

2008

The Application of Polynomial Response Surface and Polynomial Chaos Expansion Metamodels within an Augmented Reality Conceptual Design Environment

Andrew Thomas Koehring
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Koehring, Andrew Thomas, "The Application of Polynomial Response Surface and Polynomial Chaos Expansion Metamodels within an Augmented Reality Conceptual Design Environment" (2008). *Graduate Theses and Dissertations*. 11605.
<https://lib.dr.iastate.edu/etd/11605>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**The application of polynomial response surface and polynomial chaos expansion
metamodels within an augmented reality conceptual design environment**

by

Andrew Koehring

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Mechanical Engineering; Human Computer Interaction

Program of Study Committee:

Eliot Winer, Major Professor

James Oliver

Chris Harding

Mark Bryden

Iowa State University

Ames, Iowa

2008

Copyright © Andrew Thomas Koehring, 2008. All rights reserved.

TABLE OF CONTENTS

LIST OF FIGURES	iv
LIST OF TABLES	vii
ABSTRACT	viii
1 INTRODUCTION	1
1.1 The Engineering Design Process	1
1.2 Traditional Engineering Software and Conceptual Design	4
1.3 Motivation	10
1.4 Thesis Organization	13
2 BACKGROUND	14
2.1 Conceptual Design Applications	14
2.2 Augmented Reality	16
2.2.1 Registration	19
2.2.2 Display Configurations	21
2.2.3 Applications	24
2.2.4 Software Libraries	27
2.3 Metamodeling	30
2.3.1 Design of Experiments	32
2.3.2 Polynomial Response Surface	33
2.3.3 Polynomial Chaos Expansion	34
2.4 Research Issues	36

3	AUGMENTED REALITY INTERFACE ENVIRONMENT	37
3.1	Hardware	37
3.1.1	Display	37
3.1.2	Marker Boards	38
3.1.3	Wand	39
3.2	Software	41
3.2.1	Wand	41
3.2.2	Part Library	41
3.2.3	Editor	43
3.2.4	Assembly Area	44
4	METAMODEL DEVELOPMENT	48
4.1	Wheel Loading Test Case	49
4.1.1	Problem Description	49
4.1.2	Model Formulation	51
4.1.3	Experimental Setup	52
4.2	Bucket Truck Support Frame Test Case	55
4.2.1	Problem Description	55
4.2.2	Experimental Setup	59
5	RESULTS AND DISCUSSION	60
5.1	Metamodeling Results	60
5.1.1	Evaluation Procedure	60
5.1.2	Wheel Loading Results	61
5.1.3	Support Frame Results	69
6	CONCLUSIONS AND FUTURE WORK	76
6.1	Summary and Conclusions	76
6.2	Future Work	78
6.3	Acknowledgements	80
	REFERENCES	82

LIST OF FIGURES

Figure 1:	A diagram illustrating a generalized version of the engineering design process. New products must advance through these stages before being realized.	1
Figure 2:	Sample screen shot from a FLUENT analysis in CATIA. Traditional CAD and FEA/CFD software interfaces involve a large number of tools and are far too complex for conceptual design.	5
Figure 3:	Diagram illustrating the concept of mapping as it pertains to the common mouse interface. When working in a 2D environment, mouse movements correlates very directly to cursor movement. However, in a 3D environment there is no obvious cursor direction.	9
Figure 4:	This screen capture of Google SketchUp demonstrates the simplicity of the interface. SketchUp uses only a handful of tools; this is very different than crowded interface shown in Figure 2.	16
Figure 5:	An adaptation of Milgram's virtuality continuum. Augmented reality exists between the two extremes: the real world and pure virtual reality.	17
Figure 6:	Example image of AR application showing virtual objects in the context of the real world.	18
Figure 7:	One example of a video see-through display using an HMD and a camera.	23
Figure 8:	(a) A handheld video see-through device. (b) A video see-through desk configuration that uses a mirror mounted on the back of the display to augment a table top scene.	23
Figure 9:	(a) A conceptualization of a large scale optical see-through system. (b) One example of an optical see-through head mounted display.	24

- Figure 10:** A step by step breakdown of the ARToolKit tracking algorithm. 29
 (a) Video frames are captured from a camera device.
 (b) The image is converted to a binary (i.e. black and white) format.
 (c) A connected component algorithm labels connected pixels.
 (d) Edge detection is used to highlight the outer lines of the marker.
 (e) The marker edges and corners are recognized.
 (f) Pose estimation is used determine the marker's position and orientation.
- Figure 11:** A photo of the hardware interface elements (HMD, marker boards, and wand). 38
- Figure 12:** A close-up view of the wand buttons. 40
- Figure 13:** The part library contains a collection of models that can be used to create assemblies. 42
- Figure 14:** Object selection is done with the wand pointer. Increasing the model's scale provides visual feedback to the user. 43
- Figure 15:** Grabbing an object transfers it to the wand so that it may be viewed individually. 43
- Figure 16:** The editor board provides a method for altering different aspects of the selected model. 44
- Figure 17:** Copies of grabbed objects can be placed on the assembly board at any orientation. 45
- Figure 18:** Completed concept assemblies can be easily visualized. 45
- Figure 19:** When the CG button is activated, the models become transparent and the center of gravity is represented by a small red sphere. 46
- Figure 20:** Users can obtain wheel loading feedback using the built in assessment tool. 47
- Figure 21:** Screenshot of the loading rig created and analyzed using ABAQUS. 50
- Figure 22:** An example image of a bucket truck used for the second analysis case. 56
- Figure 23:** 58

- Figure 24:** This chart presents an example case of model performance in interpolating and extrapolating conditions. 63
- Figure 25:** Models can exhibit unstable behavior when presented with only a minimum of source data. 64
- Figure 26:** In this modeling configuration, interaction effects had a positive impact on model performance. 65
- Figure 27:** Interaction effects were not beneficial in this example. 66
- Figure 28:** With high levels of source data, 3rd order models offer a small improvement in model accuracy. 67
- Figure 29:** With minimal input data, 2nd order models often outperform those with 3rd order terms. 68
- Figure 30:** PCE performs better when fit to smaller distributions of data. Source data with a greater distribution is better modeled by PRS. 69
- Figure 31:** Error values are noticeably higher than those seen in the wheel loading analysis. 72
- Figure 32:** The position of the maximum displacement is difficult to approximate. This modeling configuration did not perform well. 73
- Figure 33:** Simpler, 2nd order models offered better performance than 3rd order. 74
- Figure 34:** Large improvements were obtained when 2nd order models were used to approximate maximum displacement position. 75

LIST OF TABLES

Table 1:	The parameters used to generate the simulation trials for the support frame test case.	58
Table 2:	R^2 values for 3 rd order models at different levels of input data.	71
Table 3:	R^2 values for 2 nd order models at different levels of input data.	71

ABSTRACT

The engineering design process consists of many stages. In the conceptual phase, potential designs are generated and evaluated without considering specifics. Winning concepts then advance to the detail design and high fidelity simulation stages. At this point in the process, very accurate representations are made for each design and are then subjected to rigorous analysis. With the advancement of computer technology, these last two phases have been very well served by the software community. Engineering software such as computer-aided design (CAD), finite element analysis (FEA), and computational fluid dynamics (CFD) have become an inseparable part of the design process for many engineered products and processes. Conceptual design tools, on the other hand, have not undergone this type of advancement, where much of the work is still done with little to no digital technology. Detail oriented tools require a significant amount of time and training to use effectively. This investment is considered worthwhile when high fidelity models are needed. However, conceptual design has no need for this level of detail. Instead, rapid concept generation and evaluation are the primary goals. Considering the lack of adequate tools to suit these needs, new software was created. This thesis discusses the development of that conceptual design application.

Traditional design tools rely on a two dimensional mouse to perform three dimensional actions. While many designers have become familiar with this approach, it is not intuitive to an inexperienced user. In order to enhance the usability of the developed application, a new interaction method was applied. Augmented reality (AR) is a developing research area that combines virtual elements with the real world. This capability was used to

create a three dimensional interface for the engineering design application. Using specially tracked interface objects, the user's hands become the primary method of interaction. Within this AR environment, users are able perform many of the basic actions available within a CAD system such as object manipulation, editing, and assembly. The same design environment also provides real time assessment data. Calculations for center of gravity and wheel loading can be done with the click of a few buttons. Results are displayed to the user in the AR scene.

In order to support the quantitative analysis tools necessary for conceptual design, additional research was done in the area of metamodeling. Metamodels are capable of providing approximations for more complex analyses. In the case of the wheel loading calculation, the approximation takes the place of a time consuming FEA simulation. Two different metamodeling techniques were studied in this thesis: polynomial response surface (PRS) and polynomial chaos expansion (PCE). While only the wheel loading case study was included in the developed application, an additional design problem was analyzed to assess the capabilities of both methods for conceptual design. In the second study, the maximum stresses and displacements within the support frame of a bucket truck were modeled. The source data for building the approximations was generated via an FEA simulation of digital mockups, since no legacy data was available. With this information, experimental models were constructed by varying several factors, including: the distribution of source and test data, the number of input trials, the inclusion of interaction effects, and the addition of third order terms. Comparisons were also drawn between the two metamodeling techniques.

For the wheel loading models, third order models with interaction effects provided a good fit of the data (root mean square error of less than 10%) with as few as thirty input data points. With minimal source data, however, second order models and those without interaction effects outperformed third order counterparts. The PRS and PCE methods performed almost equivalently with sufficient source data. Difference began to appear at the twenty trial case. PRS was more suited to wider distributions of data. The PCE technique better handled smaller distributions and extrapolation to larger test data. The support frame problem represented a more difficult analysis with non-linear responses. While initial third order results from the PCE models were better than those for PRS, both had significantly higher error than in the previous case study. However, with simpler second order models and sufficient input data (more than thirty trials) adequate approximation results were achieved. The less complex responses had error around 10%, and the model predictions for the non-linear response were reduced to around 20%. These results demonstrate that useful approximations can be constructed from minimal data. Such models, despite the uncertainty involved, will be able to provide designers with helpful information at the conceptual stage of a design process.

1 INTRODUCTION

1.1 The Engineering Design Process

Engineering design is a decision methodology for the creation of a new or refined product or process. Many different theories apply more specific definitions to this very broad topic: concurrent engineering; knowledge based engineering; decision based design; design by shopping¹⁻⁴. For the purposes of this thesis, a specific theory is not necessary, but rather an overall conceptual understanding of the generalized activities that occur within most design processes. Figure 1 presents a flowchart that breaks down a generic engineering design process into sequential steps.

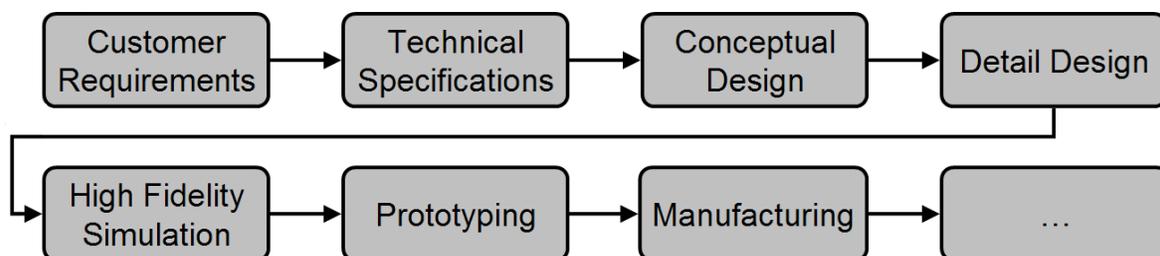


Figure 1: A diagram illustrating a generalized version of the engineering design process. New products must advance through these stages before being realized.

The first two steps require a designer to specify the need(s) that will be satisfied by the final product or process and what technically must be done to get there. He or she must research the problem to gather relevant information. The next step is the conceptual design phase. Many potential solutions are created in as short a time as possible. At this point, little thought is given to specific technical details. The large number of concepts must then be

narrowed down to a select few. In fourth stage of the process, detail design, selected concepts are made more concrete. Precise drawings are created with careful attention paid to factors such as exact dimensions, tolerances, and materials. The next step is where the proposed design(s) are subjected various methods of evaluation. This often involves computational tools such as finite element analysis (FEA) or computational fluid dynamics (CFD). Following successful evaluations, prototypes are constructed and physical performance tests can be performed to validate previous computer based analyses. The final step results in the finished product or process. It is well established that design of any kind is an iterative process⁵. If at any point in the process the design fails to meet set criteria, it may be necessary to move back one or more steps. Thus, bringing a product all the way through manufacturing is often the result of numerous redesigns and subsequent testing. In a world where the number one goal is “better, faster, cheaper,” streamlining the design process is of great interest to everyone involved.

There are certain obvious considerations for most designs, such as cost, time, quality, and performance. As a product or process moves through each of the steps, it becomes more costly and time consuming to make changes and more difficult to increase quality and improve performance. As previously mentioned, these stages are often repeated. For example, a designer may go through several iterations of prototypes before moving on to production. Each new prototype requires the designer to make changes to the detailed design, which may in turn require new simulation and testing as well. These changes, and subsequent repetitions of phases, can greatly increase the cost of the design process. The incurred cost of a design also increases greatly as it advances through the development cycle⁶. Early in the

design process, a particular change might cost one thousand dollars, for instance. The same change may cost as much as ten thousand dollars in later phases; once production has begun, such a change has the potential to cost one million dollars or more⁷. Thus, there exists a great motivation to perform changes as early in the process as possible.

For several decades, digital technology has been answering this need by providing tools to find problems and make changes on the computer rather than the shop floor. Examples of this are computer-aided design (CAD), FEA, and CFD. CAD software packages replaced pencil and paper drafting, where a single design change often necessitated an entirely new drawing. Drawings done in a digital format are much more flexible. In addition, CAD designs can be viewed and assembled in three dimensions while a two dimensional blueprint cannot. For design analysis, the numerical solution methods of FEA and CFD provide the means to analyze these CAD models, which would have otherwise been impossible. Simple models can be solved using closed form analytic equations, but as the complexity of the model increases, such equations become unmanageable. There is no refuting that CAD and FEA have revolutionized detailed design. However, if significant changes to a design must occur during or after detailed design, finalizing the alterations can still be quite costly. For example, CAD models for even a medium sized project can easily contain hundreds to thousands of parts. These have been carefully assembled with specific dimensioning and mating relationships. If a change is required in any of these parts, then a domino effect can occur requiring hours to weeks to months of time to re-create the CAD models to account for the proposed changes. This, in turn, requires new FEA and CFD models to be created and analyzed adding additional time and cost.

If more design changes could be tested and accounted for in the conceptual design stage, then significant cost and time could be removed from a design process. The paradox is that a great deal of conceptual design is still being done without technological assistance. Ideas are communicated, sometimes poorly, via simple two-dimensional drawings (e.g., “back of the envelope” drawings). Evaluating a concept is often more of an instinctual judgment than an objective analysis. Various decision systems exist to better guide designers toward optimal concepts. Pugh concept selection charts, estimating technical difficulty, and numerical scoring are three examples of these decision methods⁸. However, subjective opinion is still the foundation for these decisions, not objective analysis. While there is no substitute for experience and collaboration, designers could certainly benefit from more advanced tools targeted toward this phase of the design process. Considering the economic advantage of early design changes discussed above, such tools have the potential to dramatically impact the design process.

1.2 Traditional Engineering Software and Conceptual Design

In an effort to provide conceptual designers with the advantages of computer based tools, modern design applications have sometimes been repurposed for use in concept generation and evaluation. Unfortunately, the tools intended for detailed design are ill-suited to the task of conceptual design. A sample screen shot from a common analysis suite, FLUENT for CATIA⁹, is provided in Figure 2. It can be seen from the image that these types of programs are quite complex. These modern packages are extremely feature-rich which is beneficial, often crucial, for detail design. Unfortunately, the interface required to handle all

these features is overwhelming to anyone but an expert user. Each application can contain hundreds or even thousands of commands and options in various toolbars and intricate menu systems. The psychological sciences have coined the term “cognitive load”¹⁰, which is often used when considering interface design. A user’s ability to learn and navigate an interface is affected by the amount of information he or she is presented. This concept is related to human limitations on short-term or working memory. A casual user would be overwhelmed by the interface in Figure 2.

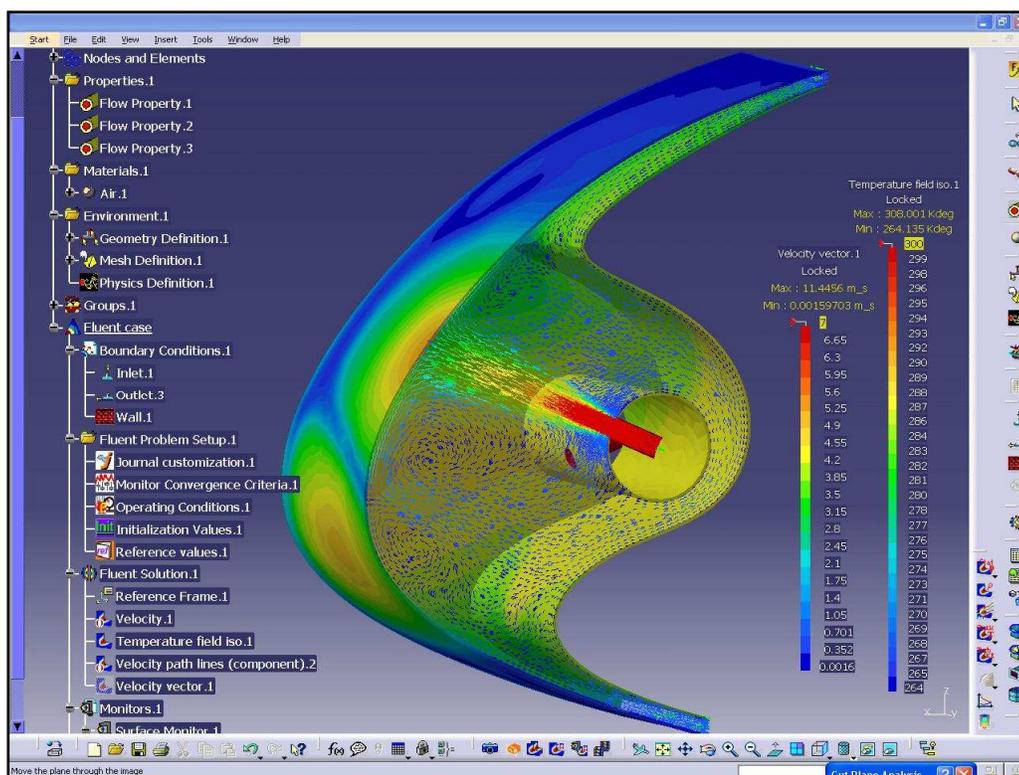


Figure 2: Sample screen shot from a FLUENT analysis in CATIA. Traditional CAD and FEA/CFD software interfaces involve a large number of tools and are far too complex for conceptual design.

Another common term encountered when discussing human-computer interaction is “usability”, which is synonymous with “ease of use”. CAD software can be considered to

have a high cognitive load which in turn results in low usability for the average person. To address this complexity, most companies employ specially trained computer drafters to take input from design engineers and create the models. Thus, the individuals with the design concepts are removed from creating the digital representation. This extra step of translation between the engineer and the drafter is a hindrance, requiring the time of two people not just one. Data is often lost in communication, as well. In the context of detail design, these drawbacks are more acceptable. There is a significant advantage to having high fidelity models and simulations, so the extra time and personnel are deemed worthwhile. Conceptual design, however, deals with much lower levels of detail. Though modern software is capable of simpler modeling, users must still navigate an interface meant for someone who has undergone training. Thus, traditional CAD and FEA packages have the same usability problems whether being used for high or low fidelity tasks. With these problems in mind, several engineering software companies have released stripped down versions of their design packages intended for use at the conceptual stage. Pro/CONCEPT¹¹ and CATIA Imagine & Shape¹² are examples of this. Unfortunately, these applications are still quite complex and require substantial time to learn and use effectively.

Even for skilled users, making changes to a detailed design can be time consuming. For all the improvements CAD offers over paper drawings, altering a single dimension can have unintended effects. Thus, experience with the software and a careful eye are needed. Analysis runs can take minutes or hours to complete, depending on complexity. The design of any product is a group effort, but entire design teams do not assemble to watch the modeling of every single component. By the time design software is being used, the team has

already developed its ideas. These decisions can be communicated to software experts without the presence of everyone involved in the design. High fidelity design operates over a much longer time frame, and workflows have adapted to this downtime. Most of the actual engineering software-related work takes place between meetings, which are scheduled for planning, coordination, or studying results.

The nature of conceptual design can be very different. Instead of a design creation and evaluation timeline that extends for weeks or even months, a great deal of conceptual design can be done within a matter of hours. In this situation, a design application would be used in a large group setting. Few have the patience to sit and wait for an object to be modeled or for analysis results that take too long to be calculated.

The physical method by which a user directs action in the software environment is also part of the interface. Without a doubt, the mouse and keyboard is the most common interaction scheme around which most software interfaces are built. It has been adapted for use in every kind of application, from business to video games. The mouse originated when the Graphical User Interface (GUI) was first created. Instead of relying upon command line arguments, a user could navigate a computer visually. The mouse pointer on the screen was controlled by moving the physical mouse on a flat surface. Then, and even today still, some individuals have difficulty learning this. While fairly straightforward, the control of the mouse pointer is not direct, as it is on a touch screen or a tablet. The user is required to understand how moving the physical mouse relates to the observed movement of the mouse pointer. This concept is known as 'mapping'¹³. Most people are able to very quickly learn to use a mouse since the mapping is very simple. Moving the mouse forward, backward, left,

and right on the table results in up, down, left, and right on the screen. Both the pointer and the mouse work in two dimensional systems. In more recent years, computers became able to handle real time interaction for three dimensional environments. Since the mouse had been the standard, its use was extended into this new dimension. Another way to describe motion is in terms of degrees of freedom. Traditionally, for object motion, there are a total of six degrees of freedom. Three of those represent positional movement in the x, y, and z axes. The remaining three represent rotational motion about each of those axes. Since the mouse only moves on a flat plane and does not capture any rotation, it is only capable of motion with two degrees of freedom. Thus, a 2D device cannot perform all the actions possible in a 3D world. Because of this shortcoming, varying methods have been employed in software to convert mouse input into 3D actions. The end result of this is a more complicated mapping for the user to learn. The diagram in Figure 3 illustrates this discussion on the subject of mapping. Black arrows indicate the directions in which motion is possible while the red arrow represents an example motion made by a user. The top half of the figure depicts common mouse usage as found in almost every 2D interface where cursor movement directly mimics the physical mouse's motion. In a 3D environment, the situation is very different. An identical movement of the physical mouse provides no obvious direction for the cursor. Product design is an inherently three dimensional task, but most interfaces are still using a two dimensional paradigm. While many people have become adept at using the mouse in this way, the fact remains that using a 2D device to work in a 3D environment is not intuitive.

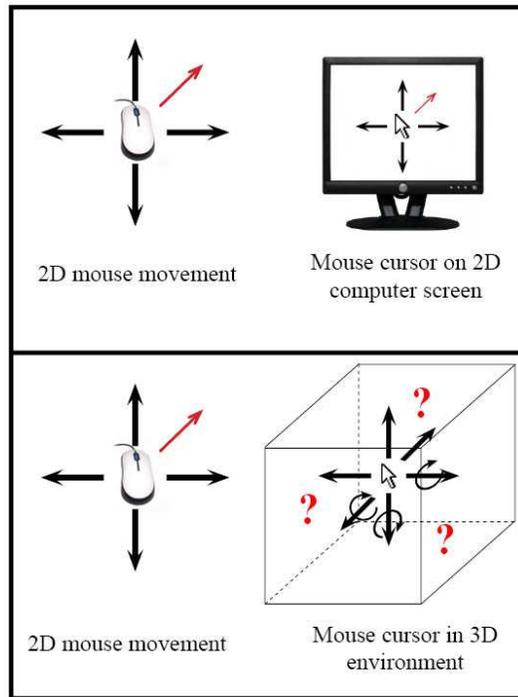


Figure 3: Diagram illustrating the concept of mapping as it pertains to the common mouse interface. When working in a 2D environment, mouse movements correlates very directly to cursor movement. However, in a 3D environment there is no obvious cursor direction.

For the reasons listed above, traditional engineering detail design software is considered to be inadequate for working with concepts. The tasks of high fidelity model creation and evaluation have very different requirements and desired outcomes than those of the conceptual design phase. The ability to improve product quality and reduce costs by leveraging the power of early design changes is not being used to its full potential. Rather than attempt to rework existing detail oriented software packages to this new domain, the focus should be to create applications with the specific goals of conceptual design in mind.

1.3 Motivation

Thus far, this discussion has dealt with the lack of suitable software packages to assist at conceptual phase and the need for the development of such a tool. At this point, the focus must turn toward the fundamental aspects that will serve as a guide for the creation of a conceptual design application. Perhaps the most important point to keep in mind is this: conceptual design is about quickly creating and assessing concepts. There are several reasons for this. First, there is never any avoiding the fact that time is money. Steep learning curves, difficult to use interfaces, and downtime spent waiting for results to finish calculating all add cost to a design as well as increase time to market. The uncertain nature of conceptual design does not warrant this kind of investment. Second, if concepts take too much time to generate, then fewer will be created, shrinking the pool of potential designs. This may result in optimal design configurations being overlooked, thus affecting product quality. Finally, conceptual design is often a group task. With a standard desktop computer interface, an application can only be operated by one person at a time. The rest of the group is likely to become impatient if their proposed ideas take a considerable amount of time to implement on the computer. Also, time consuming analysis may cause a meeting to be broken up and rescheduled for a later time, disrupting workflow. In order to avoid these issues, the rapid nature of conceptual design must be considered if an application is to be successful.

Another crucial aspect in software design of any kind is to know who the user is. The obvious answer in this case is that designers are the primary users. In most industrial settings, designers are engineers of some variety. Since many engineers have experience with CAD software, one might be led to rely upon that skill set as a prerequisite for a conceptual design

application. However, such an assumption would be premature. Recall that it is common practice for a company to employ CAD professionals. If engineers are generally not involved in this part of the design process, it is reasonable to surmise that their aptitude with the requisite software is inadequate or non-existent. Additionally, the users may not always be engineers. A design team can consist of members with diverse backgrounds (e.g. ergonomics, marketing, or administration). The team may also seek input from people involved in other segments of the product's life cycle. Customers, manufacturing workers, and service technicians all have different perspectives which could be of use to designers. It is unlikely that such a wide range of potential users all have an equal level of experience with engineering software; at best the lowest common skill set is that of the basic desktop computer interface. This must be taken into consideration when building a conceptual design application.

Serving these requirements while still providing useful design tools becomes the key issue in creating a new conceptual design application. There are many different approaches to take in order to satisfy this need. Dramatic improvements in usability can be obtained by presenting the user with a simple interface. This can be accomplished by reducing the number of options presented to the user and thus decrease the cognitive load. A small set of well designed features can be both powerful and easy to navigate. Therefore, it may be helpful to begin with the most basic functions; more can be added as needed, provided they do not complicate the interface. Following this approach, inspiration can be derived from one of the simplest design tools, the sketch. A sketch is quick, easy, and relatively without limits. The one restriction is drawing ability; not everyone is an artist. The downsides are that

sketching is two dimensional and non-interactive, sometimes making it difficult to communicate a concept. These drawbacks can be resolved in a digital application. In a sense, CAD software accomplishes this. Object creation and manipulation are fundamental components of any design software. The creation of larger, more complex objects is enabled through the use of assemblies. These operations should be the basis of a conceptual design tool, though in a more free form manner than what exists in detail oriented software. In addition to concept generation features, methods for assessment must be included. Any evaluative tools need to be on the same level of complexity as the concept models. Also, because the focus is on conceptual design, a certain degree of accuracy can be sacrificed for speed. With sufficiently fast evaluation methods, an application could be built that combines concept generation with real time assessment in the same design environment.

Additional improvements could be realized by no longer relying upon the standard 2D, mouse based interface. It is fair to assume that the vast majority of users are familiar with working in a six degree of freedom environment. Every day, real world tasks can be described in this system. Implementing a six degree of freedom interface would provide users with as direct an interaction as possible. Without a complex, non-intuitive mapping to understand, the target application would be simpler to use and more accessible to a wide audience.

While not exactly a feature, there is another item that should be taken into account when building a conceptual design application. Legacy data is a general term for any previously generated, calculated, or collected data. In the context of industrial product design, this information most often takes the form of digital models from earlier product lines or versions

and the accompanying analysis results. Most products are not designed entirely from scratch, so designers often use this data as a starting point. Current design software is not entirely ill-suited for incorporating legacy data when working on new designs with only minimal changes from previous products. However, as the pace of the market continues to accelerate, companies making small design alterations will find themselves falling behind. The ability to investigate major product advancements while taking advantage of the vast information present in legacy data will allow a company to remain competitive.

These last several paragraphs have described the direction of the research being presented. This thesis is focused on the creation of a conceptual design tool that facilitates the rapid generation, evaluation, and improvement of concepts taking into account the uncertainty of early design possibilities.

1.4 Thesis Organization

The remainder of thesis is organized as follows: Chapter 2 presents background information on the origins of this project as well as the primary research areas investigated for creating the final application: 1) augmented reality and 2) metamodeling; Chapter 3 describes the creation a three dimensional, augmented reality design environment; Chapter 4 details the procedure followed for building metamodels to approximate two example design situations; Chapter 5 presents the results of experimental trials as well as performance data of the application; finally, Chapter 5 concludes the work presented with a discussion of the research findings as well as the direction of future work.

2 BACKGROUND

2.1 Conceptual Design Applications

Improving the design process by leveraging advantages in the early stages and developing computer applications for that purpose are not novel concepts. In a paper titled “Immersive Product Configurator for Conceptual Design” the authors present a system called the Advanced Systems Design Suite (ASDS)¹⁴. The abstract of this paper is included below:

Currently, new product concepts are evaluated by developing detailed virtual part and assembly models with traditional Computer Aided Design (CAD) tools followed by appropriate analyses (e.g., finite element analysis, computational fluid dynamics, etc.). The creation of these models and analyses are tremendously time consuming. If a number of different conceptual configurations have been determined, it may not be possible to model and analyze each of them. Thus, promising concepts might be eliminated based solely on insufficient time to assess them. In addition, the virtual models and analyses performed are usually of much higher detail and accuracy than what is needed for such early assessment. By eliminating the time-consuming complexity of a CAD environment and incorporating qualitative assessment tools, engineers could spend more time evaluating additional concepts, which were previously abandoned due to time constraints. In this paper, a software framework, the Advanced Systems Design Suite (ASDS), for creating and evaluating conceptual design configurations in an immersive virtual reality environment is presented. The ASDS allows design concepts to be quickly modeled, analyzed, and visualized. It incorporates a PC user interface with an immersive virtual reality environment to ease the creation and assessment of conceptual design prototypes. The development of the modeling and assessment tools are presented along with a test case to demonstrate the usability and effectiveness of the framework.

Obviously, the motivations and goals of the ASDS project have a great deal of overlap with those of this thesis. The research presented here was done as to complement the larger, more developed ASDS framework.

Another noteworthy application is Google SketchUp¹⁵. SketchUp was intended to bring 3D design to everyone. The interface was purposefully kept very simple. Figure 4 shows an example model designed with the SketchUp application. Despite only having access to a small number of tools, users are able to create detailed models of objects and structures. The single toolbar at the top of the SketchUp screen provides a stark comparison to the interface presented in Figure 4. Goggle offers both a free and professional version of the software. The free version has been downloaded by many users around the world, and the software's community page is full of testimonials and case studies where Sketchup was successfully applied in a variety of disciplines, including engineering¹⁶. This is an excellent example of how design tools created with a focus on an intuitive interface can have a significant impact on the design process. By greatly reducing the entrance barrier to 3D design, more people are able to create and communicate physical concepts.

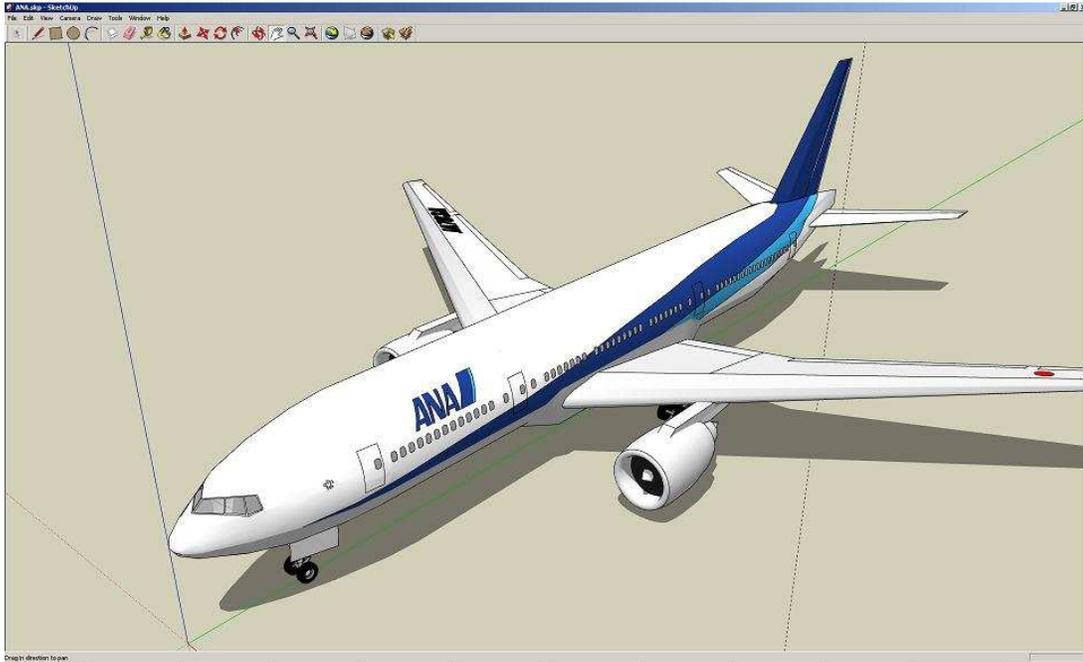


Figure 4: This screen capture of Google SketchUp demonstrates the simplicity of the interface. SketchUp uses only a handful of tools; this is very different than crowded interface shown in Figure 2.

2.2 Augmented Reality

Augmented reality is a developing research area that aims to “augment” human senses with digital information while in real world environments. AR preserves the perception of the real world while simultaneously adding computer generated enhancements. Though similar in some respects, AR is distinct from the more commonly known virtual reality (VR), an environment which is entirely synthetic. One way to conceptualize these differences is to imagine a spectrum with one extreme being the real world and the other being a pure virtual environment. Milgram and Kishino define such a concept in the virtuality continuum¹⁷. An adaptation of this continuum is pictured in Figure 5. The figure

also introduces the term mixed reality. Though sometimes used interchangeably with AR, mixed reality (MR) is a more general term defining anything between either of the two extremes. Augmented reality is placed nearer to the real world end of the spectrum due to the fact that the user is primarily sensing the real world, with less information coming from digital sources.

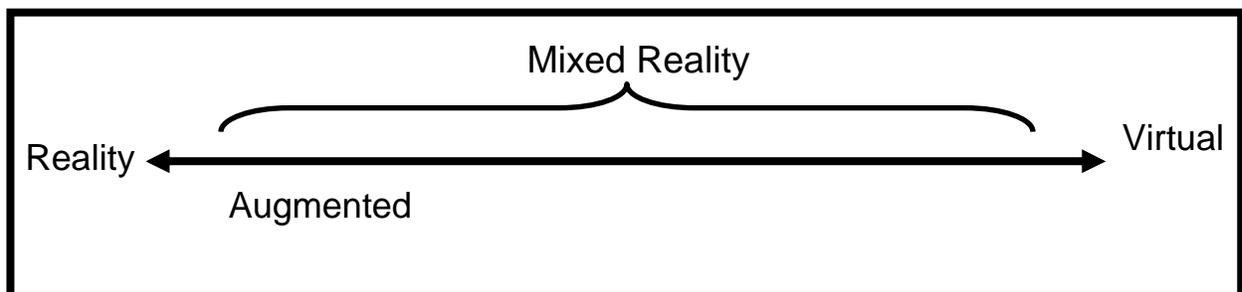


Figure 5: An adaptation of Milgram's virtuality continuum. Augmented reality exists between the two extremes: the real world and pure virtual reality.

By definition, AR can encompass the augmentation of all five senses. An increasing number of projects are focused on augmenting hearing and touch^{18,19}. However, this research, as well as the remainder of this paper, focuses on how AR systems function for the sense of sight. AR was first realized when Ivan Sutherland created the head mounted display (HMD); this system is also often considered the first VR display as well²⁰. Through the HMD, the user was able to see a wireframe model of the room in which the system operated. Augmented reality has advanced a great deal since then. Figure 6 shows an example of a modern AR application. Technological progress has allowed AR systems to take many different forms for an even greater number of applications.

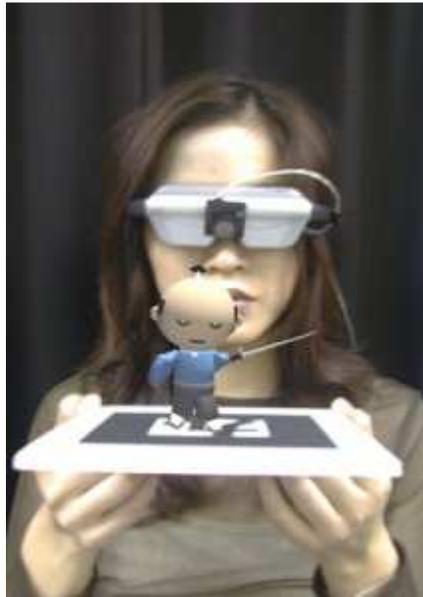


Figure 6: Example image of AR application showing virtual objects in the context of the real world.

A survey by Azuma defines three requirements for a modern system to be considered augmented reality²¹:

- Real and virtual objects are combined in a real environment.
- The system is interactive in real time.
- The system registers (aligns) real and virtual objects with one another.

The first definition is fairly obvious as it is the most fundamental idea upon which augmented reality is based. It is also concise restatement of AR's placement on the virtuality continuum in Figure 5. The next two statements place further conditions on this primary concept. By requiring real time interaction, it becomes necessary that a user is able to affect the augmentation without significant delay. For example, adding digital elements to a pre-recorded video would not be considered AR. The last requirement calls for the system to use

some manner of registration. In the context of augmented reality, registration is how a system is able to observe events in the real world and update the digital scene accordingly. While these requirements define the function of an AR system, there are no limits placed on how these goals are accomplished. The various technological approaches used to create augmented reality environments are explored in the following paragraphs.

2.2.1 Registration

Registration in augmented reality is achieved using a variety of tracking or detection technologies. As mentioned earlier, the first AR system was mechanically tracked²⁰. Using an arrangement of linkages, The Sword of Damocles system was able to determine the user's head position and orientation. With this information, the computer updated the images shown in the HMD to provide the correct perspective of the virtual scene. While this method is still technically possible, the mechanical approach has been abandoned in favor of less cumbersome methods. The fundamental goal of AR registration is to determine the position and orientation of one or more real world objects in relation to the view of the user.

Optical, or camera-based, systems are one class of technology used for the task of registration. Cameras can be employed in many different configurations to observe a real world scene. ARToolKit is a very common AR library in which a camera detects fiducials, specialized markers which must be known by the system in advance²². As ARToolKit is a major component of this work, a more detailed introduction is presented in another section. In general, optical systems can be much less expensive than other options, ARToolKit being the prime example of this. The only additional hardware required is a camera, which can be

as simple as a basic webcam. Optical systems have significant disadvantages as well; occlusion is the greatest of these. If the camera's view is obstructed in any way, the system will only have limited information about the scene or no information at all. Both dim and harsh lighting can also have an adverse effect on recognition and tracking. Other optical solutions make use of multiple cameras and the infrared (IR) spectrum. Systems such as those offered by Advanced Real-time Tracking use active markers (i.e. IR emitting) to obtain very precise and robust tracking²³. One downside to this approach is that the cameras must remain in a fixed location, so the tracking is only available within a limited range. Optical systems also have the potential to perform marker-less tracking. The Z-Cam by 3DV systems offers this kind of capability²⁴. This camera, and others like it, emit IR light and record the time it takes for the light to return to an IR sensor. What results is a depth field representing the distance of any object in the camera scene to the camera itself. Used in a multiple camera setup, this methodology eliminates the need for wearing additional tracking equipment.

A different approach to AR registration requires attaching physical sensors to the object being tracked. These sensors come in several varieties, each using different technology to obtain position and/or orientation information. Inertial sensors use accelerometers or gyroscopes to determine relative motion and rotation of an object. If desired, the relative motion can be integrated to obtain position. Magnetic-based systems operate within an electromagnetic field created by a base station. Individual sensors measure certain field properties from which position and orientation can be calculated. Ultrasonic tracking is similar to magnetic; however, the sensors measure the properties of sound waves instead. On a larger, outdoor scale, registration has been performed using a data from a GPS device²⁵.

The methods discussed here are only a brief introduction into a wide realm of possibilities. Each individual approach has distinct advantages as well as drawbacks. For example, occlusion is not an issue for most of these technologies, but magnetic/ultrasonic tracking is only valid within a certain distance of a base station and is vulnerable to interference. To address this, researchers have developed hybrid methods that combine two or more registration technologies, such as Intersense trackers that use both inertial and ultrasonic sensors²⁶. Though they add complexity, hybrid methods offer the potential of very stable tracking results.

2.2.2 Display Configurations

Creating the illusion of virtual objects existing in the real world is another aspect of augmented reality. The display systems that accomplish this task can be divided into two categories. The first is video see-through. A camera takes in images of the real world and displays them to the user after the scene has been augmented. An opaque display screen is positioned between the user and the scene. Thus, the user's vision is at least partially obstructed, and he or she must rely on the video to see the real world. Video see-through is primarily displayed using three configurations. The first arrangement uses an HMD with an attached camera (Figure 7). The user sees video captured by the camera in whatever direction he or she looks. A second method attaches a camera to a display screen. This creates a sort of lens through which a user can look, this method is very common for mobile applications using handheld devices (Figure 8a). The third method creates a "reach in" style system. This configuration uses a mirror attached to the back of a display device. The mirror is directed

down toward the surface of a table. A camera captures this reflection, and the image is displayed to the user. Figure 8b shows how this type of system is built. The second AR display category is optical see-through. In these systems, the goal is to preserve a direct view of the real world and only draw the virtual objects. Systems such as these are often seen in military applications as eyepiece and head up displays. Ideally, an optical system is preferred over a video one. Since cameras and displays screens have resolution limits well below what the human eye can see, forcing a user to view the world through such devices can interfere with natural vision. Optical see-through systems come in a variety of configurations similar to those found in video see-through systems: HMDs, handheld “lenses”, large fixed display screens (Figure 9). One drawback to most optical see-through systems is that real world view is dimmed since only a certain proportion of light is able to pass through semi-transparent materials.

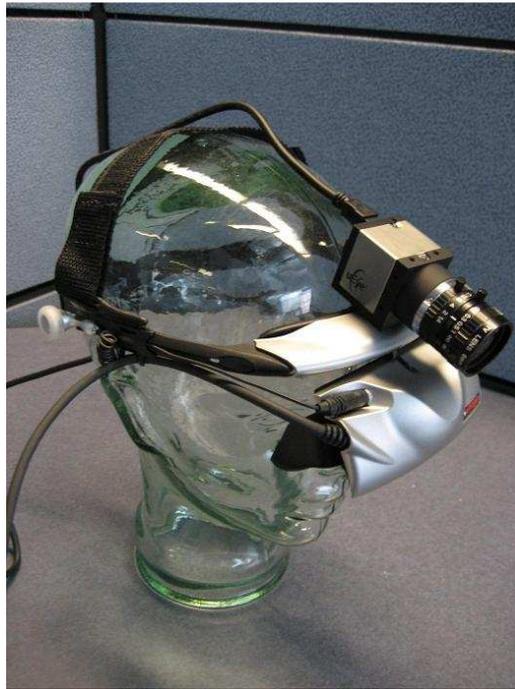


Figure 7: One example of a video see-through display using an HMD and a camera.



(a)



(b)

Figure 8: (a) A handheld video see-through device. (b) A video see-through desk configuration that uses a mirror mounted on the back of the display to augment a table top scene.

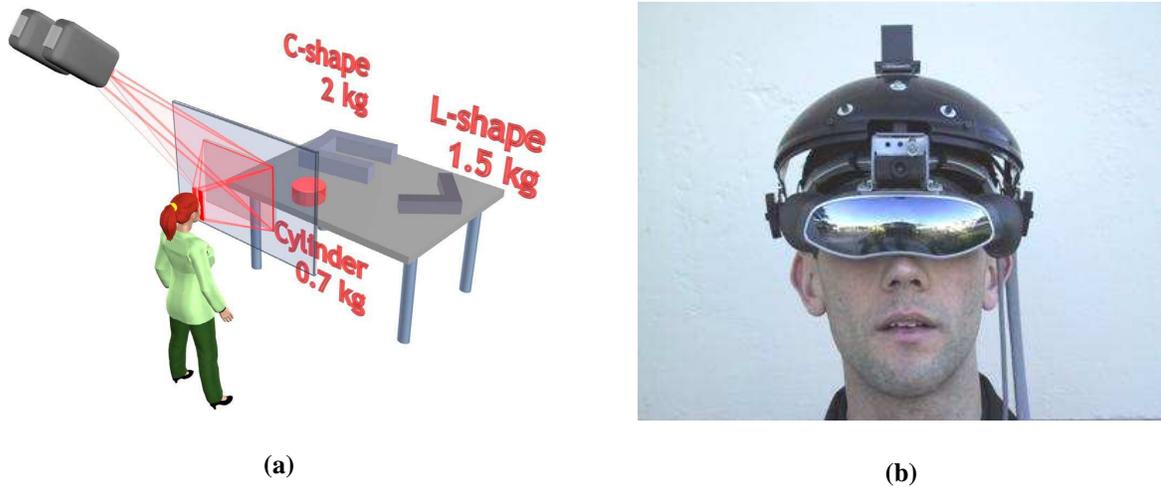


Figure 9: (a) A conceptualization of a large scale optical see-through system. (b) One example of an optical see-through head mounted display²⁷.

2.2.3 Applications

Augmented reality has been applied in many varied disciplines. As both an interaction and visualization technology, it is very flexible. Many AR applications have been created for use in the fields of entertainment and education. Museums, for example, can use AR to enhance an exhibit or guided tour²⁸. The MagicBook project takes the concept of a pop-up storybook into the realm of AR²⁹. This first person shooter concept is also used in the game ARQuake³⁰. With applications like these, the sense is that the game environment enters the player's world not vice versa as in traditional games.

More serious applications for augmented reality exist as well. One of the most rapidly developing fields is AR for use in medicine. Several projects are focused on creating intra-operative tools that enhance the surgeon's view of a patient. In a letter to the Journal of the American Medical Association, Dr. Jacques Marescaux describes the use of augmented

reality to assist in a laparoscopic adrenalectomy³¹. In this particular procedure, a digital model of the patient was generated prior to the surgery and manually registered with a live video feed during the operation. It was found that the AR imaging was very helpful in localizing the tumor, adjacent organs, vessels, and especially the main adrenal vein. While the procedure used a manual registration, other research is being done to allow automatic registration of a patient³²⁻³⁴. The fundamental goal of this research area is to essentially give the surgeon live “x-ray vision” of a patient’s anatomy. Using previously obtained scan data, computer generated medical models are aligned with a patient’s body. With semi-transparent models and a video see-through display, an operating surgeon can see both the patient and his or her internal structures simultaneously. It is hoped that this kind of advanced, real-time visualization will improve a surgeon’s performance and reduce error rates.

Augmented reality has also been applied for use in industry and research. One project used AR to prototype a cockpit interface³⁵. This example highlights many of the benefits allowed by this technology. Since the real world is preserved, it is not necessary to create digital models for the entire cockpit. The user is free to see these objects as he or she would normally, adding a greater degree of fidelity to the experience. Merging the real and virtual has other advantages as well. A user can directly interact with the entire environment. Reconfiguring the cockpit is a simple matter of moving certain markers. Evaluation is very natural as well; a pilot would be able to sit down and know if a switch is out of reach or if a particular gauge is too small to read. Augmented reality can also be used to provide assistance for real world, industrial tasks. One of the earliest examples of this type of application was begun by Boeing in the 1990s³⁶. In this project, the user is a manufacturing

technician located on the work floor. When looking at a particular work piece, relevant information appears in heads-up display. Wiring diagrams indicate the location of electrical components, and instructions are given via text. A more recent project employed AR for the task of furniture assembly. In *Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures*, the authors present an application which prompts the user with visual cues giving detailed assembly instructions³⁷. The system will indicate the next step in the procedure while highlighting the appropriate parts as well as their final arrangement. The effectiveness of AR's capability for this manner of instruction was evaluated by Pathomaree and Charoenseang in a paper titled *Augmented Reality for Skill Transfer in Assembly Task*³⁸. In this study, the task of interest was to complete a small, wooden puzzle. Participants completed both 2D and 3D puzzles with or without AR guidance. For the cases where AR was used, a video see-through display instructed the user on the next step via text while visually highlighting the next piece and its correct location. If the user places a piece incorrectly, the system determines the error and informs the user of a solution. A correct placement will prompt the next step. Experimental results from this study indicate that AR has the potential to drastically reduce both the completion time and number of steps required to reach the goal. Object manipulation and assembly is another area in which AR has been applied. Sidharta, Oliver, and Sannier published a paper presenting an alternative to 2D interfaces for design review called the Augmented Tangible Interface (ATI)³⁹. In this application, AR markers and voice command create a hands-on interface for product assembly. All available objects can be explored by shuffling through a stack of cards. Any of these objects can be transferred to a cube; all six sides of this cube are tracked and show

different orientations of an object. The objects from any two cubes can then be joined together at predetermined nodes. All of these operations are controlled using a list of voice commands issued through a microphone. The ATI is a great example of how to leverage the capability of AR technology for full 3D interaction.

This brief survey has only scratched the surface of how and where augmented reality has been applied. As previously mentioned, it is a very versatile technology that is only now seeing more mainstream attention. Though virtual reality is a generally more familiar concept, AR may become better known and more widespread in the near future. The reason being that the cost of an entry level AR system is significantly less than that of a VR setup. Recently, a game was released for the Sony Playstation 3 that features AR technology. *The Eye of Judgment*⁴⁰ is a trading card game which uses a peripheral called the Eye, essentially a webcam, to track events on the game board. The video game and consumer electronics industries can be an enormous driving force in the advancement of digital technology. On such a scale, it has the effect of greatly reducing the price of equipment. These factors may lead to a greater market for AR applications as well as better display and tracking technology.

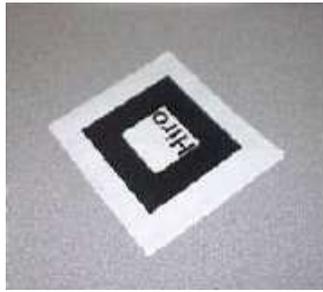
2.2.4 Software Libraries

The Augmented Reality Toolkit (ARToolKit) is one of the most common platforms for creating AR applications. At the time of writing, the project's SourceForge page had nearly 140,000 downloads⁴¹. The ARToolKit library has also been the foundation for several other popular projects including ARTag⁴², ARToolKitPlus⁴³, Studierstube ES⁴⁴, and

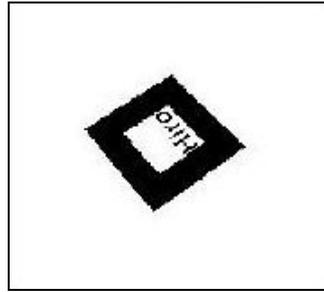
OSGART⁴⁵. Hirokazu Kato and Mark Billinghurst were the original authors of the library, first demonstrating it in 1999⁴⁶. Since then, the project has undergone numerous improvements and had many contributors. ARToolKit is a camera based system that tracks fiduciary markers in a real world scene. Each marker consists of a thick, black border surrounding a unique image at the center. In order for tracking to work, every marker must first be trained into the system. Training a marker is done using a simple program that is distributed with the library itself. A pattern file is generated for each marker that can be accessed by the toolkit during runtime. Figure 10a shows an example of an ARToolKit marker. The process by which markers are tracked is detailed in the following paragraph.

The ARToolKit detection algorithm can be generalized into several steps, each applying a different computer vision principle. Figure 10 summarizes the algorithm and shows a sample image from each step. The first step is simply the input, a single video frame. Next, the image is thresholded. Thresholding is the process of taking an image that is in color or grayscale and converting it to pure black and white. Following that, a connected components routine determines how many separate objects exist within the frame. Running a contours method on the image displays the edges of any objects in view. Provided that the contours for each object found by the connected components meet certain requirements, the edges and corners of the perceived marker are extracted. Before the final step, fitting a virtual object to the marker image, a great deal of additional calculations must be done. A full explanation of this process is not warranted in this introduction. Billinghurst and Kato's publication more fittingly details this computation. The end result of the ARToolKit

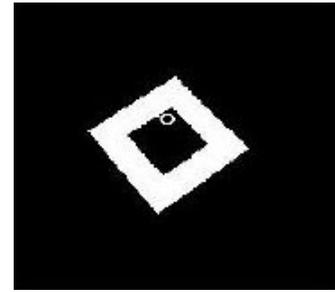
algorithm is the position and orientation, in the camera's coordinate system, of any recognized marker.



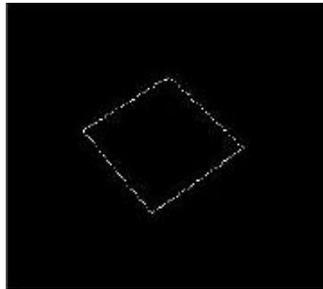
a. Original image



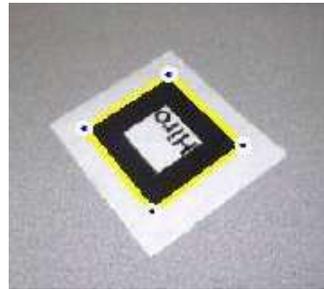
b. Thresholded image



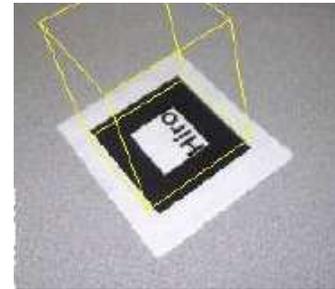
c. Connected components



d. Contours



e. Extracted marker edges



f. Fitted square

and corners

Figure 10: A step by step breakdown of the ARToolKit tracking algorithm. (a) Video frames are captured from a camera device. (b) The image is converted to a binary (i.e. black and white) format. (c) A connected component algorithm labels connected pixels.

(d) Edge detection is used to highlight the outer lines of the marker. (e) The marker edges and corners are recognized. (f) Pose estimation is used determine the marker's position and orientation.

The OpenSceneGraph (OSG) is an OpenGL-based scenegraph manager⁴⁷. It has been used extensively in the creation of graphics applications such as flight simulators, games, virtual reality, and scientific visualization. A general description of a scenegraph is that it is a

way to organize a 3D scene into a tree structure. Simply put, each of the objects being drawn is represented as a node. Matrices are applied to control object position and orientation. Various other object states like color and transparency can be controlled using the OSG library. Parent/child relationships can be created between objects. Thus, many nodes can be easily grouped to function as a single entity. Written entirely in C++, OSG creates an object oriented framework on top of OpenGL. With this capability, developers are no longer required to deal with low level graphics routines. Instead, they are able to work at a higher level, while most of the tedious code exists “behind the scenes”. OSG also does a great deal of optimization automatically. Another valuable feature is OSG’s collection of 3D database loaders. These plugins are able import from a wide variety of model formats such as 3D StudioMax (.3ds) or Alias Wavefront (.obj). The end result of the advantages listed above is that developers can create high quality graphics applications in a much shorter time frame.

Until recently, ARToolKit and OSG were not available as a single library though each had functionality that was desirable for those developing AR applications. In late 2006, ARToolworks released OSGART: ARToolKit for OpenSceneGraph. The OSGART library makes use of all the tracking capability of ARToolKit. However, developers now have access to all the features of OSG when developing applications. With this expanded capability, AR applications can be created more easily with improved functionality.

2.3 Metamodeling

As mentioned in the introduction, high fidelity simulation of digital models, FEA for example, can be expensive in regard to both time and computational resources.

Metamodeling offers a less burdensome alternative to such methods. In essence, a metamodel provides a surrogate model to be used in place of the original system. The term itself comes from the fact that metamodels are often “models of models”⁴⁸, meaning that the model is often a further approximation of a more computationally expensive numerical simulation. However, metamodels can be developed using data from any system, regardless of whether or not that system is itself a simulation. Modeling techniques vary a great deal, but each approach is based on having a finite set of known responses from the system of interest. Global methods attempt to predict the entirety of a design space, or at least a significant portion of it. Local methods are only valid in the vicinity of a specific point or within a certain confidence interval. Approximation methods provide a best fit of the source data according to some metric representing the model’s accuracy. An interpolating method will fit a solution through each known response, guaranteeing accuracy at those points. These brief descriptors provide some indication of a method’s function but are by no means an exhaustive list of all possibilities. They are merely presented to provide some familiarity with the topic. A recent survey published by Simpson et al presents a thorough review on both past and current metamodeling techniques which explains this research area in greater detail

⁴⁹.

The following sections provide background information on the specific modeling approaches followed in this research. First, the area of Design of Experiments (DOE) will be introduced. Next, the two modeling methodologies will be presented: Polynomial Response Surface (PRS) and Polynomial Chaos Expansion (PCE). Both of these methods are

considered to be global approximation techniques. Similarities do exist between PRS and PCE models, but the differences in their formulations will be explained.

2.3.1 Design of Experiments

Design of experiments (DOE) is a broadly defined term that is used in many disciplines. When used in the context of computer experiments or simulations, it refers to the data gathering method used to analyze an unknown process. How a design space is sampled can greatly impact the accuracy of the resulting model. There are many different experimental designs to choose from, such as full factorial, uniform, random, latin hypercube, and orthogonal arrays⁵⁰. In random sampling, data points are generated by randomly sampling each dimension according to a specified distribution. This strategy does not guarantee that the design space is evenly or fully sampled. However, there is no restriction on the number of data points created. An orthogonal array is a matrix that can be characterized by a few variables. The number of rows, n , is the number of experiments to be performed. The total number of columns is k , each representing a dependent variable of the experimental system. Within each cell is a value that indicates a level value for that experiment and variable. An array has l levels; these levels represent the discrete value for a particular variable. While these matrices can be generated according to desired characteristics⁵¹, they can also be chosen from large libraries of pre-generated orthogonal arrays found on the Internet⁵². Orthogonal arrays provide a more balanced sampling of a design space, but the number of experiments cannot be changed without destroying that balance.

2.3.2 Polynomial Response Surface

Response surface methodology is widely used throughout the engineering community⁵³. RS models are designed to approximate data sets using polynomial expressions. Response surface models take the form of Equation (1):

$$y(x) = f(x) + \varepsilon \quad (1)$$

where $y(x)$ is the unknown function of interest, $f(x)$ is a known polynomial function of x , and ε is random error. The random error is assumed to be normally distributed about zero. The known polynomial function, $f(x)$, is generally a low-order polynomial. In order to satisfy more non-linear behavior, higher order polynomials can be used but require large numbers of sample points to satisfy the coefficients in the polynomial equation. In Equation (2) the polynomial equation is linear. Equation (3) shows a second-order or quadratic expansion of the polynomial equation.

$$\hat{y} = \beta_0 + \sum_{i=1}^k \beta_i x_i \quad (2)$$

$$\hat{y} = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i=1}^k \sum_{j=1}^{i-1} \beta_{ij} x_i x_j \quad (3)$$

The coefficient parameters, β_0 , β_1 , β_n , and β_m , of the polynomials in Equations (2) and (3) are determined through least-squares regression. Least-squares regression first calculates the partial derivatives of the coefficients then minimizes the sum of the squares of the residuals of predicted values, $\hat{y}(\mathbf{x})$, from the actual values, $y(\mathbf{x})$. The basic formulation for least squares regression is shown in Equation (4):

$$\beta = [X'X]^{-1} X'y \quad (4)$$

where X is the design matrix of sample data points, X' is its transpose, and y is a column vector that contains the values of the response at each sample point.⁵⁴

2.3.3 Polynomial Chaos Expansion

Originally presented by Wiener, polynomial chaos expansion is a method for representing a stochastic process with orthogonal Hermite polynomials⁵⁵. The use of Hermite polynomials assumes that the process is Gaussian. In the context of this paper, the term PCE will refer to the original formulation of Wiener-Hermite chaos. The more generalized Wiener-Askey scheme can be applied to handle non-Gaussian distributions⁵⁶. Using a PCE approach, the random process $X(\theta)$ can be represented in the following form:

$$\begin{aligned} X(\Theta) = & a_0 H_0 + \sum_{i_1=1}^{\infty} a_{i_1} H_1(\xi_{i_1}(\Theta)) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} H_2(\xi_{i_1}(\Theta), \xi_{i_2}(\Theta)) \\ & + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} H_3(\xi_{i_1}(\Theta), \xi_{i_2}(\Theta), \xi_{i_3}(\Theta)) + \dots \end{aligned} \quad (5)$$

where $H_n(\xi_{i1}, \dots, \xi_{in})$ represents the Hermite-chaos of order n in the standard Gaussian variables $(\xi_{i1}, \dots, \xi_{in})$. Variables $a_0, a_{i1}, \dots, a_{ip}$ are deterministic constants. H_n are the Hermite polynomials, developed using the general expression in Equation (6):

$$H_n(\xi_{i_1}, \dots, \xi_{i_n}) = e^{\frac{1}{2}\xi^T \xi} (-1)^n \frac{\delta^n}{\delta \xi_{i_1} \dots \delta \xi_{i_n}} e^{-\frac{1}{2}\xi^T \xi} \quad (6)$$

In the one dimensional case assuming a normal distribution, the first three Hermite polynomials are found to be:

$$\begin{aligned} H_0 &= 1 \\ H_1 &= \xi \\ H_2 &= \xi^2 - 1 \\ H_3 &= \xi^3 - 3\xi \end{aligned} \quad (7)$$

ξ can be evaluated from the input variable, x , according to the following equation.

$$\xi = \frac{x - \mu_x}{\sigma_x} \quad (8)$$

In which μ_x is the mean of the distribution and σ_x is the standard deviation. Once the polynomial expression is expanded, solving for the coefficients is done using least-squares regression, Equation (4).

2.4 Research Issues

Based on the review of research in the fields of augmented reality and metamodeling as they apply to conceptual design, two research issues have been identified:

1. Investigate the use of augmented reality as an effective tool for the creation of conceptual designs.

Engineering design software has relied upon the mouse as the standard interface. This two dimensional device is not capable of performing all the actions possible within a three dimensional design environment. The ability of augmented reality to capture six degree of freedom motion and display virtual elements within the user's own world makes it an ideal choice for enhancing user interaction and creating conceptual designs in a fast and intuitive manner.

2. Investigate the use of metamodels for creating accurate approximations to provide real time assessment data of conceptual models.

Generally, most metamodeling applications are focused on replacing higher fidelity analysis for very specific, well defined problems. Early design analysis is an issue that has been relatively overlooked, mostly due to the uncertain nature of the conceptual phase. With minimal data and generalized models, valuable information can be obtained at the crucial early phases in the design process.

3 AUGMENTED REALITY INTERFACE ENVIRONMENT

Augmented reality provides the foundation for most of the application. The user interface and the majority of features are all supported by an AR environment. Augmented reality applications require a very different style of interaction than what is found in everyday software. The standard concept of a graphical user interface has evolved with the Windows Icon Menu Pointer (WIMP) model as its focus. In an AR application, the digital environment coexists with the real one. An inelegant solution would be to directly copy WIMP elements into an augmented scene. However, the user would now be forced to deal with various windows and menus obstructing his or her view of the world. While the usability principles formed over the last few decades are still applicable, new interfaces must be developed for AR environments. This section will detail the interface elements, hardware and software, that were used in this project. Since many of the application's features are directly related to AR's capabilities, these too will be explained.

3.1 Hardware

3.1.1 Display

While the application will function with various display configurations, it has been designed to be operated using a video see-through head mounted display (HMD). Pictured in Figure 11 is an eMagin z800 HMD with a Logitech QuickCam 9000 Pro. The HMD displays at a resolution of 800 x 600. The camera is typically configured to capture video at 960 x 720 or 800 x 600 pixel resolution and 15 or 20 frames per second, respectively. This particular configuration was chosen over others for several reasons. With the both the display and camera worn on the head, the user is free to use both hands for interaction. Though the

desktop configuration displayed in Figure 8b has its benefits, it also limits the space in which users can work. Such a rig is not easily portable and is only capable of capturing the table space beneath it. An HMD allows a certain degree of portability as well as a very direct view of the augmented scene.

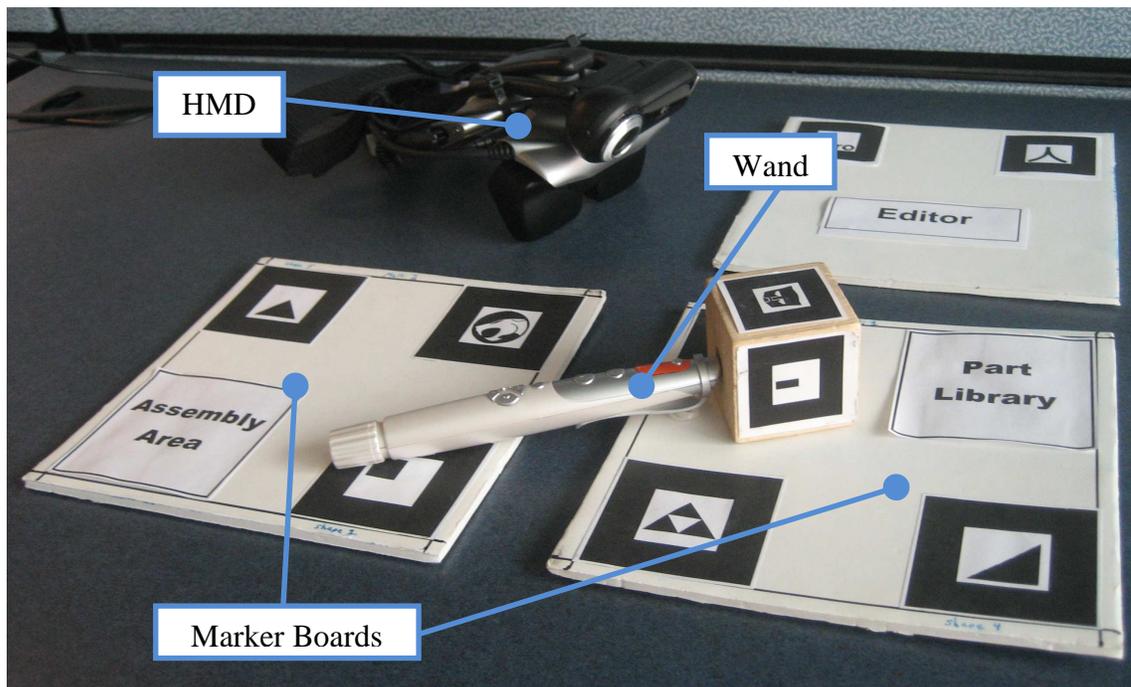


Figure 11: A photo of the hardware interface elements (HMD, marker boards, and wand).

3.1.2 Marker Boards

While the software significantly affects how the markers function, the physical configuration of the markers must also be considered. In ARToolKit, a multimarker is a collection of several single markers that function as one. As long as one of the individual markers is visible, the entire multimarker is still recognized in software. Multimarkers were

used in this application in order to prevent accidental occlusion of the interface elements. Examples of the marker boards used in this application are pictured in Figure 11. Each board is labeled to provide easy identification of its function.

3.1.3 Wand

In the AR environment created for this application, the vast majority of the interaction is performed using a device called the wand. The tip of this wand is a cube. Four faces of this cube have markers attached for tracking purposes. The handle of this wand is a presentation remote with several buttons using radio frequency (RF) technology to wirelessly communicate with the computer. Three of these buttons are used within the application to signal various actions. The specifics of these functions are described later in this section. The whole wand is shown in Figure 11, while Figure 12 displays the three buttons used to perform various actions within the application.

Two previous interaction schemes were used before the wand was implemented. First, events were triggered using keyboard input. The primary problem with this method was the difficulty involved with pressing keys while still observing the AR scene. Every time the user wished to give a command, he or she had to divert attention from the conceptual design task and locate a specific key. This action also required the use of one hand, meaning any other interaction had to be one-handed. The second interaction scheme used voice recognition technology. Short command phrases were chosen to trigger events within the application. While this approach has solved the problems of the previous method, it presented an entirely new set of difficulties. The voice recognition library did not function accurately. Spoken

commands were often misinterpreted or not detected at all. In other situations, events were triggered without any verbal command being given. Generally, the recognition only functioned in quiet environments in which the only speech was commands to the computer. Any situation with background noise or conversation rendered the application unusable. Both of these methods also shared a significant drawback; they required to remember a list of keys or phrases along with the associated function. As the number of functions increased, so did the list. This burden on memory also had a negative impact on usability. The current scheme resolves all these issues. Every function is reliably controlled by buttons which are always accessible to the user without distraction. With only three possible buttons, there is no large list of commands to commit to memory. The following section on software development will explain how these few buttons manage a larger number of functions.

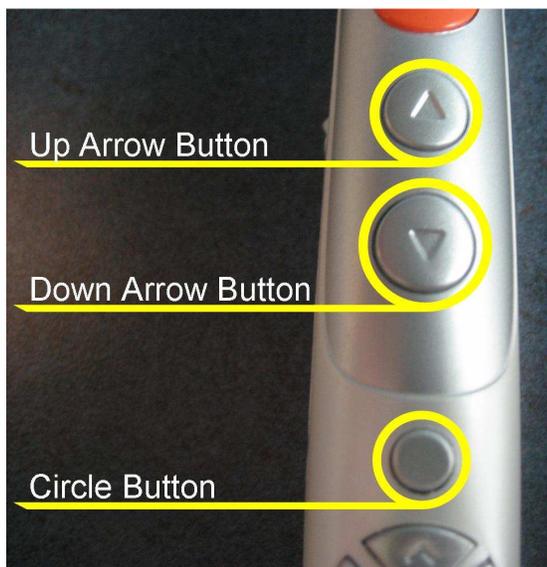


Figure 12: A close-up view of the wand buttons.

3.2 Software

3.2.1 *Wand*

As mentioned above, the wand is the primary interface method. Conceptually, the wand is designed to be used as a three dimensional mouse, keeping the familiar “point and click” functionality of the standard mouse. The primary difference is that the wand offers interaction in all six degrees of freedom. A user is free to move the wand around to any point within the AR scene in front of him or her. Visually, the user sees the physical wand in his or hand as well as an AR pointer which represents the wand’s position in the AR environment. The four markers on the tracked cube are each coded with a different rotation allowing the wand to be used at almost any orientation. The three buttons on the wand signal different operations based upon the context in which they are used. This contextual use is determined based upon the wand’s position in the augmented scene. If it is within a certain distance of one of the marker boards, it changes its behavior to reflect the tasks available for that specific board. The set of features available to users will be explained below in context with the relevant marker board.

3.2.2 *Part Library*

The first interface the user encounters is the part library (Figure 13). This multimarker board displays the various models available to the user for subsequent assembly. These models are currently loaded in from locations specified by a text file when the application launches. While many models can be loaded, only nine are displayed at a time. When interfacing with the library, the wand has the following functions: selecting objects; grabbing

objects; clearing the wand of a grabbed object; and cycling through different groups of objects within the library. Object selection is performed by simply pointing at the desired object. Whichever model is nearest to the wand's pointer is selected, and this action is displayed to the user by increasing the size of the model. This visual feedback is shown in Figure 14. Once selected, any object can be grabbed by pressing the circle button. This will attach a copy of the selected model to the wand, maintaining the orientation of the model in relation to the wand (i.e. the model is not transferred to the wand at a prescribed rotation). Figure 15 shows the wand with an attached object. Now, a user can inspect the model on the wand independent from the rest of the objects in the library. If the user is not satisfied with the grabbed orientation or if he or she wishes to grab another part, pressing the down arrow button will clear the wand of the current object. The up arrow button will cycle to another "page" of models, giving access to all the objects loaded at start up.

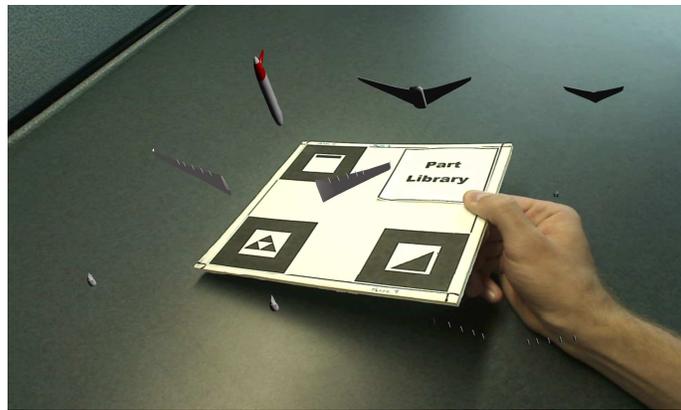


Figure 13: The part library contains a collection of models that can be used to create assemblies.

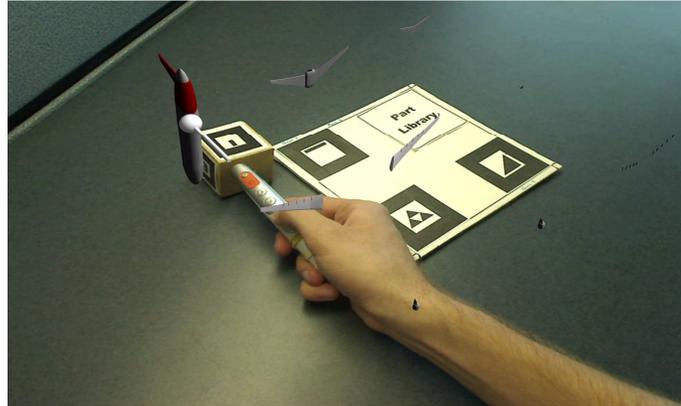


Figure 14: Object selection is done with the wand pointer. Increasing the model's scale provides visual feedback to the user.

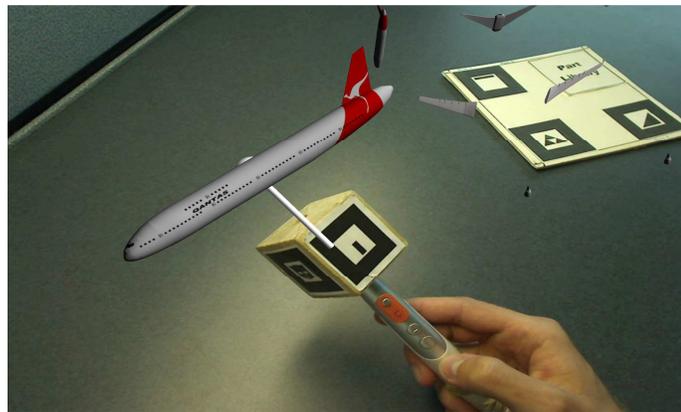


Figure 15: Grabbing an object transfers it to the wand so that it may be viewed individually.

3.2.3 Editor

After a part has been grabbed by the wand, certain characteristics can be edited by the user. The editor board displays the options available as different buttons which will uniquely affect the model (Figure 16). The scale editor will appropriately increase or decrease the size of a model when the user presses the up or down arrow buttons. Similar interactions will affect the weight associated with the current model when using the weight editor. Initial

weight is read in from a text file. This information is used later when providing the user with quantitative assessment data.

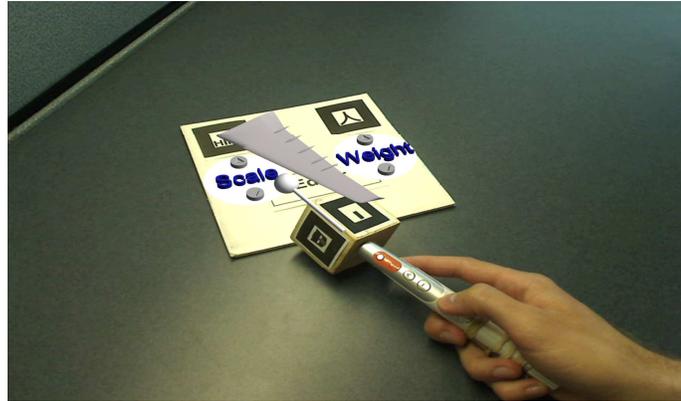


Figure 16: The editor board provides a method for altering different aspects of the selected model.

3.2.4 Assembly Area

The final marker board is the assembly area. Objects can be placed at any position or orientation as long as this board is in frame. If the wand is holding an object, pressing the circular button will drop it as shown in Figure 17. The model now belongs to the assembly. At any time, the user can clear the most recently dropped object with the down arrow button. The up arrow button will completely clear the entire assembly. Once completed, the concept can be viewed interactively according to the user's manipulations. This qualitative assessment capability is pictured in Figure 18.

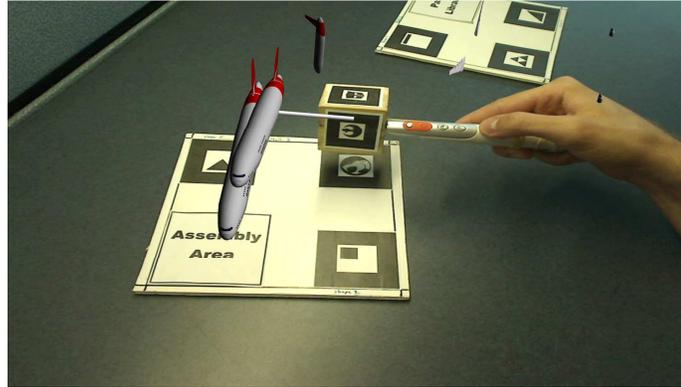


Figure 17: Copies of grabbed objects can be placed on the assembly board at any orientation.

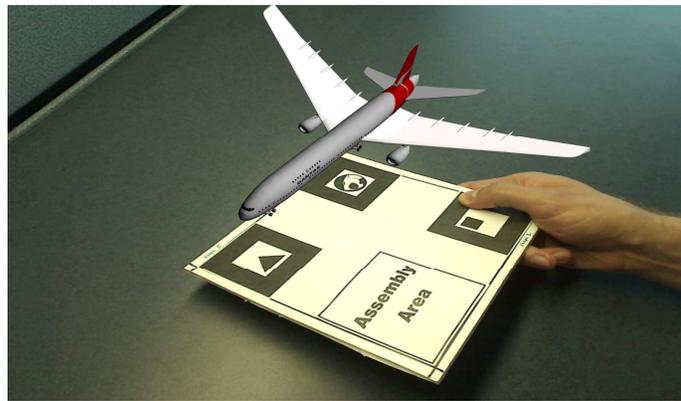


Figure 18: Completed concept assemblies can be easily visualized.

The assembly board is also where the user can obtain quantitative feedback from the model he or she has just created. Along the lower edge of the physical board are two virtual buttons. The first button reads “Toggle CG”. In order to click this virtual button, the user must point near it with an empty wand and press the circular button. This will cause the application to calculate and display the center of gravity of the specific model on the assembly board. The CG is simply calculated as the average position of all the parts on the board weighted by their respective mass or weight values. This position is represented as a

red sphere. In order to assist in viewing this sphere, all models become semi-transparent when the CG is being displayed (Figure 19). The second and final assessment tool included in the application is wheel loading. To use this feature, the user must first activate the wheel loading button. Then, he or she must click to select four “wheel” points in the assembled model. Once these four points have been selected, the loading distribution is calculated and displayed on the board. This calculation is done using metamodels that were developed as another component of this research. Specific information regarding these models can be found elsewhere in this paper. The visual result of the wheel loading calculation is provided using colored arrows (Figure 20). These arrows shift from white to red as the loading becomes more severe. In this way, a user can very quickly determine the loading distribution on the specified wheel elements.



Figure 19: When the CG button is activated, the models become transparent and the center of gravity is represented by a small red sphere.

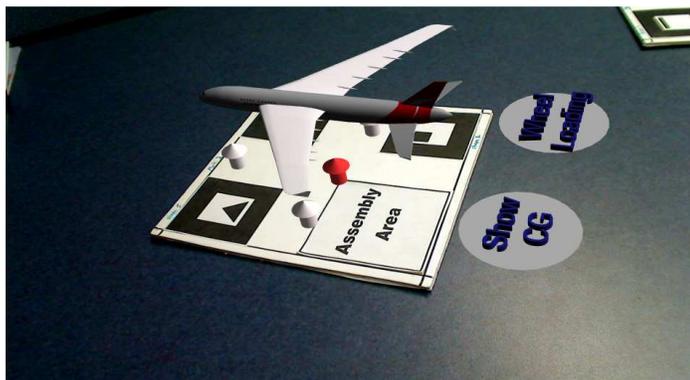


Figure 20: Users can obtain wheel loading feedback using the built in assessment tool.

4 METAMODEL DEVELOPMENT

When discussing the motivation for this research, the need for conceptual assessment tools was established. In addition, this assessment should be real-time and available within the same application as other design functions. Through the previously described augmented reality framework, users have access to a great deal of qualitative assessment. The center of gravity feature also provides a level of quantitative evaluation. In order to provide a higher level of analysis, metamodels were developed to approximate more complicated systems. Though a considerable amount of time was involved in creating these models, the final result is a set of polynomial equations that can be evaluated by a computer almost instantaneously. This time investment was mostly due to the fact that the source data for these models needed to be generated through FEA. Every single data trial required the adjustment of a CAD model followed by several minutes of waiting for the analysis results. Models were then fit to the resulting data. This is one situation where the use of legacy data would offer time savings. One of the advantages of legacy data, as discussed in the introduction, is the necessary data trials would already exist and be ready for model fitting. However, such data is often proprietary and was unavailable for use in this research. Thus, two example cases were created in a CAD environment, evaluated using FEA, and approximated with metamodels. The first case studied a generalized wheel loading situation, and the second case analyzed stresses and deformation in the support frame of a bucket truck.

4.1 Wheel Loading Test Case

4.1.1 Problem Description

One assessment tool built into the previously mentioned ASDS system is wheel loading. The term wheel loading is derived from initial work on ground vehicles, but “wheel” simply refers to any support point. If three or less support points are chosen, a statically determinant problem is created. The solution can be found by summing the forces in the y-direction and the moments in both the x and z-directions. While this is very useful, many vehicle concepts consist of more than three wheels. Such a situation is called a statically indeterminate problem. A simple set of closed form equations is insufficient to solve this system since there are too many unknowns. However, there are alternative methods that can provide a solution. One way to determine the loads at each support is to use an FEA simulation. These simulations then formed the basis for an approximation of a wheel loading system. An experiment was created to generate and test varying wheel loading conditions. First, a testing rig was constructed in ABAQUS⁵⁷ which consisted of four arms each with a support block (Figure 21). These supports can be placed at any location along the length of the arm, up to a maximum of four meters and a minimum of zero distance from the CG. A minimal load of 100 lbf was applied to the central block of the structure to simulate the CG of a concept vehicle. By varying the support-to-CG distances, different reaction forces were measured in each support, providing 4 variables and 4 outputs upon which to build a model.

Next, the DOE was formulated to specify the trials that would need to be run. Being that the final goal of this research is rapid concept evaluation, it was decided that the required number of trials be kept to a minimum. Initially, orthogonal arrays were used to sample the

design space. The L-49 array was chosen, requiring 49 test runs and 7 levels for each variable. While results from this sampling scheme were acceptable, orthogonal arrays were abandoned in favor of a basic random sampling scheme for two reasons. One, the PCE method is built to handle inputs which follow a normal, random distribution. In order to draw comparisons between the methods, the source data must be identical. Two, trials cannot be easily removed from an L-49 array without destroying its orthogonal properties; random samples can be simply removed without fundamentally affecting the sampling scheme. In total, four data sets were generated. Two of these contained 50 trials each and were used as source data. The other two were used as validation data sets with 30 trials apiece. Every distance value in this source data was distributed about a mean value of 2, half of the full leg length. The trials in one source set and one validation set were randomly generated using a standard deviation of 1. The remaining two sets used a deviation of 0.5. Details regarding this method of data generation will be discussed in Section 4.1.3 of this thesis.

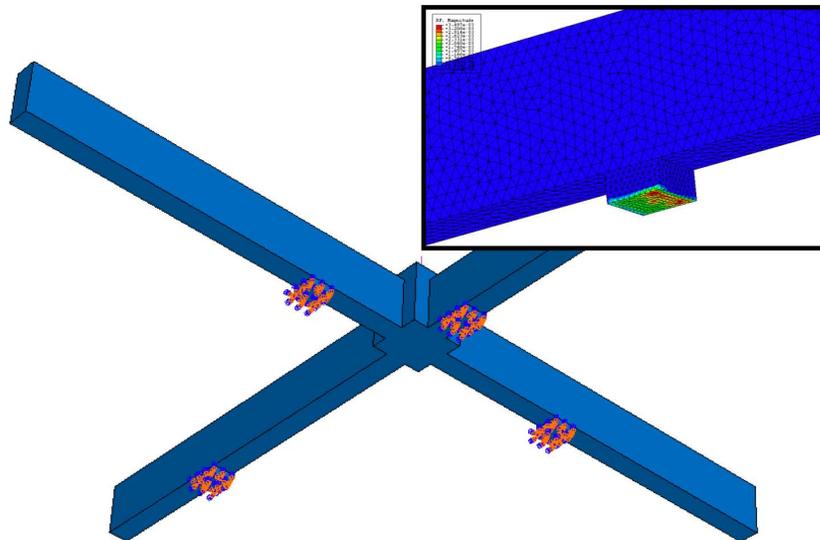


Figure 21: Screenshot of the loading rig created and analyzed using ABAQUS.

4.1.2 Model Formulation

Using the values obtained from the DOE, a series of ABAQUS analyses were performed to generate the source data upon which the metamodel is built. In every trial, the total load for each support was calculated and then represented as a percentage of the total load. For example, if all four supports were equidistant from the CG, the value at each support was 0.25, or 25%. Once all analyses were completed, metamodels were built to approximate the results. Regardless of the modeling technique used, one model was generated for each leg, using all four support distances as variables. Every loading situation was then modeled by four equations. Like the FEA results, the results of these models are formatted as a percentage of the total load. Equations (9) and (10) show the general forms of the models for the PRS and PCE methods, respectively.

$$\begin{aligned}
 \hat{y}_1(X) = & \beta_0 + \underbrace{\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4}_{\text{First order terms}} \\
 & + \underbrace{\beta_5 x_1 x_2 + \beta_6 x_1 x_3 + \beta_7 x_1 x_4 + \beta_8 x_2 x_3 + \beta_9 x_2 x_4 + \beta_{10} x_3 x_4 + \beta_{11} x_3 x_4}_{\text{Second order interaction terms}} \\
 & + \underbrace{\beta_{12} x_1^2 + \beta_{13} x_2^2 + \beta_{14} x_3^2 + \beta_{15} x_4^2}_{\text{Second order terms}} \\
 & + \underbrace{\beta_{16} x_1^3 + \beta_{17} x_2^3 + \beta_{18} x_3^3 + \beta_{19} x_4^3}_{\text{Third order terms}} \\
 \text{where } X = & [x_1 \quad x_2 \quad x_3 \quad x_4]
 \end{aligned} \tag{9}$$

$$\begin{aligned}
\hat{y}_1(\Theta) = & \beta_0 + \underbrace{\beta_1\xi_1 + \beta_2\xi_2 + \beta_3\xi_3 + \beta_4\xi_4}_{\text{First order terms}} \\
& + \underbrace{\beta_5\xi_1\xi_2 + \beta_6\xi_1\xi_3 + \beta_7\xi_1\xi_4 + \beta_8\xi_2\xi_3 + \beta_9\xi_2\xi_4 + \beta_{10}\xi_3\xi_4 + \beta_{11}\xi_3\xi_4}_{\text{Second order interaction terms}} \\
& + \underbrace{\beta_{12}(\xi_1^2 - 1) + \beta_{13}(\xi_2^2 - 1) + \beta_{14}(\xi_3^2 - 1) + \beta_{15}(\xi_4^2 - 1)}_{\text{Second order terms}} \\
& + \underbrace{\beta_{16}(\xi_1^3 - 3\xi_1) + \beta_{17}(\xi_2^3 - 3\xi_2) + \beta_{18}(\xi_3^3 - 3\xi_3) + \beta_{19}(\xi_4^3 - 3\xi_4)}_{\text{Third order terms}}
\end{aligned} \tag{10}$$

where $\Theta = [\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4]$

The above equations both calculate the predicted value, \hat{y}_1 . The ‘1’ notation indicates that this is the response for the first leg of the loading rig. The form of the equation for the other three legs is identical. Different observed values are used for the regression, however, resulting in different solutions for the coefficients, β . The values in X are normalized to the largest individual value. Θ is comprised of standard Gaussian variables calculated using Equation # and the original values generated by the DOE. With these equations formulated, the modeling methodologies can be studied within the context of the wheel loading experiment.

4.1.3 Experimental Setup

Equations (9) and (10) only represent only one feasible configuration for the polynomial response surface and polynomial chaos expansion methods. There several factors that can be altered which affect model performance. The primary goal of this segment of research was to not only build a working metamodel but also to compare many different

approaches in order to find the most robust but efficient model possible. In the following sections, these factors are explained and the approaches toward studying them are outlined.

In conceptual design, the design space for potential concepts can involve a great deal of uncertainty. Radical new ideas may be well outside the traditional understanding of a problem. Another concept may combine elements from several existing but diverse products. Modeling for both these situations involves building an approximation with source data from a design space that may be significantly different from that of the new set of concepts being evaluated. It was for this reason that two distinct sets of source and validation data were created. By altering the standard deviation of the distributions for support distances, one data set represents a larger design space, which includes more radical loading configurations. The opposite set is smaller and more conservative. This creates four possible test setups. Models can be both built and tested on data generated using the same distribution, large and small. An interpolative case is created when a model is fit to the large set and validated using the small. The reverse of this situation is the extrapolative case. Using these variations, model accuracy can be gauged when subjected to varying design spaces.

A large factor in the accuracy of any model is the number of known system responses available for fitting. Generally speaking, increasing the amount of source data will result in a better model. Unfortunately, obtaining these known values can be quite time consuming. Large amounts of existing data are not always available. The goal should be to generate reasonably accurate models with limited amounts of data. Higher error is acceptable at early stages in the design process since many issues remain undecided. The question then becomes how much the data can be limited. For this study, each set of source data contains a

maximum of 50 trials. This number was sequentially reduced to 40, 30, and 20 trials, and models were fit to each of these four situations.

While the textbook definitions of the various modeling methodologies provide a general form to be followed, the decision of which terms to include or exclude is that of the individual and the system he or she wishes to approximate. The term “interaction effects” deals with terms that in some way involve two or more variables (i.e. multidimensional). Examples of such terms are provided in Equations (9) and (10). It should be noted that interaction effects seen in Equation (10) are not true multi-dimensional Hermite polynomials. Rather, they are an alternate set of effects styled after the interaction terms in the PRS model. Adding variable interaction is not a guaranteed method to increase model accuracy. It is possible that these additional terms will capture more of the target process’s behavior; however they may also have a minimal or negative effect on model performance. This experiment was limited to second order effects, the interaction of two first order variables. Including higher order interactions would have added too many terms to the equation and affected additional components of the study.

Similar to the option of including interaction effects, model order is another choice to be made. The degree of a system’s response is not always known. Higher order models are capable of capturing more complex behavior. However, such flexibility in the model may be unneeded and result in weaker performance. Models were created that included one-dimensional terms up to the third power. The results from these will be compared to approximations built with no term higher than second order.

Finally, the two modeling methodologies will be compared. Every experimental factor explained above will be studied using both techniques. Different behaviors between the two methods may highlight certain advantages or disadvantages. Perhaps, one method performs better in all situations. Making these comparisons will not only lead to better wheel loading approximations but also provide insight into future analyses as well.

4.2 Bucket Truck Support Frame Test Case

4.2.1 Problem Description

The analyses described in the previous section focus on a generalized wheel loading situation. While such a tool is useful, the modeling challenges are relatively minimal. The loading rig has little to no complexity, and the four design variables are very similar in nature. In order to more fully assess the capability of metamodeling for conceptual design, an additional case was studied. The analysis problem was chosen to be the design of the support frame for a bucket truck. Figure 22 shows an example of this type of vehicle. The base of the arm is anchored to a custom frame in the rear of the truck. This frame is then fixed to either the vehicle itself or to outriggers that rest on the ground. Bucket trucks are manufactured in a variety of different sizes. The base vehicle used can range from medium sized, consumer level trucks all the way up to heavy duty, commercial class vehicles. Dimensions of the support frame are dependent upon vehicle choice. Boom arm length depends on the desired application; the weight to be supported at the end of the arm is also variable. Each of these conditions plays a role in the loading on the support frame. The nature of this loading is important to the designer and was chosen to be the focus of this analysis.



Figure 22: An example image of a bucket truck used for the second analysis case.

Like in the wheel loading case, no legacy data was available for use in this research. Thus, the source data was generated from finite element analysis of a CAD model. The support frame was modeled in SolidWorks⁵⁸, as shown in Figure 23. All joints in the model are assumed to be perfect; no welding assumptions were made. The outrigger plates at either end of the frame are fixed, and the loads are applied to the square plate on top. The nature of these loads was kept as simple as possible in order to stay within the confines of a useful conceptual design tool. While a potential designer will not have every detail figured out, he or she will at least know a few basic requirements such as overall dimensions and the intended usage. The five design variables were chosen based upon this assumed knowledge: length and width of the frame; length and weight of the boom arm; weight of the object to be supported at the end of the boom arm. The design of experiments for this case was done similarly to that of the wheel loading analysis. Mean values and standard deviations were set for each variable (Table 1), and data was generated through a random sampling process.

Loading was applied to the FEA model based upon these values. The sum of the two weights created a downward force. The second loading component was a moment created by the two weights. The moment caused by the boom arm alone was assumed to act at half of its length. A second moment was generated by the weight in the boom's basket and was calculated at the full length of the boom. Initial investigation of the problem determined that the maximum loading occurred with a forward facing, fully extended boom. All simulation runs were performed in CosmosWorks⁵⁹ at this maximum loading scenario. A total of 90 trials were simulated; sixty to be used as training data and an additional thirty for validation purposes. The magnitudes of the maximum stress and maximum displacement along with their respective locations would be interest to a designer. Thus, these eight different outputs were recorded for each of the FEA trials. The maximum stress was measured in pounds per square inch (p.s.i.) while the displacement and location values were measured in inches.

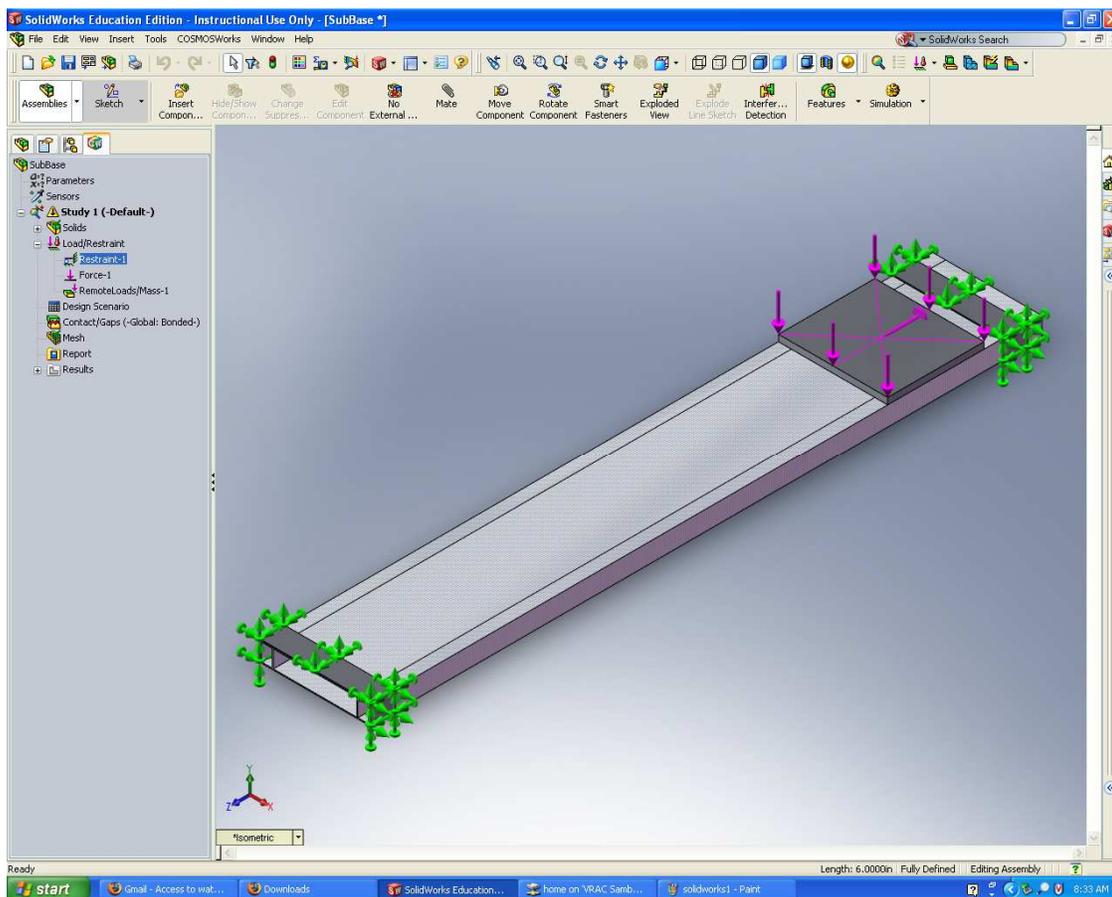


Figure 23: The support frame structure was generated in SolidWorks and simulated using CosmosWorks.

Table 1: The parameters used to generate the simulation trials for the support frame test case.

	Overall Length (in)	Overall Width (in)	Boom Length (in)	Boom Weight (lbs)	Basket Weight (lbs)
Mean	180	35	600	19000	500
St. Dev.	24	2	175	5000	150

4.2.2 *Experimental Setup*

Generating the models was again very similar to the model fitting done in the wheel loading case. Several terms were added to Equations (9) and (10) in order to accommodate a fifth design variable. The source data from the support frame analyses, both inputs and outputs, were normalized to the same ranges as the wheel loading data. This was done to allow for an accurate comparison once results were obtained. Similar experimental factors were also altered such that their effects could be studied in this new test case. Data was not generated using different distributions, however, so no investigation into interpolation or extrapolation could be performed.

5 RESULTS AND DISCUSSION

5.1 Metamodeling Results

5.1.1 Evaluation Procedure

A large number of models were created according to the procedure laid out in the development chapter. It was necessary to choose a method for measuring these metamodels so that relative performance could be determined. Though the various models were all built upon different sets of source trials, each was evaluated on common sets of validation data. As previously mentioned, each of these validation sets contained 30 data points. The reason for this additional series of trials was to assure that fair comparisons were being made. Performance metrics can be calculated to show how well a model fits the data from which it was generated. In practical applications however, models are used to approximate responses other than those used in their training. For instance, a model built on a small set of uncomplicated data may fit those points very well, but when applied to a real problem the model may fail entirely. By comparing a model's predicted responses with actual observed values in an independent data set, it is possible to obtain an unbiased assessment of the approximation.

Three performance metrics were calculated from results of the metamodeling experiments. Maximum absolute difference was the first of these. As shown in Equation (11), this value is defined quite simply as the largest difference between the predicted value, \hat{y}_i , and the observed value, y_i , in absolute terms over n samples. The second metric was root mean square error (RMSE), as defined by Equation (12). Finally, the R^2 statistic was used to

gauge a model's performance. This metric is calculated according to Equation (13), where y_{mean} is the average value of the observed responses.

$$\text{Maximum Absolute Error} = \max\left(|\hat{y}_i - y_i|_{i=1}^n\right) \quad (11)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (12)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - y_{mean})^2} \quad (13)$$

Using these calculations, the results from experimenting with various model factors will be compared to one another. The following two sections will present results from the two cases separately.

5.1.2 Wheel Loading Results

In total, 192 different models were generated in this study. Each model was then run with two distinct sets of validation data. Thus, results can be obtained from 384 different configurations. The nature of the experiment does allow the data to be condensed somewhat. Four models were generated for each test, one model for each leg on the loading rig. Since the outputs of these models are fundamentally similar (i.e. percentage of total load), they were considered as one single response for the purpose of error calculations. Comparisons

could then be made using a more manageable pool of 96 values for the RMSE and maximum absolute error metrics. The R^2 values were obtained for each regression performed. For the wheel loading case, all of these values were between 0.95 and 1.0 indicating an accurate fit of the source data. This metric, however, does not necessarily provide a good means of comparison between the various modeling configurations for reasons already explained. In order to best view the effects of the experimental factors put forth in the method development chapter, this section will present a series of charts. While both RMSE and maximum absolute error were used for analysis, only RMSE values will appear in the figures. Both metrics revealed similar trends, but RMSE is a better measure of a model's overall performance.

The first factor to be analyzed is a model's performance in situations with varying distributions of source and validation data. Figure 24 displays these results. For this configuration, 3rd order models were generated using 50 points of source data. The two columns on the left side of the chart represent models that were tested using the same validation data, generated with a 1.0 standard deviation. However, the models were fit to data of both distributions. Here, the extrapolation case has slightly higher error but still compares favorably to the models built and tested on the same data distribution. Similar results are seen on the other half of the chart. While the interpolating models are not quite as accurate, they perform nearly as well as the other models. A considerable difference in RMSE is seen between the two halves of the figure. This result is not surprising, however. The tighter, more conservative data set has fewer extreme loading conditions that are more difficult to predict.

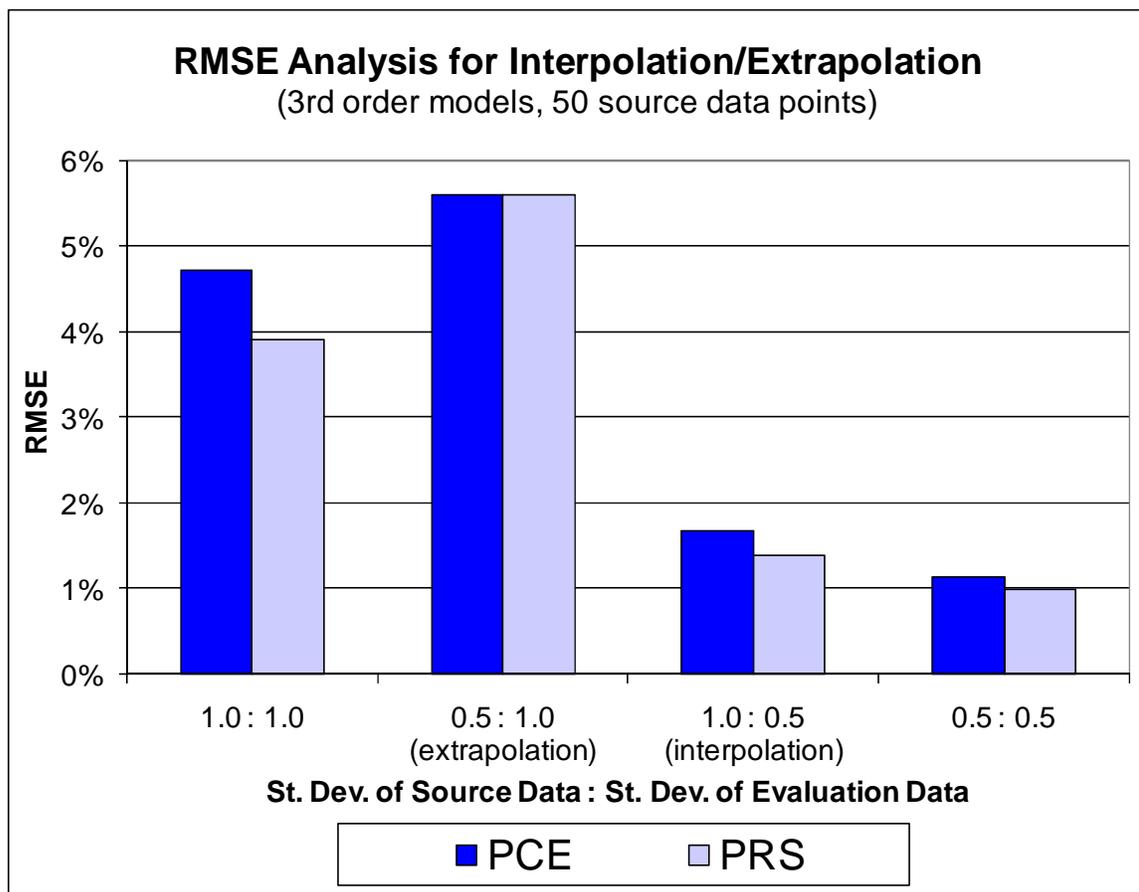


Figure 24: This chart presents an example case of model performance in interpolating and extrapolating conditions.

For every modeling configuration that was investigated, models were fit to varying levels of input data. The general trend was for error to increase as fewer data points were used for regression, as evidenced by Figure 25. For the most part, this increase was modest down to as few as 30 data points. In the chart, the error for the PCE model is still relatively low when only 20 trials were used. However, the PRS model had a noticeable reduction in accuracy. Though the error for the PRS model is not so bad as to render the approximation unusable, this behavior does indicate that more volatility can be expected when the source

data is reduced. There will be more discussion later regarding the differences in performance between the two methodologies. The point to be made here is that with minimal input data, otherwise acceptable models may no longer provide adequate results.

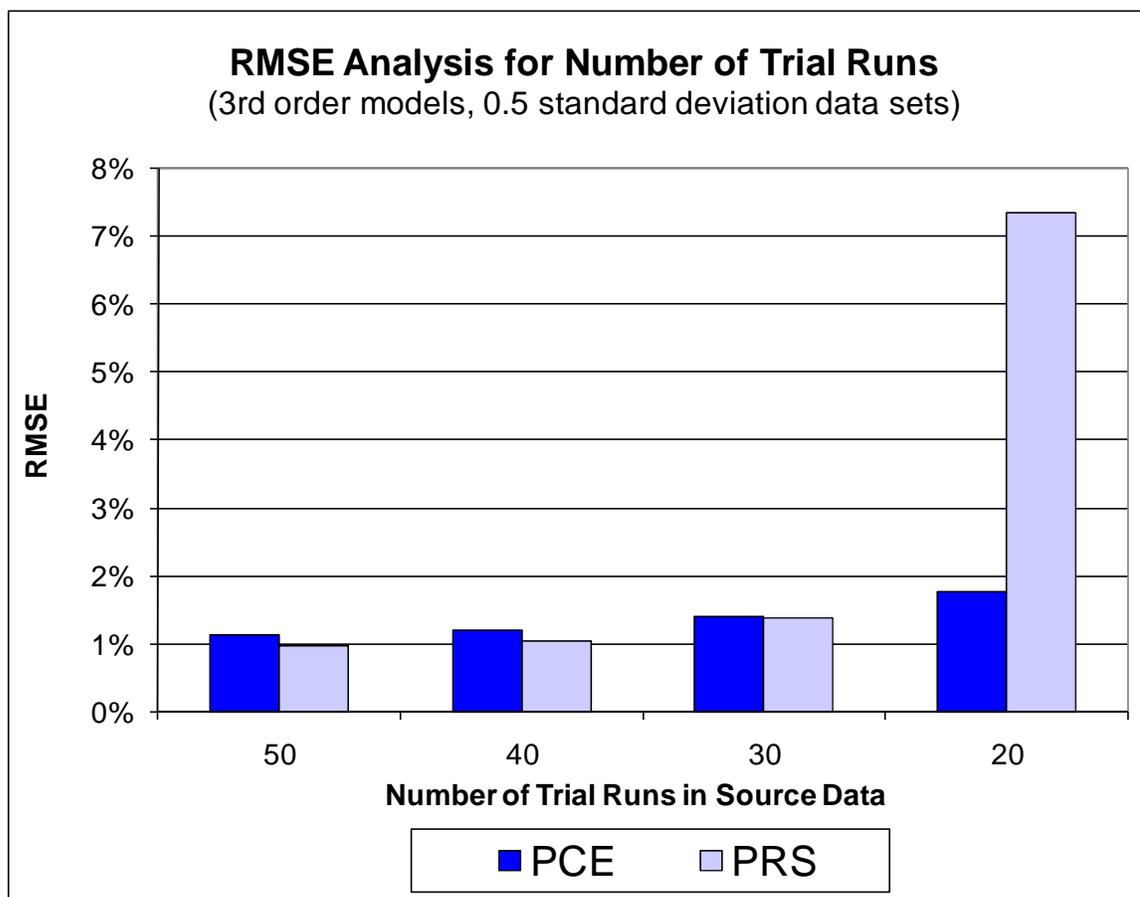


Figure 25: Models can exhibit unstable behavior when presented with only a minimum of source data.

The impact of including interaction effects in the model expression was examined. As discussed in the chapter on methodology development, the addition of these terms does not guarantee a better model. The results from this experiment affirm that claim. In Figure 26,

interaction effects provide a noticeable improvement in error at all levels of input data. However, in Figure 27, the same terms have a negative impact on model performance. The difference between these two cases is the distribution of both the source and validation data. Interaction effects appear to be beneficial in situations with more densely sampled, smaller design spaces. The additional control perhaps serves as a way to fine tune the approximation. Over larger design spaces with less dense sampling, this control becomes a drawback rather than an advantage.

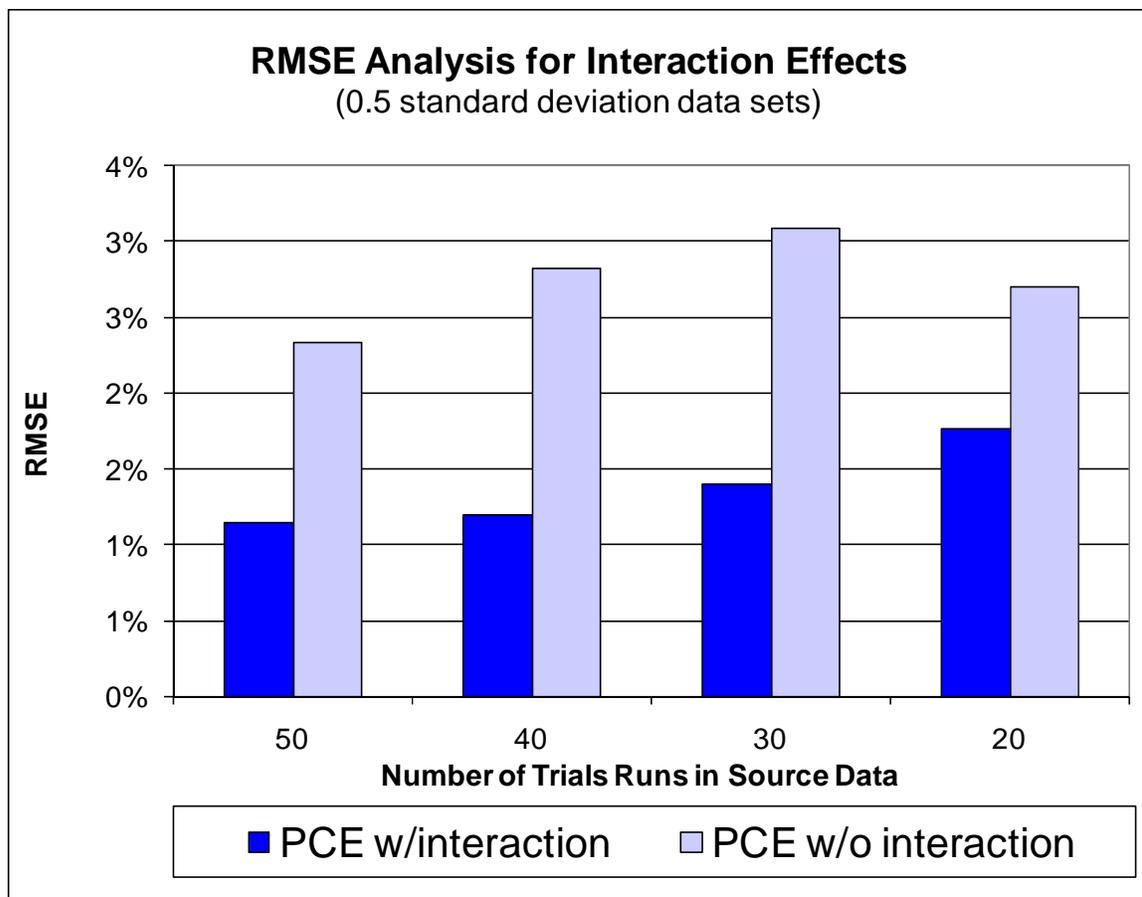


Figure 26: In this modeling configuration, interaction effects had a positive impact on model performance.

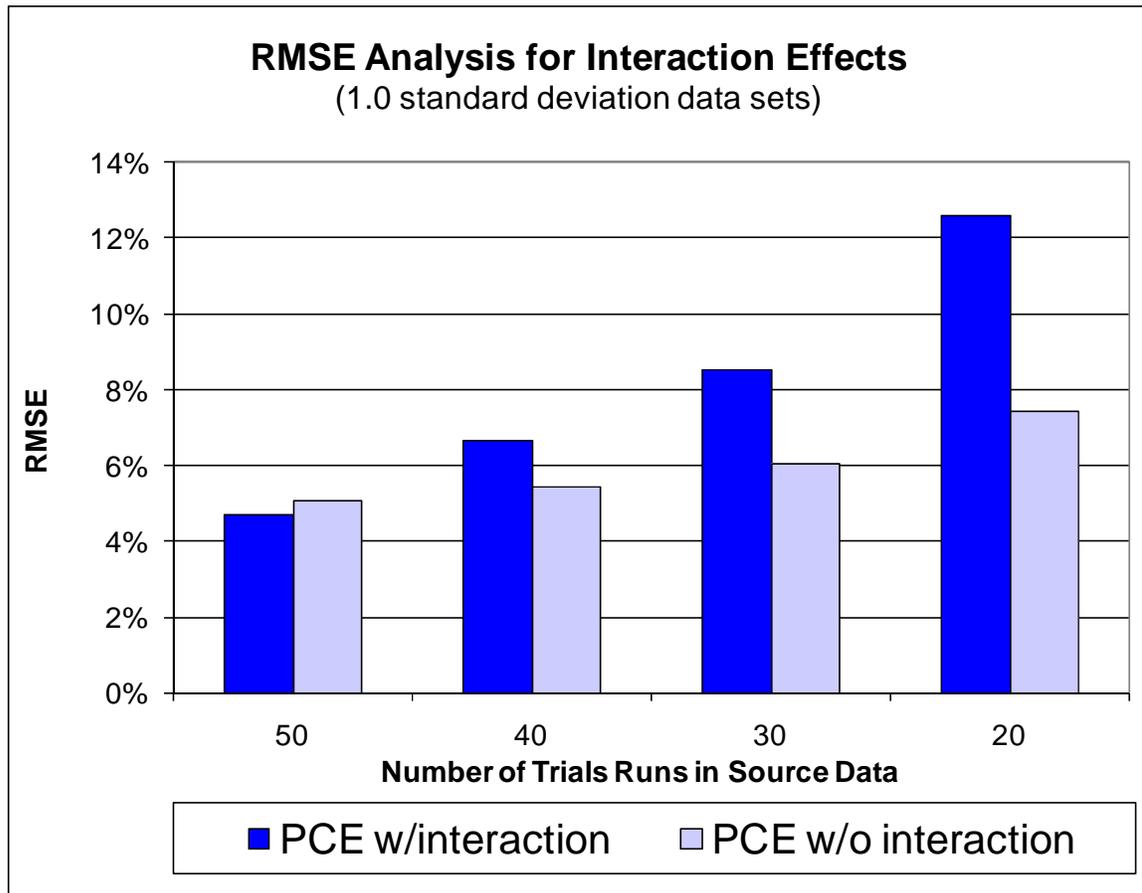


Figure 27: Interaction effects were not beneficial in this example.

Comparisons were also drawn between approximations that did or did not include third order terms. For models that were fit to all 50 data points, Figure 28, the addition of third order terms did yield a slight increase in accuracy. However, when the amount of source data was reduced to 20 trials, some second order models outperformed their third order counterparts, Figure 29. The reason for this behavior is similar to the explanation given when discussing the impact of interaction effects. For minimal sampling situations in a large design space, the gaps between regression points are much wider. Within these gaps a higher response is more likely to exhibit radical behavior than a more simple approximation.

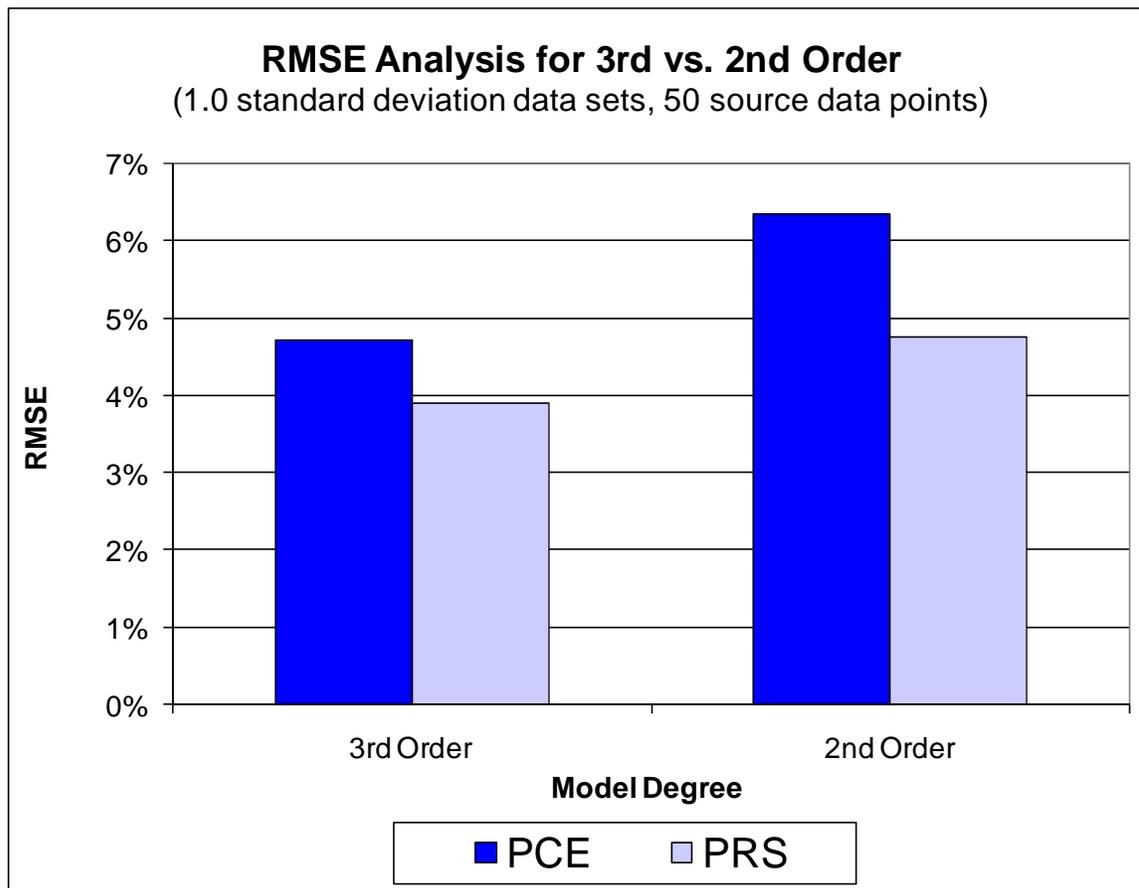


Figure 28: With high levels of source data, 3rd order models offer a small improvement in model accuracy.

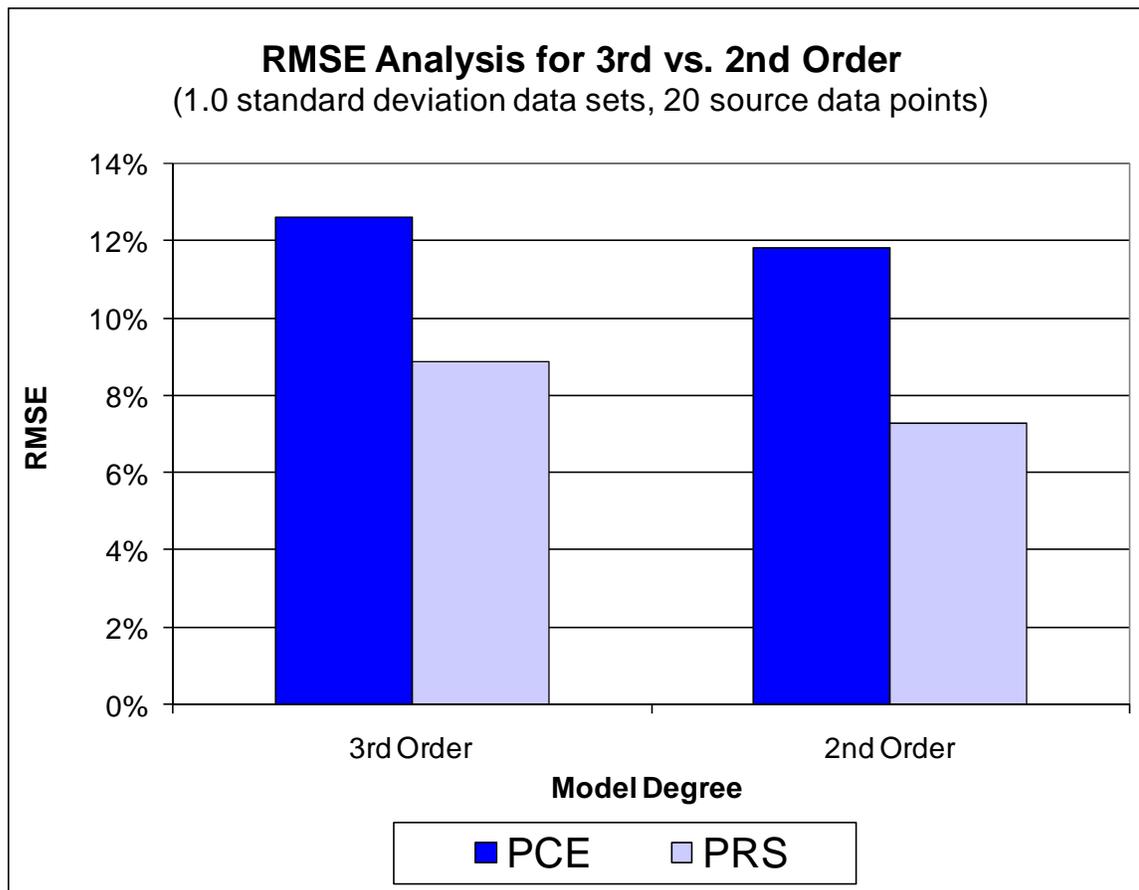


Figure 29: With minimal input data, 2nd order models often outperform those with 3rd order terms.

In many of the modeling configurations examined as part of this research, PRS and PCE models performed nearly identically. This was especially true when the models were fit to larger numbers of source data points. Differences did become apparent, however, when the models were only presented with a minimal number of data points. Figure 30 highlights these variations. The PRS model outperformed the PCE approach in both cases where the source data was distributed over a larger design space. When a smaller distribution was used as the input data, the PCE model was more accurate.

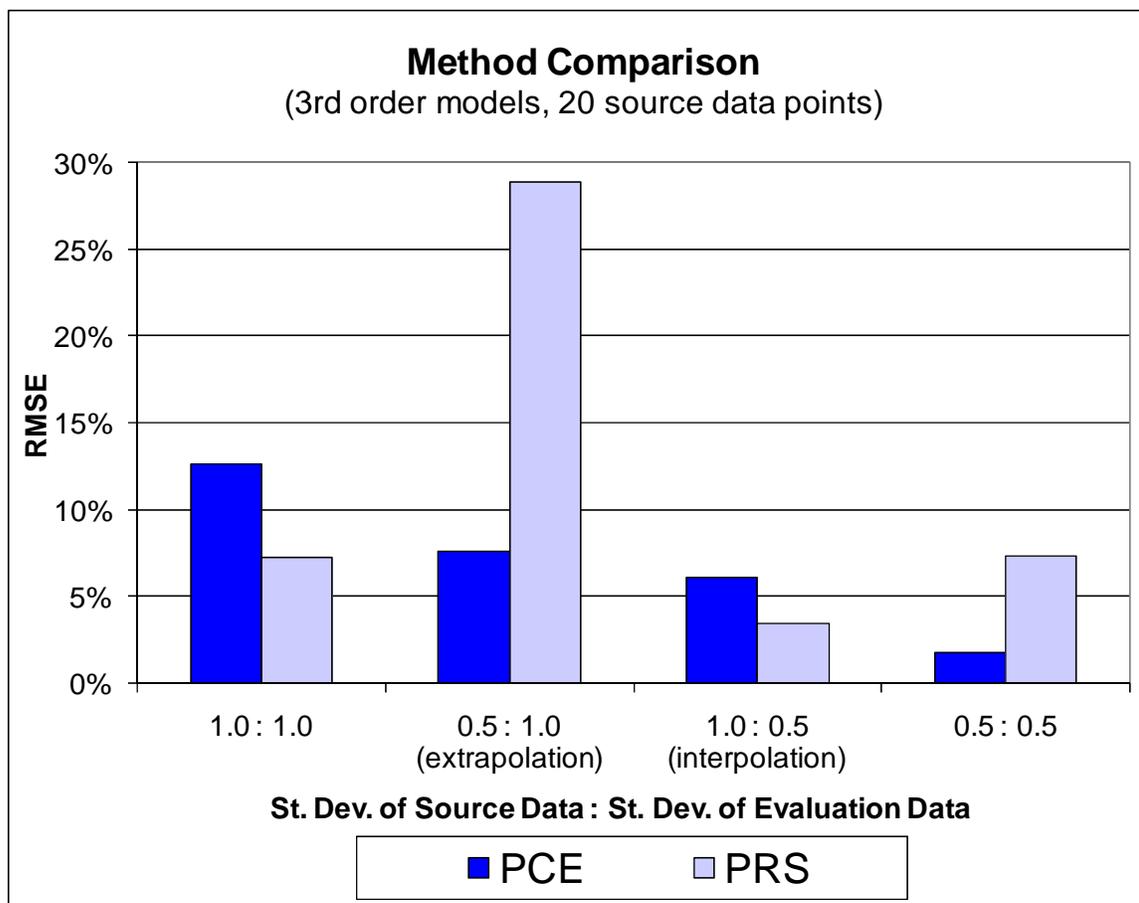


Figure 30: PCE performs better when fit to smaller distributions of data. Source data with a greater distribution is better modeled by PRS.

5.1.3 Support Frame Results

For the support frame test case, 48 different models were generated for analysis. Since each output response represented a very different measurement, the results were not combined to provide a single error value. Though eight different responses were recorded from the simulation runs, models were only constructed for six of those outputs. The y and z

location coordinates of the maximum stress location never changed, so no approximation was necessary. As before, models were built at varying levels of input data: 60, 50, 40, and 30 trials. The increase in possible data was done to accommodate the additional terms required by a five design variable problem as opposed to four.

The R^2 values for the models in this case were not as consistently high as for the wheel loading problem. The values were effectively identical between the PRS and PCE methods, however. Table 2 presents these values for 3rd order models that included interaction effects. R^2 values for 2nd order models with no interaction effects are presented in Table 3. Both tables share the general trend of increasing R^2 values as source data is removed. The values for the x-position of the maximum stress are always 1. In the previous paragraph, it was explained that y and z positions of this location never changed. While the value of the position does indeed change, it remains in relatively the same position. Its position is directly related to the overall dimensions of the frame and is thus easily predicted. Another feature to note is that the R^2 values for the x and y positions of the maximum displacement are considerably lower in the 2nd order case. This will come more into play as other results are presented.

Table 2: R² values for 3rd order models at different levels of input data.

# of trials in source data	R-squared					
	Maximum Stress		Maximum Displacement			
	Magnitude	x-position	Magnitude	x-position	y-position	z-position
60	0.7722	1	0.7792	0.8411	0.8477	0.9909
50	0.7673	1	0.7749	0.8794	0.8951	0.9928
40	0.995	1	0.9952	0.9073	0.9168	0.9967
30	0.9983	1	0.9986	0.9769	0.981	0.9975

Table 3: R² values for 2nd order models at different levels of input data.

# of trials in source data	R-squared					
	Maximum Stress		Maximum Displacement			
	Magnitude	x-position	Magnitude	x-position	y-position	z-position
60	0.757	1	0.7621	0.6094	0.6254	0.9648
50	0.7492	1	0.7546	0.6303	0.6508	0.9693
40	0.9823	1	0.982	0.6805	0.6915	0.9718
30	0.9858	1	0.9842	0.6578	0.6909	0.9762

Again, RMSE values will be used to compare the models as it is a better overall measure than maximum absolute error. Similar trends are visible in all the outputs, so only two will be presented here for discussion. The maximum stress magnitude represented some of the best modeling performance while the y-position of the maximum displacement was one of the most challenging responses to approximate. Figures 31 and 32 depict the results from these two responses for both PRS and PCE modeling approaches. In each situation, PCE outperforms PRS, but the errors in general are considerably higher than those seen for

the simpler wheel loading situation. The errors observed for the maximum displacement y-position approximation are so high as to render the model ineffective. Thus, the R^2 values were more than a little misleading. Another approach should be followed if adequate results are to be realized.

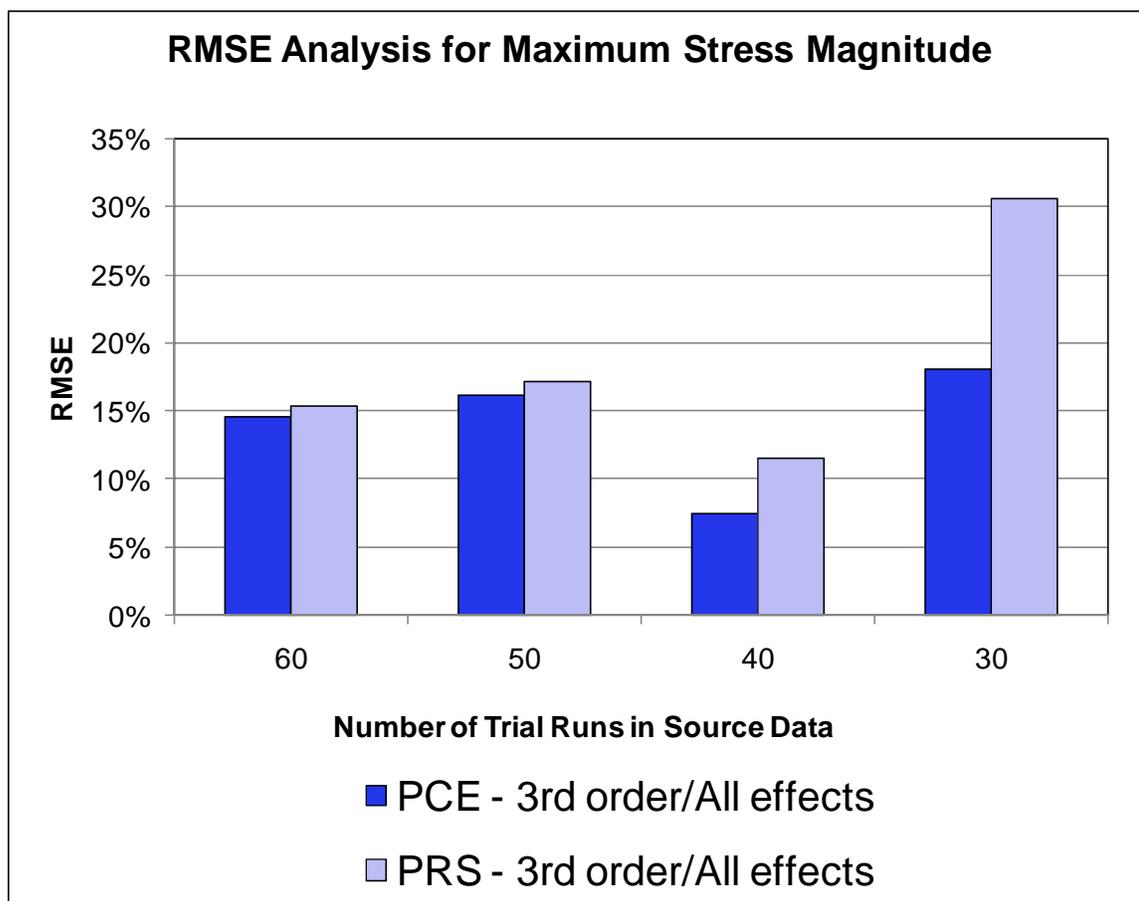


Figure 31: Error values are noticeably higher than those seen in the wheel loading analysis.

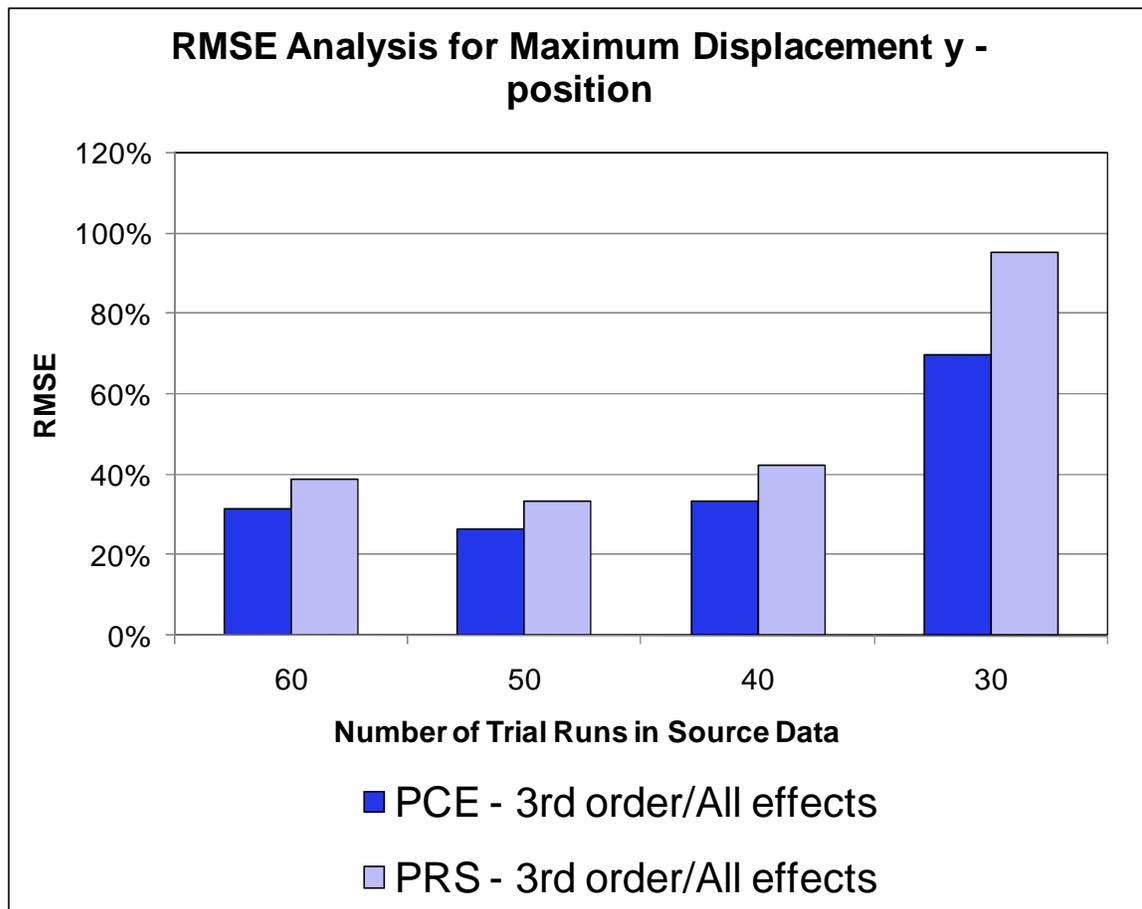


Figure 32: The position of the maximum displacement is difficult to approximate. This modeling configuration did not perform well.

The next series of charts shows a comparison between two different configurations of PCE models. The first models were built using only 2nd order effects and no interaction between variables. These are compared to the PCE models from the above charts. Figure 33 actually shows a slight trend for reduction in error as input data is removed. Moderate improvements in accuracy over the 3rd order PCE models are seen as well. Much more drastic reductions in error were observed when applying the simpler model to maximum displacement y-position response (Figure 34). Accuracy was increased to the point where the

model may be useful for rough design estimates. Recall that the R^2 values for this model type were significantly lower than for the previous one. This result indicates a situation of over-fitting the source data. The 3rd order model was able to better approximate the input data points. However, in doing so, it lost its generality and was unable to accurately predict the response elsewhere in the design space.

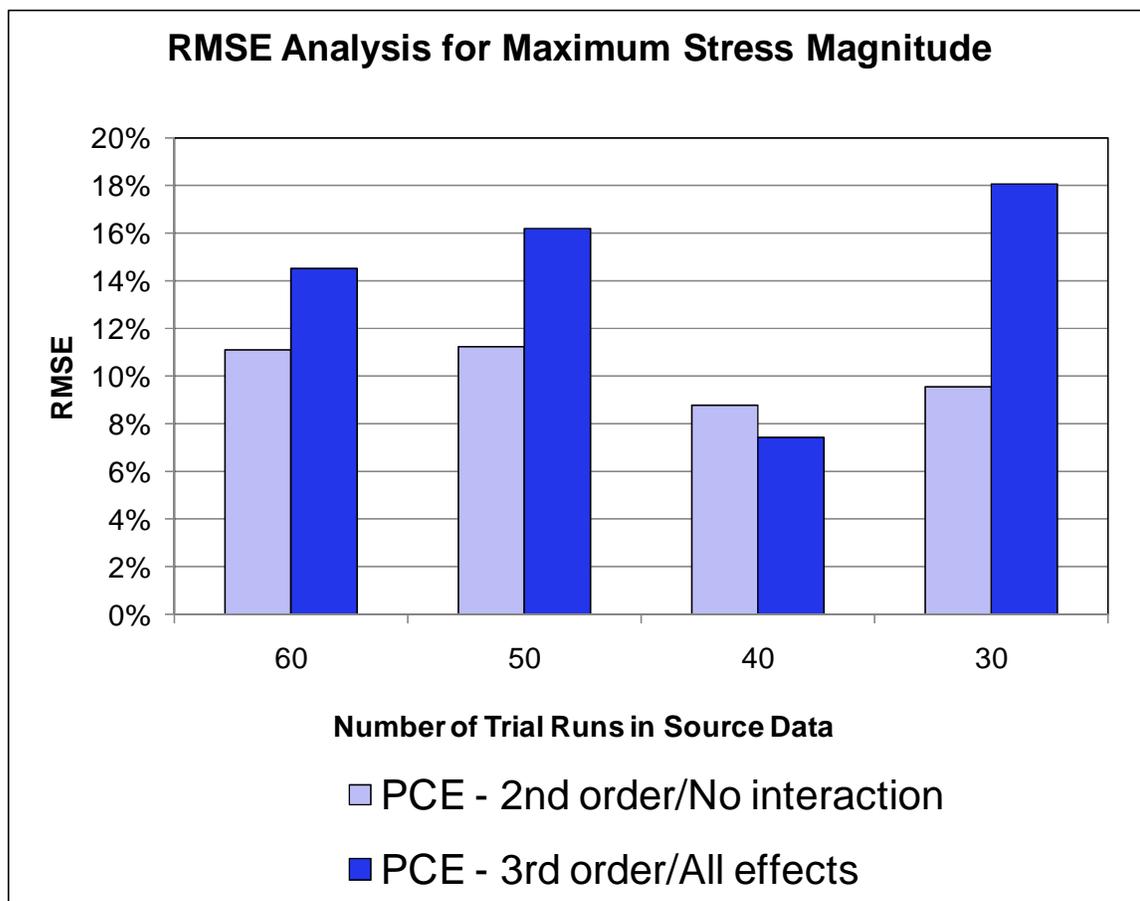


Figure 33: Simpler, 2nd order models offered better performance than 3rd order.

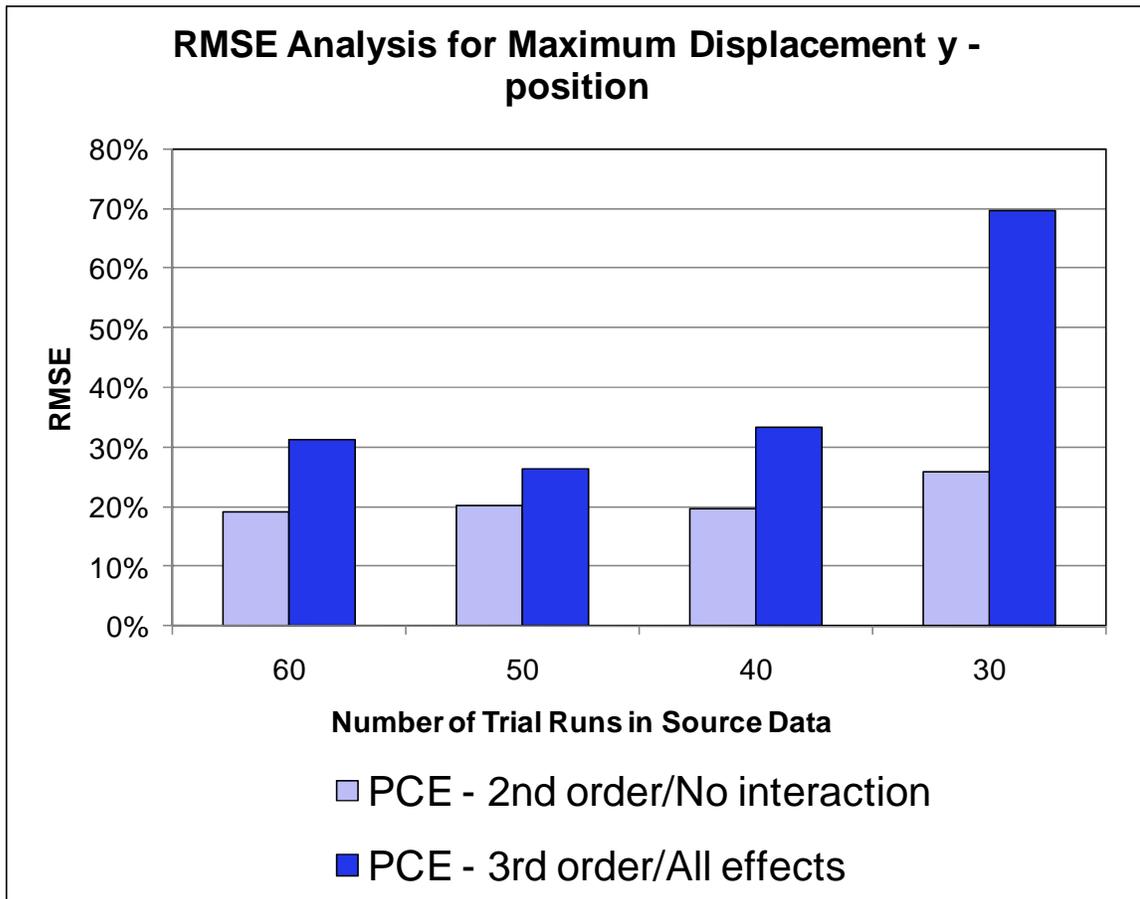


Figure 34: Large improvements were obtained when 2nd order models were used to approximate maximum displacement position.

6 CONCLUSIONS AND FUTURE WORK

6.1 Summary and Conclusions

The research presented in this thesis has described the development of a new engineering design tool. Specifically created to serve the needs of conceptual design, the application contains several features, including:

- A 3D augmented reality environment.
- Intuitive methods for object creation, manipulation, and editing.
- Quantitative assessment for user created concepts.

Augmented reality provides a new way to develop applications. Instead of the software environment existing only on a display screen, virtual elements coincide with the real world. Through the use of a video see-through HMD, a 3D environment was developed in which the user's hands are the primary mode of interaction. This creates a very direct mapping that is easily learned by most anyone.

Because of the capabilities of AR, the common tasks found in traditional design software were applied in a new, more intuitive form. Custom elements, both real and virtual, were created to develop this new interaction. A wand device became the equivalent of a 3D mouse. Various marker boards were created, each performing functions specific to the board's purpose. Virtual objects can be picked up and inspected almost as if they were actually in the user's hand. These same objects can be altered and then placed into assemblies in a similar way that one sets down a real object. These capabilities provide the user a very flexible work space to create and visualize a concept.

Within the same design environment, two quantitative analysis tools are available to provide instant feedback of a concept. Based upon the position and weights of each component on the board, a virtual button will display the center of gravity of the assembled concept. Similarly, the wheel loading button will be display the load distribution after selecting four support points. These evaluation tools aid the designer by providing more technical data at an early phase in the design process.

In support of the development of the wheel loading analysis, additional research was done to investigate the application of metamodels for use in the context of conceptual design. Two methods were studied in different case studies: polynomial response surface and polynomial chaos expansion. The first test case was the analysis situation applied in the design application, wheel loading. For both methods, error was found to be less than 10%, according to the calculated RMSE metric, when provided as few as thirty input data points. In certain test configurations, acceptable results were observed with twenty points of source data. However, model performance in these conditions was considerably more volatile and unpredictable. The inclusion of interaction effects and third order terms was found to provide a small improvement in accuracy in more densely sampled data sets. These effects had a negative impact when input data was reduced. Performance differences between the two methods were not substantial, except when only a minimum of source data was used. Based on the presented results, PRS models performed better when fit to large data distributions. PCE was better suited to modeling smaller distributions and then extrapolating.

The second case to be studied was that of the support frame for a bucket truck. This design problem was found to be considerably more difficult to model, with certain outputs

containing more non-linear behavior than the previous problem. Even for the simpler responses, third order models were only moderately sufficient for rough approximation with error values between 10% and 20%. The models for outputs that were more difficult to predict were so inaccurate as to be useless for approximation. Second order models, however, improved these numbers drastically. Prediction error for the maximum stress magnitude was much closer to 10% across all the input data conditions. Similarly, errors for the maximum displacement y-position fell to 20%, only rising to 35% at the minimum input data condition. While certainly not exact, these approximations may be able to provide rough performance estimates without the need for costly analysis.

6.2 Future Work

Continuing work for this research will advance along many paths. First and foremost, the presented application must become an official part of the ASDS project mentioned in Chapter 2. The future goals of this project are to offer a variety of interface and display options to users that are all capable of functioning as a single framework; augmented reality plays a role in this vision. Aside from this, there are a great deal of possibilities for future research in the field of metamodeling for conceptual design. This thesis has only presented two methods for consideration. A colleague is currently performing similar analyses using the radial basis function (RBF) and Kriging methods. Adaptive learning algorithms such as neural networks and support vector regression (SVR) may hold promise in building better approximations. The test cases being modeled must also be expanded. More challenging design problems will be found to better analyze the various methods. There is also a need to

better legitimize the approximations with more real world data. This can be done in a number of ways. First, the use of legacy data needs to be pursued more vigorously. In this project, legacy data was simulated by generating results from randomly chosen FEA run. However, it would be interesting to generate models using industry data from a real world product. Approximation results from the generic models could also be better validated. This could take the form of comparing the predicted model outputs to high fidelity simulation results of actual industrial models or even to physical testing of the product itself.

Advancement of the augmented reality interface is another avenue for continued research. While the theoretical benefits of AR have been proposed in this thesis, no official user studies were performed. Such testing would help validate the use of 3D interfaces as well serve as a baseline for future studies using updated hardware and software. For instance, additional methods of visual feedback could be explored to provide additional dimensions of information. Though powerful and indispensable to this project, the OSGART framework is not perfect. Even in the best conditions, the tracking results have a fair amount of noise. This could be mitigated by applying a Kalman filter, or similar method, to this data. Other problems may not be so easily solved. As more markers are used, misrecognition of markers becomes a greater issue. Sub-optimal lighting conditions cause the application to become unstable. It may become necessary to move to another tracking method. AR has many alternative solutions. Particularly, marker-less tracking holds great promise. Using infrared, depth sensing technology, it is possible to track the location and orientation of a user's hands as well as detect gestures. In this way, AR applications would no longer be tied to physical markers or sensors. With this depth information, virtual objects can be made to "hide" behind

real objects. Dual camera devices and stereoscopic displays can bring 3D vision to AR systems. Combining all these concepts, the capability exists to enhance the illusion the AR presents to a user. As this technology becomes better and smaller, the 3D AR interface may one day replace the notion of computing being confined to a desktop.

6.3 Acknowledgements

I would not be as fortunate as I am to be completing my Masters degree without the involvement of many people to whom I am very grateful. Whether directly involved in my research or not, you have gotten me to this point, and for that I would like to thank you.

To my family: I am a product of my environment, and you were the first environment I had. You provided me a home and still do. I've always had every kind of support I ever needed. I love you and am glad you remain an important part of my life to this day.

To my friends: Life is about balance, and without you I would have hit the ground years ago.

To my advisor, Dr. Eliot Winer: Your willingness to help in both the personal and academic development of your students is what makes you more than just a boss. I am extremely grateful for the opportunities and support that you have provided me, and I look forward to our future work together.

To all the faculty, staff, and students at VRAC: A great work environment can't be generated by a multi-million dollar, stereoscopic, immersive display system. Not yet, anyway. Your continued dedication to improving the lab and the projects being done there are the reason I enjoy where I work. I would like to thank the ASDS team (Ruqin Zhang,

Christian Noon, and Brandon Newendorp) for their work on that project. It served as the model upon which I built my application. Finally, thank you to Brett Nekołny for running the many, many FEA simulations which comprised the entirety of my source data.

REFERENCES

- 1 Smith, R.P., "The Historical Roots of Concurrent Engineering Fundamentals," *IEEE Transactions on Engineering Management*, Vol. 44, No. 1, February 1997, pp 67-78.
- 2 Chapman, C.B., Pinfold, M., "Design engineering – a need to rethink the solution using knowledge based engineering," *Knowledge-Based Systems*, Vol. 12, 1999, pp. 257-267.
- 3 Hazelrigg, G.A., "A Framework for Decision-Based Engineering Design," *Journal of Mechanical Design*, Vol. 120, December 1998, pp. 653-658.
- 4 Balling, R., "Design by Shopping: A New Paradigm," *Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization*, 1999, pp. 295-297.
- 5 Pahl, G., Beitz, W., *Engineering Design: A Systematic Approach*, Springer, New York, 1988.
- 6 Whitney, D.E., "Designing the Design Process," *Research in Engineering Design*, Vol. 2, No. 1, March 1990, pp. 3-13.
- 7 Ullman, D.G., *The Mechanical Design Process Third Edition*, McGraw Hill, Boston, MA, 2003.
- 8 Otto, K.N., Wood, K.L., *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River, NJ, 2000.
- 9 FLUENT for CATIA V5: Rapid Flow Modeling for PLM, *CFD Flow Modeling Software & Solutions from Fluent*, 2008, <http://www.fluent.com>, November 12, 2008.
- 10 Chalmers, P.A., "The role of cognitive theory in human-computer interface," *Computers in Human Behavior*, Vol. 19, 2003, pp. 593-607.
- 11 Pro/CONCEPT, Product Lifecycle Management (PLM) Software Solutions, Parametric Technology Corporation, 2008, <http://www.ptc.com/products/proengineer/proconcept>, November 12, 2008.
- 12 CATIA Imagine & Shape (IMA), International Business Machines, <http://www-01.ibm.com/software/applications/plm/catiav5/prods/ima/>, November 12, 2008.
- 13 Bowman, D.A., Kruijff, E., LaViola, J.J., Poupyrev, I., "An Introduction to 3-D User Interface Design," *Presence: Teleoperators and Virtual Environments*, Vol. 10, No. 1, February 2001, pp. 96-108.

- 14 Zhang, R., Noon, C. J., Winer, E., Oliver, J. H., Gilmore, B. J., and Duncan, J. R., "Immersive Product Configurator for Conceptual Design," *ASME International Design Engineering Technical Conferences 33rd Design Automation Conference*, September 2007.
- 15 Google SketchUp, Google, 2008, <http://sketchup.google.com>, November 12, 2008.
- 16 Google SketchUp, Google, 2008, <http://sketchup.google.com/community/>, November 12, 2008.
- 17 Milgram, P., Kishino, F. "A Taxonomy of Mixed Reality Visual Displays," *IEICE Transactions on Information Systems*, Vol. E77-D, No. 12, December 1994.
- 18 Hughes, D.E., Vogelpohl, S., Hughes, C.E., "Designing a System for Effective Use of Immersive Audio in Mixed Reality," *Proceeding of 2005 International Conference on Human-Computer Interface Advances in Modeling and Simulation (SIMCHI'05)*, January 2005, pp. 23-27, 51-57.
- 19 Woohun, L., Jun, P. "Augmented Foam: A Tangible Augmented Reality for Product Design," *Proceedings on the 4th IEEE and ACM International Symposium on Mixed an Augmented Reality*, October 2005, pp. 106-109.
- 20 Mazuryk, T., Gervautz, M., "Virtual Reality: History, Applications, Technology, and Future," Institute of Computer Graphics, Vienna University of Technology, Austria, 1996.
- 21 Azuma, R.T., "A Survey of Augmented Reality," *Presence: Teleoperators and Virtual Environments*, Vol. 6, No. 4, August 1997, pp. 355-385.
- 22 ARToolKit, <http://www.hitl.washington.edu/artoolkit/>, November 12, 2008.
- 23 Advanced Realtime Tracking GmbH, <http://www.ar-tracking.de/>, November 12, 2008.
- 24 3D Video Camera Technology Depth Sensing Products, 3DV Systems, 2006, <http://www.3dvsystems.com/>, November 12, 2008.
- 25 Schmeil, A., Broll, W., "MARA – A Mobile Augmented Reality-Based Virtual Assistant," *IEEE Virtual Reality Conference*, March 2007, pp. 267-270.
- 26 InterSense Inc. – Precision Motion Tracking Solutions, <http://www.isense.com>, November 12, 2008.
- 27 Olwal, A., Lindfors, C., Gustafsson, J., Kjellberg, T., Mattsson, L., "ASTOR: an autostereoscopic optical see-through augmented reality system," *Proceedings of the 4th*

- IEEE and ACM International Symposium on Mixed and Augmented Reality*, October 2005, pp. 24-27.
- 28 Wojciechowski, R., Walczak, K., White, M., Cellary, W., "Building Virtual and Augmented Reality museum exhibitions," *Proceedings of the 9th international conference on 3D Web technology*, 2004, pp. 135-144.
 - 29 Billinghamurst, M., Kato, H., Poupyrev, I., "The MagicBook – moving seamlessly between reality and virtuality," *IEEE Computer Graphics and Applications*, May 2001, Vol. 21, No. 3, pp. 6-8.
 - 30 Piekarski, W., Thomas, B., "ARQuake: the outdoor augmented reality gaming system," *Communications of the ACM*, Vol. 45, No. 1, January 2002, pp. 36-38.
 - 31 Marescaux, J., "Augmented-Reality-Assisted Laparoscopic Adrenalectomy," *Journal of American Medical Association Research Letters*, Vol. 292, No. 18, November 2004, pp. 2214-2215.
 - 32 Schwald, B., Seibert, H., "Registration Tasks for a Hybrid Tracking System for Medical Augmented Reality," *Journal of WSCG*, 2004, pp. 411-418.
 - 33 Fischer, J., Neff, M., Freudenstein, D., Bartz, D., "Medical Augmented Reality based on Commercial Image Guided Surgery" *Eurographics Symposium on Virtual Environments*, 2004.
 - 34 Bockholt, U., Bisler, A., Becker, M., Muleer-Wittig, W., Voss, G., "Augmented Reality for Enhancement of Endoscopic Interventions," *Proceedings of the IEEE Virtual Reality*, 2003.
 - 35 Persiani, F., Liverani, A., De Crescenzo, F., "Augmented reality approach to vehicle Prototyping," *Proceeding of First International Industrial Engineering Conference*, 2001, pp. 457-462.
 - 36 Sims, D., "New Realities in Aircraft Design and Manufacture," *IEEE Computer Graphics and Applications*, 2004, pp. 91.
 - 37 Zauner, J., Haller, M., Brandl, A., Hartmann, W., "Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures," *Proceedings of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2003.
 - 38 Pathomaree, N., Charoenseang, S., "Augmented Reality for Skill Transfer in Assembly Task," *IEEE International Workshop on Robots and Human Interactive Communication*, 2005.

- 39 Sidharta, R., Oliver, J., Sannier, A., “Augmented Reality Tangible Interface for Distributed Design Review,” *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, 2006.
- 40 The Eye of Judgement, Sony Computer Entertainment Inc., 2007, <http://www.us.playstation.com/EyeofJudgment/>, November 12, 2008.
- 41 SourceForge.net: ARToolKit, SourceForge, Inc., 2008, <http://sourceforge.net/projects/artoolkit/>, November 12, 2008
- 42 ARTag, <http://www.artag.net/>, November 12, 2008.
- 43 ARToolKitPlus, Graz University of Technology, 2008, http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php, November 12, 2008.
- 44 Studierstube ES, Graz University of Technology, 2008, http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbes.php, November 12, 2008.
- 45 OSGART: ARToolKit for OpenSceneGraph, HITLAB NZ, 2006, <http://www.artoolworks.com/community/osgart/>, November 12, 2008.
- 46 Kato, H., Billinghurst, M., “Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System,” *2nd IEEE and ACM International Workshop on Augmented Reality*, 1999.
- 47 OpenSceneGraph, <http://www.openscenegraph.org/projects/osg>, August, 12 2008.
- 48 Kleijnen, J. P. C., "A Comment on Blanning's Metamodel for Sensitivity Analysis: The Regression Metamodel in Simulation," *Interfaces*, Vol. 5, No. 1, 1975, pp. 21-23.
- 49 Simpson, T.W., Toropov, V., Balabanov, V., Vianna, F.A.C., “Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How Far We Have Come – or Not,” *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, September, 2008.
- 50 Simpson, T.W., Lin, D.K.J., Chen, W., “Sampling Strategies for Computer Experiments: Design and Analysis,” *International Journal of Reliability and Applications*, 2001.
- 51 Owen, A.B., "Orthogonal Arrays for Computer Experiments, Integration and Visualization," *Statistica Sinica*, Vol. 2, 1992, pp. 439-452.
- 52 Sloane, N.J., A Library of Orthogonal Arrays, <http://www.research.att.com/~njas/oadir/>, November 12, 2008.

- 53 Myers, R.H., and Montgomery, D.C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley, New York, NY, 1995.
- 54 Simpson, T.W., Mauery, T.M., Korte, J.J., and Mistree, F., “Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization,” *AIAA Journal*, Vol. 39, No. 12, December 2001, pp. 2233-2241.
- 55 Choi, S., Grandhi, R.V., Canfield, R.A., Pettit, C.L., “Polynomial Chaos Expansion with Latin Hypercube Sampling for Estimating Response Variability,” *AIAA Journal*, Vol. 42, No. 6, June 2004, pp 1191-1198.
- 56 Xiu, D., Karniadakis, G.E., “The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations,” *Society for Industrial and Applied Mathematics Journal for Scientific Computing*, Vol. 24, No. 2, October 2002, pp 619-644.
- 57 Abaqus FEA, Dassault Systèmes, 2008, http://www.simulia.com/products/abaqus_fea.html, November 12, 2008.
- 58 SolidWorks :: 3D CAD Design Software, Dassault Systèmes SolidWorks Corp., 2008, <http://www.solidworks.com/>, November 12, 2008.
- 59 Cosmos, Dassault Systèmes SolidWorks Corp., 2008, <http://www.cosmosm.com/>, November 12, 2008.