VARIABLE FIDELITY MODELING AS APPLIED TO TRAJECTORY OPTIMIZATION FOR A HYDRAULIC BACKHOE

A Thesis
Presented to
The Academic Faculty

by

Roxanne Adele Moore

In Partial Fulfillment
of the Requirements for the Degree of
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology
May 2009

COPYRIGHT 2009 BY ROXANNE A. MOORE

VARIABLE FIDELITY MODELING AS APPLIED TO TRAJECTORY OPTIMIZATION FOR A HYDRAULIC BACKHOE

Approved by:

Dr. Chris Paredis, Advisor School of Mechanical Engineering *Georgia Institute of Technology*	Dr. Seung-Kyum Choi School of Mechanical Engineering *Georgia Institute of Technology*
Dr. Bert Bras School of Mechanical Engineering *Georgia Institute of Technology*	Roger Burkhart *John Deere, Inc.*

Date Approved: April 3, 2009

ACKNOWLEDGEMENTS

I would first and foremost like to thank my parents, without whose support I certainly would not have made it this far.

I would also like to thank my fiancé, Michael Conigliaro, for his consistent love and support. He is always there to help me out at home when I get stressed out, even if that means cooking dinner, cleaning the kitchen, or doing the grocery shopping. For him I am eternally grateful.

I would like to thank my advisor Chris Paredis. Without his insights, many of the ideas appearing in this document would not exist. He is a great teacher and role model for me, and has really furthered my development as a researcher.

I would like to thank my friends and colleagues, especially Stephanie Thompson, for motivating me when I thought all hope was lost, and Alek Kerzhner, for continually solving my computer crises. I would also like to thank Suzanne Price, Rich Malak, Ben Lee, Aditya Shah, Kevin Davies, Tommy Johnson, and Jonathon Jobe for their friendship and support since my arrival at Georgia Tech.

I would like to thank my reading committee, Dr. Bert Bras and Dr. Seung-Kyum Choi as well as Roger Burkhart for taking the time to read and offer their insights on this work.

I would like to thank Deere & Company for their generous financial support of this work, along with the ERC for Compact and Efficient Fluid Power, supported by the National Science Foundation under Grant No. EEC-0540834. I would also particularly like to thank Roger Burkhart from Deere for the discussions that helped crystallize the ideas presented in this thesis.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

SUMMARY

Modeling, simulation, and optimization play vital roles throughout the engineering design process; however, in many design disciplines the cost of simulation is high, and designers are faced with a tradeoff between the number of alternatives that can be evaluated and the accuracy with which they can be evaluated. In this thesis, a methodology is presented for using models of various levels of fidelity during the optimization process. The intent is to use inexpensive, low-fidelity models with limited accuracy to recognize poor design alternatives and reserve the high-fidelity, accurate, but also expensive models only to characterize the best alternatives. Specifically, by setting a user-defined performance threshold, the optimizer can explore the design space using a low-fidelity model by default, and switch to a higher fidelity model only if the performance threshold is attained. In this manner, the high fidelity model is used only to discern the best solution from the set of good solutions, so that computational resources are conserved until the optimizer is close to the solution. This makes the optimization process more efficient without sacrificing the quality of the solution. The method is illustrated by optimizing the trajectory of a hydraulic backhoe. To characterize the robustness and efficiency of the method, a design space exploration is performed using both the low and high fidelity models, and the optimization problem is solved multiple times using the variable fidelity framework.

CHAPTER 1

INTRODUCTION

1.1 The Designer's Dilemma

Modeling, simulation, and optimization have become increasingly important to the success of design and decision making endeavors in a variety of disciplines. Although no model can ever perfectly emulate a physical system, performing simulations can be useful in design space exploration and subsequent design decisions. As models and simulation packages grow increasingly sophisticated, the error between model predictions and physical experiments has decreased. This accuracy, however, comes with a price tag of computation time. Thus, given limited computing resources, designers are often confronted with the difficult choice between the following two extremes as is depicted in Figure 1:

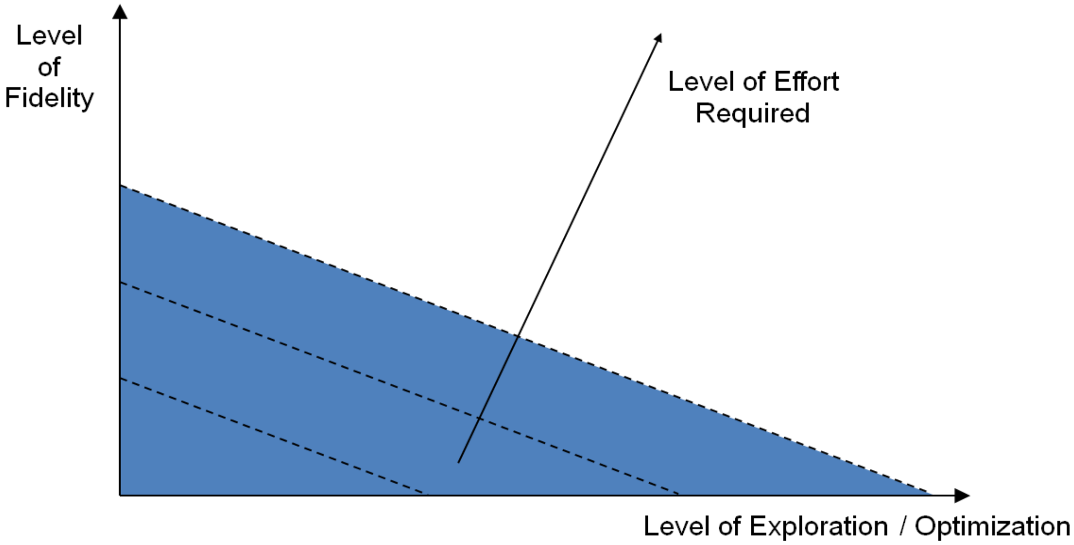


Figure 1: Designer's Dilemma:
Level of Fidelity versus Level of Exploration

1) *Explore many design alternatives with an inexpensive, low-fidelity model.* Significantly better alternatives are unlikely to be overlooked, but the best solution may not be discernable among the good solutions.

2) *Explore a smaller number of design alternatives with a very accurate but expensive high-fidelity model.* The best alternative is likely to be identified if it is among the small number of design alternatives considered, but there is no guarantee that an even better solution does not exist in the unexplored design space.

Does there exist a way to trade off broad exploration and high accuracy in a way that does not compromise the quality of the solution? To achieve this, innovative use of models is required beyond today's constant fidelity models, as is shown in Figure 2. It can be seen that very little quality and very little exploration is sacrificed at the optimal point (designated by the star) but the level of effort required is reasonable. The question is, how can this compromise be reached in practice?

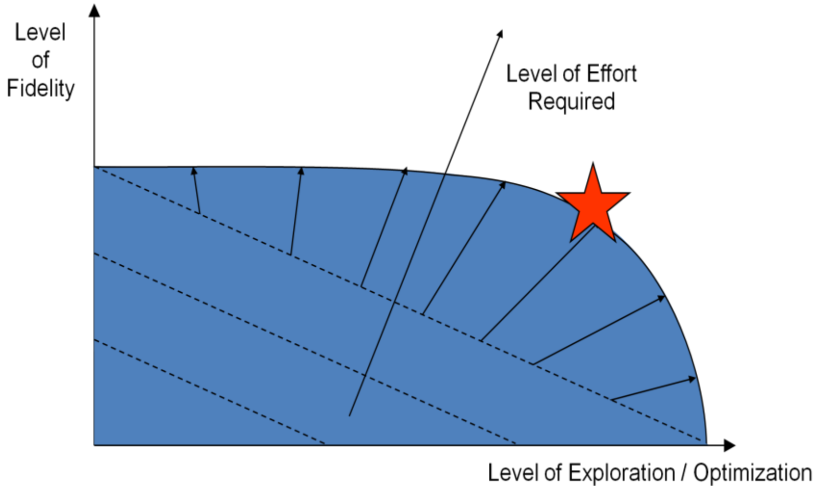


Figure 2: A Desirable Compromise Between Exploration and Accuracy

1.2 Motivating Question and Hypothesis

This designer's dilemma begs the following research question:

Is it possible to achieve both broad exploration and high accuracy by improving the way in which the available models are used?

In this thesis, it is hypothesized that using models of various levels of fidelity in the problem formulation stage of the design problem can help designers to achieve both broad exploration and high accuracy, even with limited computational resources. This hypothesis is based on the observation that high accuracy is only necessary when searching close to the optimum, as illustrated in Figure 3. In the figure, the accuracy bounds are very tight near the optimum, but grow more tolerant as we move away from the optimum. In this way, the overall behavior of the function is preserved, and the global optimum is maintained, but a lot of effort is saved by not calculating the function too accurately when far from the optimum.

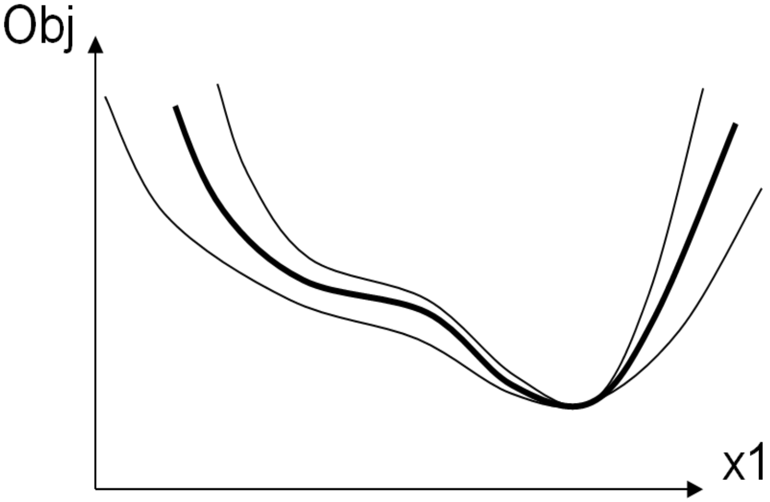


Figure 3: An objective function and its desired accuracy bounds.

Only near the optimum are very accurate predictions necessary

There is no need to accurately know how bad a poor design alternative is, so long as we can identify the general direction in which better solutions can be found. Assuming that accurate models require more computing resources than inaccurate models, a very accurate assessment of a bad solution's inadequacy is a waste of resources. Only when a design alternative is near the optimum is an accurate assessment required. In the neighborhood of the optimum, a low fidelity model would not allow one to identify the best solution from among these near optimal alternatives.

While others have proposed approaches to variable fidelity modeling in optimization [6, 13, 22, 26, 36, 44], most prior work relies on performance models that describe the same physical phenomena at different levels of fidelity (e.g. CFD with full Navier-Stokes or with Reynolds-Averaged Navier Stokes—RANS). Other approaches require the creation of surrogates (e.g. Kriging surface), and often these techniques are restricted to gradient-based optimization schemes. In this thesis, this perspective is broadened, and it is suggested that it may be acceptable to use completely different models in a variable fidelity framework as long as they are positively correlated. For example, a multi-attribute problem formulation might include a low fidelity model that completely ignores the least important attributes. Alternatively, one might approximate complex system models with idealized versions in a low fidelity system model.

As an initial step towards this goal, a framework is presented for managing two correlated models of varying levels of fidelity to create a composite cost function with the same global optimum as the high fidelity model. In the proposed framework, the function being optimized switches between a low and a high fidelity simulation in an automated fashion to navigate the design space. For this initial step, switching between the two models is regulated by a user-defined performance threshold, but part of the challenge to be addressed in the future is a more systematic approach for dynamically selecting from any number of models based on a cost-benefit tradeoff. In this thesis, the feasibility of such an approach to variable fidelity modeling in optimization is demonstrated, and a first step is made toward a thorough development of these ideas in future research.

Ultimately the focus and intended use case of this variable fidelity framework will be on exploration of systems architecture spaces to aid in concept selection, and not strictly on optimization of specific parameters for a given architecture. The difference is that the target problems resemble design space exploration, and consequently are often discrete and multi-modal. In such cases, gradient based optimization does not apply. Although the nature of this framework is that it is not restrictive to a particular type of problem, it is hypothesized that it will ultimately be most useful for these exploratory type problems that may require a lot of computational expense, broad exploration, and may not lend themselves well to standard gradient based optimization paradigms.

As a first illustration of this variable fidelity optimization framework, the approach is applied to perform a trajectory optimization on a hydraulic backhoe. While this problem is not a concept selection problem strictly speaking, it is a relevant case study for this framework because the high fidelity backhoe model is prone to system stiffness and high simulation times, due to the dynamics of the hydraulic subsystem and interaction with the trench walls. Additionally, because we are considering a space of trajectories, the fraction of acceptably good solutions (those that complete the desired task) with respect to the space of feasible solutions is relatively small.

1.3 Thesis Organization

The remainder of this thesis is organized as follows: In the next chapter, the problem background and prior work in this field is surveyed. This includes work using surrogate and other approximation based variable fidelity frameworks and feasibility tests. Some additional work in optimal motion planning is surveyed, since that is the focus of the illustrative example. In Chapter 3, the overall approach to the problem is formulated. The approach includes a mathematical validation, depiction of the framework itself, selection of appropriate high and low fidelity models, implementation of the switching function, and the choice of optimizers. In Chapter 4, the approach is demonstrated by performing a trajectory optimization on a hydraulic backhoe. This chapter details the creation of both the low and high fidelity backhoe models, and their associated objective functions. In Chapter 5 the results of the design space exploration and optimization for the trajectory optimization problem are discussed, along with contributions and future work.

CHAPTER 2

RELATED WORK

2.1 Modeling and Simulation in Design

The engineering design process involves the transformation of design requirements and objectives into a solution structure that is iteratively refined [35]. In systems engineering, there are several distinct stages to this process and many iterations. During conceptual design, system architectures are abstracted in terms of subunits and their interactions. The mere act of developing a basic solution structure is non-trivial, as the interactions of the various subsystems are vital to the success of the final design.

Once the overall system architecture is known, modeling and simulation are invaluable aids in making the final parameter selections. Selection of the best parameter values completely depends on the decision maker's preferences. While these preferences may vary based on corporate or consumer objectives and differing use cases, maximizing the overall utility of the final product requires understanding the tradeoffs that are being made.

This thesis is written from a Decision-Based Design (DBD) perspective [20, 27, 45], in that it is assumed decisions are best made using mathematically sound methods derived from decision theory. In particular, Multi-attribute utility theory (MAUT) [24], which is an extension of von Neumann and Morgenstern's utility theory [47] is the preferred method of eliciting designer preferences in spite of competing design objectives that may otherwise obfuscate which design candidate should be selected.

2.2 The Role of Optimization

Optimization is a very mature field in the engineering and mathematical communities. In systems engineering, optimization is most often employed in the later stages of design when the overall system architecture is known. An optimizer is commonly used to determine final parameter selections (e.g. cylinder diameters, gear ratios, pump sizes, etc.) based on designer preferences, which might be elicited using MAUT. In this scenario, the design space is more often continuous, making classical optimization methods fairly easy to apply. However, even in these cases where the solution structure is already known, maximizing the overall utility of the final product requires understanding the tradeoffs that are being made. In addition, the model(s) associated with the system architecture being considered can be computationally expensive, so that improving the efficiency of the optimization process without sacrificing solution quality is still a relevant research issue in this stage of the design process.

Greater difficulties arise when one considers optimization and decision making during the early stages of design or during the conceptual design process. Because of uncertainty and the cost of generating alternatives, the number of distinct concepts considered by designers is typically small; so, the 'optimization' at this stage is often done by brute force or by using a strategy like Pugh selection [39] or quality function deployment [4], or by calculating a utility associated with each alternative [29].

As we move into an age where generating design concepts and associated models of said designs is not so expensive and is becoming increasingly automated, a greater number of candidates can be considered at any stage in the design process. However, as the number of potential candidates grows, there are some inherent difficulties with performing a thorough evaluation of all potential design candidates. First, depending on the domain, performing a rigorous simulation or developing a detailed model of each candidate may be computationally prohibitive. And, even if each candidate could be modeled in a reasonable amount of time, this type of design space is often discrete and multi-modal, so conventional gradient based optimization may not be applicable.

Consequently, much attention has been given to evolutionary algorithms and other stochastic optimization algorithms in recent years due to their abilities to obtain near global optimality even in noisy design spaces exhibiting multi-modality and/or discontinuities [10, 34]. Much success has been achieved in solving complex engineering problems using evolutionary techniques [14, 50], but not much focus has been placed on using these techniques in the exploratory stages of design or during the conceptual design process.

The problem with these stochastic optimization algorithms and even with so-called classical optimization techniques is the number of function evaluations required per iteration. Depending on the algorithm parameters and the application domain, it may be computationally prohibitive to evaluate a high fidelity model for every point considered by a particular optimizer. Consequently, much work has been done to construct low fidelity approximations of high fidelity models that can either assist or take the place of these expensive simulations during the optimization process. This can be done in a variety of ways, but before discussing this prior work in detail, it seems appropriate to first discuss what is implied by the term fidelity as applied to modeling and simulation.

2.3 Fidelity and Related Terminology

Often in the literature the word *fidelity* is used interchangeably with accuracy; however, in this thesis, the terms fidelity, accuracy, resolution, and abstraction are used with the following meanings.

Fidelity refers to the degree to which a model reflects the behavior of a real system being modeled [18]. It is a property of a model. One can state that model A has higher fidelity than model B if model A includes additional phenomena beyond all the ones included B. For example, model A might be a transient model of a pendulum that includes friction at the rotational joint, while B might be a model of the same pendulum with the same properties without said friction. Note that this comparison between models A and B is a partial ordering; it is possible for A to include phenomena not included in B and vice versa. The term ‘level of fidelity’ must thus be used with caution because it is not a metric that can provide a full ordering of all models for a particular system.

Accuracy is different from fidelity in that it applies only to simulations (i.e., experiments performed on models [11]. It characterizes the degree of closeness of a prediction to its actual (true) value. Only in the context of a specific simulation can one assess accuracy. Depending on the context of the experiment, the same model can produce very different levels of accuracy.

Resolution is a special type of fidelity characterization that refers specifically to the level of discretization in either space or time. For instance, a finite element model has a higher resolution if the mesh is denser, meaning that the discretization intervals are smaller.

Finally, *abstraction* refers to the level of information content of a model. As is true for fidelity, it is a property of a model rather than of a simulation. Through a process of abstraction (or generalization), certain system properties are removed from a model so that one can no longer obtain information about these properties in an experiment or simulation [15].

Based on these definitions, the term variable fidelity modeling is somewhat of a misnomer because in the context of design optimization one is interested primarily in the *accuracy* of a model prediction, not its fidelity. Even though varying the level of fidelity is one way to influence the accuracy of a prediction, the level of fidelity does not directly characterize the accuracy. Yet, the term variable fidelity modeling is preserved in this thesis to maintain continuity with the existing literature.

2.4 Variable Fidelity Modeling in Optimization

The idea of using variable fidelity models in the optimization process for engineering design has been around for a long time [44]. In this early work, constraint deletion is employed, along with design variable linking and Taylor series of response variables to accelerate a structural system sizing optimization problem using an adaptation of the method of inscribed hyperspheres. More recent approaches to the problem include the space mapping approach [9], which attempts to create a mapping between the coarse (low fidelity) design space and the fine (high fidelity) design space that will yield the same computational outcome, i.e. apply a correction to the design variables of the low fidelity model to yield a result that more accurately depicts the desired high fidelity output. Most other work in this area can be found in either the aerospace or the multi-disciplinary optimization (MDO) literature and the most common low fidelity models are approximation or surrogate based, which will now be discussed in greater detail.

2.4.1 Approximation and Surrogate-Based Approaches

Seminal work in this field has been done by Alexandrov *et al.* [5-7]. In one of their approaches, an aerodynamic optimization is performed using the Euler equations over variable mesh sizes, effectively changing the resolution of the model. In another approach, variable-fidelity physics models are used, where the high-fidelity model is the Navier-Stokes equation and the low fidelity model is the Euler equation. In both cases, the method of correlation is a first order error function in a given trust region using augmented Lagrangian methods, which have been shown to converge to a Karush-Kuhn-Tucker (KKT) feasible point for constrained minimization problems [41]. Using the low fidelity model and this corrective factor, nested optimizations are performed on the low fidelity model, and then the trust region is adjusted based on the performance of the high fidelity model. While this method requires relatively few function calls to the high fidelity model, the method is restricted to derivative based optimization approaches. Other similar works [13, 26, 42, 48] apply a very similar trust region optimization technique successfully using either one of low fidelity model types presented by Alexandrov, or a response surface approximation as the low fidelity model. All of these works fall into the aerospace domain, typically dealing with optimization of airplane wings or other control surfaces for aerial vehicles.

Additional work has been done in the area of surrogates which can be constructed axiomatically after sampling the high fidelity model. Kriging surrogate models [28], sometimes called Design and Analysis of Computer Experiments (DACE) approximations, are an example of such a technique and are created by interpolating the sampled points in the design space. These surrogate models have been applied successfully as a replacement for a high fidelity function, but because it is an interpolation method, the high fidelity function must be called at intervals to either create the entire surface or adaptively update the surface [12]. The problem with these surface approximations is that the high fidelity design space must be sampled throughout the feasible region to construct the surface, even when far from the optimum, and the optimization is still performed over the surrogate surface, so it is not easy to guarantee that the surrogate optimum corresponds to the optimum of the original model. Some work has been done by Huang et al. [22] in integrating Kriging surfaces into a variable fidelity optimization framework. In this approach, the fidelity and location of function evaluations are selected using an expected improvement function that takes into account the evaluation costs, but there is still an initial fit involved that requires some significant computational cost.

A lot of other variable-fidelity optimization literature exploits the use of local response surface approximations (RSA's) with a variety of different sampling and interpolation techniques. [38] uses response surface approximations with optimization based sampling on two MDO test problems, in conjunction with the trust region methodology discussed previously, but does not achieve any improvement over Alexandrov et al. [8] also uses the trust region methodology with Latin-Hypercube sampling based RSA's, but does not achieve a KKT optimum point. [41, 42] use a concurrent subspace optimization technique and compare a variety of RSA constructions. [23] uses a quadratic polynomial RSA for a high-speed civil transport design problem.

Other surrogate assisted work includes the use of radial basis function surrogates [14, 34, 50] during the optimization process. Zang [49] provides a nice overview of the surrogates and optimization techniques applied to the MDO domain.

Another approach to variable fidelity modeling in optimization is to create feasibility constraints that can be tested quickly. If certain conditions are not met, then no further function evaluations take place. This type of strategy is used in conjunction with an optimization framework by Paredis [36], and by Gurnani et al [16, 17].

While much of the prior work discussed in this section makes use of gradient based or other classical optimization techniques, some work does use stochastic optimizers in conjunction with variable fidelity models [14, 16, 17, 34, 48, 50]. Most of these approaches still use local interpolation or surrogate surfaces that would imply a continuous design space, so it is not clear that the same techniques would be applicable in a more discrete design exploration stage of the problem.

2.4.2 Research Gap

The framework proposed in this work is different from other work in the variable fidelity optimization domain for a variety of reasons. First, the low fidelity model does not have to be a surrogate or other approximation based on sampling the high fidelity model. In addition, this framework is intended to be universally applicable to different system architectures in a variety of different disciplines, allowing for large design spaces with a small fraction of acceptably good solutions, as well as discrete and multi-modal design spaces where gradient based optimization does not apply. This methodology is also useful in combating stiff simulations in the design space, and does not dictate the choice of optimizers, models, design variables, or objective functions.

2.5 Optimal Motion Planning

Since the illustrative example in this thesis is essentially an optimal motion planning problem for a hydraulic backhoe, some work in this area was surveyed. Krishna [25] provides a lot of insight by working a trajectory optimization problem for a hydraulic excavator. Since this work was done about a decade ago, the models used are not as sophisticated as the ones seen in the illustrative example in this thesis. However, some ideas were provided as to how to characterize the trajectory, the environmental constraints, and how to speed up the optimization process. In his work, simulated annealing is the optimizer of choice, while the example problem seen later will make use of a genetic algorithm. Work done by Paredis [36] gives insight on how different optimizers and a variable fidelity optimization scheme could be applied to this type of problem.

2.6 Summary

This chapter provides a survey of some of the relevant literature with respect to system design and optimization, the use of variable fidelity modeling in optimization, and some optimal motion planning. It can be seen that modeling, simulation, and optimization are crucial to many design endeavors in any number of disciplines. However, as engineering problems grow increasingly complex (e.g. finite element methods, computational fluid dynamics, and systems engineering problems), the cost of simulation can quickly become unmanageable. Consequently, there is a lot of work being done in the area of surrogate modeling to keep these computation times manageable. To get better accuracy while subject to a computational budget, more work is being done using models of variable fidelity during optimization processes to maintain accuracy while doing much of the exploration with a less costly model. There are a variety of approaches to this problem, but many of them are restrictive in terms of the type of low fidelity model or the type of optimizer, or cannot guarantee global convergence. For this reason, an alternate framework will be offered in the next chapter. With some additional research and refinement, this framework should lend itself to any type of model and optimizer, and should be able to handle discontinuous and multi-modal design spaces, as might be encountered during the conceptual design phase. This new framework is then demonstrated on an optimal motion planning problem for a hydraulic backhoe in Chapter 4.

CHAPTER 3

THE VARIABLE FIDELITY MODELING APPROACH

Here we lay the framework for a method to incorporate models of varying fidelity into an optimization scheme. Under this regime, the optimizer calls an objective function, which returns the low fidelity response for a particular input by default, unless a user-defined performance threshold is met, in which case it returns the high fidelity response. In this way, the optimizer can move efficiently through the poor solutions, and only expend significant computational resources near the optimum.

3.1 Mathematical Validation- Preserving the Global Optimum

To establish mathematical validity for this approach, an unconstrained optimization problem is given:

$$f(x^*) = \min_{x \in \mathcal{D}} f(x)$$

where $\mathcal{D}$ is the allowable domain of x, x^* is the optimal solution, and $f(x)$ is a high fidelity cost function of x. Now $\tilde{f}(x)$, a less expensive, low fidelity approximation of $f(x)$, is introduced. The goal is to construct a composite cost function $F(f(x), \tilde{f}(x))$ such that x^*, the solution to the original problem, also minimizes this new cost function. It is proposed that

$$F\left(f(x), \tilde{f}(x)\right) = \begin{cases} \tilde{f}(x), \; x \in \mathcal{D}, \tilde{f}(x) \geq c_{th} \\ f(x), \; x \in \mathcal{D}, \; \tilde{f}(x) < c_{th} \end{cases}$$

where c_{th} is a user-defined cost threshold below which it is beneficial to evaluate the high fidelity cost function. It must now be verified that x^* also minimizes F, or specifically that

$$F(x^*) = \min_{x \in \mathcal{D}} F\left(f(x), \tilde{f}(x)\right) = \min_{x \in \mathcal{D}} f(x)$$

The following set of conditions is sufficient for the above assertion to be true:

1) $c_{th} > f(x^*)$
2) $\tilde{f}(x^*) < c_{th}$

These conditions guarantee that $F = f$at the global minimum and that F cannot have any minima smaller than the global minimum.

An alternate more stringent set of sufficient conditions is:

1) $\tilde{f}(x_i) \geq f(x_i)$ for all $x_i \in \mathcal{D}$
2) $\tilde{f}(x^*) < c_{th}$

In this set of conditions, the second condition from the first set is preserved, but the first condition is stronger. In this case, the low fidelity model must always be a conservative estimate of the high fidelity model, i.e., the high fidelity model always performs at least as well as the low fidelity model predicts. By adhering to this stricter set of conditions, it is less likely that artificial peaks or valleys are created in the composite design space, which is particularly important if a gradient based optimization scheme is to be used.

While these sets of sufficient conditions are useful academically in verifying the preservation of the global optimum, they are nearly impossible to test for from a practicality standpoint. Without knowing every point in the high and low fidelity design spaces, it would be nearly impossible to ensure that the low fidelity model is always conservative. It is possible, however, to quantify within a given confidence interval how often the low fidelity model is conservative using sampling techniques. In terms of verifying that the threshold is set appropriately such that the high fidelity model is evaluated at the global minimum, it may be necessary to take a small number of samples that may be in the neighborhood of the optimum and choose a sufficiently conservative threshold. Finally, it may be necessary to tune the threshold selection, just as optimization parameters often have to be tuned based on the outcomes of preliminary trials.

3.2 Overview of the Framework

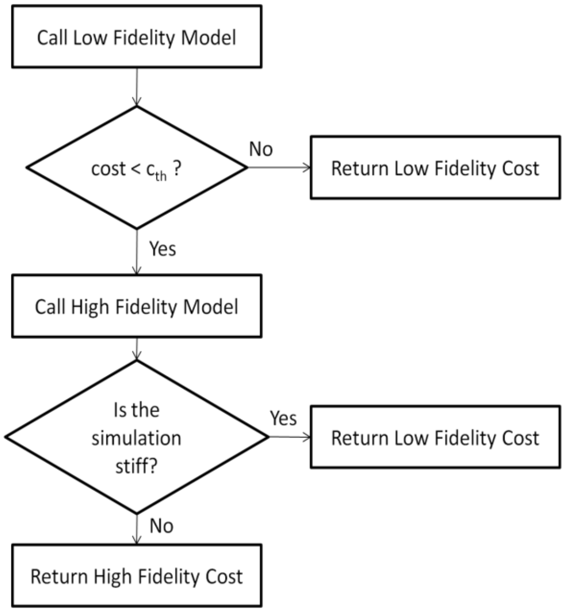


Figure 4: Variable Fidelity Cost Function Framework

This framework for the objective function is defined pictorially in Figure 4. In the figure, c_{th} is the user-defined cost threshold at which the switch from low-fidelity to high-fidelity occurs. The expected benefit results from the fact that the more expensive, high fidelity model is never evaluated for poor alternatives for which the cost threshold is not reached.

The figure also refers to infrastructure to handle failures in the high fidelity model (e.g. due to stiffness in the simulations). If the model is stiff, as may be determined by a maximum CPU time for simulation, then the low fidelity cost is returned. This additional test improves the robustness of the optimization.

3.3 Models of Various Levels of Fidelity

It is important to note that this framework is suggested under the assumption that repositories of models often exist prior to solving a particular optimization problem. If models at varying levels of fidelity were to be created from scratch every time an optimization was to be run, there would not be much or any payoff in incorporating these different models since the time recovered in the optimzation would probably be lost to model development. Assuming that multiple system models or at least a high fidelity model of the system of interest already exists, it is of value to try to make use of these available models in the most effective way possible. Because the variable fidelity optimization framework proposed in this thesis does not place any restrictions on the nature of the models and simulations it employs, it is not possible to provide a completely axiomatic approach to developing or selecting the high and low fidelity models, simulations, and/or cost metrics required to perform the optimization. However, some heuristics, suggestions, and examples can be provided to guide the selection.

Prior to constructing the relevant system models, it is important to select meaningful, mathematically sound objective functions that rely on measurable system attributes to evaluate the extent to which a design alternative meets the objective. In some cases, as in the illustrative example later in this thesis, a single attribute that can be measured using various computational outcomes may be a sufficient metric for a particular objective. In other cases, there may be competing objectives that cannot be easily combined, and it may be more appropriate to formulate the objective function as a multi-attribute utility function [24]. Either way, it is likely that the cost function associated with each individual model (previously defined as f and $\tilde{f}$) in the variable fidelity scheme will be slightly different; they may simply be functions of the same outputs of differing simulations, or the low fidelity cost may neglect or estimate certain parameters not provided by the low fidelity simulation. It should be kept in mind, however, that the low fidelity model is returning what should effectively be an estimate of the high fidelity objective function, and it is this value of the objective function that will be compared to the performance threshold.

When selecting or constructing a high fidelity model, the designer’s primary concern should be quality of the solution. It would thus be advisable to ensure that all of the computational outcomes required by the high fidelity objective function can be determined from the high fidelity model. This will ensure an accurate evaluation of the design alternative, assuming that the objective function is sufficiently comprehensive. In the case of the low fidelity model, the designer’s priority should be efficiency, since this model is simply being used to discern potentially optimal solutions from the entire design region. For this reason, the low fidelity model may not need to provide all of the computational outcomes required by the high fidelity model.

For this variable-fidelity optimization technique, it is vital that the high and low fidelity models operate over the same set of inputs, or design variables. This set of design variables must uniquely determine the outcome of the simulation in both models, or the optimization cannot be performed over this set of design variables because automated switching could not occur. Also, it is desirable for the low fidelity model be at least an order of magnitude more efficient (with respect to CPU time) than the high fidelity model, and not be prone to stiffness, timing out, or other failures in the context of the simulation environment in order to improve robustness of the optimization process.

3.4 Setting the Performance Threshold

The performance threshold, previously defined as c_{th}, is the value with which the low fidelity objective function output is compared at any new point in the design space. Selecting this threshold appropriately will depend on the range of possible outcomes of the objective function, the percentage of acceptably good solutions in the design space, and the optimizer being used. Obviously, if an unachievable threshold is selected, the high fidelity model will never be evaluated; conversely, if the threshold is too conservative, the high and low fidelity models will be evaluated at every point, effectively making the optimization process slower than optimizing the high fidelity model directly.

The range of possible outcomes addresses the order of magnitude of the objective functions; for example, if the objective function were measuring final cost, is an acceptable dollar amount, $5, $500, or $5 million? This will clearly depend on the system being optimized. Alternatively, if the objective function were a normalized multi-attribute utility function, the range of possible outcomes might be only 0 to 1. Something must be known about this range to choose a reasonable value for the performance threshold.

The percentage of acceptably good solutions will vary significantly based on the design problem and the allotted size of the design space. To recognize the acceptably good solutions, it should be ensured that the objective function returns sufficiently different values in the acceptable and unacceptable regions. Using this information, a performance threshold could be selected near or inside the boundary of the acceptably good region, so that resources are not heavily expended on truly bad solutions, but the character of the design space is maintained once the optimizer gets into the acceptable region.

The choice of optimizer may also affect the tuning of this parameter. If the threshold is set close too close to the optimum value, a gradient based optimizer or pattern search is unlikely to get to the right region without a reasonable starting point. However, under any optimization scheme, if the performance threshold is very sub-optimal, the utility of the variable fidelity scheme will be negated, as both the high fidelity and low fidelity objective functions will be evaluated at nearly every step.

3.5 Optimizer Selection

A significant benefit of this framework is that it does not limit the choice of optimizers to a particular genre. Choice of optimizer is best guided by a characterization of the design space with respect to dimensionality, local minima (maxima), flat regions, discontinuities, and the percentage of acceptably good solutions. This framework is particularly well suited to stochastic algorithms, such as genetic algorithms (GA's), evolutionary algorithms, particle swarm, or simulated annealing. Due to the randomized search efforts of these algorithms, the variable-fidelity framework can enable the early generations to move much more quickly, since many poor solutions are likely to be encountered. More significant resources are likely to be expended near the optimum, but this could be mitigated by using a hybrid search method. Any of these algorithms may require tuning or multiple starts, just as they might in the absence of variable fidelity models. However, under the variable fidelity scheme, if the algorithm is not tuned well and converges to a sub-optimal solution, the CPU time for that optimization is likely to be much smaller than if it were run using only the high fidelity simulation.

3.6 Summary

In this chapter, a variable fidelity modeling framework is presented to create a composite objective function with the same global optimum as the original high fidelity function, and sufficient conditions are provided to ensure that this condition is met. The objective function is evaluated using the low fidelity model first by default, and then proceeds to evaluate the high fidelity model for the same set of inputs if and only if the user defined performance threshold is met. The framework also includes a way of handling stiffness and unreliable simulations by returning the low fidelity approximation if the high fidelity simulation times out. Additionally, some insight is provided as to how an appropriate low fidelity model, switching threshold, and optimizer might be selected for a particular domain. Since this is a preliminary work, no rigorous means of selecting a low fidelity model or threshold can be provided at this time, and this is left for future work.

CHAPTER 4

TRAJECTORY OPTIMIZATION FOR A HYDRAULIC BACKHOE

Significant design challenges often arise in the context of systems engineering where the interactions between several subsystems yield complex system dynamics [19, 43]. It has been shown that it is preferable to optimize the composite system, and not to decouple the system into individual subsystems and optimize each system sequentially [40]. This is so because one parameter in, for example, a hydraulic subsystem is dependent upon parameters in the mechanical or electrical system with which it is coupled. Additional complications result when a system's use cases, operator performance, and obstacles or other constraints a system might encounter in a particular environment are considered. These complex models are often prone to stiffness, making it difficult to guarantee the convergence of optimizations performed on them. In this example, a hydraulic backhoe model which exhibits many of these behaviors is used as the test-bed for a variable fidelity optimization scheme. We switch between this model and a low-fidelity model containing an idealization of the hydraulic subsystem to perform a trajectory optimization.

4.1 Problem Setup

The first step in making any design decision is to identify the highest level objectives, or fundamental objectives [24]. It is assumed that our decision maker is the owner of a construction equipment contracting company, and thus her fundamental objective is to maximize profit. To do so, she negotiates contracts where she is paid for completing tasks within a certain timeframe. She owns a fixed number of backhoes, like the one depicted in Figure 5, which she contracts out for trench digging and other similar tasks.

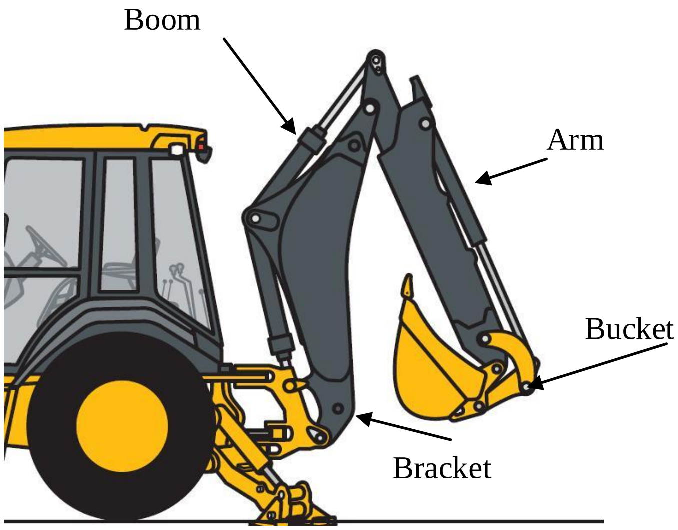


Figure 5: Backhoe Articulated Arm Linkage [1]

For each task, she hires a professional operator at some hourly wage, and a fixed hourly cost for the use of the backhoe is assumed (for maintenance, wear and tear, and the time spent on the task as opposed to being contracted out elsewhere). She also pays for the fuel consumed by the backhoe during each task. Without worrying about the fixed cost she is paid at the end of the task, we can assume that she wants to minimize her operating costs in order to maximize her overall profits. In this case, our decision maker's operating cost function would be:

$$\begin{gathered}\text{Cost} = \text{Total Task Time} \times (\text{Backhoe} + \text{Driver Hourly Costs}) \\ + \text{Fuel Consumed} \times \text{Fuel Cost}\end{gathered}$$

Now we focus on the task of trenching, where the backhoe operator is asked to dig a trench of a particular width, depth, and length in a specified location. Trenching can be broken down into *dig cycles*, where the backhoe removes soil from the trench and deposits it outside of the trench in a repetitive manner. The optimization problem is now to minimize operating costs over the space of dig cycle trajectories. The associated influence diagram [21] for this problem is depicted in Figure 6.

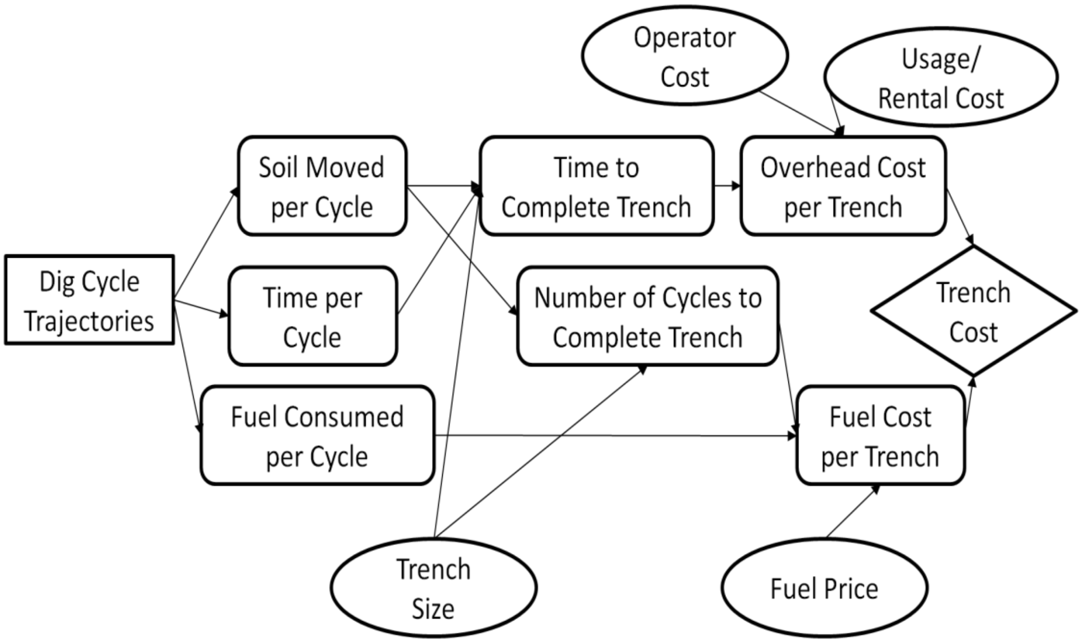


Figure 6: Influence Diagram for Trenching Operating Costs

4.2 Defining the Dig Cycle Trajectory

A dig cycle consists of three individual segments:

1. The soil is gathered into the bucket at the desired dig site.
2. The full bucket is moved to the dump site, whereupon the soil is posited.
3. The backhoe manipulator moves back into position for the next dig.

Due to the complex nature of soil-tool interaction and the variability of soil behavior (ranging from sand to rock to mud and everything in between), we assume that the first trajectory segment is known. Because the third portion of the trajectory is dependent upon the final state and time for the second portion of the trajectory, it is more practical to optimize the third segment after the second segment has been optimized. This may be revisited in future work. In this work we focus only on the second portion of the dig cycle, where the full bucket starts in the trench, as is shown in Figure 7. The bucket is lifted from the trench, positioned over the dump site, shown in Figure 8, and the soil is released. For this problem, it is assumed that the excavated soil is deposited along the edge of the trench in a designated dump zone, though it would certainly be viable to consider depositing the soil in a truck bed or other dumping ground. It is also assumed that the bucket starts this segment of the dig cycle loaded with 100kg of soil.

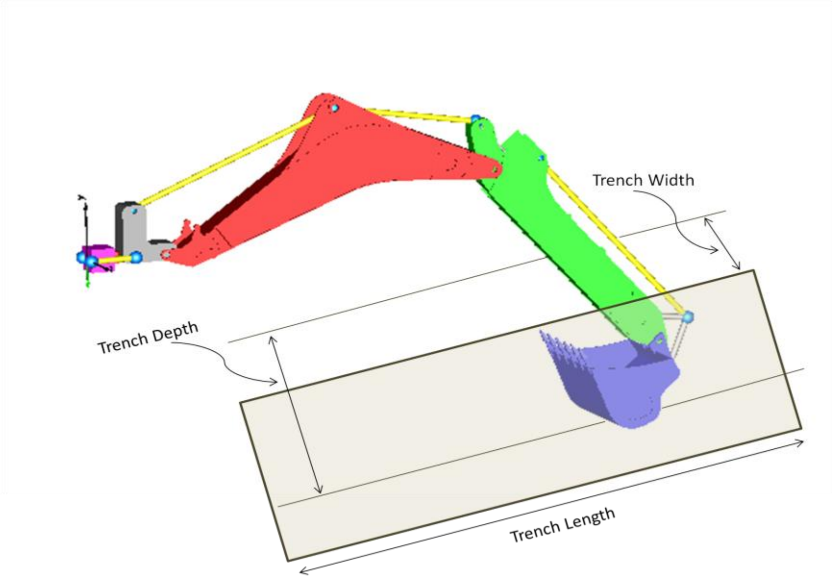


Figure 7: Starting Position of the Backhoe Manipulator- Full Bucket, In Trench

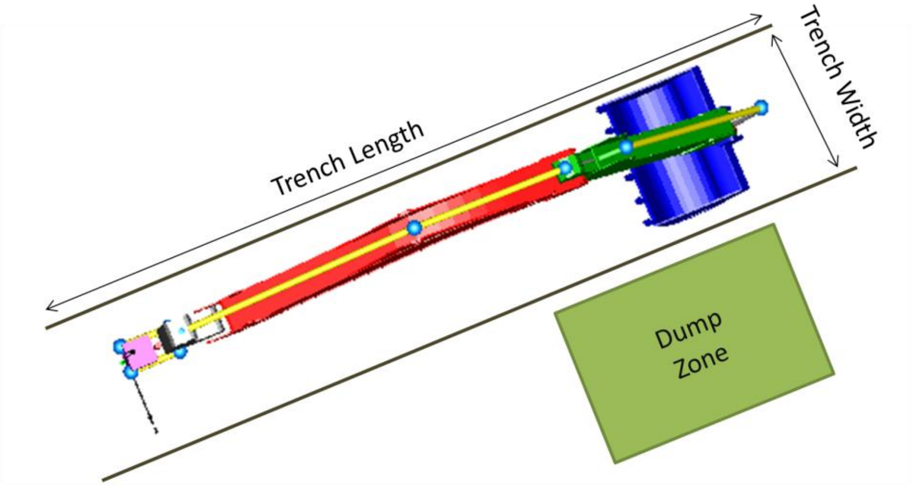


Figure 8: Position of Dump Zone with Respect to Trench

4.3 Design Variables

Assuming known starting and ending positions for each joint in the backhoe manipulator, we seek to optimize the trajectory from post-dig to post-dump using switching times as design variables. That is, we assume that each of the four joint angles is known at the start time and at the time of completion, and the only variables that are introduced are the times at which switches occur between the two. This keeps the number of design variables to four, one switching time for each degree of freedom, and these variables are set to range from 0-10 seconds. While many other means of characterizing the trajectory were considered, it is assumed that there is value in trajectory simplicity, so that such behavior could conceivably be emulated by a backhoe operator in practice. By employing a minimum number of switches, we also simplify the optimization problem considerably.

4.4 The Cost Function

Assume that we are asked to dig a trench that is 1m wide, 2m deep, and 15m long, where the total soil removed would subsequently be 30m^3. Assuming an average soil particle density of 2.35 g/cm^3, the trench necessitates the removal of 70,500kg of soil. If we assume the hourly rate of the backhoe and driver to be $\$50/\mathrm{hr}$, and the cost of diesel fuel to be $\$3/\mathrm{gal}$, we can obtain the total task time and fuel consumed using a detailed simulation of the backhoe's behavior for a task.

Because the simulation only accounts for a small portion of one dig cycle, we scale the results in terms of completing the whole task. Using the amount of soil deposited in the dump zone (m_{soil}) and the time to complete the task (T_{stop}) we can estimate the number of required dig cycles to finish the trench, and the time required to do so. Since we are only dealing with one of three segments of the overall dig cycle trajectory, we multiply the time from this segment by three to get a more realistic sense of the time required to finish the job. Using this time, and the amount of fuel consumed during the dig cycle (f_c), we can estimate the total fuel consumed during the entire trenching project in a similar matter. These data are sufficient to calculate the total cost of the project, based on the formula given in the problem setup. The final cost formula is:

$$\text{Cost}=\frac{3\times 70{,}500\text{ kg}}{m_{soil}}\left(\$50\times T_{stop}\ +3\times f_c\times {}^{\$3}/_{\text{gal}}\right)$$

Now, the only problem with this formula is that it would return an infinite value or an error in the case where $m_{soil}=0$, which occurs when no soil actually gets deposited into the acceptable dump region during the dig cycle. This would imply that the trench is never actually completed, and thus costs an infinite amount of money. Because this is not numerically pliable, a complementary cost function is provided to account for these infinite cost cases:

$$\text{Cost}=10{,}000\times\left(\$50\times T_{stop}\ +3\times f_c\times {}^{\$3}/_{\text{gal}}\right)$$

In this equation, 10,000 is simply an arbitrarily high number used to scale the infinite cost case to something that is an obviously poor solution, but T_{stop} and f_c are used to guide the optimizer toward better solutions.

4.5 The High Fidelity Backhoe Model

Having identified the key modeling issues using the influence diagram, we create an energy based model of the important subsystems of the backhoe using the Modelica modeling language [31, 46] in the Dymola software package [2]. We focus on the backhoe manipulator, neglecting the motion of the cab and the front loader mechanism. The manipulator is an articulated arm consisting of three links with four degrees of freedom: three rotational joints on the linkage, and a mounting bracket which is capable of swiveling along a vertical axis. The linkage and component definitions are given in Figure 5.

To encompass all of the relevant modeling issues shown in the influence diagram, the high fidelity simulation includes a mechanical subsystem, a hydraulic subsystem, a signaling sequence to generate the trajectory, and a feedback control loop to help the system track the prescribed input trajectory, in lieu of a professional operator.

4.5.1 Mechanical Subsystem

The mechanical subsystem is modeled using the MultiBody Mechanics library, which is part of the Modelica standard library. The main links are connected using actuated revolute joints. Mass and inertia properties are included for each significant component, and each joint is subject to Coulomb friction. There is also a simple soil model to determine spillage from the bucket based on the mass of soil in the bucket, and the absolute angle of the bucket. The mechanical subsystem is depicted in Figure 9.

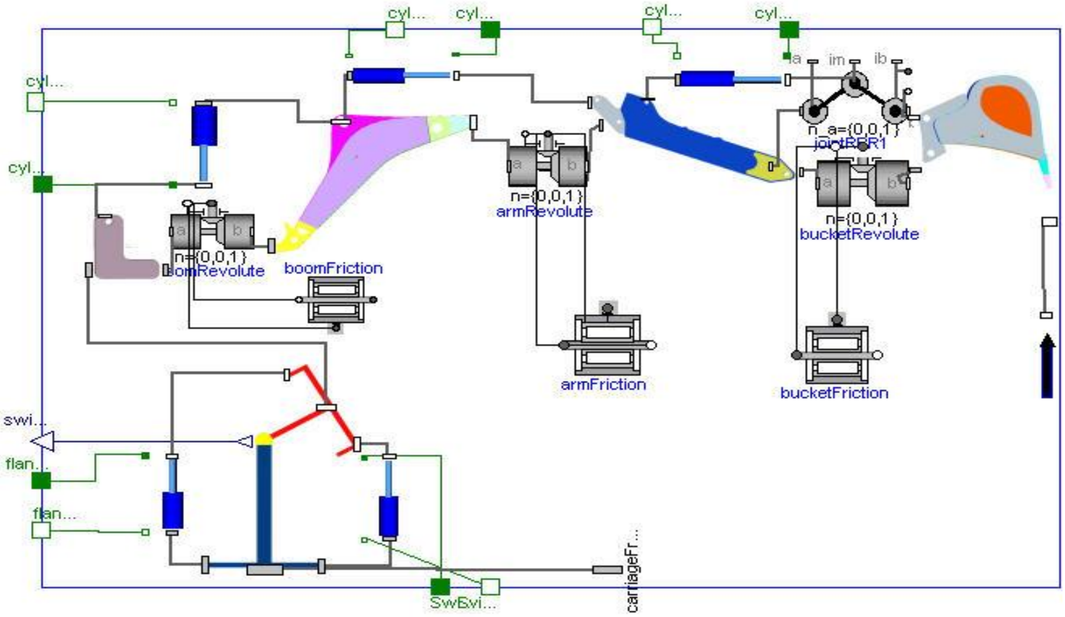


Figure 9: Mechanical Subsystem for High Fidelity Backhoe Model

The main components (boom, arm, bucket, and bracket) were modeled as CAD files in Pro/ENGINEER [3] and were used to extract reasonable inertia tensors for each component. The assembled manipulator using CAD models for the main links is depicted in Figure 10.

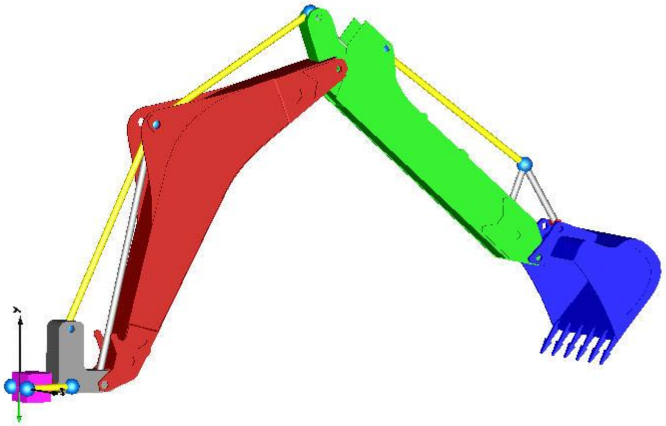


Figure 10: Assembled Backhoe Manipulator using CAD Parts

4.5.2 Hydraulic Subsystem

The hydraulic subsystem is modeled using an open source fluid power library [37]. The backhoe arm has three hydraulic cylinders which actuate the crowd, bucket, and boom. There are two more cylinders which control the swing of the mounting bracket. There are four-way three-position valves controlling the actuators. All of the actuators are powered by a variable displacement pump, which is controlled by a pressure compensating load sensing (PCLS) circuit. Under the PCLS paradigm, the pump generates the highest output pressure as required by the valves. The pump is powered by a diesel engine, which is approximated using a constant speed source.

Additional assumptions for the hydraulic system are made:

- The diesel engine is approximated using a constant speed source.
- Thermal aspects of the system are neglected.
- The variable displacement pump's control system is approximated using a pressure sensor and a PID controller.

The hydraulics detail model is shown in Figure 11.

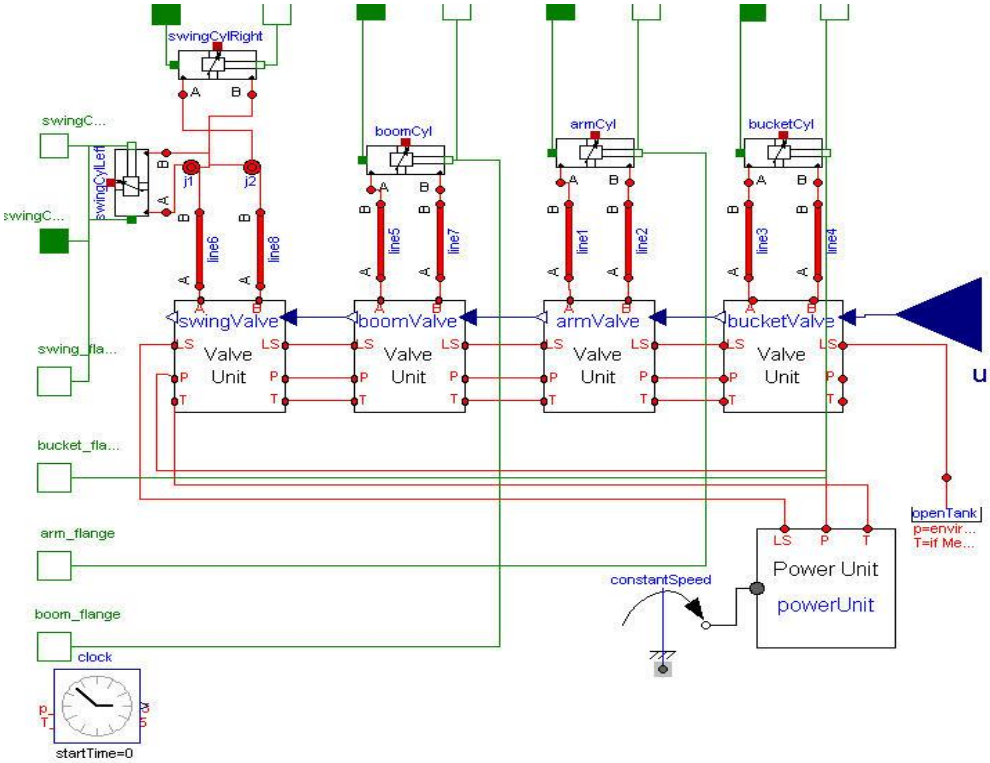


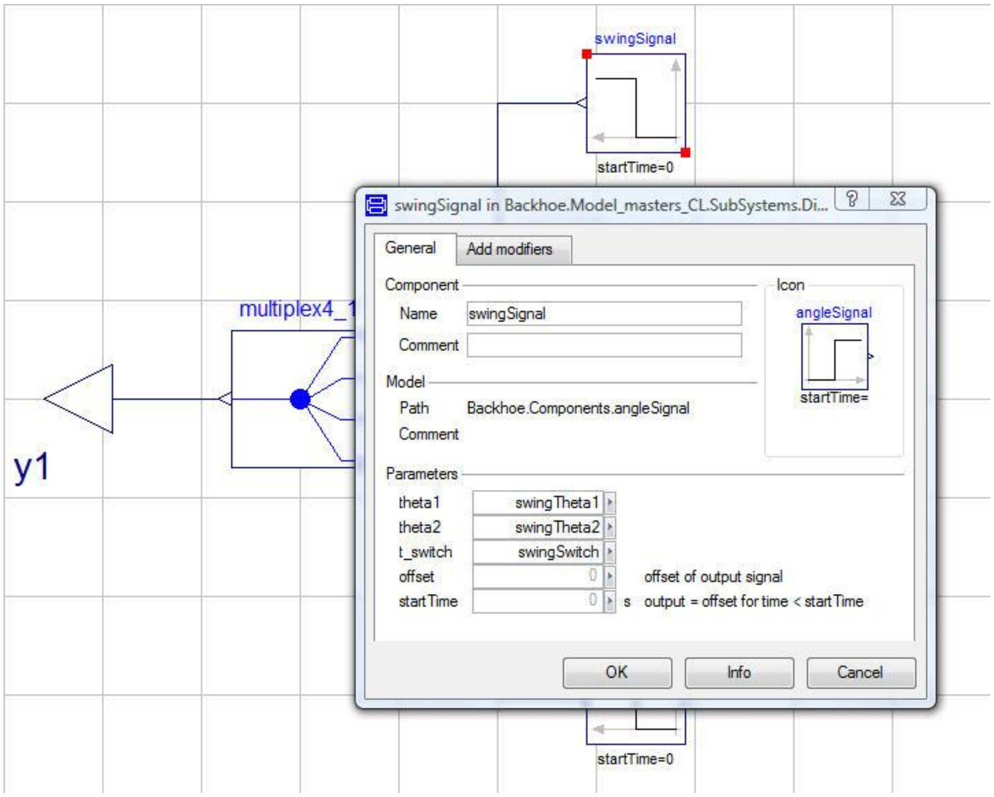
Figure 11: Hydraulic Subsystem for High Fidelity Backhoe Model

4.5.3 Landscape and Penalty Models

The trench walls are included in the high fidelity model, since the bucket starts in the trench, as is depicted in Figure 7. The trench model is created using nonlinear functions of position and velocity of the bucket joint; forces are applied to the manipulator when the bucket position sensor is detected at the trench walls. The nonlinear function effectively creates a stiff spring, which prevents the walls from being violated, but also creates simulation stiffness.

4.5.4 Trajectory Specification

To create the trajectory per the technique specified in the previous section, step functions are supplied to the high fidelity model at the specified times. A basic input model is shown in Figure 12. The signals provided are step functions that start at the initial angle (which is assumed to be known) and end at the final angle. The switching time between the two is the design variable for each degree of freedom. At this time, it is assumed that the final angles positioning the manipulator over the dump zone are known, but it is a logical extension of this problem to consider optimizing over the space of the final angles as well as the switching times.



swingSignal in Backhoe.Model_masters_CL.SubSystems.Di...

Name swingSignal

Comment

Path Backhoe.Components.angleSignal

Comment

theta1	swingTheta1		
theta2	swingTheta2		
t_switch	swingSwitch		
offset	0		offset of output signal
startTime	0	s	output = offset for time < startTime

Figure 12: Sample Input Signal to Specify Trajectory

4.5.5 Controller Subsystem

A proportional-integral-derivative (PID) control scheme acts as the backhoe's operator, allowing the manipulator to track the provided input signals. Separate controllers are used for each degree of freedom, as is depicted in Figure 13. However, these PID controllers are specially designed with a dead zone to prevent chattering about the set point, and they also include input limiters to scale the inputs to the valves to have an absolute value between zero and one. The structure of one of these PID controllers is given in Figure 14.

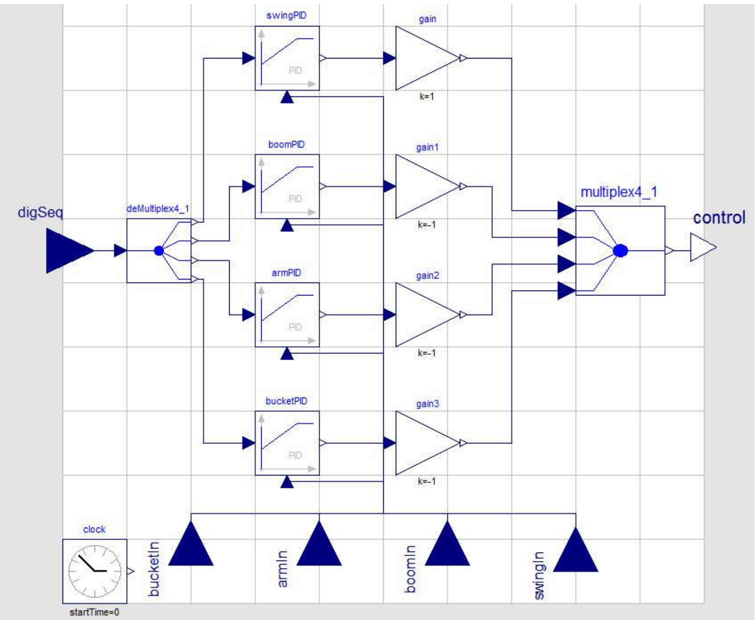


Figure 13: High Fidelity Control Subsystem

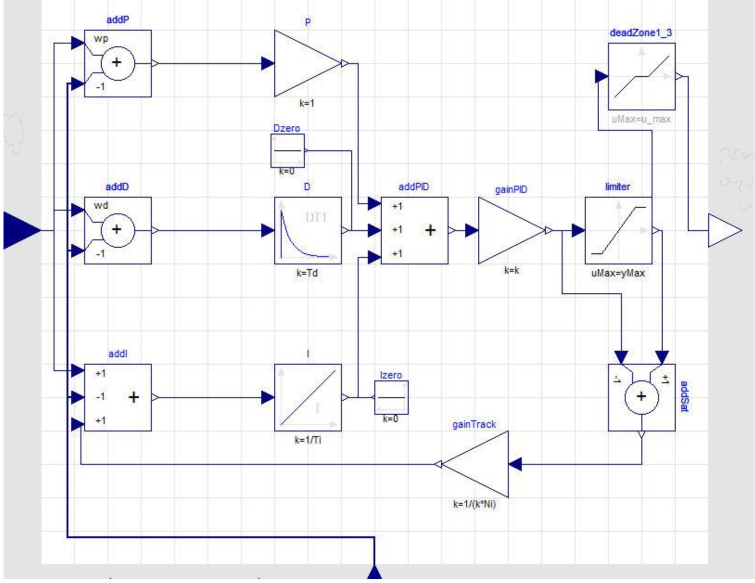


Figure 14: Example of Limited PID Control Structure with Dead Zone

The dead zones implemented in the PID controllers for this feedback regime are different from a standard hysteresis in that they are not path dependent, for simplicities sake. For each controller, a tolerance about the set point is given, and if the control signal is within that margin, it is set to zero by the dead band. While this means that the final angles may not be exact in this model, the tolerances ensure that the manipulator gets close.

The tuning of the PID controllers is a compromise between response speed and system stiffness. Because the larger hydraulic cylinders are not really designed for speed, often a slower controller response time is desirable to prevent the system from becoming stiff. An additional challenge is posed because the time at which each cylinder is to move is unknown. There are significant differences in the system response when cylinders are asked to move simultaneously, as opposed to sequentially, because the pump can only provide so much pressure and flow rate. The system makes compromises if the demand exceeds the possible supply, but this often slows down the simulation. This is an unfortunate difference between the simulated backhoe and an actual backhoe.

4.5.6 Composite High Fidelity Model

The composite high fidelity model depicting the connections between the various subsystems and source signals is shown in Figure 15.

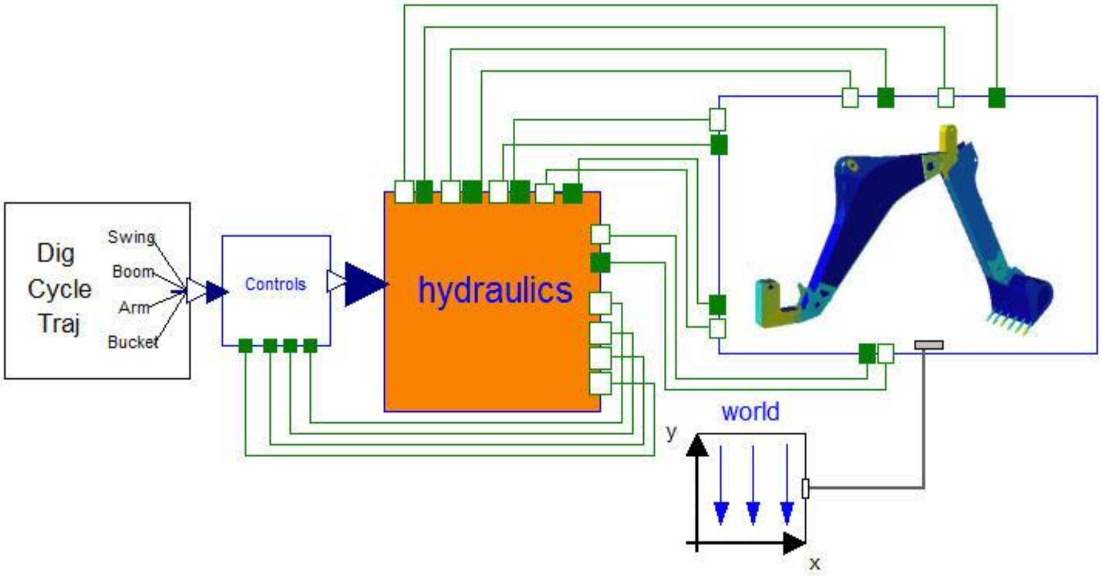


Figure 15: Composite High Fidelity Backhoe Model

4.6 The Low Fidelity Backhoe Model

The low fidelity model is constructed with the intent of discerning the acceptably good trajectories from the ones that do not complete the task. To make the simulations fast, we must identify the most significant sources of stiffness and computational expense in the high fidelity model, and then pare down the information contained in the model. Based on debugging statistics provided by the Dymola simulation environment, it is found that the hydraulics and the trench walls are the primary sources of computational expense. Thus, it is logical to simply remove the hydraulic subsystem and actuate the joints directly. Additionally, the trench walls are removed and replaced by a computationally simpler penalty function. The low fidelity model information is thus a subset of the high fidelity modeling information, and we can capture most of the relevant information for the cost function except for fuel consumption as is shown in Figure 16.

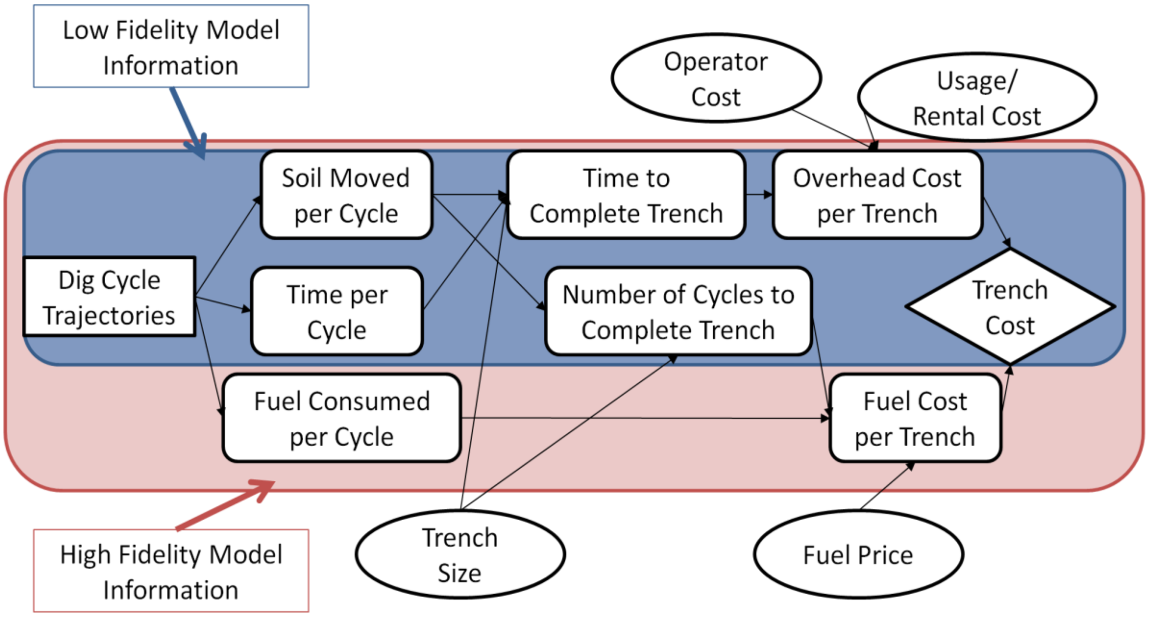


Figure 16: High and Low Fidelity Modeling Information

To conserve modeling resources, the same mechanics subsystem from the high fidelity model is used in the low fidelity model, except that the friction in the joints is removed. Instead of using the same step input signals used in the high fidelity model, ramp signals of appropriate durations with the same start times are used to test the prescribed trajectory. These ramp signals take into account the inherently slower response time of the high fidelity model.

The low and high fidelity simulations both use the same simulation parameters, and are allowed to run for 10 (simulated) seconds, unless the bucket is emptied beforehand, in which case the simulation is terminated. The fuel consumed, the time it takes to complete this portion of the cycle, and the amount of soil that makes it into the designated dump zone are subsequently used to calculate the total cost for the trenching project. For the low fidelity model, since there is no fuel consumption output, a very conservative estimate of the maximum fuel consumed during the cycle is used in an effort to reduce the number of artificial peaks in the design space. The composite low fidelity model is shown in Figure 17.

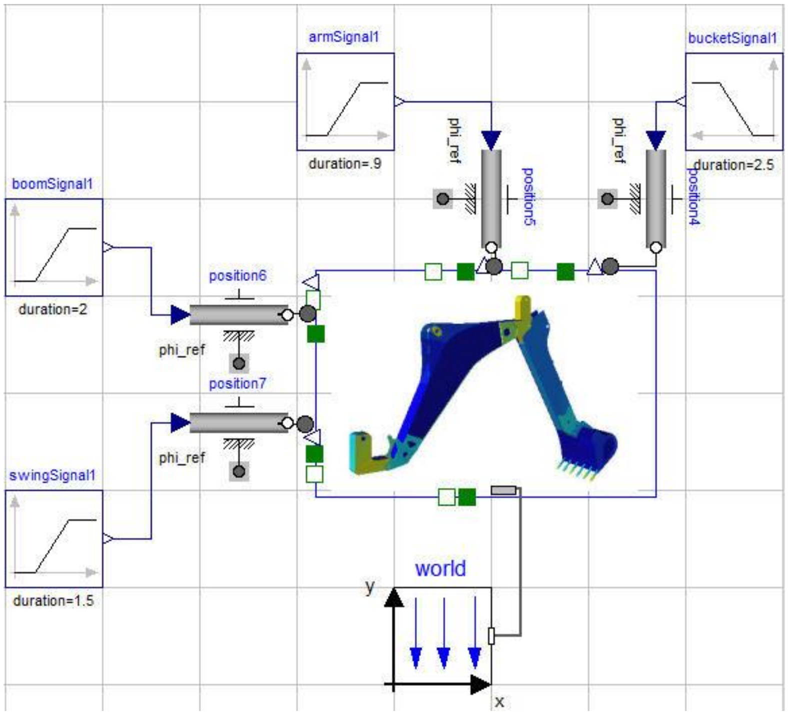


Figure 17: Low Fidelity Backhoe Model

4.7 Selection of the Performance Threshold

The performance threshold (c_{th}) is selected based on knowledge about the feasible range of the cost function and the range of costs associated with acceptably good solutions. Because of the way the cost function is structured for the cases where no soil lands in the designated dump zone, there is a large discontinuity of several orders of magnitude between the cycles that are productive and the ones that are not. This discontinuity is readily apparent in Figure 18 in the next section. Obviously, it is logical to choose a performance threshold that is inside of this discontinuity, so that only the low fidelity model is called in this far-from-optimal region.

From having some familiarity with the model, and knowing a few productive trajectories, it is known prior to optimization that the high fidelity model is capable of achieving a task cost of approximately $200. Without knowing the minimum cost that the model can attain, it is assumed that this cost will lie somewhere below this $200 benchmark. To ensure that the design region where the high fidelity model is evaluated is not too small to be found c_{th} is first selected as $500. However, to test the robustness of the framework, tests are also run with c_{th} selected as $250 (very close to the low fidelity optimum of $228), $350, and $1000.

4.8 Selection of the Optimizer

Due to the presence of discontinuities, weak local minima, and strong local minima in the 4-dimensional design space, a genetic algorithm (GA) is selected to perform the optimization. However, this selection was prompted by some experimentation with a pattern search optimizer. The pattern search optimization revealed that the convergence point was highly sensitive to the starting point, and so a global optimization scheme is used for this example. For the GA, a population size of 20 is used, with a crossover rate of 0.9 and a Gaussian mutation function.

4.9 Summary

In this chapter, a trajectory optimization problem for a hydraulic backhoe is set up. This includes the problem definition, specifying the trajectory, and the relevant high and low fidelity system models. The high fidelity model contains a means of providing trajectory input signals, a feedback control loop to regulate tracking of the input signals, a mechanical linkage, and a hydraulic subsystem. The low fidelity model can provide an estimate of the required time to finish the task, as well as how much fuel is consumed during the process. The low fidelity model contains a frictionless version of the high fidelity mechanics body and joint actuators to test out potential trajectory candidates. This model can only provide an estimate of the time to complete the trenching project, so a conservative estimate is used for fuel consumption. For this problem, a cost threshold of $500 is selected, but will also be performed with a few other thresholds to test the robustness of the framework. A genetic algorithm is selected to perform the optimization. A design exploration, sensitivity analysis, and the optimization results for the various thresholds are presented in the next section.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Computational Resource Allocation

To test the success of the variable fidelity design space with respect to computational resource allocation, a Latin-Hypercube Sampling (LHS) is performed over the whole space. LHS is a stratified sampling technique first presented by McKay et al. [32], and it serves to ensure that all portions of the design space are represented. For this experiment, 1000 samples are used over the full available range of each of the design variables, i.e. each of the four switching times is allowed to vary between 0 and 10 seconds. One set of inputs is generated, and that same set of randomly generated inputs is used on both the low and high fidelity models, and the results are plotted against each other so that the correlation between low and high fidelity cost can be visualized. The high fidelity stiff simulations, i.e. the trials that timed out after 200 CPU seconds, are assigned an arbitrarily high cost of $\$4.5 \times 10^5$ so that they are easily discernable.

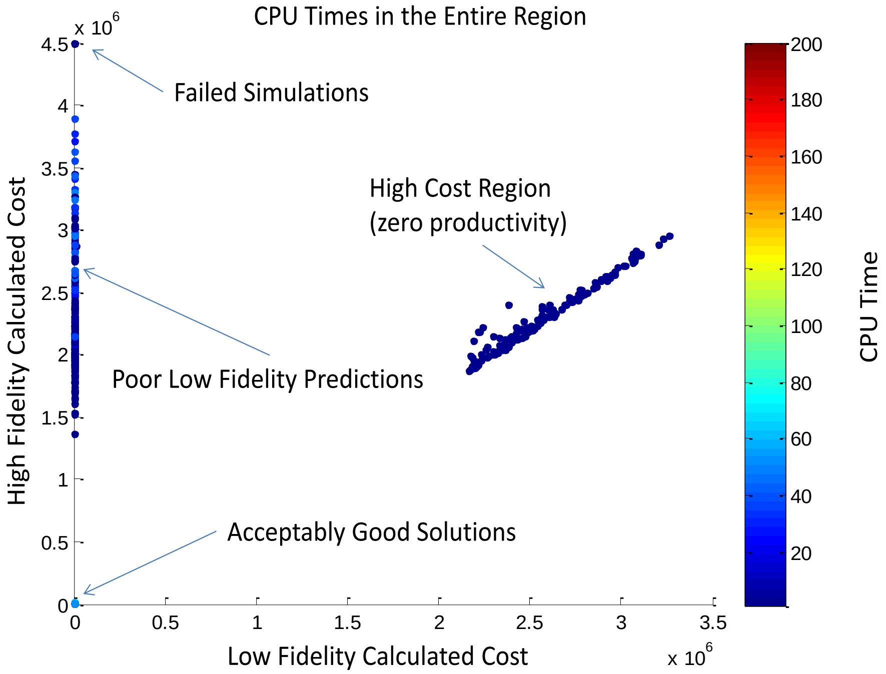


Figure 18: LHS Samples Spanning the Design Space

Figure 18 depicts the samples taken over the whole range of the design space. The color scale, representing CPU time for a given simulation, is determined based on the $500 performance threshold. Any cost below this threshold is assigned the high fidelity CPU time, while those that lie below are assigned the low fidelity CPU time for the given inputs. The four distinct regions of the design space are readily apparent in this figure: the failed high-fidelity simulations, the set of unproductive solutions, the set of poor predictions, and the set of acceptably good solutions. It can be seen that the CPU times in the unproductive solutions are very low, and are mostly low in the set of poor predictions.

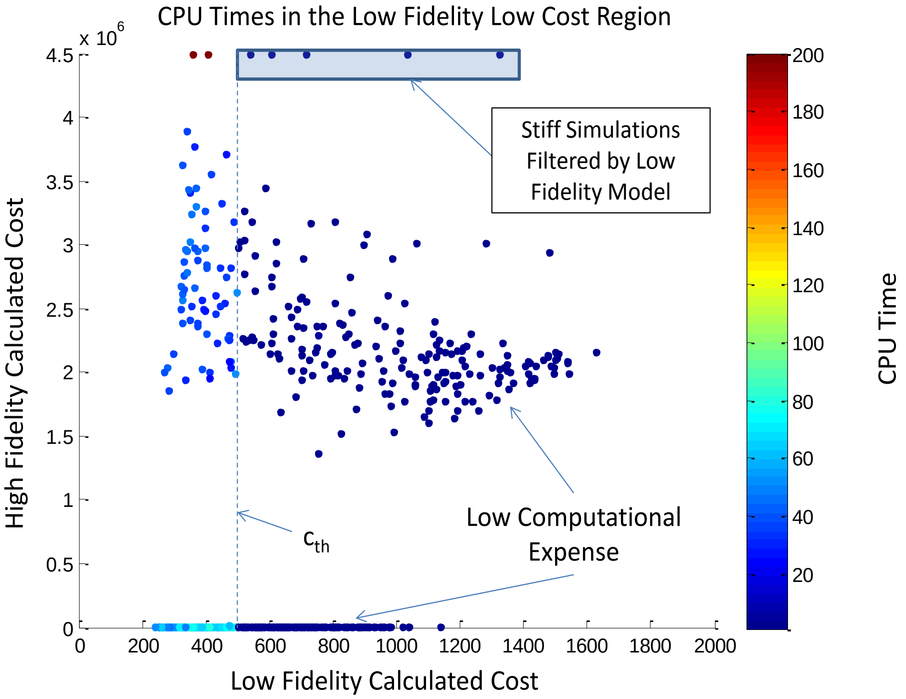


Figure 19: LHS Samples in the Low Cost Region

Figure 19 shows a close up view of the same LHS sample space in the low cost range as predicted by the low fidelity model. Here, the poor predictions are visible, and the good predictions are clustered along the x-axis. In this region, it is clear how the choice of the performance threshold impacts the computation time throughout the region. Clearly, the stiff high fidelity simulations that lie above this threshold would be filtered out by the low fidelity model, whereas those that lie below the threshold would not. Ideally, there would be far fewer poor predictions in this region, but these are a function of the assumptions made in the low fidelity model, not the variable fidelity framework itself.

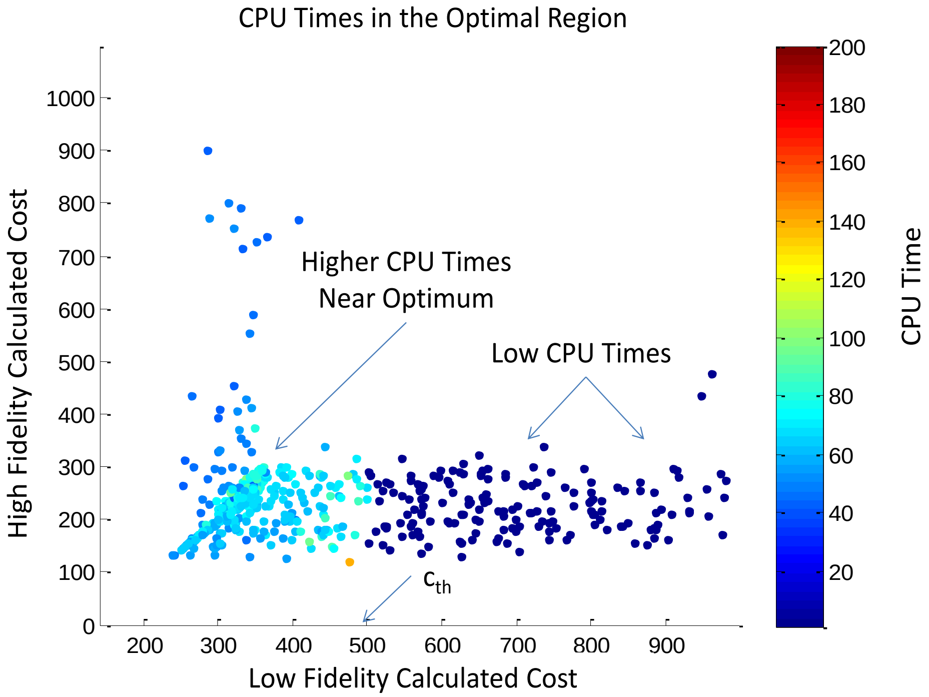


Figure 20: LHS Samples in the Acceptably Good Region

In Figure 20 the plot is scaled to show the acceptably good region of the LHS sample space. In this region we see the highest concentration of high CPU times in the optimal region, with low simulation times for all instances above the cost threshold. All of the figures collectively show that the variable fidelity framework is successful in reducing the CPU times for instances in the design space that lay far from the optimum. This is especially useful in design spaces such as this, where there are discontinuities and the acceptably good region is small (~40% according to this LHS).

5.2 Characterization of Design Spaces and Conservativeness

5.2.1 Correlation of High and Low Fidelity Models

In this thesis, it is suggested that the use of variable fidelity models does not have to be restricted to surrogates or other approximation based methods; rather, it may be possible to use any types of models so long as they are positively correlated. This is actually most important when far from the global optimum and above the performance threshold. In this far region of the design space, it is the job of the low fidelity model to guide the optimizer toward better solutions (of the high fidelity problem) while using few resources. Once the threshold is reached and the high fidelity model is being evaluated, the correlation between the two models is not important.

For the backhoe models, correlation information is gathered from the same LHS samples used in the previous section. The correlation coefficient between the low and high fidelity backhoe models over the entire design space is 0.4544. However, it makes sense to look at the correlation coefficients by region, as shown in Figure 18 (high cost region, acceptably good solutions, and poor predictions). In the high cost region, the correlation coefficient is 0.98, very close to 1. This is exactly as desired, because the low fidelity model is simply guiding the optimizer toward better solutions in the same manner as the high fidelity model would but at much lower cost. For the acceptably good solutions, the correlation coefficient is 0.0125, very close to zero. This makes sense also: at this point, the low fidelity model has reached its accuracy limits and any deviation from the low-fidelity prediction looks like random noise. However, it does not much matter here that the models are poorly correlated, because for most appropriate threshold selections the high fidelity model will be used. As long as the low fidelity model provides a prediction that is below the threshold, it no longer matters how closely the two models are correlated. Similarly, in the region of poor predictions, the correlation coefficient is -0.195. Even in this region, the high fidelity model will have to be evaluated some of the time, depending on the threshold. Because this is the area where omitting some physical phenomena affects the overall performance of the backhoe, it is expected that the correlation between the two models would be close to zero in this region.

5.2.2 Design Space Characterization- Histograms

To get a better sense of what the design space looks like, histograms are provided for both the high and low fidelity design spaces. The high fidelity and low fidelity histograms use the same LHS samples from the previous sub-sections, and are shown in Figure 21 and Figure 22, respectively. In both figures, the bins are costs in dollars for the trenching project, and the frequencies for each bin are given. The total number of samples is 1000 (per the LHS samples). The bins selected include all of the threshold costs to be used in the optimization (Section 5.4). While these histograms give some sense of what regions one is likely to end up in and how often the high fidelity model would be evaluated if the design space were sampled randomly, they do not give much indication of the probability of ending up in particular region or the frequency of high fidelity evaluations when a genetic optimizer is being used. To get a better sense of how 'difficult' the design problem is, it would be necessary to gain an understanding of the size of the region of attraction around the global optimum. However, this parameter is not a useful (or measurable) metric when using a stochastic optimization algorithm.

Bin	*Frequency*
125	1
250	362
350	194
500	12
1000	12
10000	6
100000	2
1000000	0
2000000	90
3000000	286
4000000	28
5000000	7
More	0

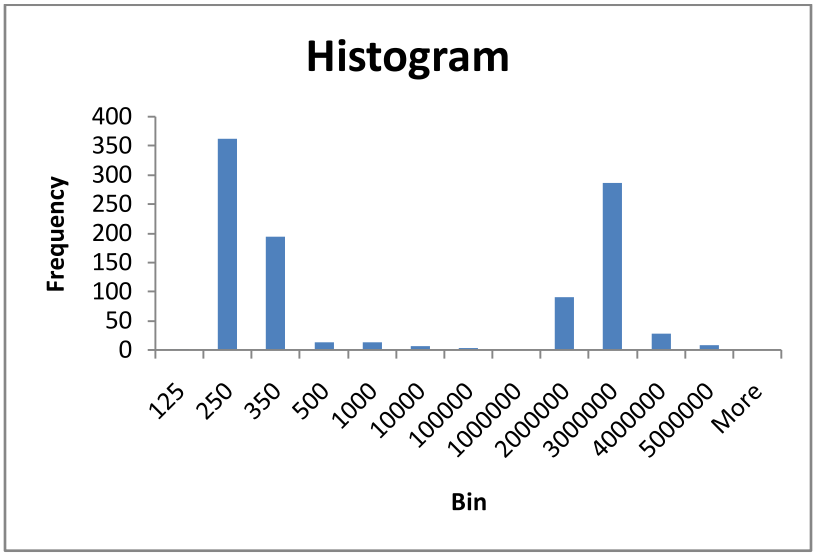


Figure 21: High Fidelity Design Space Histogram

Bin	*Frequency*
125	0
250	3
350	303
500	177
1000	263
10000	112
100000	0
1000000	0
2000000	0
3000000	127
4000000	15
5000000	0
More	0

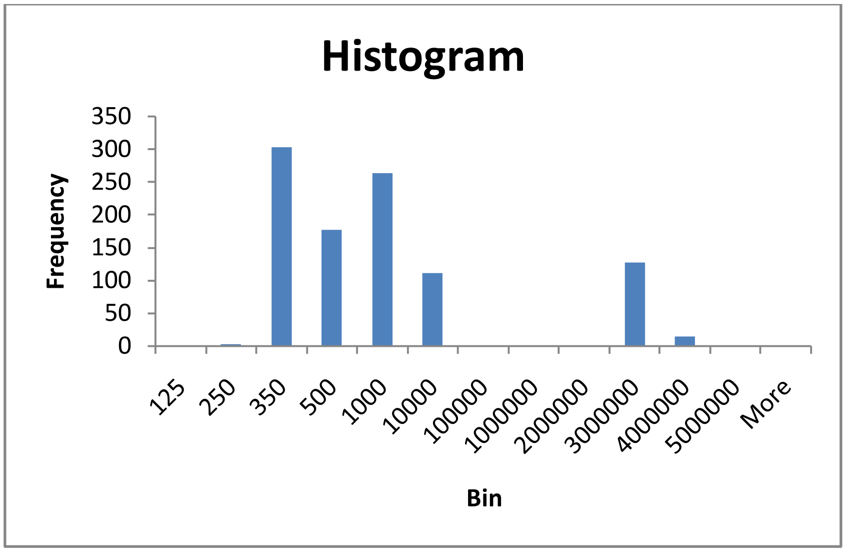


Figure 22: Low Fidelity Design Space Histogram

5.2.3 Conservativeness of the Low Fidelity Model

For the backhoe models, the low fidelity model is conservative 69.3% of the time, based on the LHS samples. In the mathematical validation of the method (Section 5.4.2), it is suggested that having the low fidelity model always be conservative will prevent the creation of artificial peaks in the design space, or stopping prematurely in the neighborhood of the threshold. Because a GA is used in this example, reaching the global optimum is still possible, but may not always happen reliably.

5.3 Sensitivity Analysis

To gain a better understanding of how each of the switching times impacts the attributes which determine cost and how these might differ between the low and high fidelity models, a Method of Morris (MOM) [33] sensitivity analysis is performed on both the low and high fidelity models using ModelCenter [30] from Phoenix Integration.

The MoM examines the changes in an output based on experimental plans composed of randomized designs for the input factors. Only one input factor is changed at a time allowing change in output to be unambiguously attributed to change in that input. This is done by estimating the mean and variance of elementary effects attributed to input factors. One sample of the elementary effect for the i^{th} input factor is defined as:

$$d_i(x) = \frac{\left[y(x_1, \cdots, x_{i-1}, x_i + \Delta, x_{i+1}, \cdots, x_k) - y(x) \right]}{\Delta}$$

where x is a k-dimensional vector of model inputs, and Δ is often chosen as:

$$\Delta = \frac{p}{\left(2 \left(p - 1 \right) \right)}$$

where p is the number of grid levels in the region of experimentation. The finite distribution of elementary effects associated with the i^{th} input factor, obtained by randomly sampling different x's, is denoted by F_i. We take the mean, μ, and the standard deviation, σ, of F_i to be informative sensitivity measures. Input factors with large μ are likely to have an overall important influence on the output, while input factors with large σ may have interaction with other factors or may have non-linear effects.

Twenty five random observations for each of four input variables were taken, requiring 125 total function evaluations. The four design variables are numbered accordingly:

1. Swing Switching Time
2. Boom Switching Time
3. Arm Switching Time
4. Bucket Switching Time

Figure 23 shows the high fidelity MOM experiment with respect to the total project cost as an output. The points that lie within the dashed 'V' have an effective mean for the particular output that is not statistically significant from aero. The arm switching time (#3), for example, has a small mean and a small standard deviation, so its impact on the overall cost is minimal. The boom switching time (#2) has the largest mean, followed by the swing switching time (#1) and the bucket switching time (#4), so these variables are likely to dominate the cost response. This means that increasing the boom switching time will cause the greatest increase in cost. This happens because the boom has to lift early (before the swing and the bucket) in order for the manipulator to successfully get out of the trench and land in the dump zone. If the soil does not land in the dump zone, the dig is not productive, and the cost of completing the trench becomes essentially infinite.

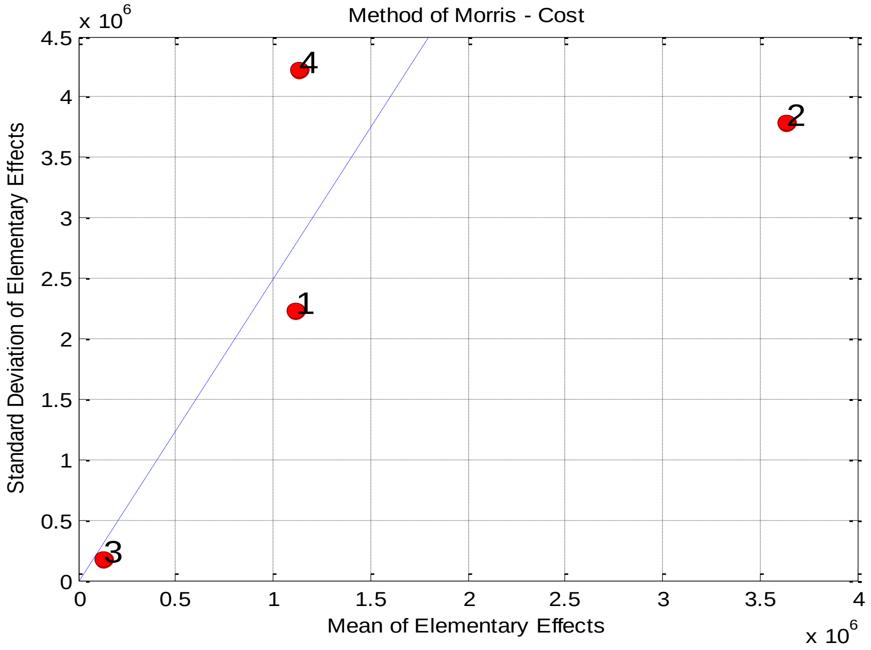


Figure 23: High Fidelity MOM Sensitivity Analysis: Cost ($)

The variables having a high standard deviation are again the boom and swing, but this time the bucket is included as well. A high standard deviation implies a likelihood of interaction effects. This is logical because the sequencing of the boom, arm, and bucket determines where the bucket dumps out soil, and whether or not the bucket actually makes it to the dump zone. That is, if the swing switches before the boom lifts, the manipulator hits the side of the trench, and may not get out in time to reach the dump zone before the soil is released from the bucket.

The low fidelity counterpart MOM for cost is shown in Figure 24. This figure is somewhat different from the high fidelity version (Figure 23). In this case, the arm still has very little effect on the overall cost, but the bucket has a negative mean, and the boom and swing have essentially switched roles. These discrepancies will be accounted for as the attributes that make up the cost of the project are examined in the next figures.

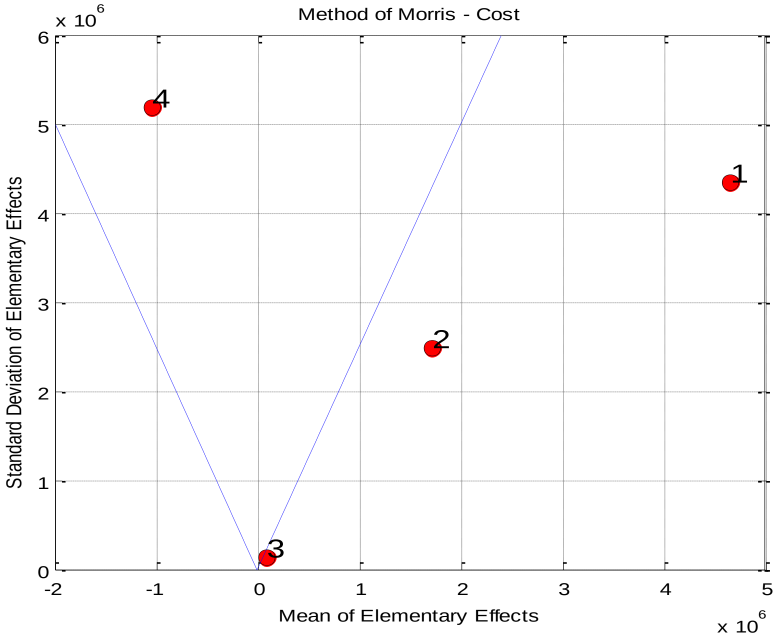


Figure 24: Low Fidelity MOM Sensitivity Analysis: Cost ($)

Figure 25 and Figure 26 show the high and low fidelity MOM experiments with respect to productivity. In this case, productivity is strictly a measurement of how much soil is deposited in the dump zone and is not a rate (i.e. it is not time dependent). In Figure 25 it can be seen that the boom and swing switching times have negative means. This implies that increasing these input times tends to decrease the overall productivity. On the other hand, increasing the bucket switching time tends to lead to increased productivity. This is because the bucket is only apt to dump soil in the dump zone after the boom and arm have completed their motions, so making the bucket dump later is a good way to ensure that the sequencing is correct. The arm has little to no bearing on the success in this case.

The primary difference between the low and high fidelity sensitivity analyses with respect to productivity is the boom switching time (#2). In the high fidelity model, the boom switching time has a much higher mean and standard deviation than in the low fidelity model. This difference is due to the lack of a trench model in the low fidelity model. By taking out the trench walls and only giving a penalty for violation of these walls (which is not considered in the productivity metric), trajectories that would fail to reach the dump zone in the high fidelity model are successful in the low fidelity model. Therefore, the boom does not need to lift before the swing occurs in the low fidelity model, whereas that trajectory would result in collision with the trench walls in the high fidelity model.

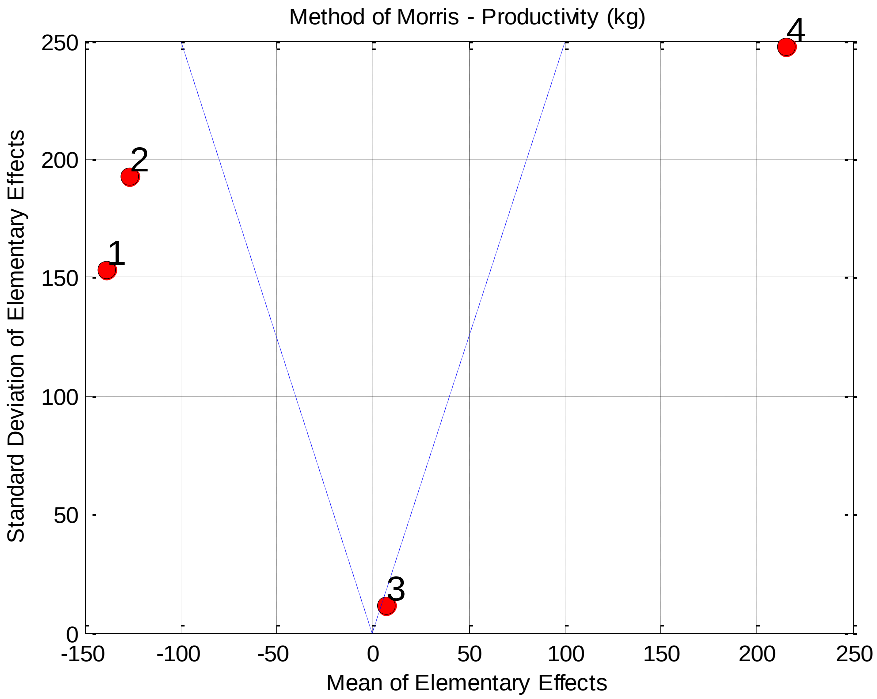


Figure 25: High Fidelity MOM Sensitivity Analysis: Productivity

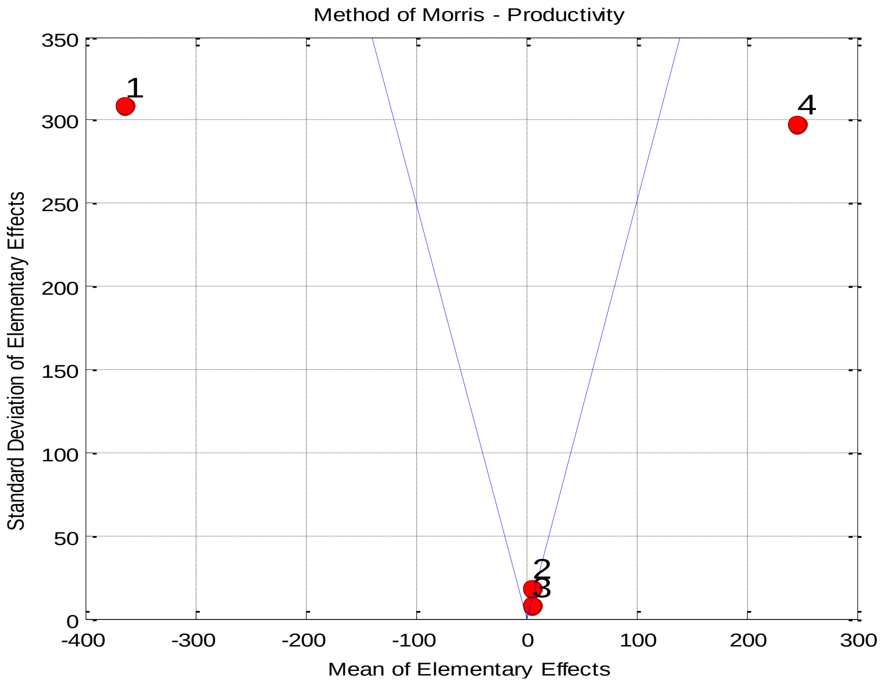


Figure 26: Low Fidelity MOM Sensitivity Analysis: Productivity

Figure 27 and Figure 28 show the high and low fidelity MOM experiments with respect to time. In this case, time is strictly the measurement of when the simulation terminates, (i.e. when the bucket empties) and not the overall project time. In the high fidelity plot, we see that all of the design variables seem to have some effect on the overall outcome. This is somewhat counter-intuitive, since the emptying of the bucket (when the bucket switches) uniquely determines the end of the simulation. The fact that all of the variables seem to have an effect on this is likely due to the hydraulic response time and the bucket angle. The hydraulic response time will be much slower if multiple actuators are trying to do work at once—the pump can only provide so much pressure and flow rate to the system. In the cases where a lot of switches occur nearly simultaneously, it will take longer for the bucket to finish its motion and therefore take longer to deposit all of the soil. Additionally, the angle of the bucket is also affected by the angle of the boom and arm joints. If the boom is lifted, the bucket joint itself does not have to go as far to dump out all of the soil, so this too affects the overall dig cycle time.

In the low fidelity plot, the swing and arm switching times have little effect on the cycle time. This difference between the low and high fidelity models is likely due to the lack of hydraulic dynamics in the low fidelity model—actuations occurring simultaneously do not matter. Also in this plot, we see that the boom has a much greater impact on the cycle time, and this is likely to do the fact that the angle of the bucket is affected by the boom angle. However, in both the low and the high fidelity plots, the bucket switching time is the primary contributor to the cycle time, since it is the emptying of the soil from the bucket that determines when the cycle terminates.

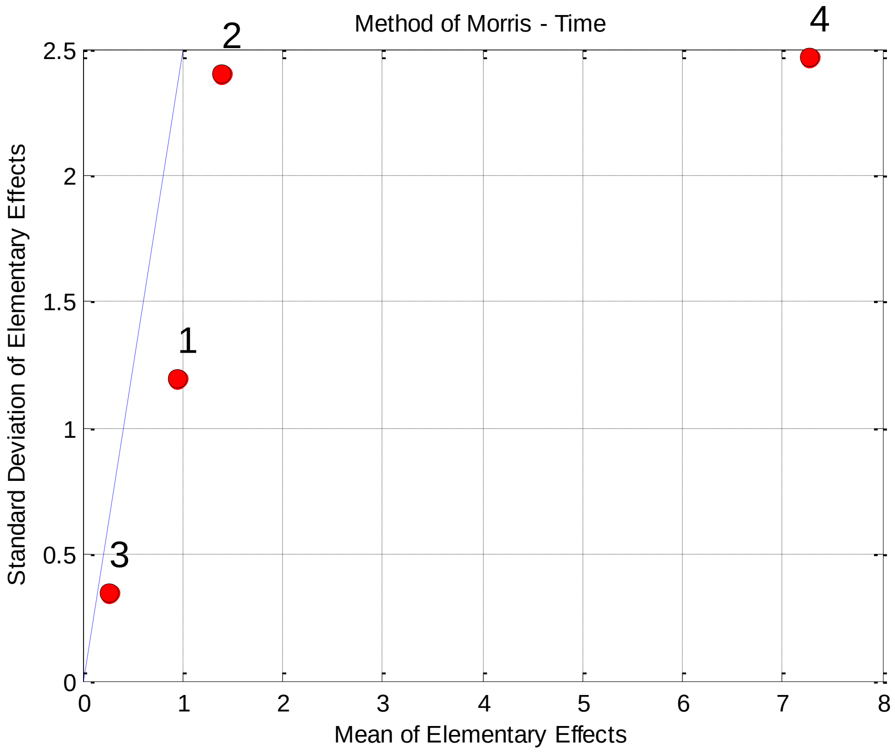


Figure 27: High Fidelity MOM Sensitivity Analysis: Time

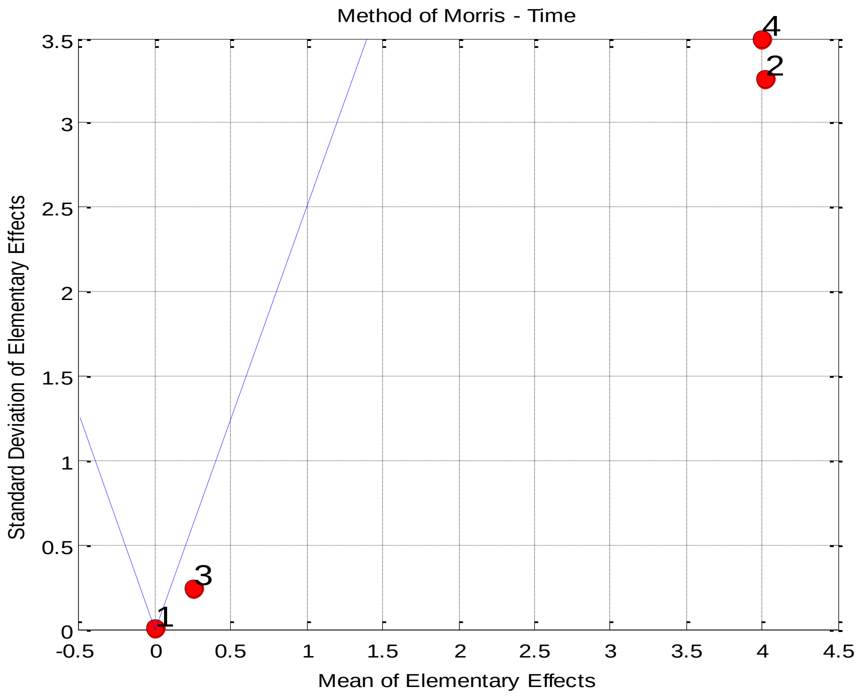


Figure 28: Low Fidelity MOM Sensitivity Analysis: Time

The final MOM experiment is performed only on the high fidelity model with respect to fuel consumption. There is no low fidelity comparison for this output because the low fidelity model does not contain a hydraulic system or any other means of quantifying the energy consumed over the course of the cycle. This final MOM plot is given in Figure 29. The bucket, swing, and boom all have a large effect on the fuel consumption. Since the arm moves very little during the course of the dig cycle, it does not have a big impact on the fuel consumed. The reason that the mean of the swing is negative is due to the fact that the boom must lift before the swing occurs to prevent energy losses while colliding with the trench wall.

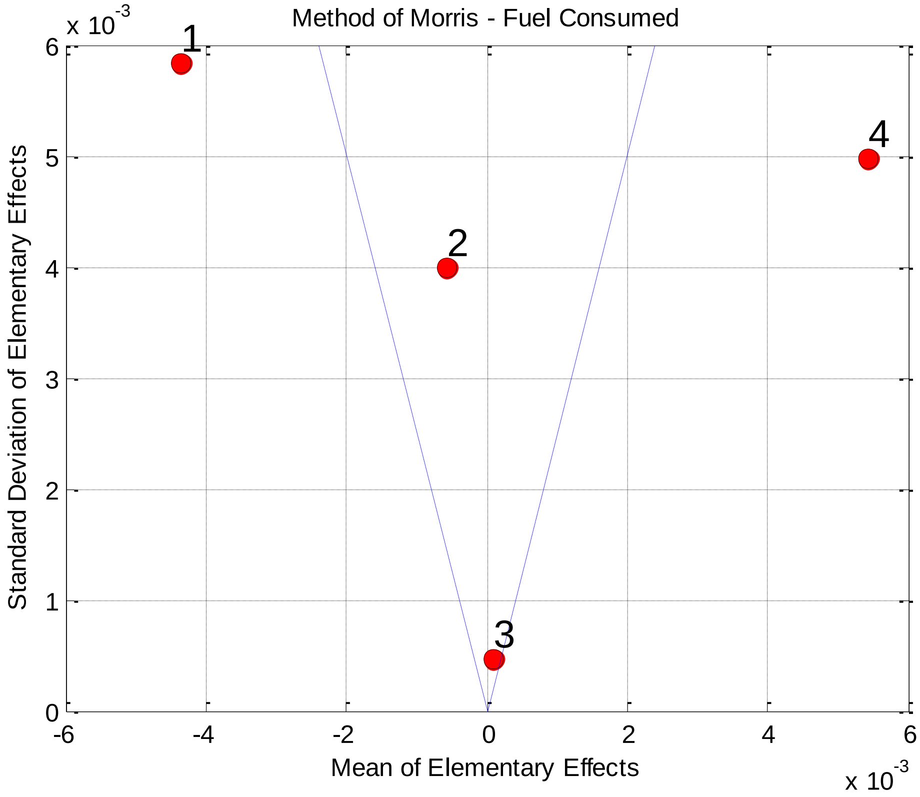


Figure 29: High Fidelity MOM Sensitivity Analysis: Fuel Consumption (kg)

5.4 Optimization Results

5.4.1 Computational Savings and Robustness

In Table 1, the results for the optimization problem with a $350 threshold are presented. We see that the optimizer reached a near globally optimal solution for five of the eight trials. The three that failed converged prematurely around the switching point. This is due to the low fidelity model's inability to provide a consistently conservative estimate of the cost, so artificial peaks are created in the design space.

Table 1: Genetic Algorithm Optimization Results for $350 Threshold

GA Optimization Results, c_th=$350				
Final Cost	Total Calls	Hi Calls	Est. Savings (Hrs)	% Savings
Successful Trials				
$110.00	420	272	2.13	35.24%
$110.09	560	385	2.46	31.25%
$110.78	580	465	1.71	19.83%
$110.86	700	606	1.46	13.43%
$111.24	800	619	2.67	22.62%
Unsuccessful Trials				
$350.04	340	46	2.24	86.47%
$350.00	340	38	2.36	88.82%
$350.04	380	31	3.26	91.84%

A genetic algorithm is run five times with random seeding populations with a threshold of $500. Three of these trials converge to solutions in the optimal range, while two converge prematurely in the neighborhood of the performance threshold. The results are depicted in Table 2. Also depicted in the table are the total number of function calls, and the number of high fidelity function calls. Although the high fidelity model is called more often than not in the successful trials, we still see a significant time savings. This is because the high fidelity model averages ~55 seconds, and the low fidelity model only takes 1-2 seconds. In the table, the estimated time savings is based on average high fidelity CPU time for the trial, and the number of times the high fidelity model was not called. The percent savings is the expected savings over the total time it would have taken to call the high fidelity model at all of the points called in the design space. This percentage savings ranges from 14-26%.

Table 2: Genetic Algorithm Optimization Results for $500 Threshold

GA Optimization Results, *c_th*=$500				
Final Cost	Total Calls	Hi Calls	Est. Savings (Hrs)	% Savings
Successful Trials				
$109.25	780	671	1.67	13.97%
$115.92	500	398	1.62	20.40%
$110.00	560	412	2.1	26.43%
Unsuccessful Trials				
$500.03	760	91	7.65	88.03%
$500.00	340	59	2.94	82.65%

The two unsuccessful trials failed to converge to solutions in the neighborhood of the optimum; however, the computational savings in these cases is much higher: 80-90%. Because the optimizer never really got into the region of acceptable solutions, very few calls were made to the high fidelity model. These unsuccessful trials could be avoided in the future by tuning the genetic algorithm to converge more slowly and deliberately; however, if solving a problem such as this with a genetic algorithm typically requires multiple runs and some tuning, it is still more efficient for the unsuccessful trials to take less time.

In Table 3 the optimization results for the $1000 threshold are depicted. In this case, the region over which the high fidelity model is evaluated seems to be too large because the time and percentage savings is much smaller than any of the previous cases (3-14% for the successful trials). Three out of the five attempts converged to solutions in the neighborhood of the global optimum, but they are not better than the results achieved by the other trials with smaller thresholds. In addition, since the high fidelity function is called more often than not, and the low fidelity function is always called, this threshold is close to borderline in terms of being beneficial when compared to simply solving the high fidelity optimization problem without using the low fidelity model at all.

Table 3: Genetic Algorithm Optimization Results for $1000 Threshold

GA Optimization Results, c_th=$1000				
Final Cost	Total Calls	Hi Calls	Est. Savings (Hrs)	% Savings
Successful Trials				
$111.65	360	347	0.24	3.61%
$111.23	660	565	1.69	14.39%
$111.85	420	377	1.08	10.24%
Unsuccessful Trials				
$1,000.14	580	215	3.33	62.93%
$1,000.09	500	225	2.62	55.00%

In Table 4 the optimization results for the $250 threshold are presented. This threshold is very close to the optimum for the low fidelity model alone, where the lowest observed cost is $228. However, the low fidelity minimum is not very close to the global optimum, which actually occurs at around $334 for the best optimum found for the $500 case. This choice of threshold actually violates the sufficient conditions. While the percentage savings still seems to be higher than in the previous two cases, the overall hourly savings is much higher only because the optimization runs took a lot longer in general. A lot more function calls are necessary in this case, and the final cost results are not as good as those in the previous cases, even though only one of six trials converged prematurely around the switching point.

Table 4: Genetic Algorithm Optimization Results for $250 Threshold

GA Optimization Results, c_th=$250				
Final Cost	Total Calls	Hi Calls	Est. Savings (Hrs)	% Savings
Successful Trials				
$113.96	1000	523	6.91	47.70%
$117.63	840	480	5.21	42.86%
$119.15	1340	1041	4.59	22.31%
$124.87	540	263	4.31	51.30%
$142.34	880	386	7.31	56.14%
Unsuccessful Trials				
$250.04	640	29	7.09	95.47%

The failure to achieve solutions closer to the global optimum is due to misalignment of the nadirs for the low and high fidelity models and the subsequent violation of the sufficient condition by picking a threshold where the high fidelity model would not be evaluated at the global minimum. By making the threshold so low, some of the good high fidelity solutions are never run, so the best solutions are actually filtered out by making the tolerance so tight. This behavior is shown in Figure 30. And, even though intuition would have it that the optimizations would take less time with so many low fidelity function calls, the optimization took more time because it required so many more function calls to converge successfully.

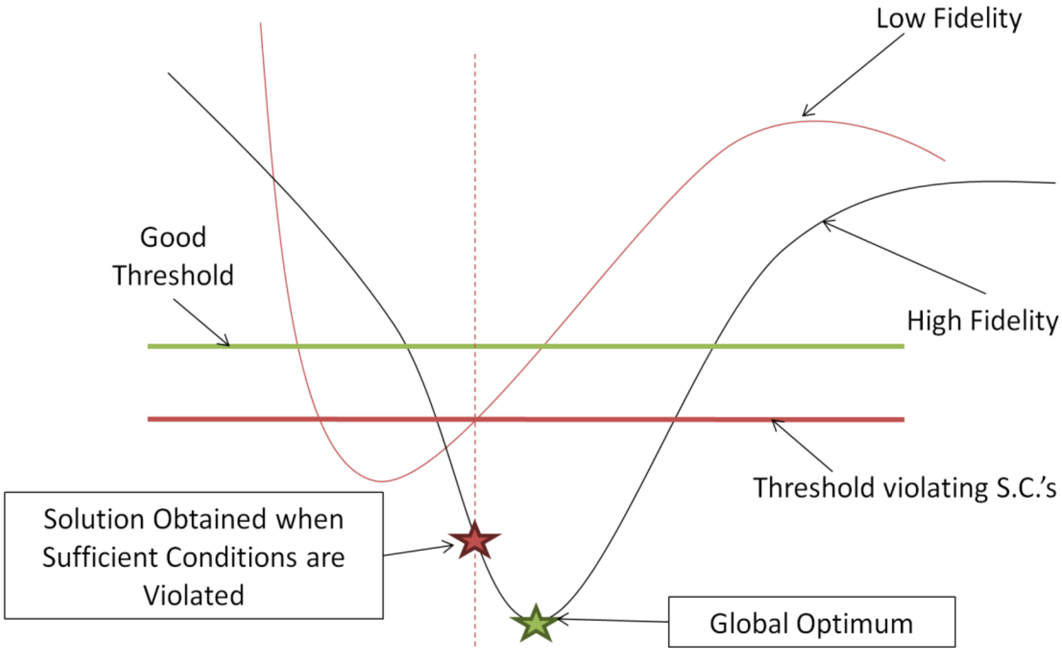

Low Fidelity
Good Threshold
High Fidelity
Threshold violating S.C.'s
Solution Obtained when Sufficient Conditions are Violated
Global Optimum

Figure 30: Violation of Sufficient Conditions-- Threshold set too low

5.4.2 The Impact of Conservativeness

To see how using a more conservative low fidelity model might affect the robustness of the optimization, additional GA runs are made using a $500 threshold. To make the low fidelity model always conservative, a large constant (5e6) is added to all low fidelity cost estimates and to the cost threshold. These results are compared with results obtained using the original low-fidelity model using the Fisher Test [36]. The test relies on pair-wise comparisons to determine a criterion for statistical significance. The pair-wise comparisons are shown in Table 5. In the table, *f* is the resulting cost output using the variable fidelity framework and the original low fidelity model for a particular run, and *f_cons* is run using the variable fidelity framework and the conservative low fidelity model. *Z* is the difference between *f* and *f_cons*, where smaller is better, since we are attempting to minimize cost. Then, *phi* is selected as 0 or 1: 0 if the conservative version obtains a better result and 1 if the regular version obtains a better result.

It can be seen in the table that 5 of the 14 runs using the regular framework converge prematurely in the neighborhood of the threshold cost ($500). While this phenomenon does not occur using the conservative low fidelity model, 4 of the 14 runs fail to get into the neighborhood of the global optimum, since effectively by shifting the low fidelity design surface, the global minimum is now in a narrower nadir. Since the conservative version comes up with a better solution exactly 50% of the time, it can be concluded that for this particular example, the conservativeness of the low fidelity model does not have a statistically significant bearing on the robustness of the optimization process.

Table 5: Fisher Test Results

Run	f ($)	f_cons ($)	Z=f_cons-f	phi
1	109.25	109.94	0.69	1
2	110	109.99	-0.01	0
3	115.92	9.54E+05	953884.08	1
4	500.03	114.16	-385.87	0
5	500	112.11	-387.89	0
6	111.49	111.02	-0.47	0
7	500.5	9.45E+05	944499.50	1
8	110.41	117.05	6.64	1
9	109.96	110.49	0.53	1
10	115.91	114.54	-1.37	0
11	500	110.04	-389.96	0
12	116.67	9.71E+05	970883.33	1
13	500.55	109.75	-390.80	0
14	110.74	1.03E+06	1030889.26	1

5.4.3 Interpretation of the Solution to Trajectory Optimization Problem

The best results from the optimization using the various thresholds are combined and presented in Table 6. The results are fairly similar for the five cases presented, at least in terms of the most significant variables (shown previously by the sensitivity analyses). The switches happen in sequence, i.e., not simultaneously, but there is some overlap in movement due to the response time of the hydraulics. The bucket switch times are very similar, which makes sense since the cycle time is determined almost exclusively by the time at which the bucket releases all of the soil. Releasing the bucket any earlier would result in missing the dump zone, while waiting longer would extend the cycle time. The arm switch time is not very important, and so the results there are less similar. This is because the arm does not actually have to move to perform the trajectory successfully, so the time at which it does so has very little bearing on the overall cost of the trenching project. The swing switching times are mostly similar, always occurring after the boom lifts. The boom times vary some but they are all close to zero, which is desirable since this is the first even that has to transpire for the manipulator to clear the trench walls.

Table 6: Genetic Algorithm Solutions to Trajectory Optimization Problem

GA Optimization Solutions				
Cost	*Swing Switch Time (s)*	*Boom Switch Time (s)*	*Arm Switch Time (s)*	*Bucket Switch Time (s)*
$109.25	0.4577	0.0103	0.9824	1.5847
$110.00	0.4526	0.0194	0.3291	1.6073
$110.00	0.4455	0.0150	0.4232	1.5990
$110.09	0.4710	0.0217	0.2813	1.6204
$110.78	0.6430	0.0325	1.0606	1.6460

Looking more closely at the best solution found, Figure 31 shows the valve openings and the control input to the valves for this trajectory with respect to time (swing, boom, arm and bucket, from top to bottom). The valve openings vary between -1 and 1, where an absolute value of 1 indicates fully open in one direction or the other, and zero is completely closed. Here it can be seen that there is a bit of overlap between the valve opening events due to the hydraulic response time and the nature of the feedback controller, but most of the actions occur in sequence so as not to strain the hydraulic system past its capabilities, slowing down the overall motion.

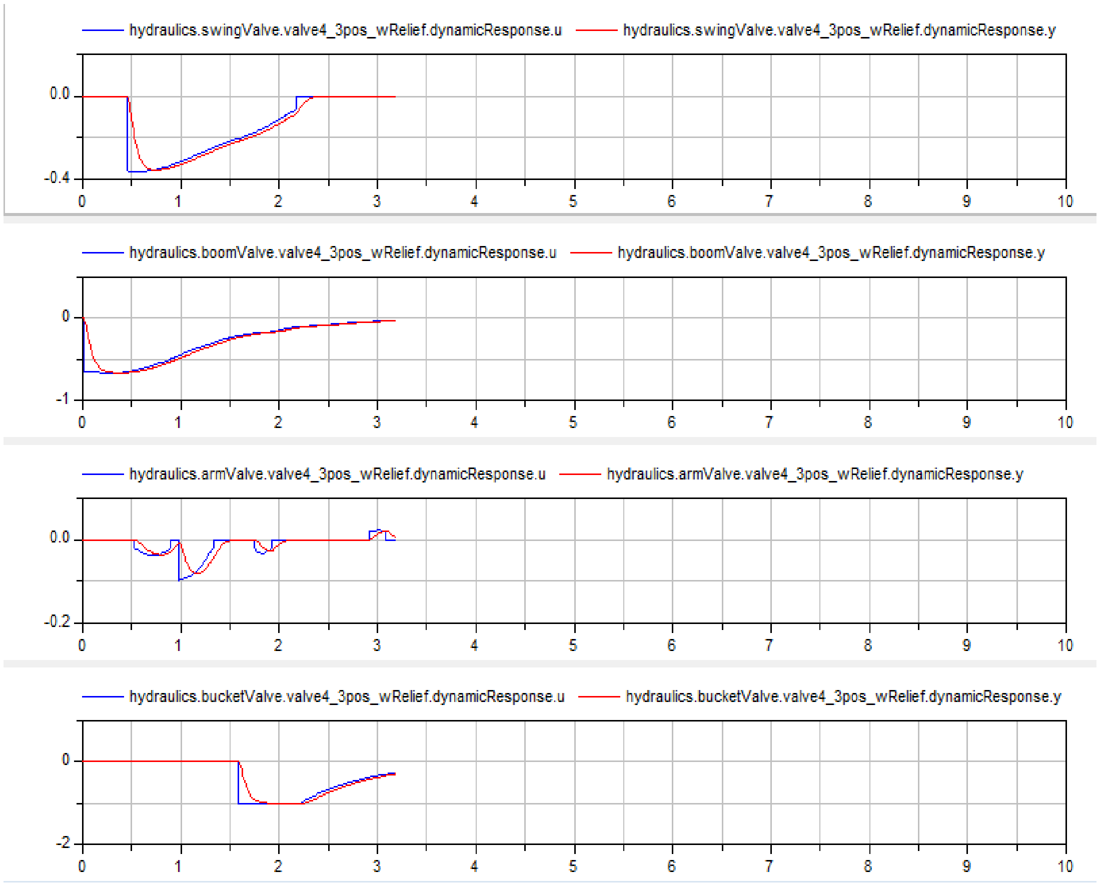


Figure 31: Valve Openings (Swing, Boom, Arm, Bucket) for the Optimal Trajectory

CHAPTER 6

SUMMARY AND FUTURE WORK

6.1 Review and Evaluations

Modeling, simulation, and optimization play vital roles throughout the engineering design process; however, in many design disciplines the cost of simulation is high, and designers are faced with a tradeoff between the number of alternatives that can be evaluated and the accuracy with which they are evaluated. This begs the following research question:

> *Is it possible to achieve both broad exploration and high accuracy by improving the way in which we use the available models?*

It is hypothesized that using models of various levels of fidelity in the problem formulation stage of the design problem can help designers to achieve both broad exploration and high accuracy, even with limited computational resources. This thesis provides a preliminary framework for incorporating two positively correlated models of varying levels of fidelity into an optimization process by switching between the models based on a user-defined performance threshold. The primary research contributions of this work are:

1) A simplistic framework that serves as a starting point for handling a collection of models of varying levels of fidelity in the formulation of the objective function to be optimized
2) A mathematical validation of this framework and sufficient conditions for preservation of the global optimum

3) A characterization of this framework in terms of improved performance during the optimization process

The framework is applied a trajectory optimization problem for a hydraulic backhoe. The design space is examined using a Latin Hypercube Sampling technique, and it is shown that the low fidelity model helps to reduce stiffness in the design space and channels the computational resources toward the optimum. The trajectory optimization problem is then solved using a genetic algorithm under the proposed framework. For each trial, computational time is saved compared to the high fidelity case, but the overall savings and overall solution qualities for each trial are dependent on the choice of the performance threshold. Under this framework, there is probably an optimal choice of performance threshold for a particular problem and optimizer, depending on the strength of the correlation of the high and low fidelity models, the dimensionality of the design space, and the fraction of acceptably good solutions.

6.2 Limitations

This thesis provides only a first step toward the overall goal of managing any number of models of varying levels of fidelity in an algorithmic fashion to efficiently solve otherwise costly optimization problems. In particular, the framework provided is currently limited to only two models—one and high and one low fidelity and the selection of appropriate models is left to designer intuition. In addition, switching between the two is based on a user-defined threshold. Selection of this threshold is not automated or axiomatic. While efficiency is improved by at least some percentage in every trial of the optimization problem, statistical significance has not yet been shown, but this may be done in the very near future. Finally, while the proposed framework does not dictate the type of optimizer used or the type of low fidelity moved, the framework could be expanded to include multiple types of optimizers or hybrid approaches in conjunction with a more robust model selection process to improve both efficiency and robustness of the overall process.

6.3 Future Work

In future work, the selection of the performance threshold will be investigated for its effects on computational efficiency and convergence under various optimization schemes. Ideally, switching between models might not even be based purely on a threshold. Rather, switching between models could be algorithmic, based on the available models, the nature of the design space, the type of optimizer(s), and the proximity to the optimum. As a next step, it may be possible to optimize the selection of the performance threshold and automate its selection based on minimal sampling of the design space, or dynamically select the performance threshold based on the cost and value of gathering information at a particular data point. In addition to refining the heuristics for selecting the performance threshold, this framework could be applied to a variety of problems, and tested with a variety of optimization schemes. The framework could be expanded to accommodate any number of models of varying levels of fidelity, as well as combinations of optimization schemes of varying fidelity, such as a GA with a high tolerance followed by a second GA with a much tighter tolerance, or other hybrid methods. Another method worth investigating might be a more spiral approach, where the GA is run on a low fidelity model and promising solutions are used as starting points for a high fidelity model. Ideally, future refinements of this framework will allow for dynamic switching among any number of models based on a cost-benefit tradeoff.

APPENDIX

Relevant source code

Backhoe.m

```
function cost=backhoe(x)
%
% Programmed by Roxanne Moore
%
% The appropriate input is a 1x4 vector of switching times, where the
% first entry is the swing joint switching time, the second is the boom
% joint switching time, the third is the arm joint switching time, and
% the fourth is the bucket joint switching time.  These times uniquely
% define the backhoe trajectory for the dump portion of a dig cycle.
%
% This function calls the appropriate backhoe simulations and returns
% the appropriate cost which serves as the objective function to be
% minimized by the optimizer.  It first calls a low fidelity backhoe
% model, and checks whether or not the cost estimated by this low
% fidelity model achieves a user defined performance threshold.  If it
% does not, the low fidelity cost is returned to the optimizer.  If the
% threshold is met, then a high fidelity backhoe model is run with the
% same set of inputs and the high fidelity cost is returned to the
% optimizer.  If the high fidelity model is stiff and times out, then
% the low fidelity cost is returned.
%%
tic
global INPUT

% Get input script into a usable, editable format

[status, result]=system('"c:\Program Files\Dymola\Mfiles\alist"-b
dsin_good_Lo.txt input_Lo.mat');

% Load the editable input file

load input_Lo.mat

% Store the input variables

INPUT=[INPUT; x];

% Set the design variables based on their indices in the .mat array:

% swingSwitch is 952
initialValue(952,2) = x(1);
% boomSwitch is 953
initialValue(953,2) = x(2);
% armSwitch is 954
```

```
initialValue(954,2) = x(3);
% bucketSwitch is 955
initialValue(955,2) = x(4);

% Declare the remaining global variables to track all relevant
statistics:

global HI_COUNT
global LoSimTime
global HiSimTime
global Cost_History
global Cost_History_Hi
global Cost_History_Lo
global Prod_History_Hi
global Prod_History_Lo
global Fuel_History_Hi
global Time_History_Hi
global Time_History_Lo

%%
% Save the new input.mat file and put it back into the text format used
% by the simulation software

save -v4 input_modified_Lo.mat Aclass
save -v4 input_modified_Lo.mat experiment -append
save -v4 input_modified_Lo.mat method -append
save -v4 input_modified_Lo.mat settings  -append
save -v4 input_modified_Lo.mat initialName  -append
save -v4 input_modified_Lo.mat initialValue -append
save -v4 input_modified_Lo.mat initialDescription  -append
[status, result]=system('"c:\Program Files\Dymola\Mfiles\alist" -a
input_modified_Lo.mat dsin_Lo.txt');

%%
% Run the Low Fidelity Simulation

[status, result]=system('dymosimLo dsin_Lo.txt');



%%
% Load the results, but first make sure the results file is there (This
% is just for robustness- sometimes the timeout script for the high
% fidelity model can delete the low fidelity results file if the low
% fidelity model finishes while the script is still closing from the
% previous function call.  This causes the optimizer to halt.)

    try
        [s,n]=tload;
    catch a
        status=1;
    end

    % If a problem is detected with the results file, re-run the low
    % fidelity simulation until there is no longer a problem.
    while status~=0
        [status, result]=system('dymosimLo dsin_Lo.txt');
        try
            [s,n]=tload;
        catch a
            status=1;
        end
    end

    % Get the relevant output values from the results matrix

        D=size(s);
        costLo=s(D(1),295);
        TimeLo=s(D(1),1);
        ProdLo=s(D(1),1728);
        Cost_History_Lo=[Cost_History_Lo costLo];
        Time_History_Lo=[Time_History_Lo TimeLo];
        Prod_History_Lo=[Prod_History_Lo ProdLo];
        toc
        LoSimTime=[LoSimTime toc];

%%
% Set the performance threshold here: If it is not met, return the low
% fidelity cost value.

if costLo>1000
    cost=costLo;

   % If the performance threshold is met, continue on.

else
    tic
    %%
    % Load the high fidelity input file into an editable form

    [status, result]=system('"c:\Program Files\Dymola\Mfiles\alist" -b
dsin_good_Hi.txt input_Hi.mat');
    load input_Hi.mat

    % Set the design variables in the input array
    %(from the same input vector used for the low fidelity model)

    % swingSwitch is 12075
    initialValue(12075,2) = x(1);
    % boomSwitch is 12076
    initialValue(12076,2) = x(2);
    % armSwitch is 12077
    initialValue(12077,2) = x(3);
    % bucketSwitch is 12078
    initialValue(12078,2) = x(4);

    % Save the new array into a text file used by the high fidelity
    % simulation

    save -v4 input_modified_Hi.mat Aclass
    save -v4 input_modified_Hi.mat experiment -append
    save -v4 input_modified_Hi.mat method -append
    save -v4 input_modified_Hi.mat settings  -append
    save -v4 input_modified_Hi.mat initialName  -append
    save -v4 input_modified_Hi.mat initialValue -append
    save -v4 input_modified_Hi.mat initialDescription  -append
    [status, result]=system('"c:\Program Files\Dymola\Mfiles\alist" -a
input_modified_Hi.mat dsin_Hi.txt');

    % Run the scheduling script to set the priority and the timeout

    [status, result]=system('start /b sched.exe');

    % Run the high fidelity simulation

    [status, result]=system('dymosimHi dsin_Hi.txt');

    % Make sure the results file exists- it will not if the simulation
    % times out.
    try
        [q,r]=tload;
    catch a
        status=1;
    end

    % If the results do not exist, return the low fidelity cost.

    if status~=0
        toc
        HiSimeTime=[HiSimTime toc];
        cost=costLo;
    else

    % Extract the relevant data from the high fidelity results.

    D=size(q);
    costHi=q(D(1),295);
    TimeHi=q(D(1),1);
    FuelHi=q(D(1),308);
    ProdHi=q(D(1),14859);
    HI_COUNT= HI_COUNT+1;
    cost=costHi;
    Cost_History_Hi=[Cost_History_Hi costHi];
    Time_History_Hi=[Time_History_Hi TimeHi];
    Fuel_History_Hi=[Fuel_History_Hi FuelHi];
    Prod_History_Hi=[Prod_History_Hi ProdHi];
    toc
    HiSimTime=[HiSimTime toc];
    end
end

Cost_History=[Cost_History cost];
```

GA_Optimize.m

```
%%
% Programmed by Roxanne Moore
%
% This script calls the genetic algorithm optimizer for the backhoe
% model (backhoe.m).  The global variables are tracked and the
% optimization parameters can be set here.

 clear

 format long e


global INPUT
global HI_COUNT
global LoSimTime
global HiSimTime
global Cost_History
global Cost_History_Hi
global Cost_History_Lo
global Prod_History_Hi
global Prod_History_Lo
global Fuel_History_Hi
global Time_History_Hi
global Time_History_Lo
global Iter_Result
global Hi_Result
global Cost_Result
global Input_Result
global Hi_Sim_Total
global Lo_Sim_Total
global Sim_Time_Total

% Keep track of some totals for relevant output variables and
% simulation times.

Iter_Result=[];
Hi_Result=[];
Cost_Result=[];
Input_Result=[];
Hi_Sim_Total=[];
Lo_Sim_Total=[];
Sim_Time_Total=[];

% Set the genetic algorithm options.

options =
gaoptimset('PopulationSize',20,'CrossoverFraction',0.9,'StallGenLimit',
15,'TolFun',1e-
5,'StallTimeLimit',inf,'PlotFcns',{@gaplotbestf,@gaplotbestindiv});

% Loop the optmization with the same input parameters any number of
% times for experimentation purposes.

for i=1:6
    close all
    INPUT=[];
    HI_COUNT=0;
    LoSimTime=[];
    HiSimTime=[];
    Cost_History=[];
    Cost_History_Hi=[];
    Cost_History_Lo=[];
    Prod_History_Hi=[];
    Prod_History_Lo=[];
    Fuel_History_Hi=[];
    Time_History_Hi=[];
    Time_History_Lo=[];

% Run the optimization

[X1,Fval,Exitflag,Output] = ga(@backhoe,4, [], [], [], [],
[0.01,0.01,0.01,0.01], [10,10,10,10], [], options);

% Return other results not accounted for in the backhoe.m file

iter=Output.generations;
f_eval=Output.funccount;
costFinal=Fval;

% Save the results at each iteration, the name shown here reflects the
% threshold (set in backhoe.m) and each one is numbered for the i'th
% iteration.

save(['1000GAOpt',num2str(i)])
saveas(gcf, ['1000GA',num2str(i),'.fig'])

% Save the final results.

Iter_Result=[Iter_Result iter];
Hi_Result=[Hi_Result HI_COUNT];
Cost_Result=[Cost_Result costFinal];
Input_Result=[Input_Result X1'];
Hi_Sim_Total=[Hi_Sim_Total sum(HiSimTime)];
Lo_Sim_Total=[Lo_Sim_Total sum(LoSimTime)];
Sim_Time_Total=[Sim_Time_Total sum(HiSimTime)+sum(LoSimTime)];

end

save('1000GAfinal')
```

REFERENCES

[1] 1996-2008, John Deere Backhoes Brochure, http://www.deere.com/en_US/cfd/construction/deere_const/media/pdf/backhoe/JSeries.pdf January, 2008.

[2] 2004, Dymola Multi-Engineering Modeling and Simulation, http://www.dynasim.se/dymola.htm January, 2008.

[3] 2008, PTC- Pro/Engineer, http://www.ptc.com/products/proengineer/ April, 2008.

[4] Akao, Y., 2004, *Quality Function Deployment: Integrating Customer Requirements into Product Design*, Productivity Press, New York.

[5] Alexandrov, N. M., Lewis, R. M., 2001, "An Overview of First-Order Model Management for Engineering Optimization," *Optimization and Engineering*, **2**, pp. 413-430.

[6] Alexandrov, N. M., Lewis, R. M., Gumbert, C.R., Green, L.L., and Newman, P.A., 1999, "Optimization with Variable-Fidelity Models Applied to Wing Design," NASA, Langley Research Center, Hampton, VA

[7] Alexandrov, N. M., Lewis, R. M., Gumbert, C.R., Green, L.L., and Newman, P.A., 2001, "Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models," *Journal of Aircraft*, **38**(6), pp. 1093-1101.

[8] Baker, M. L., Munson, Michael J., Duchow, Eric, Hoppus, George W., Alston, Katherine Y., 2003, "System Level Optimization in the Integrated Hypersonic Aeromechanics Tool (Ihat)," in *AIAA International Space Planes and Hypersonic Systems and Technologies*, Norfolk, VA.

[9] Bakr, M. H., Bandler, J.W., 2000, "Review of the Space Mapping Approach to Engineering Optimization and Modeling," *Optimization and Engineering*, **1**, pp. 241-276.

[10] Belegundu, A. D., and Chandrupatla, T. R., 1999, *Optimization Concepts and Applications in Engineering*, Prentice-Hall, Upper Saddle River, NJ.

[11] Cellier, F. E., 1991, *Continuous System Modeling*, Springer-Verlag, New York.

[12] Conigliaro, R., Kerzhner, A., Paredis, C., 2009, "Model-Based Optimization of a Hydraulic Backhoe Using Multi-Attribute Utility Theory," in *SAE World Congress*, Detroit, MI.

[13] Gano, S. E., Perez, V. M., Renaud, J. E., Batill, S. M., 2004, "Multilevel Variable Fidelity Optimization of a Morphing Unmanned Aerial Vehicle," in

AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, Palm Springs, CA.

[14] Giannakoglou, K. C., 2002, "Design of Optimal Aerodynamic Shapes Using Stochastic Optimization Methods and Computational Intelligence," *Progress in Aerospace Sciences*, **38**, pp. 43-76.

[15] Giese, H., Levendovszky, T., Vangheluwe, H., 2007, "Summary of the Workshop on Multi-Paradigm Modeling: Concepts and Tools," *Models in Software Engineering*, Springer Berlin, Heidelberg, **4364/2007**, pp. 252-262.

[16] Gurnani, A., Donndelinger, J., Kemper, L., 2005, "Feasibility Assessment in Preliminary Design Using Pareto Sets," in *ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Long Beach, CA.

[17] Gurnani, A., Ferguson, S., Lewis, K., Donndelinger, J., 2006, "A Constraint-Based Approach to Feasibility Assessment in Preliminary Design," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **20**, pp. 351-367.

[18] Hays, R., Singer, M., 1989, *Simulation Fidelity in Training System Design: Bridging the Gap between Reality and Training*, Springer-Verlag, New York.

[19] Hazelrigg, G. A., 1996, *Systems Engineering: An Approach to Information-Based Design*, Prentice-Hall, Upper Saddle River, NJ.

[20] Hazelrigg, G. A., 1998, "A Framework for Decision-Based Engineering Design," *ASME Journal of Mechanical Design*, **120**, pp. 653-658.

[21] Howard, R. A., 1988, "Decision Analysis: Practice and Promise," *Management Science*, **34**(6), pp. 679-695.

[22] Huang, D., Allen, T. T., Notz, W.I., Miller, R. A., 2006, "Sequential Kriging Optimization Using Multiple-Fidelity Evaluations," *Structural and Multidisciplinary Optimization*, **32**(5), pp. 369-382.

[23] Kaufman, M., Balabanov, V., Burgee, S. L., Giunta, A. A., 1996, "Variable-Complexity Response Surface Approximations for Wing Structural Weight in Hsct Design," in *AIAA, Aerospace Sciences Meeting and Exhibit*, Reno, NV.

[24] Keeney, R. L., and Raiffa, H., 1993 (1976), *Decisions with Multiple Objectives*, Cambridge University Press, Cambridge, UK.

[25] Krishna, M., 1999, *Optimal Motion Computation for Hydraulic Robots*, Thesis, Robotics, Carnegie Mellon University, Pittsburgh.

[26] Le Moigne, A., Qin, N., 2004, "Variable-Fidelity Aerodynamic Optimization for Turbulent Flows Using a Discrete Adjoint Formulation," *AIAA Journal*, **42**(7), pp. 1281-1292.

[27] Lewis, K., Chen, W., and Schmidt, L. C., 2006, *Decision Making in Engineering Design*, American Society of Mechanical Engineers, New York.

[28] Lophaven, S. N., Nielsen, H. B., and Sondergaard, J., 2002, "DACE: A Matlab Kriging Toolbox," Technical Report IMM-TR-2002-12, Technical University of Denmark

[29] Malak, R. J., and Paredis, C. J. J., 2008, "Modeling Design Concepts under Risk and Uncertainty Using Parameterized Efficient Sets," *SAE World Congress*, Detriot, MI.

[30] Malone, B., and Papay, M., 1999, "Modelcenter: An Integration Environment for Simulation Based Design," *Simulation Interoperability Workshop*.

[31] Mattsson, S. E., Elmqvist, H., and Otter, M., 1998, "Physical System Modeling with Modelica," *Control Engineering Practice*, **6**, pp. 501-510.

[32] McKay, M. D., Beckman, R. J., and Conover, W. J., 1979, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, **22**(2), pp. 239-245.

[33] Morris, M. D., 1991, "Factorial Sampling Plans for Preliminary Computational Experiments," *Technometrics*, **33**(2), pp. 161-174.

[34] Ong, Y. S., Nair, P. B., Keane, A. J., Wong, K. W., 2004, "Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems," *Knowledge Incorporation in Evolutionary Compuation*, Jin, Y. eds., Springer Verlag, pp. 307-322.

[35] Pahl, G., and Beitz, W., 1996, *Engineering Design: A Systematic Approach*, Springer-Verlag, London.

[36] Paredis, C. J. J., 1996, *An Agent-Based Approach to the Design of Rapidly Deployable Fault Tolerant Manipulators*, Thesis, Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh.

[37] Paredis, C. J. J., 2008, "An Open-Source Modelica Library of Fluid Power Models," in *Bath/ASME Symposium on Fluid Power and Motion Control*, Bath, UK.

[38] Perez, V. M., Renaud, John E., Gano, Shawn E., 2000, "Constructing Variable Fidelity Response Surface Approximations in the Usable Feasible Region," in *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA.

[39] Pugh, S., 1991, *Total Design: Integrated Methods for Successful Product Engineering*, Addison-Wesley, Reading, MA.

[40] Reyer, J. A., Papalambros, Panos Y., 2002, "Combined Optimal Design and Control with Application to an Electric Dc Motor," *Journal of Mechanical Design*, **124**, pp. 183-191.

[41] Rodriguez, J. F., and Renaud, J.E., 1998, "Convergence of Trust Region Augmented Lagrangian Methods Using Variable Fidelity Approximation Data," *Structural Optimization*, **15**, pp. 141-156.

[42] Rodriguez, J. F., Perez, V.M., Padmanabhan, D., and Renaud, J.E., 2001, "Sequential Approximate Optimization Using Variable Fidelity Response Surface Approximations," *Structural and Multidisciplinary Optimization*, **22**, pp. 23-34.

[43] Sage, A. P., Armstrong, J, 2000, *Introduction to Systems Engineering*, John Wiley & Sons, Inc., New York.

[44] Schmit, L. A., and Farshi, B., 1974, "Some Approximation Concepts for Structural Synthesis," *AIAA Journal*, **12**(5), pp. 692-699.

[45] Thurston, D. L., 1991, "A Formal Method for Subjective Design Evaluation with Multiple Attributes," *Research in Engineering Design*, **3**(2), pp. 105-122.

[46] Tiller, M. M., 2001, *Introduction to Physical Modeling with Modelica*, Kluwer, Boston, MA.

[47] von Neumann, J., and Morgenstern, O., 1980, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ.

[48] XIA Lu, G. Z.-h., 2006, "Application Fo Variable-Fidelity Models to Aerodynamic Optimization," *Applied Mathematics and Mechanics*, **27**(8), pp. 1089-1095.

[49] Zang, T. A., Green, L. L., 1999, "Multidisciplinary Design Optimization Techniques: Implications and Opportunities for Fluid Dynamics Research," in *30th AIAA Fluid Dynamics Conference*, Norfolk, VA.

[50] Zhou, Z., Nair, P.B., Keane, A. J., Lum, K. Y., 2007, "Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization," *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews*, **37**(1), pp. 66-76.