\hat{z}$  is the predicted attribute,  $u$  is an objective function,  $r_{LR}(\tilde{\mathbf{z}})$  is the distance between the hypersphere center and  $\tilde{\mathbf{z}}$ ,  $r_{SVLR}$  is the support vector radius, and  $\mathbf{z}^*$  is the solution. The only difference between this problem and the interpolation problem is the predictive model used is a regression model instead of an interpolation model (this difference is indicated by the change in subscripts in Equation 10).

## 4. EXAMPLE PROBLEM

To demonstrate and compare the above TCM methods, I conduct an example study. Since my goals are to show how TCMs can be used in systems design problems and also to quantitatively compare each of the methods to each other, I need a problem that is complex and quantitative. Also, my problem needs to involve a comparison of competing technologies that perform the same function or task using different fundamental models or behaviors. This will allow us to show how TCMs permit systems designers to easily compare different component technologies in the same search space, using the same objective function. The problem I choose is the selection of a heat exchanger to be used in a steam power plant's Rankine cycle because it is sufficiently complex and large to be customized, involves multiple competing technologies or types, has a well-defined system-component interaction and hierarchy, and can be easily quantified. TCMs can help power plant designers compare different heat exchanger types and select the best one for their application.

Heat exchangers are important devices in conventional power plant operation because they directly affect the overall plant efficiency and other key system characteristics. Some of the common types are: parallel flow concentric tube, counter-flow concentric tube, shell and tube, multi-shell and tube, cross-flow, and finned. Selecting a heat exchanger type and its dimensions in order to improve the overall power plant performance can be a difficult task because there are many complex relationships between heat exchanger variables and system level variables. Designing a heat exchanger without considering its affects on the system as a whole disregards important relationships and could lead to a suboptimal design. The system-oriented nature of this design problem is noted by Shah and Sekulić [26]: "If the heat exchanger is one component of a system or a thermodynamic cycle, an optimum system design is necessary rather than just an optimum heat exchanger."

My study uses this example problem to compare the results of a system optimization using an approach that starts with the lowest-level design variables (I will call this method the all-at-once, or AAO method) with the four methods of TCM using attributes instead of design variables. I show how the four methods of TCM differ in accuracy (relative to the AAO solution), computation time, and scalability (to data sets of different size).

#### **4.1 System Design Problem**

The systems design scenario involves the selection of a heat exchanger technology to be used as a condenser in a steam power plant non-ideal Rankine cycle. The cycle is shown in Figure 4-1 with assumed state values and relationships indicated. I select the pressure and temperature at state one, the isentropic turbine and pump efficiencies and other assumed values and assumptions as shown in Table 4-1. All of my assumptions are representative of typical power plant systems of this type and scale [27, 28]. Additionally, I assume steady state, steady flow conditions, turbulent flow in all pipes, and negligible kinetic energy and potential energy effects.



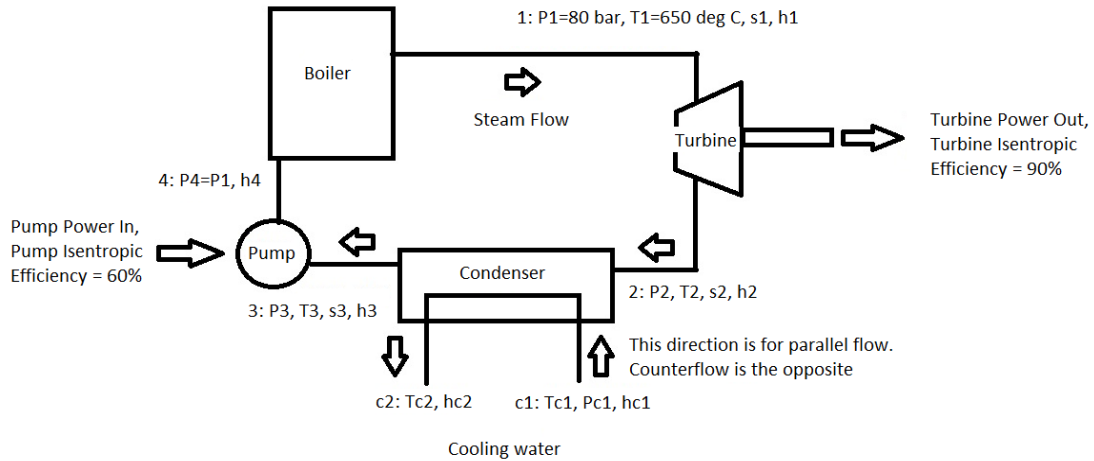


Figure 4-1 - Non-ideal Rankine Cycle

Table 4-1 - Cycle Assumptions

Variable	Assumed Value
Pressure at state 1	80 bar
Temperature at state 1	650 °C
Isentropic Turbine Efficiency	90%
Isentropic Pump Efficiency	60%
$\Delta P$ Across Boiler	0 bar
Cooling Water Velocity	5 m/s
Steam Velocity	60 m/s
Ambient Pressure	1 bar
Ambient Temperature	25 °C
Cooling Water Source and Sink Temperature	15 °C
Cooling Water Pressure	1 bar

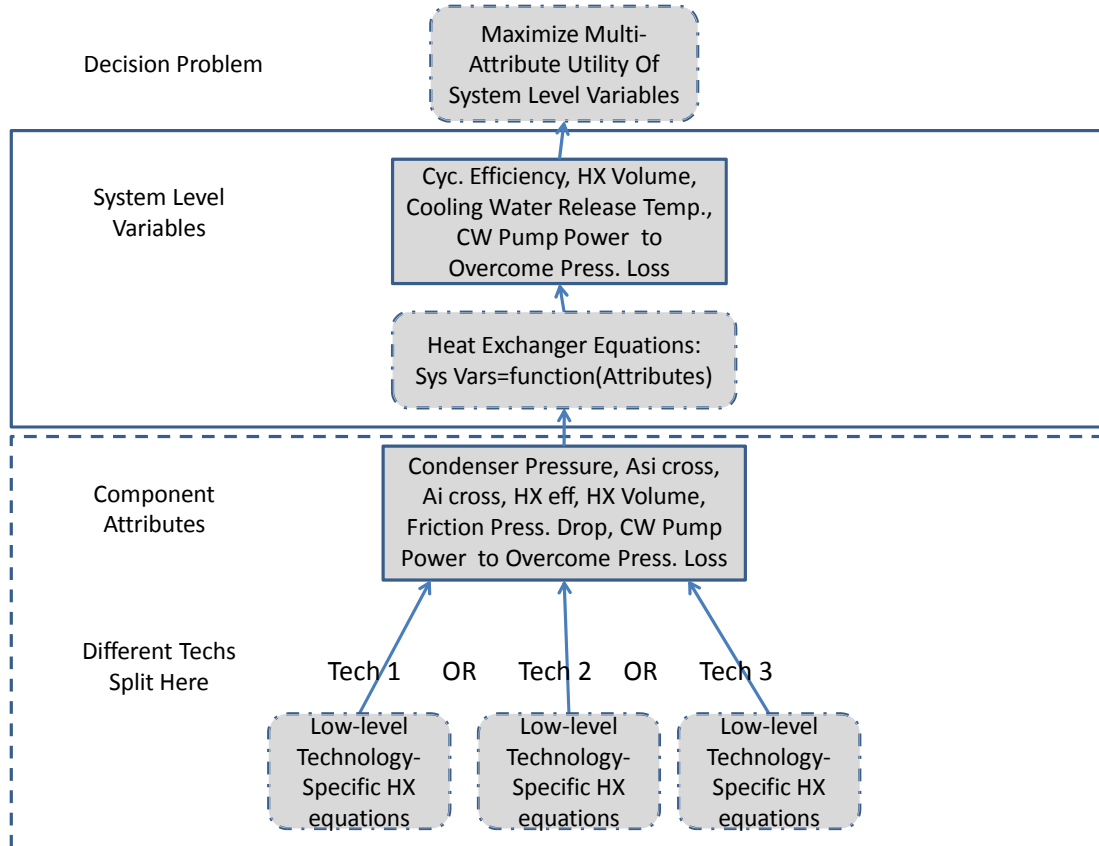


Figure 4-2 - Systems Design Problem Layout

My design objectives are to maximize: cycle efficiency, condenser volume, cooling water release temperature, and cooling water pumping power. I select these parameters because all are affected by the condenser design and would be important to a power plant designer. This importance is due to cycle efficiency being directly related to operation cost, condenser volume being related to condenser purchase cost, land use, and construction costs, cooling water release temperature being regulated by environmental laws if the water is returned to natural bodies of water, and cooling water pumping power being directly related to operating and installation costs. The layout of the entire example problem is shown in Figure 4-2.

The objective function I use to optimize these system-level properties is a utility function, with preferences characterized using multi-attribute utility analysis [29, 30]. The objective is to maximize utility within the feasible domain. Using my own preferences for each system level property's values, I develop individual utility functions as shown in Figure 4-3. These functions are used to convert each design's parameter values to a value between 0 and 1 to be used in the overall utility objective function for each iteration of the search.

I combine individual utility function output values by using this equation:

$$U(X) = k_e u_e(x_e) + k_{wp} u_{wp}(x_{wp}) + k_V u_V(x_V) + k_{Tc} u_{Tc}(x_{Tc})$$

$$X = [x_e \ x_{wp} \ x_V \ x_{Tc}], \quad (11)$$

where  $X$  is the set of system-level values being evaluated, the  $x_i$  are the respective elements of  $X$ , the  $k_i$  are the scaling factors ( $\sum_{i=1}^n k_i = 1$ ), and the  $u_i$  are the individual utility function output values for the design being evaluated.  $U$  is the total utility for each design (optimization step). This is the value that I wish to maximize. I use equal scaling factors in this example.

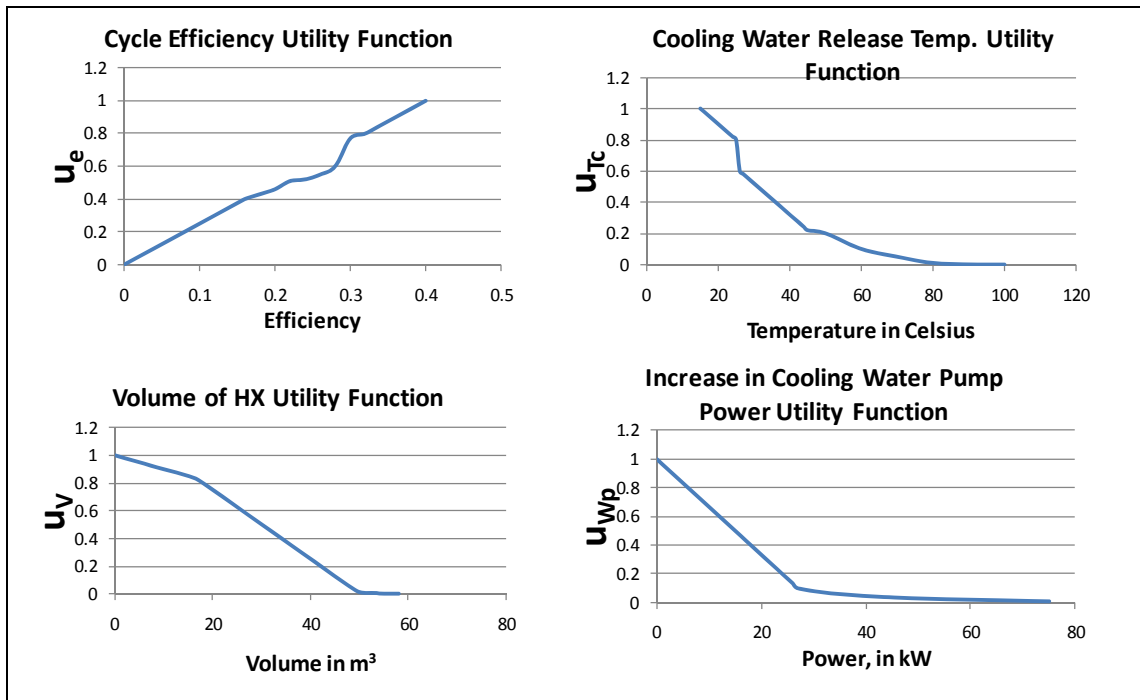


Figure 4-3 - Individual Utility Functions

## 4.2 Heat Exchanger Technologies Studied and Design Variables

I consider three main “technologies” or types of heat exchanger: parallel flow concentric tube, counter-flow concentric tube, and shell and tube. Diagrams of each of these heat exchanger technologies are shown in Figure 4-4. The structural differences between the technologies—flow direction and number of tubes—are readily apparent in the diagrams. Also indicated are the five design variables used in this example:  $Do$ ,  $Di$ ,  $Dsi$ ,  $Dso$ , and  $L$ . These represent, respectively, the outer and inner diameters of the inner tubes/pipe, the inner and outer diameters of the shell, and the length of each tube/pipe and the entire heat exchanger itself. By varying the five design variables, an infinite number of heat exchanger designs is possible. However many of these designs are not feasible due to thermodynamic or physical limits and geometric constraints. It should be noted here that for the shell and tube heat exchanger type, I calculate the number of tubes ( $N$  in the computations) by computing the maximum number of pairs of tubes that

would fit within the shell given the diameters of the shell and tubes. Thus,  $N$  is not an independent design variable like the other five.

Standard engineering design optimization using these five variables would require a different model for each technology because each has its own unique equations that relate these variables to attributes and system-level variables. For example, the heat exchanger effectiveness (a measure of the efficiency of a heat exchanger's heat transfer) is related to the heat capacity of the fluid passing through the device and the number of heat transfer units (NTU) by a unique equation for every type of heat exchanger, thus one engineering model cannot be used to optimize the design of all three technologies. Although each technology has the same design variables, they relate to different elements (diameter of multiple tubes in shell-and-tube versus the diameter of only the one inner concentric tube in the others) and use different models to arrive at attributes. Additionally, in the optimization method of Section 4.5 below, the shell-and-tube technology requires an additional constraint to ensure there is enough space between the shell's inner diameter and the tube's outer diameter to fit at least one pair of tubes inside the shell. Getting from these five design variables and the earlier assumptions to attributes and important system level parameters requires many steps of calculation. The calculation methodology is summarized in Table 4-2.

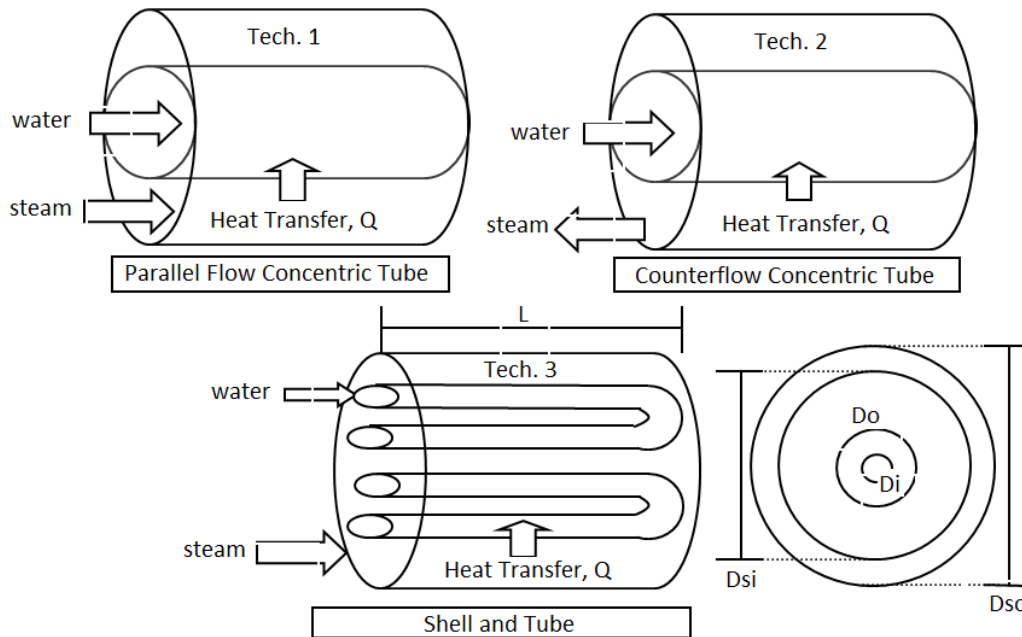


Figure 4-4 - Heat Exchanger Technologies

This table displays the steps taken and the equations used to compute the desired system level values: cycle efficiency, condenser volume, cooling water release temperature, and cooling water pumping power. Also, this table demonstrates two other important items:

- The TCM approach involves only computing system-level values from high-level attributes, while the method using the design variables involves all of the computation steps (although the early computation steps are eventually used on the “winning” condenser technology to determine its design variables once the winner is determined from the attribute-level problem)
- The primary computation difference between the technologies is highlighted in step 12 and demonstrates why optimizing with only design variables would require three separate models to arrive at the desired system-level objectives

Table 4-2 - Heat Exchanger Calculation Procedure

Step	Computations	Used in	Level of Detail
1	Compute cross-sectional and surface areas and number of tubes, N, that will fit, if necessary	AAO	Design Vars
2	Lookup properties at state 1 [31]	AAO	Design Vars
3	Iteratively solve for pressure at state 2 using state 1 and isentropic efficiency of turbine to get enthalpy at state 2 (h2)	AAO	Design Vars
4	Lookup properties at state 2 using P2, h2	AAO	Design Vars
5	Lookup up properties at cooling water inlet state (c1)	AAO	Design Vars
6	Using fluid velocities, densities and pipe areas, compute mass flow rates of both fluids	AAO	Design Vars
7	Compute Reynolds number at states 2 and c1 [32]	AAO	Design Vars
8	Compute convective heat transfer coefficients, hi and ho, inside and outside tube using correlations [33-35]	AAO	Design Vars
9	Compute heat capacities of hot and cold fluids (Ch, Cc) using specific heats and mass flow rates	AAO	Design Vars
10	Compute overall heat transfer coefficient, U, from fouling coefficients, tube conductivity, hi, ho, Ai, Ao, and L [32]	AAO	Design Vars
11	Compute num. of transfer units, NTU, from tot. transfer area, $A=\pi \cdot D_o \cdot L \cdot N$ , U and C values [32]	AAO	Design Vars
12	Use correlations for C and NTU <b>by technology</b> to compute HX effectiveness [32]	AAO	Design Vars
13	Compute max. heat transfer, $q_{max} = \text{mass flow} \cdot (h_2 - h_{c1})$	TCM, AAO	Attributes
14	Compute state c2 from HX water side energy balance	TCM, AAO	Attributes
15	Compute Tc2 (cooling water release temp.) from hc2 and Pc2	TCM, AAO	Attributes/Sys. Objective
16	Fix state 3 by iteratively solving for P3 and T3, given h3 from steam side HX energy balance	TCM, AAO	Attributes
17	Compute average HX temperature and pressure by averaging states 2 and 3 and repeat steps 4 – 16 using Pavg, Tavg for steam properties	TCM, AAO	Attributes
18	Compute friction factor, f, and equivalent length, Le, for tubes and shell [32, 36, 37]	AAO	Design Vars
19	Use Darcy-Weisbach equation to compute friction pressure drops in tubes and shell, then pump power to overcome [36]	AAO	Attribute/Sys. Objective
20	Compute enthalpy at state 4 with isentropic pump efficiency	TCM, AAO	Attributes
21	Compute turbine power, boiler heat transfer	TCM, AAO	Attributes
22	Compute total cycle pump power by adding initial pump power from states 3 and 4 to pump power to overcome press. drop in shell	TCM, AAO	Attributes
23	Compute total cycle thermodynamic efficiency	TCM, AAO	Sys. Objective
24	Compute volume of HX	AAO	Attribute/Sys. Objective

### 4.3 Heat Exchanger Attributes

As mentioned in Section 1, low-level design variables and models can contain proprietary information about a product and/or can add domain-specific knowledge requirements to the systems design problem. Therefore, I seek to use higher-level variables in developing the TCMs. This process involves using the component level model to take the low-level design variables ( $D_o$ ,  $D_i$ ,  $L$ ,  $D_{so}$ , and  $D_{si}$  in this example) and abstract them to a level suitable for the given systems design scenario and that allows different technologies to be compared in the same variable space. In other words, this process takes low-level component variables and turns them into only the variables that the systems designer needs to optimize his or her system and select the best component design while being able to compare technologies with different low-level models. These higher level variables needed for the system optimization are called “attributes” of the component for use in a TCM. In this problem, I use some of the steps in Table 4-2 to calculate the seven attributes needed for the system design optimization.

I select these seven attributes (condenser inlet pressure, shell cross-sectional area, tube cross-sectional area, condenser effectiveness, condenser volume, friction pressure loss across condenser, and the pump power needed to overcome cooling water friction pressure loss) because they are the minimum variables necessary to complete the computation of the system-level variables of interest. Two of these system-level variables are identical to their corresponding attributes, but the other two require additional calculation steps to arrive at the system-level values. However, they require only steps 13-17 and 19-24 of Table 4-2’s calculations to reach the desired system values, resulting in less effort on the part of the system designer in his or her system optimization (only need 11 of the 24 calculation steps).



#### 4.4 TCM Generation and Optimization

I generate potential condenser designs (combinations of the five design variables) by randomly generating diameter values selected from standard ASME NPS pipe sizes for the design variables (within reasonable constraints shown in Table 4-3) and running them through a set of constraints to remove infeasible designs such as ones where the tube outer diameter exceeds the shell outer diameter. From the remaining designs, (those that are feasible) I select 100 designs at random for each of the three technologies. These designs are displayed in Table A-5 in the appendix. I then perform the necessary calculations to convert the design variable information to attribute data for each design. These 100 designs and their attributes then constitute my “catalog” of discrete potential condenser designs of each type. I then use these 100 designs to develop the TCMs for each technology.

Table 4-3 - Design Generation Limits

Design Variable	Bounds/Constraints	Design Variable	Bounds/Constraints
Do	Ranging from 0.01 to 2.48 m	Dsi	Maximum of 2.49 m
Dsi-Do	Ranging from 0.01 to 2.47 m	Dso	Maximum of 2.5 m
Inner and Outer Wall Thickness	= $0.0919 * Do$ (or $Dso$ ) + 0.0033 based on NPS data on wall thickness vs. diameter with minimum of 0.01 and maximum of 0.2 m	L	Randomly generated between 1 and 20 m
Di	Minimum of 0.01 m		

With data for all 7 attributes in numerous configurations, I proceed to develop my technology characterization models for each condenser type just as described in Section 3. However, I do add one additional physical linear constraint on the optimizer forcing the cross-sectional area of the shell to be greater than that of the tube inside it to prevent the impossible situation where the inner tube is larger than the outer shell. This is

necessary because some of the designs have values near or at the constraints and when a TCM is fit to attributes of these designs, it may extend just beyond the feasible domain near that point due to small inherent errors in the model. The optimizer may search in this small infeasible part of the TCM and find an infeasible solution.

#### 4.4.1 SVDD on Example Problem

Following the steps of Table 3-1, I first centralize (re-scale to a range of -1 to 1) the attribute data for each technology and solve for the maximum  $q$  value such that the data is in one cluster by fitting domain descriptions and counting the number of clusters at each  $q$  value attempt in an optimization scheme. The resulting  $q$  value is then used to fit the SVDD around the attribute data. Table A-3 in the appendix displays the  $q$  values I used for fitting each of the TCMs, technologies, and data sets. Table 4-4 shows the number of support vectors found for each technology for all four TCMs. Table A-4 in the appendix shows the number of support vectors for different sizes of attribute data, while Table A-6 in the appendix shows the actual support vectors (and their coefficients) used for each TCM and technology for the full dataset. The number of support vectors is always less than or equal to the number of data points and often far less. In this example, there are only 20-40 support vectors for 100 data points, meaning that a minority of points assist in defining the SVDD boundary, while the others lie inside (or outside if some are excluded) the domain. These support vectors and their radii are then used as constraints in the optimization problem defined in Equation 7, with the utility function (Equation 11) as the objective function.

Table 4-4 - Number of Support Vectors by TCM and Technology

TCM	Tech 1	Tech 2	Tech 3
SVDD	38	41	29
PPS SVDD	34	39	29
Interpolation	33	34	21
Regression	33	34	21

#### 4.4.2 PPS SVDD on Example Problem

The first step in this method is to use parameterized Pareto dominance. I use this procedure to eliminate points from the condenser technology attribute data sets that are dominated by other points. The results of this analysis for all three technologies are shown in Table 4-5. I define the cross-sectional areas ( $A_i$  and  $A_s$ ) as the parameters (attributes that I do not yet have a preference for one way or another) and the other attributes as dominance attributes (subjected to dominance). I prefer to minimize the pressure drop, maximize the effectiveness, minimize volume, minimize pump power, and minimize operating pressure. The dominance process eliminates only a few points from each data set in this example, making it less useful. This is because the two parameters make it more difficult for a design point to be dominated (there are few points with equal parameter values for both parameters, a necessary condition for parameterized Pareto dominance as defined in Section 2.1). From here, the process proceeds just as with the SVDD method. The number of support vectors for each technology after dominance analysis is shown in Table 4-4. Equation 9 in Section 3.3 describes the optimization problem for this TCM. The support vector radius is used as a constraint on the optimizer and the utility function (Equation 11) is the objective function.

Table 4-5 - Number of Non-Dominated Points by Technology

Tech 1	Tech 2	Tech 3
92	94	95

#### 4.4.3 Interpolation on Example Problem

This method begins with parameterized Pareto dominance just as the previous section defines. The results for all three technologies are the same as in Table 4-5, as they are unchanged from the PPS SVDD method. Step 2 of Table 3-1 is accomplished by using DACE Kriging to model the Pareto frontier data by predicting the value of one attribute

based on the values of the other six. This reduces the dimensionality of the problem by one. DACE Kriging selects the appropriate fitting parameters by running its own internal optimization within user-defined bounds. This tool also requires the user to select what it calls “correlation functions” and “regression functions.” I select the Gaussian correlation function and the “regpoly2” regression function. Once the model is fit, I complete the steps of Table 3-1 by centralizing the remaining non-dominated data (the six attributes that are not predicted by the Kriging model), choosing appropriate SVDD parameters as before and computing the SVDD of this data. The optimization problem then looks like Equation 8, with the objective function replaced by my utility function (Equation 11). The number of support vectors for this method for the three technologies are also shown in Table 4-4. In this case, the support vectors are only six-dimensional because one attribute is predicted and is not part of the SVDD used to constrain the optimizer.

#### 4.4.4 Linear Regression on Example Problem

This method starts with the same dominance analysis as the previous two and the number of non-dominated points remains the same as in Table 4-5. Step 2 of Table 3-1 for Linear Regression states that the Pareto frontier should be modeled with a linear regression model. As mentioned in Section 3.4, I assume a full quadratic function with cross-terms as my regression function and use the stepwisefit function to eliminate unnecessary terms. This function uses a statistical p-value test to determine which terms improve the model correlation significantly and which have little effect. If a term’s p-value is larger than 0.1, it is removed from the model and if it’s p-value is less than 0.05, it is kept in the model. Equations 12, 13, and 14 detail the linear regression models I use in this problem for each technology after stepwisefit has eliminated some terms.

$$\begin{aligned} \text{Tech 1: } V = & 1 + 2.9469w - 1.8498p - 0.0031dh - 6.3189E^{-4}dw + 0.4037dA_s + \\ & 0.00533dA_i + 2.492hA_s - 6.2879wp + 0.521wA_s - 1.5541A_sA_i + \\ & 8.2876E^{-7}d^2 + 0.0207w^2 \end{aligned} \quad (12)$$

$$\text{Tech 2: } V = 1 - 0.0115dp + 0.3481dA_s + 0.0415dA_i + 1.1693hp + 0.8547wA_s - 2.1963A_sA_i + 1.4060E^{-6}d^2 \quad (13)$$

$$\text{Tech 3: } V = 1 - 2.3633w - 6.9753E^{-5}dp + 1.1685dA_s + 2.4427hw - 1.2461E^3hA_s + 10.5588wA_i + 1.2034E^3pA_s - 9.2102E^{-6}w^2 + 1.1088E^3A_i^2, \quad (14)$$

where (for all three equations)  $V$ = volume,  $w$ =pump power,  $d$ =pressure drop,  $p$ =condenser pressure,  $h$ =heat exchanger effectiveness,  $A_i$ =tube cross-sectional area, and  $A_s$ =shell cross-sectional area. All three achieve high correlation coefficient ( $R^2$ ) values. These values are 1, 1, and 0.9148 for technology 1, 2 and 3, respectively. From this step on, the process is nearly identical to that of the Interpolation method. The only difference is that the optimization formulation is in Equation 10 instead of Equation 8. The key difference between these two optimization problems is the Pareto frontier model is a regression model instead of an interpolated model. The number of support vectors used as a non-linear constraint on the optimizer is shown once again in Table 4-4.

#### 4.5 All-at-Once Optimization (AAO)

In order to compare the TCM optimized designs and overall utility values with an objective standard, I conduct an all-at-once optimization (AAO). The AAO method involves optimizing the design of a system by varying the lowest-level variables of the components of the system. In this case, I vary the five design variables of the condenser to optimize the utility of the power plant system. This entails placing bounds on the design variables equal to those used to generate the attribute data above for each condenser type and allowing the optimizer to compute the system level variables from these design variable values (the AAO setup can be seen in Table 4-6). For this AAO approach optimization, I use Boeing's design explorer optimization algorithm (within Phoenix Integration's ModelCenter software) because it is a surrogate-based search that optimizes globally, can handle non-smooth or noisy design spaces and can handle analysis code failures [38]. I use this algorithm along with Matlab code to compute the system level variables and utility to find my optimal design for each condenser type.

Table 4-6 - AAO Constraints

Design Variable/Output Variable	Lower Bound	Upper Bound
Do	0.01 m	2.48
Di	0.01 m	2.47
Dso	0.01 m	2.5
Dsi	0.01 m	2.49
L	1 m	20
Dso-Dsi	0.01 m	0.2
Dsi-Do	0.02 m	2.47
Do-Di	0.01 m	0.2
Cooling Water Release Temperature	15 °C	None
Cycle Efficiency	0	1
Calculated Boiler Heat Transfer	0 kW	None
Increase in Cooling Water Pump Power	0 kW	None

#### 4.6 Results of Example Problem

Are attribute solutions from optimization of a TCM feasible? (Do feasible design variables corresponding to these attributes exist?) The optimal designs resulting from applying all four TCM methods and the AAO approach to the condenser technology selection problem are shown in Table 4-7. The TCM design variables are derived by taking the best attribute vector of ten optimization runs with random starting points and running a separate optimization problem using the component-level model to find design variables that come closest to generating attributes matching the attribute values provided by the TCM solution. If the TCM solution cannot be built (i.e. constraints on design variables are violated), the feasible design closest to the TCM solution is used. I use the Design Explorer optimizer again (with the same design variable constraints as used in the AAO approach) to vary the design variables in order to minimize the Euclidean distance between the attribute targets from the TCM solutions and the

attribute values computed from the design variables the optimizer is testing. The last column of Table 4-7 is a measure of how close the design comes to generating attribute values equal to those of the TCM solution. It is a Euclidean distance from the vector of attributes from the TCM solution to the closest feasible design's attributes. The larger the number, the farther away the TCM's solution is from a feasible design. This table shows that the first two technologies have relatively small distances for all four TCM methods. The third technology, however, shows a dramatic difference in accuracy between the SVDD-based methods and the regression and interpolation methods. This is possibly due to the model of the Pareto frontier being jagged or fitting poorly in certain regions. Interpolation may create anomalous curve-fits between data points and regression, as mentioned earlier, can under- or over-estimate the frontier. Table 4-7 also shows that TCMs can produce attribute results that correspond to feasible designs with only a small amount of error. It should also be noted that for all three technologies, the SVDD-based methods (SVDD, PPS SVDD) are the most consistent and never exceed a distance of 100 from the feasible design solution.

Table 4-8 shows the best utility values found for each technology and method. It also shows which technology had the highest utility and therefore, is the preferred technology for this problem for each method. From these results, technology 2 (counter flow concentric tube) is the best technology for this application. The AAO and SVDD-based TCMs both agree that technology 2 is superior based on a rank-ordering of the total utilities of each technology. The other two TCM methods, regression and interpolation, select technology 1 as the winner. This is likely due to the relative inaccuracies of these methods as mentioned above. The deficiencies of these methods extend further and will be discussed in Section 5. The fact that the SVDD-based methods are most consistent in terms of the distance from Table 4-7 and that they lead to the same conclusion as the AAO solution indicates they may be better methods of TCM than the other two.

Table 4-7 - Example Problem Results

<b>Tech 1</b>	Do	Di	Dso	Dsi	L	Eucl. Dist. To Solution
SVDD	1.27877	1.15352	1.66352	1.61813	10.3516	32.7353
PPS SVDD	1.87697	1.68684	1.94072	1.89906	15.8438	87.5904
Interpolation	1.78049	1.70605	2.5	2.48516	20	125.024
Regression	0.71916	0.5193	1.49816	1.48734	20	34.5875
AAO	0.51172	0.50008	0.83189	0.75109	1.51953	N/A
<b>Tech 2</b>	Do	Di	Dso	Dsi	L	Eucl. Dist. To Solution
SVDD	0.68057	0.46645	1.25014	1.20641	12.0586	37.4966
PPS SVDD	0.7674	0.56734	1.26959	1.19188	9.53516	34.3449
Interpolation	1.47656	1.27844	2.15471	1.99109	19.1094	15.3238
Regression	0.78188	0.7307	1.41063	1.23063	3.67188	17.7023
AAO	0.57443	0.5193	0.63736	0.61063	1	N/A
<b>Tech 3</b>	Do	Di	Dso	Dsi	L	Eucl. Dist. To Solution
SVDD	0.2162	0.07246	0.36502	0.31516	19.7031	24.8111
PPS SVDD	0.02447	0.01	0.67141	0.46531	11.168	95.7214
Interpolation	0.84459	0.625	2.10607	2.08313	20	8820.12
Regression	0.63232	0.50488	1.3085	1.29359	19.8516	954.882
AAO	0.02447	0.01	2.5	2.49	1	N/A

Table 4-8 - More Example Problem Results

Technique	Tech 1 Util.	Tech 2 Util.	Tech 3 Util.	Max Util	Rank Order Winner
SVDD	0.831	0.892	0.871	0.892	tech2
PPS SVDD	0.882	0.911	0.871	0.911	tech2
Interpolation	0.935	0.930	0.921	0.935	tech1
Linear Reg	0.931	0.931	0.911	0.931	tech1
AAO	0.914	0.927	0.640	0.9271	tech2

Table A-1 in the appendix shows the results of the system optimization including the best condenser designs (in terms of attributes) found in each TCM and the AAO method for each technology and the full data set. Table A-2 shows the utility values of each of



these designs. This thesis is focused on the comparison of TCMs and not on the optimal condenser design, so the results of this optimization are not a definitive statement about heat exchanger technologies in general, or which is best. These tables are presented to show the differences between the solutions of the different TCMs and the AAO approach visually, while the numeric differences are quantified in Section 5.1.

#### **4.7 Summary of Example Problem**

This example shows how TCMs can be used to make decisions in systems design. Using the design of a condenser for a steam power plant, I show that TCMs are readily applicable to engineering problems and can easily compare different technologies. In examining the results of the example problem, the counter flow concentric tube technology stands out by having the highest utility for two of the TCM methods and the AAO approach. The remaining TCMs disagree with this conclusion due to error in their model fits that can lead to poor representations, infeasible designs, and inaccurate solutions. From heat transfer theory, it makes sense that the counter flow heat exchanger would be more effective at transferring heat because the temperature difference between the hot and cold fluids is larger at the heat exchanger entrance than in the parallel flow heat exchanger. The SVDD-based methods both correctly identify this outcome (although because heat transfer effectiveness affects only part of the utility function, it alone does not determine which technology is better). Because this example is sufficiently complex, produces good results, and demonstrates TCMs well, I re-use it in a parametric study to determine what happens when the amount of available attribute data to fit the models to changes.

## 5. COMPARISON STUDY RESULTS

I now take the above example problem and change the number of attribute data points used to fit the TCM to see how the TCMs are affected. This is a parametric study of the example problem under varying conditions. Specifically, I seek to answer questions 5-8 of Section 1.7 in doing this. These questions are important to answer because they lead to selecting the best representation of TCM, which is the key contribution of this research. With this information, one can focus on using this type of TCM over the others on other problems.

### 5.1 Accuracy

Which TCM is the most accurate when its solution is compared to a trusted, well-defined method's solution? To answer this question and compare the four TCM representation methods and objectively show how one is superior, I compute a distance metric. This distance metric is a measure of the Euclidean distance ( $\sqrt{(a_{AAO}-a_{TCM})^2+(b_{AAO}-b_{TCM})^2+\dots}$ ) between the normalized (all values re-scaled to a 0 to 1 range) vectors of the optimized designs (attribute values) for each condenser type and TCM and the corresponding optimized design (in attributes) from the AAO method. This is accomplished by using the optimal feasible design (based on total utility) using each method for each condenser type and the AAO solution. These values are the best of 10 optimization runs for each condenser type and method. Figure 5-1 show the values of both the distance metric and the next metric, computation time, for each of the methods and technology for the full data sets. These figures show that on average, the SVDD-based methods are clearly superior to the others by having lower metric values (their data points are farther down and to the left of the graph). The smaller the distance metric, the closer the TCM solution is to the AAO solution, and the more accurate the TCM solution is.

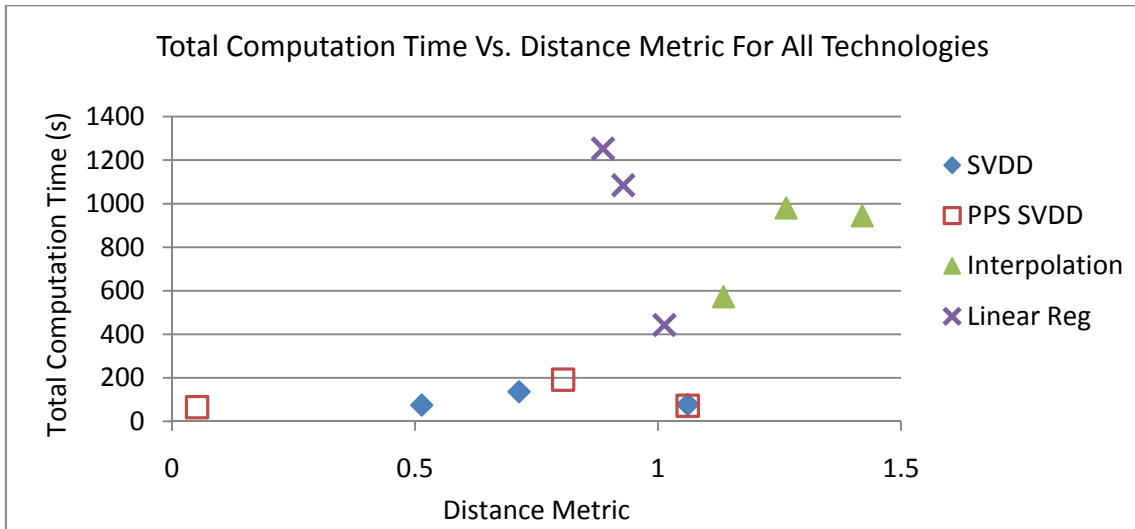


Figure 5-1 - Comparison Study Metrics

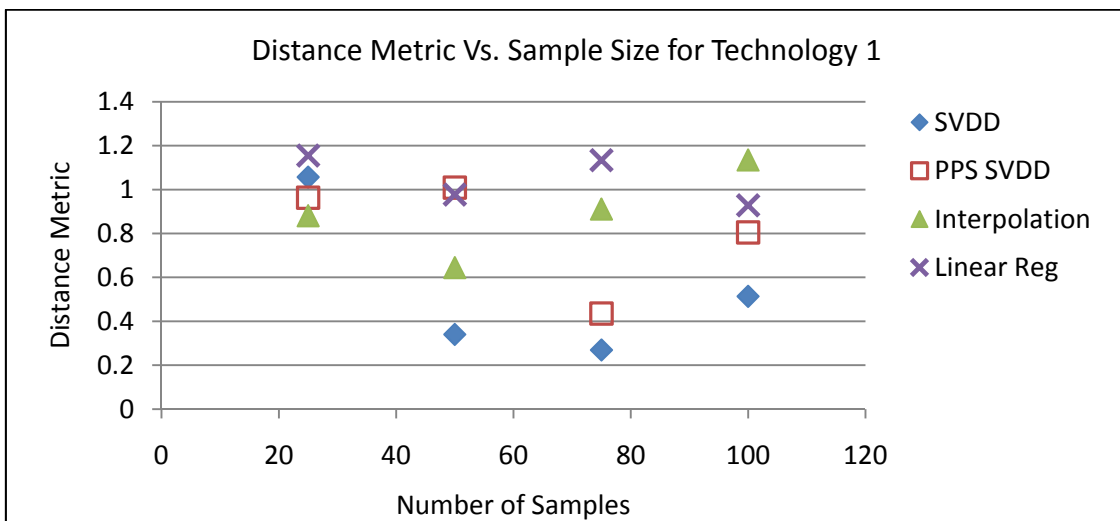


Figure 5-2 - Distance Metric for Different Sample Sizes

Table 5-1 - Breakdown of Computation Time

Method	Domination Time			Model Time			SVDD Time		
	Tech 1	Tech 2	Tech 3	Tech 1	Tech 2	Tech 3	Tech 1	Tech 2	Tech 3
SVDD	0	0	0	0	0	0	44.21	39.19	47.66
PPS SVDD	0.047	0.008	0.005	0	0	0	26.24	33.74	40.63
Interpolation	0.015	0.005	0.006	0.370	0.154	0.156	32.19	30.94	25.54
Linear Reg	0.006	0.005	0.006	0.676	0.007	0.008	30.72	33.57	26.80
AAO	0	0	0	0	0	0	0	0	0

Method	Optimization Time			Total Time		
	Tech 1	Tech 2	Tech 3	Tech 1	Tech 2	Tech 3
SVDD	30.42	96.98	29.78	74.63	136.17	77.44
PPS SVDD	165.18	31.51	31.29	191.47	65.25	71.93
Interpolation	540.17	949.02	918.61	572.74	980.11	944.31
Linear Reg	1053.19	1219.33	415.99	1084.59	1252.91	442.81
AAO	807	1530	443	807	1530	443

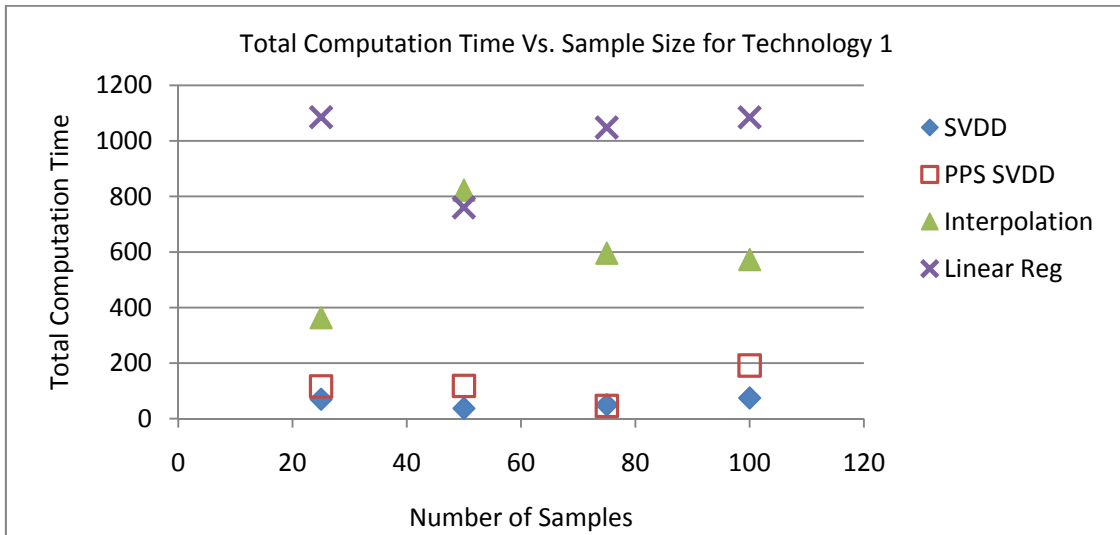


Figure 5-3 - Computation Time for Different Sample Sizes

## 5.2 Computation Time

Which TCM type requires the least computation time? To answer this question I use a second metric, computation time, which is a measure of the total amount of time needed to complete the TCM fitting and optimization on a 2.93 GHz, quad-core processor using 8GB of RAM. This metric is also broken down into smaller segments based on individual tasks within the process: dominance, model fitting, SVDD, and optimization. Figure 5-1 shows this metric plotted against the distance metric so the values of these two metrics for each TCM can be easily compared for the full data set.

Table 5-1 shows the breakdown of the time metric into the individual computational steps. This table shows that although the SVDD method takes the longest to produce the TCM, its optimization time is much shorter than both Interpolation and Linear Regression, and slightly shorter than the PPS+SVDD method on average. Work by Roach, Malak, and Parker describes a new SVDD algorithm that drastically reduces the SVDD model-forming time, making it even more appealing and reducing its primary weakness: high computation time with large datasets [39].

## 5.3 Scalability

How do all four types scale with the size of the attribute data? To answer this question and test the effects of changing data size and of less information on these TCM methods, I conduct the same analysis as above using data sets of size:  $G/2$ ,  $G/4$  and  $3G/4$ . These data sets are randomly sampled from the original data set for each technology with original size,  $G=100$  designs. This analysis is particularly important given the data size dependency of the SVDD method. However, prior work on SVDD shows that this effect may be most extreme on datasets larger than 100 or 200 points, so this case does not represent an extreme case [18].

Figure 5-2 shows an example of this analysis for the parallel flow concentric tube technology for the distance metric. Figure 5-3 shows the same analysis for the time

metric. Neither graph shows an identifiable trend relating sample size to accuracy or computation time. However, both of these graphs show that the SVDD method often has the lowest computation time and the lowest distance, even compared to the PPS+SVDD method. Combining this information with that of Figure 5-1, it seems the SVDD method provides the best combination of accuracy and computational time.

#### **5.4 Summary**

Which TCM is the best overall? This can be answered by looking holistically at all of the metrics for all of the data sizes and drawing conclusions from them. Figures 5-4, 5-5 and 5-6 show the same metrics displayed in Figure 5-1 for the other data sizes. These figures show that the same trends emerge at different sample sizes. The SVDD and PPS+SVDD methods have, on average, the lowest values of computation time and often the lowest values of distance (their points lie closest to the lower left corner of the graph). Figures 5-7 and 5-8 show the effects of changing sample size on the accuracy of the other two technologies (Technology 1 is in Figure 5-2). Figure 5-7 furthers my conclusion that SVDD-based methods are superior, while Figure 5-8 shows that technology 3's results are inconclusive and have little variation in accuracy between three of the TCMs (with Interpolation being the only one significantly less accurate than the others). Figures 5-9 and 5-10 demonstrate once again that the SVDD-based methods consistently require less computation time (all three technologies and all four data sizes exhibit this same trend).

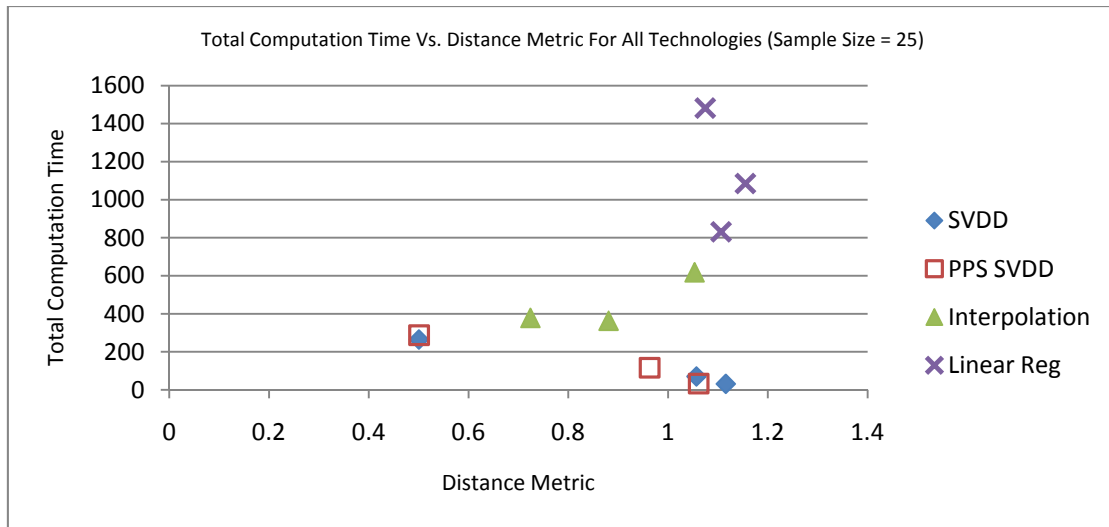


Figure 5-4 - Comparison Study Metrics for Sample Size 25

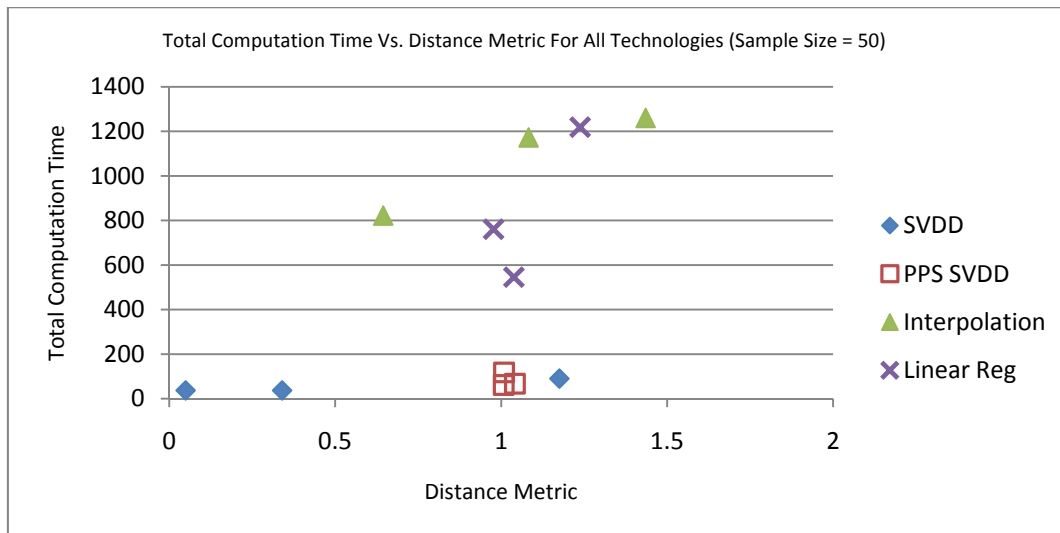


Figure 5-5 - Comparison Study Metrics for Sample Size 50

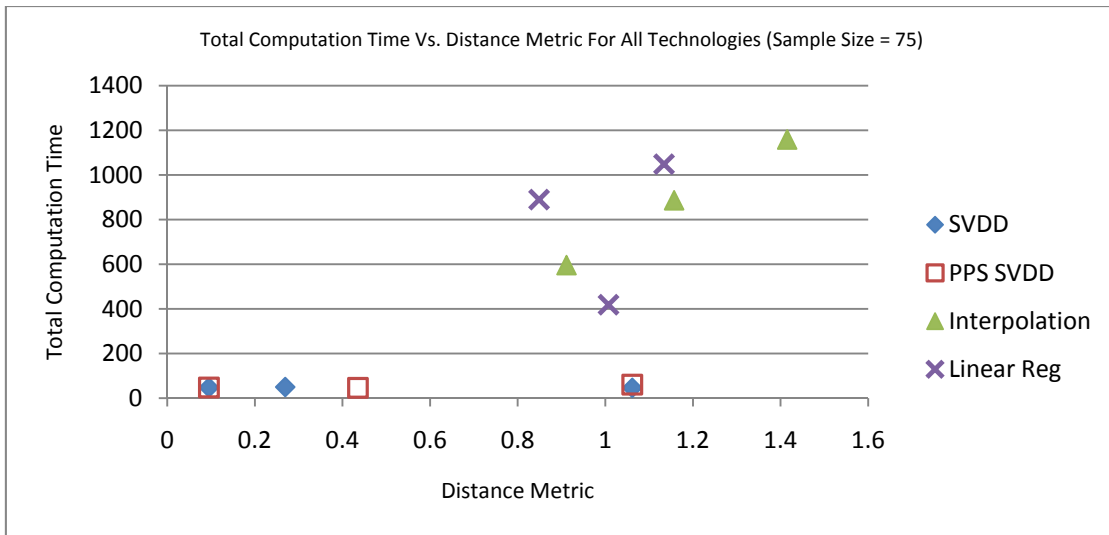


Figure 5-6 - Comparison Study Metrics for Sample Size 75

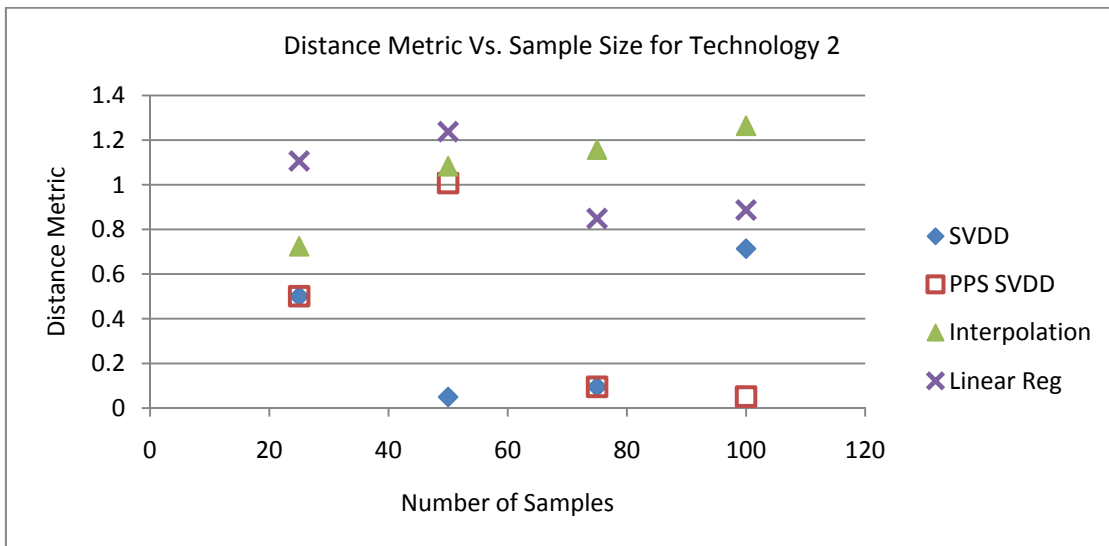


Figure 5-7 - Distance Metric for Different Sample Sizes (Tech 2)



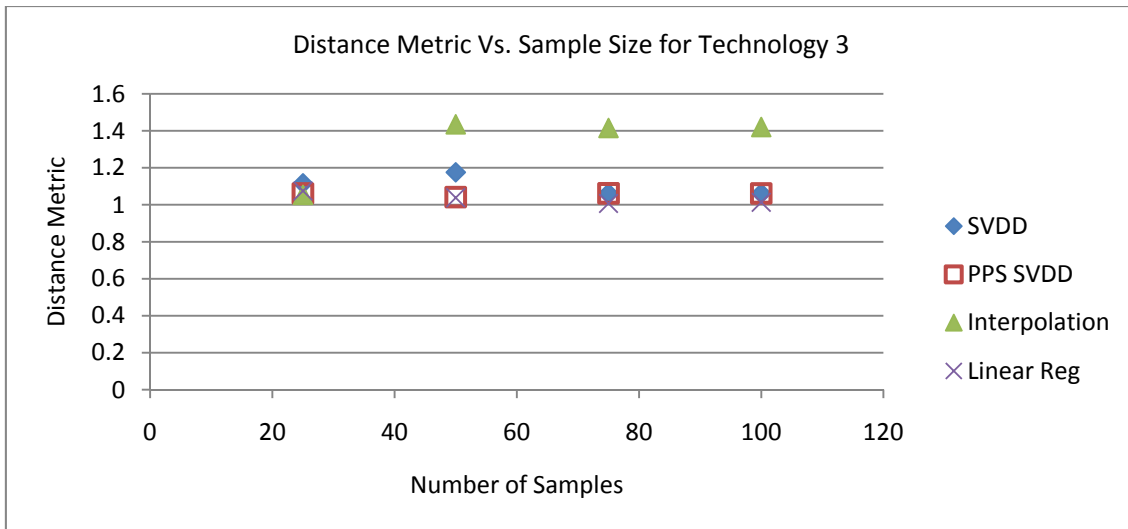


Figure 5-8 - Distance Metric for Different Sample Sizes (Tech 3)

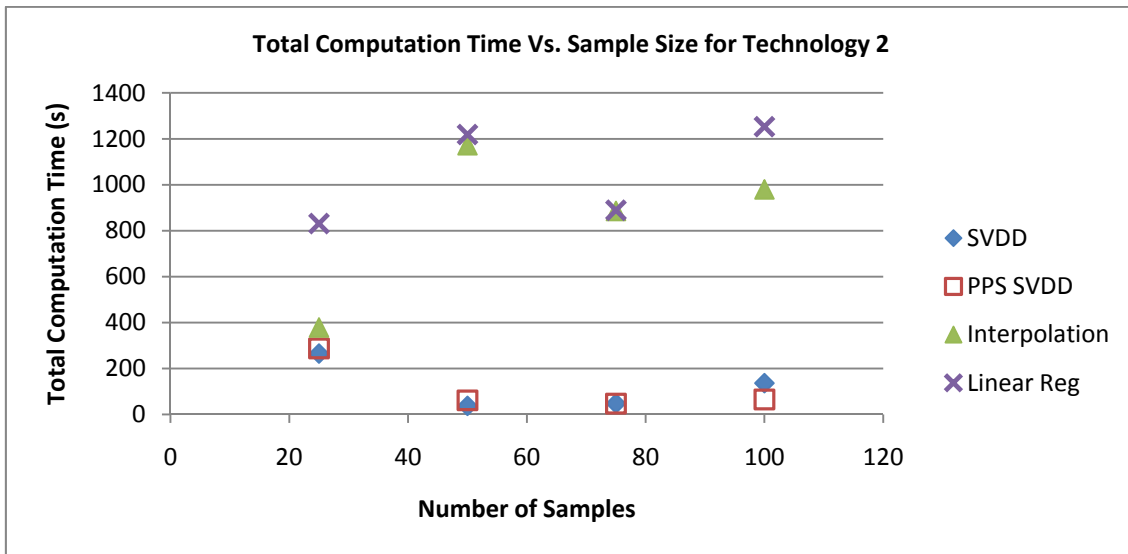


Figure 5-9 - Computation Time for Different Sample Sizes (Tech 2)

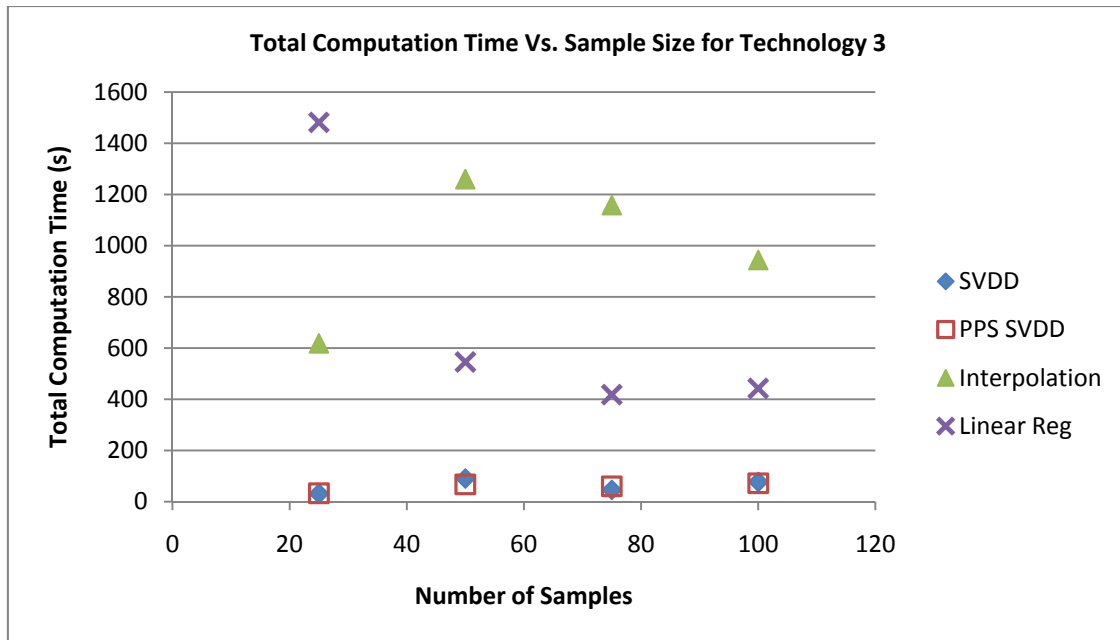


Figure 5-10 - Computation Time for Different Sample Sizes (Tech 3)

Qualitatively, both SVDD methods are superior because they produce consistent results (consistently find the same solution from different starting points), have shorter optimization times, and produce no unfeasible solutions. The interpolation and regression methods, at least in this example, occasionally provide infeasible solutions (negative values of the Volume attribute in this case), especially with smaller data sets. This is likely due to noise in the model fit that may contain sharp drops or increases along the fit surface. This is most pronounced in the 50-point dataset using interpolation, where out of the ten optimization runs, not a single one resulted in a feasible solution (all had negative predicted condenser volumes).

Unexpectedly, performing dominance analysis prior to fitting a SVDD model affected the solution quality. This result is possibly due to the fact that limiting the SVDD to the Pareto frontier reduced the number of solutions available to the optimizer and restricted its movement and prevented it from reaching better solutions. This example problem is

seven-dimensional, so trying to visually understand the domain and the Pareto frontier is difficult, making diagnosing the real cause of the loss of solution quality in the PPS+SVDD method challenging. Whatever the cause, the PPS+SVDD method was a very close second to the SVDD-only method and it may be reasonable to consider the PPS+SVDD method equal to the SVDD method for all intents and purposes.

Although this analysis and example problem point to the SVDD method of TCM being dominant, some of this result is problem-specific and this conclusion should not be viewed as definitive and general. However, this result does show that the SVDD method has great potential and that TCMs in general are useful and powerful tools in systems design.

## 6. CONCLUSIONS AND FUTURE WORK

In addition to the use of TCMs in systems design as technology selectors, I believe with further research they can be used in systems design problems for product design, technology development, set-based design and innovation. These potential uses are shown in Figure 6-1.

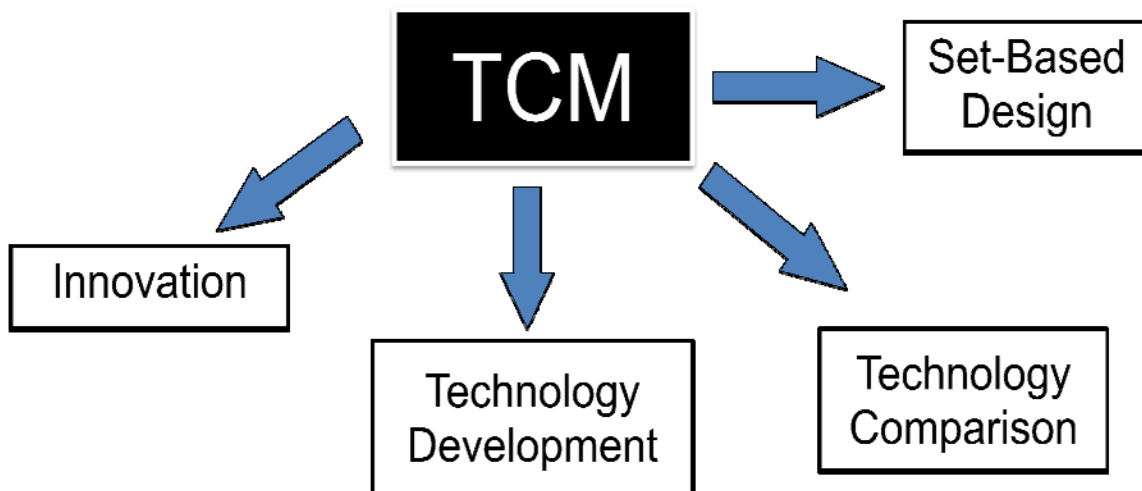


Figure 6-1 - Future Applications of TCMs

The continuous, set-based nature of a TCM lends itself to use as a feasible space for set-based design methods and a thorough study of this application is needed. Sobek, Ward, and Liker show that set-based concurrent design can be useful but they focus on simpler bounded variables (i.e. the diameter should be between 5 and 10 mm) to represent their sets instead of a richer, more complex set such as those determined by a TCM [15]. This added richness in the feasible domain description may allow designers of different sub-systems to better communicate their individual design capabilities and feasibilities.

Further study is also needed to determine the mechanics of finding TCM unions and intersections when two or more TCMs must be compared. These techniques would be useful in set-based design where different groups (manufacturing, engineering, testing, etc.) each develop their own TCMs or feasible sets and need to find the union or intersection sets of their designs so a design optimization can be performed.

Additionally, the fact that TCMs can be used to define a technology's current range of performance allows technology developers to search the TCM space to find performance regions that have not yet been explored, perhaps leading to improved designs. The TCM allows developers to drive the direction of technology development toward the untried designs within the feasible space for the technology. Figure 6-2 shows this process visually.

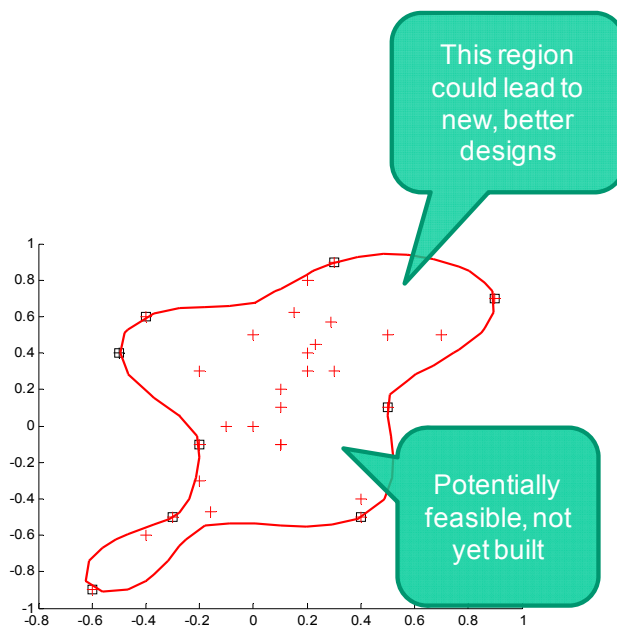


Figure 6-2 - Technology Development With TCMs

This application requires further study in the form of an industry partnership where TCMs can be tested on a real-world technology development problem.

I believe that TCMs are especially useful for technology comparisons as outlined earlier in this thesis. By abstracting individual technologies into a single space and fitting models to each, it allows systems designers to readily compare the performance of each technology and make decisions using their preferences on which technology is best for their system. An example of a simple technology comparison is shown in Figure 6-3.

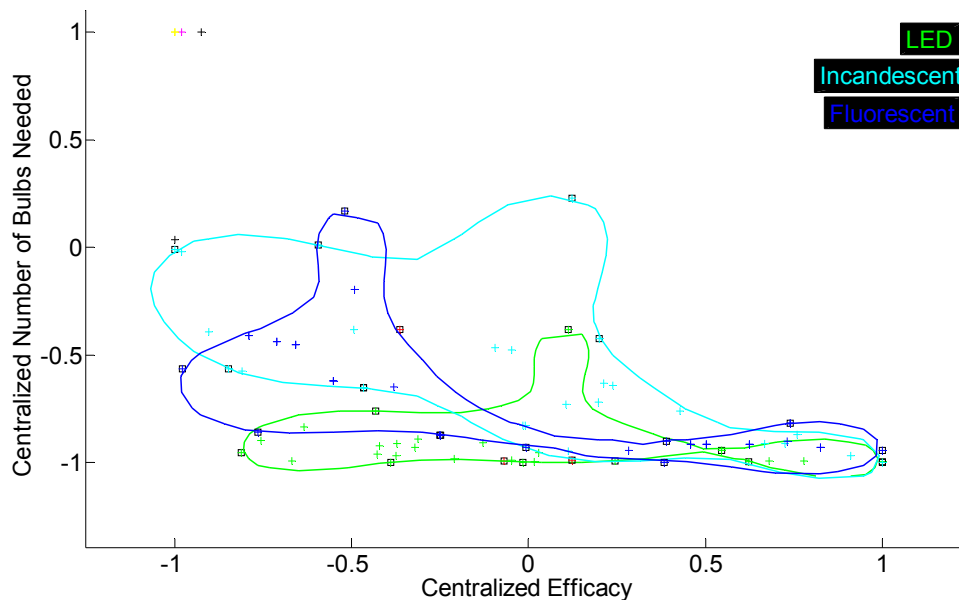


Figure 6-3 - Lighting Technology Comparison

Here, three lighting technologies are compared based on their efficacy and number of bulbs needed to light a certain size room. It is clear that each technology occupies a different part of the space and different technologies would be selected by designers with

different preferences. The TCMs allow the designers to make comparison decisions easily and directly among technologies with different design variables but the same attributes. It should be noted here that this method of abstraction does not work for all possible technologies because there are cases where several technologies that perform the same function may be too behaviorally different to abstract to the same attributes. For example, the electro-chromic windows mentioned in Section 1.4 can be abstracted with the photo-chromic ones, but not with standard windows because standard windows do not have dynamic behavior (their properties do not change with time). Though not uncommon, design problems like these are still in the minority. Future research is needed to extend current methods of abstraction to include a method for dealing with these exceptional problems.

I believe TCMs can be valuable tools to inventors because it may be possible to combine them with technology forecasting tools to project a feasible technology domain into the future and determine where new designs and products may lie in the design space to guide inventors in the innovation process. This too, would require a partnership with a company that strives for innovation to test this TCM methodology on a real technology through an industrial innovation process. Further, I believe that TCMs will mutually benefit systems designers and inventors in three ways: they allow for complete customer customization of products to suit their specific needs, they reduce limitations on customers' models and optimizers for component selection, and they encourage early adoption of new technologies by providing customers with a richer, more useful and flexible set of performance data. An interaction between a system designer and an inventor using TCMs may look like this (Figure 6-4):

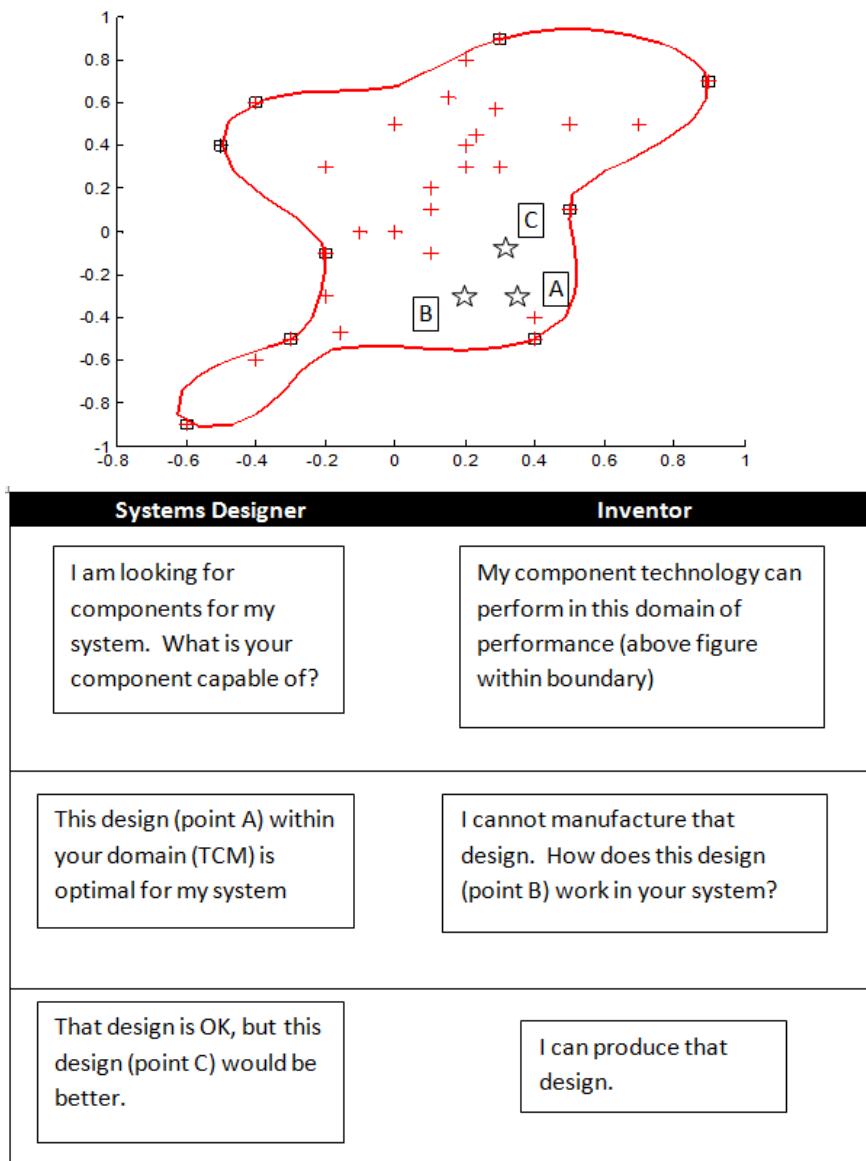


Figure 6-4 - Hypothetical System Designer-Inventor Interaction

In this figure, the practicalities of what designs can be manufactured are balanced with the optimal component designs needed for the system as a whole. This give-and-take accounts for the fact that in reality, the space of manufacturable designs for a product



may not be truly continuous (when costs of re-tooling, new equipment, etc. are considered), but a continuous TCM is still useful as a tool for the system designer and as a way to communicate overall feasibility and performance capabilities to systems designers. This figure shows the effects of *manufacturability* of designs, not to be confused with their *design* (behavioral engineering model) feasibility. Thus, TCMs can be used on products that are not practically customizable at every point within the domain by facilitating interaction and compromise between the inventor and the system designer.

Additional work is also needed to develop TCMs to use on design problems under uncertainty. Work is also needed on problems using different design scenarios and data sets to see if the results of this comparison study are consistent. I believe personal beliefs about the bias of a given inventor or product designer (because he or she is ultimately trying to sell his/her product and may distort information in his/her favor) may be important to study in a design problem because the TCM could be modified by the system designer to characterize this belief mathematically. This may result in a more accurate TCM because exaggerations of product performance can be accounted for.

I believe that TCMs represent an opportunity to communicate design information in a new and more efficient way that benefits both systems designers and inventors. With the future work above completed, the full range of TCM applications will be known and we will better understand their limitations and benefits in engineering design. My study will hopefully lay groundwork for further investigation into types of TCM, which type is the most useful, the future opportunities in design using TCMs, and the tools and techniques needed to use TCMs in real-world design situations.

## REFERENCES

- [1] Da Silveira, G., Borenstein, D., and Fogliatto, F.S., 2001, "Mass Customization: Literature Review and Research Directions," *International Journal of Production Economics*, **72**(1), pp. 1-13.
- [2] Denning, P.J. and Dunham, R., 2006, "Innovation As Language Action - By Learning Seven Foundational Practices, Anyone Can Become A Skillful Innovator," *Communications of the ACM*, **49**(5), pp. 47-52.
- [3] Rogers, E.M., 1983, *Diffusion of Innovations*, 3 ed., The Free Press, New York.
- [4] Orlikowski, W.J., 1992, "The Duality of Technology: Rethinking the Concept of Technology in Organizations," *Journal of Organizational Science*, **3**(3), pp. 398-427.
- [5] "Technology," [cited 2010 September 20], Merriam-Webster Online Dictionary; Available from: <http://www.merriam-webster.com/dictionary/technology>.
- [6] Rose, L.C. and Dugger Jr., W.E., 2002, "ITEA/Gallup Poll Reveals What Americans Think About Technology: A Report of the Survey Conducted by the Gallup Organization for the International Technology Education Association," *The Technology Teacher*, **61**(6), pp. 1-8.
- [7] Heidegger, M., 1977, *The Question of Technology and Other Essays*, Harper and Row, New York.
- [8] Ferguson, S., Gurnani, A., Donndelinger, J., and Lewis, K.E., 2005, "A Study of Convergence and Mapping in Preliminary Vehicle Design," *International Journal of Vehicle Systems Modelling and Testing*, **1**(1/2/3), pp. 192-215.
- [9] Gurnani, A., Ferguson, S., Lewis, K.E., and Donndelinger, J., 2006, "A Constraint-Based Approach to Feasibility Assessment in Preliminary Design," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **20**(4), pp. 351-367.

- [10] Mattson, C.A. and Messac, A., 2003, "Concept Selection Using S-Pareto Frontiers," *AIAA Journal*, **41**(6), pp. 1190-1198.
- [11] Mattson, C.A. and Messac, A., 2005, "Pareto Frontier Based Concept Selection Under Uncertainty With Visualization," *Optimization and Engineering*, **6**(1), pp. 85-115.
- [12] Malak, R.J. and Paredis, C.J.J., 2010, "Using Parameterized Pareto Sets to Model Design Concepts," *Journal of Mechanical Design*, **132**(4), pp. 0410071-04100711.
- [13] Malak, R.J. and Paredis, C.J.J., 2009, "Modeling Design Concepts under Risk and Uncertainty using Parameterized Efficient Sets," *Society of Automotive Engineers International Journal of Materials and Manufacturing*, **1**(1), pp. 339-352.
- [14] Malak, R.J., Tucker, L., and Paredis, C.J.J., 2008, "Composing Tradeoff Models for Multi-Attribute System-Level Decision Making," *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, New York, ASME Paper No. DETC2008-49970.
- [15] Sobek, D.K., Ward, A.C., and Liker, J.K., 1999, "Toyota's Principles of Set-based Concurrent Engineering," *Sloan Management Review*, **40**(2), pp. 67-83.
- [16] Malak, R.J., Jr., 2008, "Using Parameterized Efficient Sets to Model Alternatives for Systems Design Decisions", Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA.
- [17] Tax, D. and Duin, R., 1999, "Data Domain Description Using Support Vectors," *European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp. 251-256.
- [18] Malak, R.J. and Paredis, C.J.J., 2010, "Using Support Vector Machines to Formalize the Valid Input Domain of Predictive Models in Systems Design Problems," *Journal of Mechanical Design*, **132**(10), pp. 1010011-10100114.

- [19] Scholkopf, B. and Smola, J.A., 2002, *Learning with Kernels*, MIT Press, Cambridge, MA.
- [20] van Beers, W.C.M. and Kleijnen, J.P.C., 2004, "Kriging Interpolation in Simulation: A Survey," Proceedings of the 36th Conference on Winter Simulation, Washington, D.C., pp. 113-121.
- [21] Lophaven, S.N., Nielsen, H.B., and Sondergaard, J., 2002, "DACE: A Matlab Kriging Toolbox," Informatics and Mathematical Modelling Department, Technical University of Denmark, Lyngby, Denmark, Available from: <http://www2.imm.dtu.dk/~hbn/dace/>.
- [22] Chapra, S.C. and Canale, R.P., 2010, *Numerical Methods for Engineers*, 6th ed., McGraw Hill Higher Education, New York.
- [23] Jun, S.-H. and Oh, K.-W., 2006, "A Competitive Co-evolving Support Vector Clustering," in *Neural Information Processing*, I. King, J. Wang, L. Chan, D. Wang, Eds., Lecture Notes in Computer Science, Springer Berlin, Berlin, Germany, pp. 864-873.
- [24] Lee, S.-H. and Daniels, K., 2005, "Gaussian Kernel Width Generator for Support Vector Clustering," in *Advances in Bioinformatics and Its Applications*, I.M. He, G. Narasimhan, and S. Petoukhov, Eds., World Scientific, Fort Lauderdale, FL, pp. 151-162.
- [25] Ben-Hur, A., Horn, D., Siegelmann, H.T., and Vapnik, V., 2002, "Support Vector Clustering," *Journal of Machine Learning Research*, **2**(2), pp. 125-137.
- [26] Shah, R.K. and Sekulic, D.P., 2003, *Fundamentals of Heat Exchanger Design*, John Wiley and Sons, Inc., Hoboken, NJ.
- [27] Moran, M.J. and Shapiro, H.N., eds. 2004, *Fundamentals of Engineering Thermodynamics*, 5 ed., John Wiley & Sons, Inc., Hoboken, NJ.
- [28] "Steam Pipe Sizing," [cited 2010 November 10], The Engineering Toolbox; Available from: [http://www.engineeringtoolbox.com/steam-pipe-sizing-d\\_105.html](http://www.engineeringtoolbox.com/steam-pipe-sizing-d_105.html).

- [29] Thurston, D.L., 1991, "A Formal Method for Subjective Design Evaluation with Multiple Attributes," *Research in Engineering Design*, **3**(2), pp. 105-122.
- [30] Keeney, R. and Raiffa, H., 1993, *Decisions With Multiple Objectives: Preferences and Value Tradeoffs*, 2 ed., Cambridge University Press, Cambridge, United Kingdom.
- [31] Holmgren, M., 2006, "X Steam for Matlab," Available from: <http://www.x-eng.com>.
- [32] Incropera, Dewitt, Bergman, and Lavine, eds. 2007, *Fundamentals of Heat and Mass Transfer*, 6 ed., John Wiley & Sons, Hoboken, NJ.
- [33] McCabe, W.L., Smith, J.C., and Harriott, P., 1993, *Unit Operations of Chemical Engineering*, 5 ed., McGraw-Hill, Singapore.
- [34] Singh, P., Walker, J., Lee, H.S., Gharfeh, S., Thomason, B., and Blumer, D., 2007, "An Application of Vacuum-insulated Tubing (VIT) for Wax Control in an Arctic Environment," *SPE Drilling & Completion*, **22**(2), pp. 127-136.
- [35] Petukhov, B.S. and Roizen, L.I., 1964, "Generalized Relationships for Heat Transfer in a Turbulent Flow of Gas in Tubes of Annular Section," *High Temperature*, **2**, pp. 65-68.
- [36] Kreider, J.F., Curtiss, P.S., and Rabl, A., 2010, *Heating and Cooling of Buildings Design for Efficiency*, 2 ed., Mechanical Engineering Series, F. Kreith and R. Mahajan, eds., CRC Press, Boca Raton, FL.
- [37] "Resistance and Equivalent Length of Fittings," [cited 2010 November 16], The Engineering Toolbox; Available from: [http://www.engineeringtoolbox.com/resistance-equivalent-length-d\\_192.html](http://www.engineeringtoolbox.com/resistance-equivalent-length-d_192.html).
- [38] Booker, A.J., Meckesheimer, M., and Torng, T., 2004, "Reliability Based Design Optimization Using Design Explorer," *Optimization and Engineering*, **5**(2), pp. 179-205.
- [39] Roach, E., Parker, R., and Malak, R.J., 2011, under review, "An Improved Support Vector Domain Description Method for Modeling the Valid Domain of Predictive Engineering Models," *ASME International Design Technical*

Conferences and Computers and Information in Engineering Conference,  
Washington, D.C.

## APPENDIX

Table A-1 - Results of Optimization of Condenser Design

**Optimization results (tech1)**

Technique	Data Size	delta_p	hxeff	Volume	Wdotpwater	P2	Asicross	Aicross
SVDD	25	192.214	0.830	8.472	0.584	0.975	0.098	0.074
PPS SVDD	25	308.459	0.812	24.719	1.576	0.933	0.155	0.124
Interp	25	29.940	0.940	5.036	0.421	0.848	2.211	1.493
Linear Reg	25	308.107	0.898	127.970	0.421	0.898	1.015	1.009
SVDD	50	1374.281	1.000	4.630	0.974	0.733	0.042	0.044
PPS SVDD	50	232.508	0.787	14.922	1.195	0.942	0.158	0.159
Interp	50	528.624	0.836	-5.222	0.421	0.826	0.045	0.045
Linear Reg	50	254.359	0.895	127.878	0.421	0.781	1.101	1.100
SVDD	75	231.422	0.996	19.019	2.568	0.740	0.223	0.187
PPS SVDD	75	1046.815	0.953	1.206	0.194	0.782	0.006	0.002
Interp	75	308.665	0.938	131.007	0.194	0.745	0.912	0.912
Linear Reg	75	341.991	0.870	128.447	0.194	0.887	0.834	0.833
SVDD	Full Set	241.240	0.957	22.212	2.432	0.819	0.195	0.163
PPS SVDD	Full Set	2714.341	0.953	10.512	1.184	0.856	0.030	0.023
Interp	Full Set	342.069	0.871	128.144	0.194	0.888	0.855	0.855
Linear Reg	Full Set	238.525	0.956	127.876	0.265	0.749	1.188	1.149
AAO	Full Set	46.749	0.997	3.971	0.574	0.668	0.237	0.196

Table A-1 continued

**Optimization results (tech2)**

Technique	Data Size	delta_p	hxeff	Volume	Wdotpwater	P2	Asicross	Aicross
SVDD	25	181.736	0.954	10.114	0.841	0.822	0.131	0.093
PPS SVDD	25	181.739	0.954	10.113	0.841	0.823	0.130	0.093
Interp	25	46.274	0.889	0.365	0.481	0.882	0.404	0.404
Linear Reg	25	158.908	0.955	127.890	7.325	0.746	1.994	1.330
SVDD	50	46.193	0.999	3.611	0.481	0.667	0.186	0.175
PPS SVDD	50	72.636	0.915	22.551	1.913	0.978	0.559	0.539
Interp	50	141.863	0.818	127.891	0.138	0.727	1.670	1.349
Linear Reg	50	125.734	0.828	130.333	0.138	0.810	2.747	0.066
SVDD	75	273.633	1.000	5.235	0.505	0.667	0.038	0.029
PPS SVDD	75	273.633	1.000	5.235	0.505	0.667	0.038	0.029
Interp	75	164.217	0.724	128.080	0.138	0.763	1.426	0.674
Linear Reg	75	716.014	0.946	127.898	0.138	0.686	0.472	0.472
SVDD	Full Set	181.735	0.947	10.111	0.841	0.892	0.108	0.092
PPS SVDD	Full Set	168.087	1.000	4.174	0.724	0.669	0.381	0.336
Interp	Full Set	224.507	0.856	127.891	0.138	0.940	1.121	0.757
Linear Reg	Full Set	63.862	0.899	0.536	0.138	0.939	0.009	0.009
AAO	Full Set	189.951	1.000	2.002	0.389	0.668	0.293	0.212



Table A-1 continued

**Optimization results (tech3)**

Technique	Data Size	delta_p	hxeff	Volume	Wdotpwater	P2	Asicross	Aicross
SVDD	25	766.756	1.000	28.430	21.624	0.719	0.028	0.0215
PPS SVDD	25	3175.732	0.998	3.985	1.680	0.774	0.004	0.0029
Interp	25	1300.449	0.998	2.027	0.818	0.727	0.004	0.0038
Linear Reg	25	517.872	0.999	0.532	0.818	0.723	0.003	0.0029
SVDD	50	3553.234	0.995	7.910	11.747	0.915	0.002	0.0008
PPS SVDD	50	198.721	0.997	13.106	3.297	0.787	0.208	0.0950
Interp	50	25.251	0.998	3796.069	0.818	0.699	0.017	0.0172
Linear Reg	50	1501.159	0.998	1.467	0.818	0.701	0.001	0.0008
SVDD	75	3175.732	0.998	3.985	1.680	0.774	0.003	0.0036
PPS SVDD	75	3175.732	0.998	3.985	1.680	0.774	0.003	0.0036
Interp	75	6.396	0.996	944.412	0.818	0.699	0.011	0.0111
Linear Reg	75	377.702	0.996	3.748	47.979	0.699	0.001	0.0012
SVDD	Full Set	3175.732	0.998	3.985	1.680	0.774	0.005	0.0024
PPS SVDD	Full Set	3175.732	0.998	3.985	1.680	0.774	0.005	0.0024
Interp	Full Set	97.426	0.996	8947.095	0.818	0.694	0.041	0.0408
Linear Reg	Full Set	1501.158	0.996	1.415	0.818	0.693	0.001	0.0008
AAO	Full Set	568.593	0.996	7.854	372.914	0.730	4.870	0.0001

Table A-2 - Utility Values of Optimization Results

Technique	Utility Values Tech1						Utility Values Tech2				
	Data Size	eff util	Tc util	Vol util	Wdotp Util	Tot. Util	eff util	Tc util	Vol util	Wdotp Util	Tot. Util
SVDD	25	0.783	0.948	0.915	0.981	0.907	0.760	0.946	0.899	0.972	0.894
PPS SVDD	25	0.785	0.954	0.632	0.947	0.830	0.760	0.946	0.899	0.972	0.894
Interp	25	0.769	0.941	0.950	0.986	0.911	0.783	0.961	0.996	0.984	0.931
Linear Reg	25	0.780	0.960	1.000	0.986	0.931	0.773	0.947	1.000	0.756	0.869
SVDD	50	0.699	0.967	0.954	0.968	0.897	0.750	0.967	0.964	0.984	0.916
PPS SVDD	50	0.782	0.964	0.852	0.960	0.890	0.765	0.955	0.686	0.936	0.836
Interp	50	0.791	0.965	1.000	0.986	0.935	0.800	0.961	1.000	0.995	0.939
Linear Reg	50	0.788	0.964	1.000	0.986	0.934	0.793	1.000	1.000	0.995	0.947
SVDD	75	0.718	0.960	0.775	0.914	0.842	0.747	0.959	0.948	0.983	0.909
PPS SVDD	75	0.766	0.869	0.988	0.994	0.904	0.747	0.959	0.948	0.983	0.909
Interp	75	0.780	0.964	1.000	0.994	0.934	0.782	0.940	1.000	0.995	0.929
Linear Reg	75	0.785	0.962	1.000	0.994	0.935	0.782	0.966	1.000	0.995	0.936
SVDD	Full	0.757	0.955	0.695	0.919	0.831	0.747	0.951	0.899	0.972	0.892
PPS SVDD	Full	0.726	0.947	0.895	0.961	0.882	0.747	0.964	0.958	0.976	0.911
Interp	Full	0.785	0.962	1.000	0.994	0.935	0.783	0.942	1.000	0.995	0.930
Linear Reg	Full	0.772	0.963	0.999	0.991	0.931	0.776	0.958	0.995	0.995	0.931
AAO						0.914					0.927

Technique	Utility Values Tech3						Utility Rank Order Winner		Name
	Data Size	eff util	Tc util	Vol util	Wdotp Util	Tot. Util	Max Util		
SVDD	25	0.715	0.957	0.539	0.279	0.623	0.907	tech1	
PPS SVDD	25	0.656	0.954	0.960	0.944	0.879	0.894	tech2	
Interp	25	0.706	0.964	0.980	0.973	0.906	0.931	tech2	
Linear Reg	25	0.717	0.964	0.995	0.973	0.912	0.931	tech1	
SVDD	50	0.606	0.916	0.921	0.608	0.763	0.916	tech2	
PPS SVDD	50	0.690	0.922	0.869	0.890	0.843	0.890	tech1	
Interp	50	0.739	0.967	1.000	0.973	0.920	0.939	tech2	
Linear Reg	50	0.719	0.956	0.985	0.973	0.908	0.947	tech2	
SVDD	75	0.656	0.969	0.960	0.944	0.882	0.909	tech2	
PPS SVDD	75	0.656	0.969	0.960	0.944	0.882	0.909	tech2	
Interp	75	0.741	0.967	1.000	0.973	0.920	0.934	tech1	
Linear Reg	75	0.738	0.960	0.963	0.033	0.673	0.936	tech2	
SVDD	Full Set	0.656	0.924	0.960	0.944	0.871	0.892	tech2	
PPS SVDD	Full Set	0.656	0.924	0.960	0.944	0.871	0.911	tech2	
Interp	Full Set	0.744	0.967	1.000	0.973	0.921	0.935	tech1	
Linear Reg	Full Set	0.727	0.957	0.986	0.973	0.911	0.931	tech1	
AAO						0.640	0.9271	tech2	

Table A-3 - Gaussian Width Parameters ( $q$ ) Used in This Research (Max value such that data is contained in one SV cluster)

	Data Size			
Technology	25	50	75	100
Tech 1	0.9359	0.7799	1.2478	1.2647
Tech 2	0.585	0.6825	1.3842	1.4361
Tech 3	0.5265	0.702	1.1113	1.0932
C values are always kept at 0.99				

Table A-4 - Number of Support Vectors by TCM, Technology and Data Size

TCM	Data Size	Tech 1	Tech 2	Tech 3
SVDD	25	19	14	11
PPS SVDD	25	18	12	11
Interpolation	25	16	10	9
Regression	25	16	10	9
SVDD	50	19	19	19
PPS SVDD	50	19	21	18
Interpolation	50	19	18	13
Regression	50	19	18	13
SVDD	75	36	35	28
PPS SVDD	75	34	35	27
Interpolation	75	31	31	21
Regression	75	31	31	21

Table A-5 - 100 Designs Used to Generate TCM Attribute Data (Units in Meters)

Tech 1					Tech 2					Tech 3						
Do	Di	Dso	Dsi	L	Do	Di	Dso	Dsi	L	Do	Di	Dso	Dsi	L		
0.51	0.46	0.67	0.61	19.66		1.42	1.29	1.61	1.47	8.31		0.19	0.17	2.50	2.30	19.53
1.07	0.97	1.22	1.12	19.75		0.46	0.41	1.81	1.66	9.76		0.22	0.20	0.35	0.32	13.31
0.27	0.24	0.31	0.28	2.84		0.13	0.11	0.63	0.58	11.69		0.22	0.20	1.33	1.22	7.87
1.63	1.47	2.50	2.30	12.63		1.07	0.97	1.99	1.82	16.33		0.04	0.03	0.76	0.69	9.70
0.36	0.32	0.40	0.37	3.66		0.27	0.24	0.79	0.72	11.17		0.02	0.01	0.52	0.47	17.57
1.42	1.29	1.57	1.43	4.96		1.63	1.47	1.83	1.68	7.68		0.06	0.05	1.00	0.91	14.86
0.81	0.73	1.11	1.01	16.46		0.22	0.20	0.68	0.62	8.82		0.19	0.17	1.31	1.19	16.17
1.22	1.10	1.83	1.67	2.46		0.71	0.64	2.31	2.11	8.42		0.10	0.09	1.21	1.10	4.92
1.02	0.92	1.66	1.52	18.87		0.41	0.37	0.46	0.42	12.09		0.03	0.02	0.30	0.28	5.57
0.51	0.46	1.65	1.51	4.68		1.83	1.66	2.33	2.13	3.82		0.02	0.01	0.24	0.22	17.39
0.46	0.41	0.56	0.51	13.87		0.04	0.03	0.22	0.19	2.68		0.27	0.24	1.23	1.12	14.89
0.17	0.15	1.01	0.92	7.96		1.02	0.92	2.50	2.30	11.71		0.04	0.03	0.22	0.19	11.65
0.76	0.69	1.16	1.06	5.97		1.73	1.57	2.50	2.30	19.77		0.02	0.01	1.34	1.22	10.43
0.41	0.37	1.21	1.11	14.75		0.36	0.32	1.32	1.21	16.46		0.17	0.15	0.41	0.37	13.22
0.86	0.78	2.26	2.06	4.23		0.66	0.60	0.78	0.71	2.08		0.66	0.60	2.50	2.30	14.12
1.73	1.57	2.22	2.03	11.61		2.03	1.84	2.50	2.30	12.77		0.66	0.60	1.65	1.51	17.83
2.24	2.04	2.50	2.30	15.04		0.51	0.46	1.32	1.21	5.43		0.19	0.17	0.60	0.54	8.01
0.41	0.37	0.83	0.76	19.41		0.10	0.09	0.33	0.30	5.01		0.41	0.37	1.27	1.16	14.61
1.22	1.10	1.99	1.82	5.54		0.46	0.41	1.10	1.01	1.27		0.02	0.01	0.63	0.57	1.50
0.66	0.60	2.50	2.30	12.33		0.76	0.69	1.00	0.91	2.07		0.10	0.09	1.04	0.95	11.05
0.22	0.20	1.17	1.07	19.16		1.12	1.01	1.77	1.62	17.58		0.04	0.03	1.80	1.64	16.51
0.91	0.83	1.82	1.66	9.19		1.32	1.20	2.26	2.07	12.78		0.32	0.29	2.32	2.12	2.74
1.22	1.10	2.50	2.30	16.91		0.36	0.32	0.88	0.81	10.46		0.27	0.24	2.05	1.87	1.47
0.27	0.24	0.79	0.72	4.76		2.24	2.04	2.49	2.29	3.41		0.02	0.01	0.46	0.42	4.87
0.41	0.37	0.50	0.46	14.74		1.12	1.01	2.50	2.30	4.84		0.05	0.04	0.39	0.35	1.82
0.17	0.15	0.90	0.82	16.27		2.24	2.04	2.50	2.30	1.00		0.06	0.05	1.16	1.06	15.68
1.02	0.92	1.66	1.52	14.06		0.41	0.37	1.54	1.41	7.39		0.71	0.64	1.87	1.71	19.36
0.32	0.29	1.12	1.02	6.29		0.27	0.24	1.23	1.12	11.88		0.11	0.10	1.44	1.31	18.25
1.63	1.47	2.50	2.30	12.40		1.93	1.75	2.48	2.28	5.29		0.19	0.17	1.96	1.79	5.49
0.14	0.13	0.38	0.34	15.65		0.51	0.46	0.94	0.86	16.51		0.22	0.20	2.50	2.30	10.95
0.27	0.24	0.47	0.42	8.95		1.07	0.97	2.50	2.30	11.60		0.17	0.15	2.50	2.30	5.28
1.12	1.01	2.32	2.12	17.89		2.24	2.04	2.50	2.30	2.01		0.13	0.11	1.89	1.73	19.42
1.83	1.66	2.48	2.28	12.35		0.32	0.29	0.90	0.82	6.43		0.36	0.32	2.50	2.30	4.48
1.63	1.47	1.94	1.78	15.24		0.13	0.11	0.52	0.48	5.20		0.02	0.01	0.46	0.42	3.44
1.52	1.38	2.50	2.30	3.23		1.32	1.20	2.50	2.30	7.17		0.03	0.02	0.85	0.78	6.03
0.51	0.46	0.83	0.76	7.29		0.91	0.83	1.06	0.96	12.28		0.07	0.06	0.68	0.62	7.18
1.63	1.47	2.50	2.30	8.91		1.02	0.92	2.50	2.30	4.88		0.05	0.04	0.33	0.30	13.36
1.52	1.38	2.50	2.30	12.78		0.76	0.69	0.89	0.81	19.88		0.06	0.05	1.82	1.66	10.49
1.07	0.97	1.60	1.47	2.16		1.52	1.38	2.10	1.92	14.82		0.07	0.06	0.63	0.57	2.38
1.22	1.10	2.50	2.30	6.29		0.17	0.15	0.30	0.27	10.34		0.07	0.06	0.14	0.12	9.22
0.41	0.37	0.50	0.46	11.48		1.02	0.92	1.33	1.22	6.43		0.10	0.09	0.71	0.65	17.69
1.52	1.38	1.72	1.57	15.11		1.52	1.38	2.50	2.30	18.01		0.46	0.41	2.46	2.26	3.55
1.73	1.57	2.50	2.30	11.44		0.71	0.64	1.43	1.31	1.29		0.10	0.09	2.50	2.30	19.15
1.07	0.97	1.28	1.17	6.20		1.93	1.75	2.50	2.30	8.30		0.02	0.01	1.55	1.42	13.50
1.02	0.92	1.93	1.77	5.22		0.66	0.60	1.49	1.36	1.39		0.02	0.01	0.57	0.52	3.57
2.24	2.04	2.50	2.30	15.47		0.61	0.55	0.78	0.71	7.07		0.05	0.04	0.66	0.60	18.06
0.17	0.15	0.41	0.37	18.55		1.52	1.38	2.32	2.12	12.72		0.14	0.13	1.25	1.14	10.96
0.10	0.09	0.44	0.40	4.17		0.66	0.60	1.05	0.96	16.99		0.09	0.08	0.86	0.79	1.00
0.04	0.03	0.16	0.14	19.64		0.17	0.15	0.57	0.52	5.65		0.04	0.03	0.11	0.09	6.22
0.24	0.22	0.87	0.79	12.32		0.17	0.15	0.41	0.37	15.89		0.07	0.06	0.85	0.77	15.86

Table A-5 continued

Do	Di	Dso	Dsi	L		Do	Di	Dso	Dsi	L		Do	Di	Dso	Dsi	L
1.73	1.57	2.50	2.30	3.21		1.63	1.47	2.38	2.18	10.44		0.71	0.64	2.50	2.30	19.80
1.63	1.47	1.83	1.68	15.67		0.71	0.64	1.27	1.16	15.07		0.36	0.32	1.16	1.06	6.46
1.42	1.29	2.50	2.30	8.92		1.93	1.75	2.50	2.30	7.41		0.46	0.41	1.32	1.21	5.56
1.93	1.75	2.17	1.98	16.82		0.24	0.22	0.49	0.44	9.14		0.07	0.06	0.63	0.57	10.77
0.32	0.29	0.52	0.47	14.50		1.73	1.57	2.16	1.98	10.52		0.22	0.20	1.55	1.42	1.47
1.12	1.01	2.50	2.30	19.51		2.24	2.04	2.50	2.30	7.28		0.10	0.09	1.64	1.50	6.00
0.41	0.37	0.72	0.66	14.67		0.81	0.73	1.11	1.01	10.58		0.04	0.03	0.75	0.69	19.17
0.32	0.29	0.96	0.87	6.65		1.73	1.57	2.00	1.83	19.75		0.05	0.04	0.39	0.35	5.74
0.13	0.11	0.52	0.48	11.60		1.52	1.38	1.94	1.77	19.10		0.14	0.13	0.54	0.49	19.27
1.07	0.97	2.50	2.30	16.13		1.22	1.10	1.88	1.72	4.59		0.19	0.17	1.53	1.39	2.28
0.76	0.69	2.50	2.30	8.56		1.02	0.92	1.12	1.03	1.18		0.71	0.64	2.50	2.30	18.14
0.17	0.15	0.46	0.42	19.01		1.83	1.66	2.22	2.03	4.77		0.10	0.09	0.39	0.35	15.54
1.52	1.38	2.27	2.07	4.49		1.83	1.66	2.50	2.30	10.85		0.07	0.06	0.14	0.12	18.79
0.66	0.60	1.32	1.21	8.41		0.41	0.37	1.21	1.11	17.71		0.32	0.29	0.85	0.77	7.16
0.46	0.41	1.32	1.21	13.52		1.63	1.47	2.16	1.98	2.43		0.41	0.37	0.83	0.76	4.93
0.14	0.13	0.81	0.74	18.64		0.10	0.09	0.28	0.25	19.70		0.03	0.02	0.85	0.78	11.97
0.19	0.17	0.82	0.74	12.58		0.81	0.73	2.20	2.01	5.20		0.04	0.03	0.75	0.69	1.21
0.27	0.24	0.31	0.28	11.39		0.46	0.41	0.83	0.76	16.61		0.27	0.24	1.07	0.97	7.52
1.32	1.20	2.16	1.97	8.57		2.24	2.04	2.50	2.30	9.50		0.06	0.05	0.89	0.81	11.84
0.91	0.83	1.93	1.76	13.34		1.83	1.66	2.05	1.88	14.92		0.02	0.01	0.73	0.67	17.94
1.32	1.20	1.50	1.37	7.25		1.02	0.92	1.88	1.72	2.01		0.04	0.03	0.10	0.09	14.82
1.07	0.97	2.47	2.27	5.42		1.07	0.97	1.88	1.72	2.08		0.10	0.09	0.93	0.85	5.52
1.73	1.57	2.50	2.30	12.42		0.86	0.78	1.00	0.91	12.74		0.19	0.17	0.98	0.89	12.32
1.22	1.10	2.42	2.22	15.82		1.07	0.97	1.71	1.57	12.98		0.09	0.08	2.28	2.09	13.31
0.71	0.64	1.00	0.91	14.78		0.19	0.17	1.14	1.04	10.79		0.02	0.01	2.50	2.30	19.21
0.61	0.55	0.83	0.76	14.90		0.81	0.73	2.50	2.30	15.13		0.27	0.24	0.68	0.62	10.51
0.76	0.69	2.36	2.16	13.20		1.07	0.97	2.10	1.92	1.44		0.41	0.37	1.54	1.41	9.13
1.42	1.29	1.57	1.43	15.74		1.63	1.47	2.48	2.28	6.90		0.22	0.20	1.01	0.92	17.53
1.42	1.29	1.99	1.82	3.90		0.05	0.04	0.22	0.20	8.52		0.13	0.11	0.96	0.88	17.80
2.03	1.84	2.50	2.30	1.97		0.81	0.73	2.20	2.01	11.97		0.46	0.41	0.83	0.76	16.97
0.19	0.17	0.43	0.39	15.57		0.66	0.60	1.54	1.41	6.08		0.61	0.55	1.76	1.61	10.89
1.42	1.29	1.67	1.52	15.31		0.76	0.69	1.76	1.61	17.16		0.17	0.15	0.57	0.52	15.90
0.36	0.32	0.77	0.71	9.14		0.32	0.29	0.68	0.62	3.61		0.51	0.46	0.94	0.86	11.33
0.27	0.24	0.90	0.82	12.23		0.61	0.55	1.11	1.01	17.45		0.07	0.06	0.57	0.52	17.87
0.61	0.55	1.11	1.01	8.90		1.02	0.92	1.12	1.03	13.00		0.03	0.02	0.25	0.23	12.39
2.03	1.84	2.50	2.30	19.90		0.24	0.22	0.60	0.54	5.34		0.17	0.15	0.46	0.42	2.09
1.22	1.10	1.61	1.47	6.66		0.36	0.32	0.83	0.76	6.39		0.17	0.15	2.50	2.30	5.55
2.24	2.04	2.50	2.30	1.65		0.66	0.60	0.78	0.71	5.63		0.03	0.02	0.69	0.63	1.74
0.14	0.13	0.65	0.59	12.13		1.07	0.97	1.82	1.67	14.94		0.05	0.04	1.59	1.45	18.20
0.05	0.04	0.11	0.10	3.39		0.24	0.22	0.71	0.64	6.65		0.32	0.29	1.29	1.17	1.60
0.91	0.83	1.55	1.41	14.75		1.02	0.92	1.44	1.32	9.39		0.51	0.46	0.94	0.86	12.50
0.76	0.69	1.65	1.51	3.61		0.81	0.73	1.49	1.36	4.58		0.51	0.46	1.38	1.26	5.97
0.71	0.64	1.27	1.16	3.29		0.71	0.64	1.38	1.26	7.62		0.36	0.32	2.50	2.30	3.53
0.71	0.64	1.33	1.21	15.39		0.91	0.83	1.11	1.01	4.61		0.03	0.02	2.50	2.30	19.20
2.03	1.84	2.50	2.30	19.94		0.46	0.41	1.16	1.06	16.92		0.05	0.04	0.66	0.60	1.92
0.81	0.73	1.49	1.36	5.37		0.46	0.41	0.78	0.71	1.48		0.10	0.09	0.77	0.70	16.49
0.86	0.78	2.04	1.86	13.53		0.46	0.41	1.32	1.21	10.13		0.17	0.15	0.51	0.47	2.09
2.24	2.04	2.50	2.30	2.82		0.71	0.64	1.71	1.56	11.45		0.13	0.11	0.96	0.88	18.23
0.71	0.64	1.05	0.96	17.84		0.71	0.64	1.54	1.41	16.75		0.24	0.22	0.82	0.74	16.80
0.46	0.41	1.16	1.06	17.56		0.46	0.41	1.43	1.31	14.79		0.04	0.03	0.54	0.49	8.52

Table A-6 - Support Vectors (and Coeff.) Used for Full Data Set by Technology and TCM

SVDD Tech 1 xSV								SVDD Tech 2 xSV								bSV	
Element								Element								Coeff.	
1	2	3	4	5	6	7	Coeff.	1	2	3	4	5	6	7	Coeff.		
-0.4534	0.9988	-0.0391	0.1685	-0.6943	-0.9573	-0.5504	0.0240	-0.9741	-0.3622	-0.2993	-0.6717	-0.6854	0.0946	-0.9187	0.0245		
-0.5096	0.9991	-0.9797	-0.9603	-0.8437	-1.0000	-0.9717	0.0298	-0.8070	1.0000	-0.4471	-0.2694	-0.7967	-0.9321	0.0472	0.0141		
-0.7897	1.0000	-0.7009	-0.6421	0.1954	-0.9905	-0.1989	0.0195	-0.9820	-0.4908	-0.2282	-0.5948	0.3002	0.7066	-0.8012	0.0448		
-0.9110	0.9811	0.2500	0.0720	-0.5416	-0.4801	-0.5924	0.0106	1.0000	1.0000	-0.7970	-0.6273	-0.7967	-1.0000	-0.9360	0.0609		
-0.9948	-0.7361	-0.7033	-0.8626	-0.0133	-0.1707	-0.8993	0.0208	-0.8698	0.5246	-1.0000	-1.0000	0.8382	-0.9880	-1.0000	0.0249		
-0.3516	0.9969	-0.7030	-0.5937	-1.0000	-0.9824	-0.9187	0.0163	-0.9067	1.0000	1.0000	1.0000	-0.7967	-0.0052	0.1826	0.0292		
-0.9714	-1.0000	-0.6919	-0.9066	-0.5066	-0.6661	-0.9898	0.0531	-0.8936	1.0000	0.2839	0.4709	-0.7967	-0.5011	0.6381	0.0017		
-1.0000	-0.5429	-0.6295	-0.8037	-0.5294	0.4467	-0.7060	0.0281	-0.8686	0.9994	-0.9555	-0.9601	-1.0000	-0.9687	-0.9968	0.0118		
-0.7929	0.9994	0.5043	0.6324	-0.8168	-0.8811	1.0000	0.0402	-0.9965	-0.8031	-0.9665	-0.9704	-0.5355	-0.6550	-0.9187	0.0288		
-0.9874	-0.5474	0.2311	-0.5133	-0.1091	1.0000	-0.8288	0.0450	-0.8933	1.0000	0.2505	0.2457	-0.7967	-0.4117	-0.5063	0.0039		
-0.9214	0.1125	-0.1085	-0.6884	0.9227	-0.5506	-0.9821	0.0364	-0.9918	0.1529	-0.5282	-0.6679	-0.5667	0.7453	-0.5063	0.0347		
-0.9581	0.9519	0.6593	0.0976	0.6052	0.3323	-0.5063	0.0318	-0.9877	1.0000	-0.9213	-0.8887	-0.7967	-0.8764	1.0000	0.0553		
-0.9515	0.9919	0.2224	0.1292	-0.7358	-0.2398	0.3262	0.0060	-0.9364	0.0458	-0.4255	-0.7360	0.9425	-0.4900	-0.9717	0.0252		
-0.8440	0.9975	0.1810	0.2682	-0.9997	-0.7918	0.0472	0.0216	-0.9785	0.9358	-0.2901	-0.4271	1.0000	0.5309	-0.3096	0.0429		
-0.9984	0.8366	-0.6890	-0.7595	0.7878	0.2218	-0.0799	0.0413	-0.9927	-0.2788	-0.5245	-0.6911	-0.5638	0.8392	-0.5924	0.0139		
-0.9922	0.6432	-0.3796	-0.5974	0.8887	0.5667	-0.4121	0.0094	-0.0962	1.0000	-0.2994	0.0334	-0.7967	-0.9695	-0.7715	0.0411		
-0.9630	0.9877	0.1410	-0.0036	-0.3202	-0.0507	0.1826	0.0100	-0.9354	1.0000	0.8201	0.6437	-0.7967	0.2807	-0.0799	0.0348		
-0.9417	0.2529	-0.9409	-0.9785	0.9752	-0.9400	-0.9968	0.0387	-0.9980	-0.3559	-0.9482	-0.9508	-0.5622	-0.4784	-0.8012	0.0107		
0.1628	0.7653	-0.8872	-0.9229	0.4073	-0.9945	-1.0000	0.0473	-0.9981	-1.0000	-0.9388	-0.9494	-0.0634	-0.3913	-0.8288	0.0156		
-0.9969	0.9857	-0.6914	-0.7339	-0.3237	-0.0507	0.1826	0.0230	-0.9148	0.9106	-0.8920	-0.9238	0.9211	-0.8996	-0.9898	0.0172		
-0.9859	0.9759	-0.1138	-0.3424	-0.9090	0.3453	-0.1989	0.0391	-0.8701	1.0000	-0.2787	-0.0967	-0.7967	-0.8764	1.0000	0.0056		
-0.9651	0.9439	0.9566	0.1987	-0.1050	0.6647	-0.5063	0.0483	-0.6570	1.0000	0.5925	0.9981	-0.7967	-0.8497	0.1826	0.0445		
-0.9940	-0.5533	-0.1507	-0.6256	-0.4783	0.9404	-0.7715	0.0118	-0.9217	1.0000	-0.9693	-0.9365	-0.9870	-0.9947	-0.5924	0.0075		
-0.9466	0.8608	-0.2932	-0.6046	0.8591	-0.4874	-0.9187	0.0005	-0.9427	1.0000	0.0870	0.1435	-0.7967	-0.1616	0.3262	0.0114		
1.0000	0.9991	-0.8716	-0.7856	-0.8437	-1.0000	-0.9717	0.0631	-0.9079	0.9991	-0.1455	-0.4474	-1.0000	-0.5452	-0.9360	0.0111		
-0.9971	0.1523	-0.4754	-0.6929	-0.1650	0.6479	-0.5504	0.0201	-0.9880	1.0000	-0.8083	-0.7787	-1.0000	-0.4581	0.0472	0.0218		
-0.9791	0.2347	0.2450	-0.4125	1.0000	0.6864	-0.7715	0.0436	-0.2468	1.0000	-0.7995	-0.7994	-0.7967	-0.9806	-0.9968	0.0219		
-0.3115	1.0000	-0.0181	0.1744	0.1954	-0.9905	-0.1989	0.0443	-0.8285	1.0000	-0.0513	0.1835	-0.7967	-0.8764	1.0000	0.0445		
-0.9965	0.9960	-0.8161	-0.8186	-0.7022	-0.5234	0.6381	0.0046	-0.6551	1.0000	0.2317	0.5782	-0.7967	-0.9233	0.3262	0.0059		
-0.8775	0.9962	0.9963	0.9962	-0.9956	-0.5234	0.6381	0.0194	-0.9945	0.1531	-0.8692	-0.8815	0.9507	-0.1759	-0.5924	0.0304		
-0.9862	0.9994	-0.8484	-0.8367	-0.8168	-0.8811	1.0000	0.0472	-0.9482	-0.6979	-0.5186	-0.8209	-0.4831	-0.5484	-0.9862	0.0118		
-0.6025	0.9356	-1.0000	-1.0000	0.0587	-0.9992	-0.9998	0.0088	-0.9755	0.9339	0.5253	-0.1746	0.9793	1.0000	-0.7398	0.0564		
-0.9861	0.9551	-0.8465	-0.8755	0.8443	-0.6547	-0.8012	0.0158	-1.0000	-0.8842	-0.8998	-0.9153	-0.0469	0.0940	-0.5504	0.0263		
-0.8772	0.9962	1.0000	1.0000	-0.9956	-0.5234	0.6381	0.0277	-0.9757	1.0000	-0.3246	-0.3457	-1.0000	0.0939	0.0472	0.0298		
-0.9841	0.9729	-0.6917	-0.7588	-0.9436	-0.5087	-0.7398	0.0261	-0.9367	0.9999	0.2150	-0.1101	-1.0000	-0.1302	-0.7715	0.0348		
-0.9685	0.9122	0.0997	-0.3319	0.6774	0.1227	-0.7060	0.0014	-0.0755	1.0000	-0.4257	-0.1519	-0.9870	-0.9947	-0.5924	0.0040		
-0.9694	0.9994	-0.7308	-0.7091	-0.8168	-0.8811	1.0000	0.0131	-0.9229	0.9840	-0.8311	-0.8716	0.2639	-0.8497	-0.9775	0.0237		
-0.9109	0.9225	-0.1938	-0.4810	0.2738	-0.6275	-0.9187	0.0120	-0.9831	0.9966	-0.9767	-0.9628	-1.0000	-0.8776	-0.9187	0.0254		
								-0.9529	0.9825	-0.2232	-0.4435	0.2159	-0.1676	-0.8012	0.0248		
								-0.9279	0.9275	-0.1572	-0.4947	0.8701	-0.3545	-0.9187	0.0225		

Table A-6 Continued

SVDD								PPS SVDD							
Tech 3								Tech 1							
xSV								bSV							
Element								Element							
1	2	3	4	5	6	7	Coeff.	1	2	3	4	5	6	7	Coeff.
-0.9973	0.6079	0.9715	-0.8111	-0.8954	-0.9153	-0.8562	0.0645	-0.1565	0.9991	-0.9797	-0.9603	-0.8437	-1.0000	-0.9717	0.0228
-0.9491	-1.0000	-0.6540	-0.9232	0.7609	-0.9983	-0.9998	0.0284	-0.3162	0.9356	-1.0000	-1.0000	0.0587	-0.9992	-0.9998	0.0262
-0.2024	0.9411	-0.9577	-0.9924	-0.5761	-0.9999	-0.9992	0.0481	1.0000	0.7653	-0.8870	-0.9227	0.4073	-0.9945	-1.0000	0.0665
-0.8840	-0.2373	-0.8539	-0.9824	-0.8262	-0.9991	-1.0000	0.0578	-0.6383	1.0000	-0.7003	-0.6414	0.1954	-0.9905	-0.1989	0.0281
-0.9981	0.9444	0.4182	-0.9996	0.0400	-0.8752	0.7231	0.0346	0.1152	0.9969	-0.7024	-0.5929	-1.0000	-0.9824	-0.9187	0.0334
-0.9982	0.4948	0.1793	-0.9977	-0.2903	0.1863	0.7231	0.0274	-0.0598	0.9988	-0.0373	0.1707	-0.6943	-0.9573	-0.5504	0.0431
0.5979	0.6179	-0.9890	-0.9867	0.6939	-0.9998	-1.0000	0.0407	-0.8997	0.2529	-0.9408	-0.9785	0.9752	-0.9400	-0.9968	0.0422
-0.9838	0.2703	0.1891	-0.5493	0.7422	-0.9949	-0.9946	0.0012	-0.5069	0.9994	0.4597	0.6172	-0.6586	-0.9216	0.4781	0.0326
-0.9873	0.9211	-0.9985	-0.9973	0.8067	-0.9993	-0.9925	0.0016	-0.9763	0.9994	-0.8481	-0.8364	-0.8168	-0.8811	1.0000	0.0615
-0.9985	0.2271	0.4562	-0.9989	0.0698	1.0000	1.0000	0.0724	-0.7318	0.9975	0.1832	0.2706	-0.9997	-0.7918	0.0472	0.0120
-0.9969	0.8938	0.0938	-0.9018	0.8605	-0.9769	-0.8150	0.0106	-0.9508	-1.0000	-0.6913	-0.9065	-0.5066	-0.6661	-0.9898	0.0548
-0.9949	0.6010	0.4745	-0.8231	-0.2111	-0.9673	-0.9397	0.0044	-0.9761	0.9551	-0.8463	-0.8753	0.8443	-0.6547	-0.8012	0.0162
-0.9890	-0.8076	-0.9619	-0.9867	0.7802	-0.9984	-0.9998	0.0367	-0.8468	0.9225	-0.1923	-0.4801	0.2738	-0.6275	-0.9187	0.0173
-0.9985	0.1681	-0.9658	-0.9935	0.8799	-0.9814	-0.9812	0.0125	-0.8649	0.1125	-0.1068	-0.6878	0.9227	-0.5506	-0.9821	0.0372
-0.9996	0.8214	-0.6698	-0.9989	0.8153	-0.8272	-0.1785	0.0261	-0.9939	0.9960	-0.8158	-0.8183	-0.7022	-0.5234	0.6381	0.0051
-0.9907	0.7577	0.9329	-0.5944	0.9794	-0.9893	-0.9621	0.0624	-0.7893	0.9962	1.0000	1.0000	-0.9956	-0.5234	0.6381	0.0551
-0.5083	-0.5063	-0.1682	-0.4123	0.5480	-0.9991	-1.0000	0.0495	-0.9726	0.9729	-0.6912	-0.7584	-0.9436	-0.5087	-0.7398	0.0359
-1.0000	1.0000	-0.9437	-0.9773	0.7805	-1.0000	-1.0000	0.0442	-0.9081	0.8608	-0.2919	-0.6038	0.8591	-0.4874	-0.9187	0.0003
-0.9984	0.8013	1.0000	-0.9961	0.1075	-0.4874	1.0000	0.0452	-0.8470	0.9811	0.2523	0.0740	-0.5416	-0.4801	-0.5924	0.0165
-0.9985	0.7948	0.8291	-0.9989	0.3471	-0.4874	1.0000	0.0113	-0.9165	0.9919	0.2247	0.1314	-0.7358	-0.2398	0.3262	0.0123
1.0000	0.6230	-0.4876	-0.8280	0.5885	-0.9998	-1.0000	0.0573	-0.9910	-0.7361	-0.7027	-0.8624	-0.0133	-0.1707	-0.8993	0.0215
-0.8475	0.9520	0.2179	-0.7177	-0.7964	-0.9979	-0.9714	0.0305	-0.9364	0.9877	0.1431	-0.0017	-0.3202	-0.0507	0.1826	0.0103
-0.7267	0.2633	0.9391	1.0000	0.6742	-0.9994	-0.9998	0.0565	-0.9947	0.9857	-0.6908	-0.7334	-0.3237	-0.0507	0.1826	0.0225
-0.9990	0.0986	-0.4517	-0.9916	0.4173	-0.2788	-0.3523	0.0339	-0.9459	0.9122	0.1017	-0.3306	0.6774	0.1227	-0.7060	0.0012
-0.9989	0.5814	-0.2423	-1.0000	0.0458	-0.2019	0.4667	0.0108	-0.9973	0.8366	-0.6885	-0.7590	0.7878	0.2218	-0.0799	0.0425
-0.9975	0.6770	-0.5909	-0.9987	-0.8300	-0.5137	0.0160	0.0450	-0.9279	0.9519	0.6624	0.0997	0.6052	0.3323	-0.5063	0.0335
-0.9727	0.7425	-0.9780	-0.9863	-0.8725	-0.9992	-0.9984	0.0206	-0.9757	0.9759	-0.1122	-0.3411	-0.9090	0.3453	-0.1989	0.0380
-0.8222	0.8717	0.1557	-0.6602	0.8221	-0.9988	-0.9925	0.0090	-1.0000	-0.5429	-0.6288	-0.8033	-0.5294	0.4467	-0.7060	0.0287
-0.9985	0.9436	-0.6907	-0.9997	-0.0489	-0.9245	0.0160	0.0172	-0.9866	0.6432	-0.3785	-0.5966	0.8887	0.5667	-0.4121	0.0098
-0.9658	0.8069	0.9383	0.7476	0.6998	-0.9997	-0.9992	0.0399	-0.9950	0.1523	-0.4744	-0.6923	-0.1650	0.6479	-0.5504	0.0207
								-0.9400	0.9439	0.9603	0.2010	-0.1050	0.6647	-0.5063	0.0490
								-0.9640	0.2347	0.2474	-0.4114	1.0000	0.6864	-0.7715	0.0448
								-0.9897	-0.5533	-0.1491	-0.6249	-0.4783	0.9404	-0.7715	0.0122
								-0.9782	-0.5474	0.2334	-0.5124	-0.1091	1.0000	-0.8288	0.0463





Table A-6 Continued

Interpolation Tech 1							Interpolation Tech 2							
xSV						bSV	xSV						bSV	
Element							Element							
t	1	2	3	4	5	6	Coeff.	1	2	3	4	5	6	Coeff.
-0.1565	0.9991	-0.9603	-0.8437	-1.0000	-0.9717	0.0233	1.0000	1.0000	-0.6273	-0.7967	-1.0000	-0.9360	0.0724	
-0.3162	0.9356	-1.0000	0.0587	-0.9992	-0.9998	0.0256	-0.9217	1.0000	-0.9365	-0.9870	-0.9947	-0.5924	0.0078	
1.0000	0.7653	-0.9227	0.4073	-0.9945	-1.0000	0.0793	-0.8698	0.5246	-1.0000	0.8382	-0.9880	-1.0000	0.0191	
-0.6383	1.0000	-0.6414	0.1954	-0.9905	-0.1989	0.0352	-0.2468	1.0000	-0.7994	-0.7967	-0.9806	-0.9968	0.0221	
0.1152	0.9969	-0.5929	-1.0000	-0.9824	-0.9187	0.0355	-0.0962	1.0000	0.0334	-0.7967	-0.9695	-0.7715	0.0486	
-0.0598	0.9988	0.1707	-0.6943	-0.9573	-0.5504	0.0450	-0.8686	0.9994	-0.9601	-1.0000	-0.9687	-0.9968	0.0245	
-0.4436	0.9011	-0.7511	0.5653	-0.9418	-0.9898	0.0083	-0.8070	1.0000	-0.2694	-0.7967	-0.9321	0.0472	0.0161	
-0.8997	0.2529	-0.9785	0.9752	-0.9400	-0.9968	0.0441	-0.6551	1.0000	0.5782	-0.7967	-0.9233	0.3262	0.0018	
-0.5069	0.9994	0.6172	-0.6586	-0.9216	0.4781	0.0327	-0.9357	1.0000	-0.5847	-0.7967	-0.9057	1.0000	0.0188	
-0.9763	0.9994	-0.8364	-0.8168	-0.8811	1.0000	0.0738	-0.9148	0.9106	-0.9238	0.9211	-0.8996	-0.9898	0.0299	
-0.7318	0.9975	0.2706	-0.9997	-0.7918	0.0472	0.0094	-0.9831	0.9966	-0.9628	-1.0000	-0.8776	-0.9187	0.0195	
-0.9508	-1.0000	-0.9065	-0.5066	-0.6661	-0.9898	0.0660	-0.9877	1.0000	-0.8887	-0.7967	-0.8764	1.0000	0.0537	
-0.9761	0.9551	-0.8753	0.8443	-0.6547	-0.8012	0.0226	-0.6570	1.0000	0.9981	-0.7967	-0.8497	0.1826	0.0505	
-0.8468	0.9225	-0.4801	0.2738	-0.6275	-0.9187	0.0004	-0.9229	0.9840	-0.8716	0.2639	-0.8497	-0.9775	0.0240	
-0.9575	0.0863	-0.8714	0.9465	-0.6131	-0.9598	0.0050	-0.9965	-0.8031	-0.9704	-0.5355	-0.6550	-0.9187	0.0416	
-0.8649	0.1125	-0.6878	0.9227	-0.5506	-0.9821	0.0167	-0.8936	1.0000	0.4709	-0.7967	-0.5011	0.6381	0.0289	
-0.9939	0.9960	-0.8183	-0.7022	-0.5234	0.6381	0.0027	-0.9364	0.0458	-0.7360	0.9425	-0.4900	-0.9717	0.0288	
-0.7893	0.9962	1.0000	-0.9956	-0.5234	0.6381	0.0601	-0.9980	-0.3559	-0.9508	-0.5622	-0.4784	-0.8012	0.0120	
-0.9726	0.9729	-0.7584	-0.9436	-0.5087	-0.7398	0.0417	-0.9880	1.0000	-0.7787	-1.0000	-0.4581	0.0472	0.0192	
-0.8470	0.9811	0.0740	-0.5416	-0.4801	-0.5924	0.0075	-0.9981	-1.0000	-0.9494	-0.0634	-0.3913	-0.8288	0.0250	
-0.9165	0.9919	0.1314	-0.7358	-0.2398	0.3262	0.0052	-0.9279	0.9275	-0.4947	0.8701	-0.3545	-0.9187	0.0066	
-0.9910	-0.7361	-0.8624	-0.0133	-0.1707	-0.8993	0.0285	-0.9945	0.1531	-0.8815	0.9507	-0.1759	-0.5924	0.0298	
-0.9364	0.9877	-0.0017	-0.3202	-0.0507	0.1826	0.0041	-0.9529	0.9825	-0.4435	0.2159	-0.1676	-0.8012	0.0325	
-0.9947	0.9857	-0.7334	-0.3237	-0.0507	0.1826	0.0214	-0.9367	0.9999	-0.1101	-1.0000	-0.1302	-0.7715	0.0333	
-0.9973	0.8366	-0.7590	0.7878	0.2218	-0.0799	0.0436	-0.9067	1.0000	1.0000	-0.7967	-0.0052	0.1826	0.0217	
-0.9279	0.9519	0.0997	0.6052	0.3323	-0.5063	0.0271	-0.9757	1.0000	-0.3457	-1.0000	0.0939	0.0472	0.0297	
-0.9757	0.9759	-0.3411	-0.9090	0.3453	-0.1989	0.0403	-1.0000	-0.8842	-0.9153	-0.0469	0.0940	-0.5504	0.0204	
-1.0000	-0.5429	-0.8033	-0.5294	0.4467	-0.7060	0.0158	-0.9741	-0.3622	-0.6717	-0.6854	0.0946	-0.9187	0.0205	
-0.9950	0.1523	-0.6923	-0.1650	0.6479	-0.5504	0.0135	-0.9354	1.0000	0.6437	-0.7967	0.2807	-0.0799	0.0389	
-0.9400	0.9439	0.2010	-0.1050	0.6647	-0.5063	0.0455	-0.9785	0.9358	-0.4271	1.0000	0.5309	-0.3096	0.0379	
-0.9640	0.2347	-0.4114	1.0000	0.6864	-0.7715	0.0545	-0.9820	-0.4908	-0.5948	0.3002	0.7066	-0.8012	0.0487	
-0.9897	-0.5533	-0.6249	-0.4783	0.9404	-0.7715	0.0313	-0.9918	0.1529	-0.6679	-0.5667	0.7453	-0.5063	0.0392	
-0.9782	-0.5474	-0.5124	-0.1091	1.0000	-0.8288	0.0343	-0.9927	-0.2788	-0.6911	-0.5638	0.8392	-0.5924	0.0208	
							-0.9755	0.9339	-0.1746	0.9793	1.0000	-0.7398	0.0556	

Table A-6 Continued

Interpolation							Regression								
Tech 3							Tech 1								
xSV							bSV								
Elemen							Elemen								
t	1	2	3	4	5	6	Coeff.	t	1	2	3	4	5	6	Coeff.
-1.0000	1.0000	-0.9773	0.7805	-1.0000	-1.0000	0.0555	-0.1565	0.9991	-0.9603	-0.8437	-1.0000	-0.9717	0.0232		
-0.2024	0.9411	-0.9924	-0.5761	-0.9999	-0.9992	0.0585	-0.3162	0.9356	-1.0000	0.0587	-0.9992	-0.9998	0.0254		
0.5979	0.6179	-0.9867	0.6939	-0.9998	-1.0000	0.0341	1.0000	0.7653	-0.9227	0.4073	-0.9945	-1.0000	0.0794		
1.0000	0.6230	-0.8280	0.5885	-0.9998	-1.0000	0.0802	-0.6383	1.0000	-0.6414	0.1954	-0.9905	-0.1989	0.0350		
-0.9658	0.8069	0.7476	0.6998	-0.9997	-0.9992	0.0608	0.1152	0.9969	-0.5929	-1.0000	-0.9824	-0.9187	0.0354		
-0.7267	0.2633	1.0000	0.6742	-0.9994	-0.9998	0.0691	-0.0598	0.9988	0.1707	-0.6943	-0.9573	-0.5504	0.0450		
-0.8840	-0.2373	-0.9824	-0.8262	-0.9991	-1.0000	0.0801	-0.4436	0.9011	-0.7511	0.5653	-0.9418	-0.9898	0.0085		
-0.5083	-0.5063	-0.4123	0.5480	-0.9991	-1.0000	0.0461	-0.8997	0.2529	-0.9785	0.9752	-0.9400	-0.9968	0.0436		
-0.9417	0.9179	-0.9423	-0.7970	-0.9984	-0.9876	0.0027	-0.5069	0.9994	0.6172	-0.6586	-0.9216	0.4781	0.0328		
-0.9491	-1.0000	-0.9232	0.7609	-0.9983	-0.9998	0.0797	-0.9763	0.9994	-0.8364	-0.8168	-0.8811	1.0000	0.0742		
-0.8475	0.9520	-0.7177	-0.7964	-0.9979	-0.9714	0.0357	-0.8623	0.9809	-0.7756	-0.5395	-0.8717	-0.8993	0.0002		
-0.9907	0.7577	-0.5944	0.9794	-0.9893	-0.9621	0.0184	-0.7318	0.9975	0.2706	-0.9997	-0.7918	0.0472	0.0094		
-0.9784	0.9530	-0.9982	-0.8017	-0.9871	-0.8150	0.0044	-0.9508	-1.0000	-0.9065	-0.5066	-0.6661	-0.9898	0.0661		
-0.9981	0.9444	-0.9996	0.0400	-0.8752	0.7231	0.0416	-0.9761	0.9551	-0.8753	0.8443	-0.6547	-0.8012	0.0228		
-0.9993	0.1159	-0.9599	1.0000	-0.8535	-0.8562	0.0140	-0.8468	0.9225	-0.4801	0.2738	-0.6275	-0.9187	0.0004		
-0.9996	0.8214	-0.9989	0.8153	-0.8272	-0.1785	0.0306	-0.9575	0.0863	-0.8714	0.9465	-0.6131	-0.9598	0.0063		
-0.9975	0.6770	-0.9987	-0.8300	-0.5137	0.0160	0.0600	-0.8649	0.1125	-0.6878	0.9227	-0.5506	-0.9821	0.0155		
-0.9985	0.7948	-0.9989	0.3471	-0.4874	1.0000	0.0546	-0.9939	0.9960	-0.8183	-0.7022	-0.5234	0.6381	0.0024		
-0.9990	0.0986	-0.9916	0.4173	-0.2788	-0.3523	0.0474	-0.7893	0.9962	1.0000	-0.9956	-0.5234	0.6381	0.0601		
-0.9982	0.4948	-0.9977	-0.2903	0.1863	0.7231	0.0301	-0.9726	0.9729	-0.7584	-0.9436	-0.5087	-0.7398	0.0417		
-0.9985	0.2271	-0.9989	0.0698	1.0000	1.0000	0.0964	-0.8470	0.9811	0.0740	-0.5416	-0.4801	-0.5924	0.0075		
							-0.9165	0.9919	0.1314	-0.7358	-0.2398	0.3262	0.0054		
							-0.9910	-0.7361	-0.8624	-0.0133	-0.1707	-0.8993	0.0285		
							-0.9364	0.9877	-0.0017	-0.3202	-0.0507	0.1826	0.0038		
							-0.9947	0.9857	-0.7334	-0.3237	-0.0507	0.1826	0.0216		
							-0.9973	0.8366	-0.7590	0.7878	0.2218	-0.0799	0.0436		
							-0.9279	0.9519	0.0997	0.6052	0.3323	-0.5063	0.0271		
							-0.9757	0.9759	-0.3411	-0.9090	0.3453	-0.1989	0.0404		
							-1.0000	-0.5429	-0.8033	-0.5294	0.4467	-0.7060	0.0156		
							-0.9950	0.1523	-0.6923	-0.1650	0.6479	-0.5504	0.0136		
							-0.9400	0.9439	0.2010	-0.1050	0.6647	-0.5063	0.0455		
							-0.9640	0.2347	-0.4114	1.0000	0.6864	-0.7715	0.0547		
							-0.9897	-0.5533	-0.6249	-0.4783	0.9404	-0.7715	0.0315		
							-0.9782	-0.5474	-0.5124	-0.1091	1.0000	-0.8288	0.0340		

Table A-6 Continued

Regression Tech 2 xSV							bSV	Regression Tech 3 xSV							bSV
Element								Element							
t	1	2	3	4	5	6	Coeff.	1	2	3	4	5	6	Coeff.	
1.0000	1.0000	-0.6273	-0.7967	-1.0000	-0.9360	0.0725	-1.0000	1.0000	-0.9773	0.7805	-1.0000	-1.0000	0.0560		
-0.9217	1.0000	-0.9365	-0.9870	-0.9947	-0.5924	0.0078	-0.2024	0.9411	-0.9924	-0.5761	-0.9999	-0.9992	0.0584		
-0.8698	0.5246	-1.0000	0.8382	-0.9880	-1.0000	0.0192	0.5979	0.6179	-0.9867	0.6939	-0.9998	-1.0000	0.0341		
-0.2468	1.0000	-0.7994	-0.7967	-0.9806	-0.9968	0.0221	1.0000	0.6230	-0.8280	0.5885	-0.9998	-1.0000	0.0802		
-0.0962	1.0000	0.0334	-0.7967	-0.9695	-0.7715	0.0487	-0.9658	0.8069	0.7476	0.6998	-0.9997	-0.9992	0.0609		
-0.8686	0.9994	-0.9601	-1.0000	-0.9687	-0.9968	0.0239	-0.7267	0.2633	1.0000	0.6742	-0.9994	-0.9998	0.0691		
-0.8070	1.0000	-0.2694	-0.7967	-0.9321	0.0472	0.0161	-0.8840	-0.2373	-0.9824	-0.8262	-0.9991	-1.0000	0.0803		
-0.6551	1.0000	0.5782	-0.7967	-0.9233	0.3262	0.0020	-0.5083	-0.5063	-0.4123	0.5480	-0.9991	-1.0000	0.0463		
-0.9357	1.0000	-0.5847	-0.7967	-0.9057	1.0000	0.0188	-0.9417	0.9179	-0.9423	-0.7970	-0.9984	-0.9876	0.0035		
-0.9148	0.9106	-0.9238	0.9211	-0.8996	-0.9898	0.0298	-0.9491	-1.0000	-0.9232	0.7609	-0.9983	-0.9998	0.0796		
-0.9831	0.9966	-0.9628	-1.0000	-0.8776	-0.9187	0.0200	-0.8475	0.9520	-0.7177	-0.7964	-0.9979	-0.9714	0.0358		
-0.9877	1.0000	-0.8887	-0.7967	-0.8764	1.0000	0.0536	-0.9907	0.7577	-0.5944	0.9794	-0.9893	-0.9621	0.0180		
-0.6570	1.0000	0.9981	-0.7967	-0.8497	0.1826	0.0504	-0.9784	0.9530	-0.9982	-0.8017	-0.9871	-0.8150	0.0032		
-0.9229	0.9840	-0.8716	0.2639	-0.8497	-0.9775	0.0242	-0.9981	0.9444	-0.9996	0.0400	-0.8752	0.7231	0.0416		
-0.9965	-0.8031	-0.9704	-0.5355	-0.6550	-0.9187	0.0415	-0.9993	0.1159	-0.9599	1.0000	-0.8535	-0.8562	0.0140		
-0.8936	1.0000	0.4709	-0.7967	-0.5011	0.6381	0.0288	-0.9996	0.8214	-0.9989	0.8153	-0.8272	-0.1785	0.0306		
-0.9364	0.0458	-0.7360	0.9425	-0.4900	-0.9717	0.0287	-0.9975	0.6770	-0.9987	-0.8300	-0.5137	0.0160	0.0602		
-0.9980	-0.3559	-0.9508	-0.5622	-0.4784	-0.8012	0.0120	-0.9985	0.7948	-0.9989	0.3471	-0.4874	1.0000	0.0546		
-0.9880	1.0000	-0.7787	-1.0000	-0.4581	0.0472	0.0192	-0.9990	0.0986	-0.9916	0.4173	-0.2788	-0.3523	0.0471		
-0.9981	-1.0000	-0.9494	-0.0634	-0.3913	-0.8288	0.0249	-0.9982	0.4948	-0.9977	-0.2903	0.1863	0.7231	0.0300		
-0.9279	0.9275	-0.4947	0.8701	-0.3545	-0.9187	0.0067	-0.9985	0.2271	-0.9989	0.0698	1.0000	1.0000	0.0964		
-0.9945	0.1531	-0.8815	0.9507	-0.1759	-0.5924	0.0297									
-0.9529	0.9825	-0.4435	0.2159	-0.1676	-0.8012	0.0325									
-0.9367	0.9999	-0.1101	-1.0000	-0.1302	-0.7715	0.0333									
-0.9067	1.0000	1.0000	-0.7967	-0.0052	0.1826	0.0218									
-0.9757	1.0000	-0.3457	-1.0000	0.0939	0.0472	0.0298									
-1.0000	-0.8842	-0.9153	-0.0469	0.0940	-0.5504	0.0203									
-0.9741	-0.3622	-0.6717	-0.6854	0.0946	-0.9187	0.0206									
-0.9354	1.0000	0.6437	-0.7967	0.2807	-0.0799	0.0388									
-0.9785	0.9358	-0.4271	1.0000	0.5309	-0.3096	0.0379									
-0.9820	-0.4908	-0.5948	0.3002	0.7066	-0.8012	0.0487									
-0.9918	0.1529	-0.6679	-0.5667	0.7453	-0.5063	0.0392									
-0.9927	-0.2788	-0.6911	-0.5638	0.8392	-0.5924	0.0208									
-0.9755	0.9339	-0.1746	0.9793	1.0000	-0.7398	0.0556									

## VITA

Name: Robert Reed Parker

Address: 3123 TAMU  
Department of Mechanical Engineering  
Texas A&M University,  
College Station, TX 77843

Email Address: [rrparker@tamu.edu](mailto:rrparker@tamu.edu)

Education: B.S., Mechanical Engineering, Iowa State University, 2008