MULTIDISCIPLINARY DESIGN OPTIMIZATION OF DYNAMIC SYSTEMS USING SURROGATE MODELING APPROACH

BY

ANAND PRAMOD DESHMUKH

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Industrial Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Assistant Professor James T. Allison
Abstract

This thesis aims to explore a new paradigm for efficient solution of multidisciplinary design optimization (MDO) of dynamic systems. Many of the MDO problems in dynamic systems design often involve computationally expensive system simulations, severely limiting their design optimization. This work demonstrates a novel method which approximates the expensive system dynamics by cheap-to-evaluate surrogate models for system derivative functions. This is advantageous to do, since it preserves the inherent nature of dynamic system to certain accuracy and enables the efficient solution of MDO problems at the same time. The proposed method is demonstrated on a real world example of wind turbine design and obtained results are very encouraging.
To my parents, for their love and support.
I wish to express my sincere appreciation and gratitude toward Professor James T. Allison for his invaluable guidance and motivation throughout this thesis work. In addition, special thanks to Ashima and my family who have helped me through the sometimes trying process of earning my Master’s Degree.

I am also thankful to Ashish, Jeff, Dan and Tinghao for the wonderful discussions in the basement of Transportation Building.

Last but not the least, I am immensely grateful to the Department of Industrial & Enterprise Systems Engineering and Professor Allison for the continued financial support through Teaching and Research Assistantship, without which this work would not have come to fruition.
Table of Contents

List of Tables ................................................................. vi
List of Figures ................................................................. vii

Chapter 1 Introduction ...................................................... 1

Chapter 2 Co–Design of Dynamic Systems ......................... 3
  2.1 Sequential System Design ........................................... 3
  2.2 Nested Co-Design ..................................................... 5
  2.3 Simultaneous Co-Design ........................................... 6

Chapter 3 Surrogate Modeling ........................................... 7
  3.1 Sampling the Design Space ......................................... 8
  3.2 Constructing the Surrogate Model ............................... 8
  3.3 Model Validation ...................................................... 9

Chapter 4 Surrogate Modeling for Design of Dynamic Systems .. 10
  4.1 Constructing the Sampling Plan ................................... 13
  4.2 Surrogate Model Construction ................................... 13
  4.3 Model Validation ..................................................... 16
  4.4 Direct Transcription ................................................ 17

Chapter 5 Wind Turbine System Co–Design .......................... 19
  5.1 Wind System Optimization ......................................... 20
  5.2 Performance Characteristics of Wind Turbine .................. 22
  5.3 Sequential Design Formulation ................................... 26
  5.4 Nested Co-Design Formulation .................................... 28
  5.5 Simultaneous Co-Design Formulation ............................. 29
  5.6 Problem Setup ........................................................ 29
  5.7 Interfacing FAST with MATLAB .................................. 30

Chapter 6 Results and Discussion ...................................... 33
  6.1 Wind Turbine Co–Design ............................................ 33
  6.2 Surrogate Modeling based Design Optimization ............... 37
  6.3 Discussion ............................................................ 44

References ........................................................................... 48
List of Tables

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Optimal plant design vector for each of the formulations</td>
</tr>
<tr>
<td>6.2</td>
<td>Bounds on constraints and design variables for wind turbine co-design problem</td>
</tr>
<tr>
<td>6.3</td>
<td>Example 1: Optimal plant design vector</td>
</tr>
<tr>
<td>6.4</td>
<td>Example 2: Optimal plant design vector</td>
</tr>
<tr>
<td>6.5</td>
<td>Optimal plant design vector</td>
</tr>
<tr>
<td>6.6</td>
<td>Solution characteristics</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Surrogate modeling based optimization process for dynamic systems</td>
<td>14</td>
</tr>
<tr>
<td>4.2</td>
<td>Evaluation of sample training points</td>
<td>15</td>
</tr>
<tr>
<td>5.1</td>
<td>Variation of power coefficient ($C_p$)</td>
<td>24</td>
</tr>
<tr>
<td>5.2</td>
<td>Variation of power coefficient ($C_p$), for a fixed $\beta = 0.2$</td>
<td>25</td>
</tr>
<tr>
<td>5.3</td>
<td>Sketch of wind turbine system with considered modes of vibration</td>
<td>25</td>
</tr>
<tr>
<td>5.4</td>
<td>Optimization process using FAST</td>
<td>30</td>
</tr>
<tr>
<td>5.5</td>
<td>Output window of FAST.exe called through MATLAB system command</td>
<td>31</td>
</tr>
<tr>
<td>5.6</td>
<td>Input file for FAST executable</td>
<td>32</td>
</tr>
<tr>
<td>6.1</td>
<td>Input wind profile</td>
<td>35</td>
</tr>
<tr>
<td>6.2</td>
<td>Power output for each of the formulations</td>
<td>35</td>
</tr>
<tr>
<td>6.3</td>
<td>Blade flap-wise tip deflection</td>
<td>36</td>
</tr>
<tr>
<td>6.4</td>
<td>Optimal tip-speed ratio over time</td>
<td>36</td>
</tr>
<tr>
<td>6.5</td>
<td>Example 1 solution with non surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control</td>
<td>39</td>
</tr>
<tr>
<td>6.6</td>
<td>Example 1 solution with surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control</td>
<td>40</td>
</tr>
<tr>
<td>6.7</td>
<td>Example 2 Solution with non surrogate approach (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control</td>
<td>42</td>
</tr>
<tr>
<td>6.8</td>
<td>Example 2 Solution with surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control</td>
<td>43</td>
</tr>
<tr>
<td>6.9</td>
<td>Example 3 Solution: (a) Input wind speed (b) Optimal torque trajectory using non surrogate approach (c) Optimal torque trajectory using surrogate approach</td>
<td>45</td>
</tr>
</tbody>
</table>
Dynamic engineering systems are ubiquitous. They play a significant role in our day–to–day life. These systems are inherently multidisciplinary in nature, requiring knowledge from multiple disciplines. Moreover, many dynamic systems are actively controlled to make them perform in a desired manner. For example, a suspension of a car might be actively controlled to provide desired ride comfort to passengers on an uneven road surface. The Multidisciplinary Design Optimization (MDO) of such dynamic systems is challenging due to the lack of mature methods that address specific needs of dynamic systems. Most of the MDO methods have been developed for static MDO problems. Also, very often the design of the dynamic systems involves expensive system dynamic simulations (e.g. Multi–body Dynamic Analysis, Computational Fluid Dynamics etc.). There appears to be a significant need to propose smart methods for efficient solution of these integrated MDO problems for dynamic systems.

This thesis work is motivated by this need to advance the state of the art in field of MDO for dynamic engineering systems. It is author’s hope and belief that the fundamental methods and case studies presented in this thesis provide an exciting step toward achieving this goal. This work is focused on proposing an efficient solution method for ‘co–design’ problems for dynamic systems involving computationally expensive system simulations (or state derivative evaluations). Co–design is a special class of MDO problems for dynamic systems in which plant and control design problems are considered together to exploit synergy between both the disciplines to arrive at system optimal solutions.

In many dynamic system design problems, the associated state derivative evaluations are computationally expensive, resulting in simulations that are significantly slower than real–time. This makes the use of optimization techniques in the design of such systems impractical. In this work, an effi-
cient two-loop method, based on surrogate modeling, is proposed for solving
dynamic system design problems with computationally expensive derivative
functions. A surrogate model is constructed for only the derivative function
instead of the complete system analysis, as is the case in previous studies.
This approach addresses one of the most expensive element of system analy-
sis (i.e., the derivative function), while limiting surrogate model complexity.
Simulation is performed based on the surrogate derivative functions, preserv-
ing the nature of the dynamic system, and improving estimation accuracy.

This thesis work is broadly focussed on two themes: Co–design formu-
lations and Efficient solution of co–design problems for dynamic systems.
Chapter 2 and 3 discuss the preliminaries of co–design formulations and
surrogate modeling for approximating the unknown function behavior, re-
spectively. Chapter 4 discusses the core method for dynamic system design,
based on the surrogate models of derivative functions. Chapter 5 is focussed
on describing the case study on co–design of wind energy system. Finally,
Chapter 6 is focussed on the results and discussion of few case studies. Chap-
ter 6 is divided into two parts: first part is focussed on the general solution
of co–design for wind turbine system and second part applies the method de-
scribed in Chapter 4 to three different case studies including the wind turbine
design problem.
Co-design is a class of design optimization methods for actively-controlled dynamic systems. Design of physical systems and their associated control systems are often coupled tasks; design methods that manage this interaction explicitly can produce system-optimal designs, whereas conventional sequential processes (i.e., plant design followed by control design) may not [1]. In this section, we review the sequential design process, followed by a discussion of two co-design formulations: nested and simultaneous.

2.1 Sequential System Design

In design practice, the sequential design approach is used most often when developing actively controlled engineering systems. This involves designing the physical system first, and then designing the control system without modifying the plant design. When optimization is used, the sequential approach produces optimal solutions with respect to individual disciplines, plant and control design, but normally will not produce a system-optimal solution. The mathematical formulation for sequential system design includes both the plant and control design optimization problem. Here the plant design optimization problem is formulated as follows:

$$\begin{align*}
\min_{x_p} & \quad \psi(\xi(t), x_c, x_p) \\
\text{s.t.} & \quad g_p(\xi(t), x_p) \leq 0,
\end{align*}$$

(2.1)

where, $x_p$ is the vector of plant design variables, $\psi(\cdot)$ is the plant design objective, and $g_p(\cdot)$ is the vector of plant constraint functions and $x_c$ is the control design vector. The solution to Prob. (2.1)—i.e., the optimal plant design vector $x_{p*}$—is used as the basis for the optimal control design problem.
The objective and constraint functions in the optimal control problem depend on $x_{p^*}$, but its value is held fixed during the solution of optimal control problem:

$$\min_{x_c} \psi(\xi(t), x_c, x_{p^*})$$

s.t. $g_p(\xi(t), x_{p^*}) \leq 0$

$$\dot{\xi}(t) - f(\xi(t), x_c, x_{p^*}) = 0.$$  \hspace{1cm} (2.2)

The last term in Prob. (2.2) represents system dynamics, where $\xi(t)$ are time dependent system states and $f(\cdot)$ is the system derivative function.

The sequential design problem may be formulated in several ways. One approach for characterizing the difference between these formulations is to identify the nature of the objective function in each problem. Allison and Herber[2] presented a taxonomy of sequential design formulations. In much of the literature a distinct objective function is formulated for the plant design problem. The plant objective in this case is often an approximation of the dynamic system objective used in the optimal control problem. In the formulation presented here, we assume that the objective function for the plant and control design problems is identical. The only difference is what varies in each problem formulation.

In more complete co-design formulations, the plant design depends on state. For example, stress values in a turbine tower or blades depend on system state. While both the plant objective function and constraints will depend directly on state, only the objective function will depend directly on control design. Plant constraints, however, will depend indirectly on control design since state is influenced by control design. A co-design problem formulated in this way exhibits bi-directional coupling, i.e., plant design depends on control design, and vice versa. One consequence of plant constraints depending on state is the need to include plant constraints in the control optimization problem. Otherwise feasibility issues may arise.

Several options exist for solving the optimal control problem. A classical or ‘indirect’ approach is to apply optimality conditions such as Pontryagin’s Maximum Principle (PMP)[3 4], and then solve for the optimal control trajectory $x_c^*$ that minimizes $\psi(\cdot)$. If a closed-form solution to the optimality conditions cannot be found, the resulting boundary value problem (BVP) often can be solved numerically. This approach is therefore known as an ‘optimize-then-discretize’ approach, since optimality conditions are applied...
first to obtain a BVP, which is then discretized and solved\[5\]. One significant challenge in utilizing indirect optimal control methods in co-design is the need to satisfy inequality plant constraints. This is not possible in the general case. Other methods are needed that are more naturally suited for solving Prob. (2.2).

Optimal control problems may also be solved using direct methods, where an infinite-dimensional optimal control problem, such as the one given in Prob. (2.2), is ‘transcribed directly’ into a finite-dimensional nonlinear program (NLP). The discretized optimization problem can then be solved numerically using appropriate NLP algorithms, and can easily accommodate inequality plant constraints. This approach, known as Direct Transcription (DT)[5], is classified as a ‘discretize-then-optimize’ method, since discretization is performed before optimization.

2.2 Nested Co-Design

Allison and Herber[2] identified the nested co-design formulation as a special case of the Multidisciplinary Design Feasible (MDF) formulation. This formulation has two loops: an outer loop solves the plant design optimization problem, and an inner loop generates the optimal control for each plant design considered by the outer loop. The outer loop formulation is:

\[
\begin{align*}
\min_{x_p} \ & \psi^* (\xi(t), x_c, x_p) \\
\text{s.t.} \ & g_p (\xi(t), x_p) \leq 0,
\end{align*}
\]  

(2.3)

where, \(x_p\) is the plant design vector, \(g_p (\cdot)\) are the plant design constraints, and \(\psi^* (\cdot)\) is an optimal value function that depends only on \(x_p\). This optimal value function is evaluated by solving the inner loop optimal control problem, i.e., for a given plant design, it finds the optimal control and returns the objective function value. For every outer loop function evaluation, the inner loop is solved for the optimal control design vector \(x_c^*\):

\[
\begin{align*}
\min_{x_c} \ & \psi (\xi(t), x_c, x_p) \\
\text{s.t.} \ & g_p (\xi(t), x_p) \leq 0 \\
& \dot{\xi}(t) - f (\xi(t), x_c, x_p) = 0.
\end{align*}
\]  

(2.4)
As can be seen from above formulation, the plant design is held fixed during the inner loop solution. Plant design constraints $g_p(\cdot)$ are imposed in both loops to ensure system-level design feasibility. As with sequential system design, the optimal control problem must be solved using an optimization method that can accommodate inequality plant design constraints.

### 2.3 Simultaneous Co-Design

The simultaneous co-design problem formulation is:

$$
\begin{align*}
\min_{x_p, x_c} & \quad \psi(\xi(t), x_c, x_p) \\
\text{s.t.} & \quad g_p(\xi(t), x_p) \leq 0 \\
& \quad \dot{\xi}(t) - f(\xi(t), x_c, x_p) = 0
\end{align*}
$$

(2.5)

The solution to Prob. (2.5) yields the system-optimal design because it accounts for all dynamic system interactions and plant-control design coupling, resulting in a minimum $\psi(\cdot)$ that is lower than what could be achieved using the sequential approach. This formulation is often referred to as the simultaneous co-design method, as plant and control design decisions are made simultaneously.

With the preliminaries on co-design, we now move on to the discussion of surrogate modeling and its application to the co-design of dynamic systems.
Chapter 3

Surrogate Modeling

Surrogate modeling is a well known method for mathematically approximating the input–to–output mapping of the unknown function behavior. The surrogate models of unknown functions are obtained by fitting an approximate surface through the pairs of input–output sample points. Surrogate model construction needs relatively few function evaluations which makes them desirable for optimization problems.

The optimization algorithms iteratively arrive at optimal solution by computing the objective function multiple times. The optimization process becomes extremely time intensive if the objective function evaluation is computationally expensive. The surrogate model approximations of objective functions are very useful in such cases, to reduce the computational expense. The surrogate modeling based optimization process has mainly following four steps:

1. **Sampling the Design Space**: The sampling of design space is the key starting step in surrogate modeling based design optimization. The sampling of design space is achieved by means of design of experiments. The important consideration in this step is the goodness of such designs, considering the number of samples is severely limited by the computational expense of each sample.

2. **Surrogate Model Construction**: Once the design space is sampled with initial points, the obtained input–output response is used to fit a surface through the observed data. This step concerns itself with the questions such as: (a) What surrogate model(s) should be used? (b) How do we find the corresponding model parameters?

3. **Model Validation**: This step is necessary to establish the predictive capabilities of the surrogate model for general (non-sample) inputs within reasonable accuracy.
4. **Design Space Exploration**: Once the reasonably accurate surrogate model is obtained, the design space can be explored to get deeper insights into the function behavior. The surrogate model can also be used as an objective or constraint function in conjunction with optimization algorithm.

3.1 Sampling the Design Space

This section deals with the groundwork for construction of surrogate model $\hat{f}$, that emulates the expensive response of some black-box function or system $f$. where $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous function defined over the design space $D \subset \mathbb{R}^n$. With this assumption of continuity, we proceed to make observations of $f$ through the evaluation of some sample (discrete) points $x^{(i)} \in D : x^{(i)} \rightarrow y^{(i)} = f(x^{(i)})$, for $i = 1, 2 \ldots k$. The origins of design space sampling can be found in classical design of experiments (DOE). Some of the widely used DOE techniques include factorial or Fractional Factorial Design, Central Composite Design (CCD), Box-Behnken Design. These methods tend to spread the sample points around boundaries of the design space and leave a few at the center of the design space. Where as, some other DOE techniques focus on space filling designs. Examples of space filling DOEs include, Maximum Entropy Design, Mean Squared-Error Designs, Minimax and Maximin Designs, Latin Hypercube Designs, Orthogonal Arrays (see [6],[7],[8],[9] for review).

3.2 Constructing the Surrogate Model

Construction of surrogate modeling evolves naturally from classical design of experiments (DOE) theory, in which polynomial functions are used as response surfaces. Besides the commonly used polynomial functions, there have been many statistical methods for modeling the response, such as Kriging Method, Neural Networks etc. Other types of deterministic surrogate models include radial basis functions, multivariate adaptive regression splines, or a combination of polynomial functions and artificial neural networks. No decision can me made about superiority of one surrogate modeling approach
over other. The detailed description of many of these methods can be found in various sources (see [6], [7], [8], [9] for review).

### 3.3 Model Validation

This step is needed to ensure that surrogate models are accurate within a desired tolerance limit. There have been numerous metrics for error computation in literature (see [7]). One of the widely used metric of error is the cross-validation error. To compute the cross validation error, the surrogate model training data are split randomly into $k$ roughly equal subsets and each of the subset is removed turn-by-turn and surrogate model is fitted using the remaining data-sets. The error at a test point is then calculated as $\|y(x_s) - \hat{y}_{-i}(x_s)\|$, where $y(x_s)$ is the actual function response at the test point $x_s$ and $\hat{y}_{-i}(x_s)$ is the predicted function response using the surrogate model constructed after removing $i^{th}$ subset from the training points. The total cross-validation error is then obtained by summing up all the $k$ such errors:

$$
\epsilon_{cv} = \sum_{i=1}^{k} \|y(x_s) - \hat{y}_{-i}(x_s)\| \quad (3.1)
$$

The cross-validation method essentially quantifies the degree of insensitivity of a surrogate model to the lost information[6], however an insensitive surrogate model is not necessarily an accurate one. There are few other model validation techniques employing additional points, one of those is the Root Mean Square Error (RMSE). RMSE is evaluated for $m$ test points as:

$$
\epsilon_{rmse} = \sqrt{\frac{\sum_{i=1}^{m} (y^{(i)} - \hat{y}^{(i)})^2}{m}} \quad (3.2)
$$

where, $y^{(i)}$ and $\hat{y}^{(i)}$ are the actual and predicted function responses at $i^{th}$ test point, respectively. The surrogate modeling description in this chapter is not meant be exhaustive, readers are referred to any standard text on surrogate modeling based optimization (such as [7]) for detailed treatment on this topic. With the basic preliminaries on surrogate modeling, we now move on to the application of surrogate modeling to design of nonlinear dynamic systems, in Chapter 4.
Chapter 4

Surrogate Modeling for Design of Dynamic Systems

Optimization of nonlinear (or linear state-dependent) dynamic systems often requires system simulation. In many cases the associated state derivative evaluations are computationally expensive, resulting in simulations that are significantly slower than real-time. This makes the use of optimization techniques in the design of such systems impractical. Optimization of these systems is particularly challenging in cases where control and physical systems are designed simultaneously. In this work, an efficient two-loop method, based on surrogate modeling, is proposed for solving dynamic system design problems with computationally expensive derivative functions. A surrogate model is constructed for only the derivative function instead of the complete system analysis, as is the case in previous studies. This approach addresses one of the most expensive element of system analysis (i.e., the derivative function), while limiting surrogate model complexity. Simulation is performed based on the surrogate derivative functions, preserving the nature of the dynamic system, and improving estimation accuracy.

The objective here is to apply this surrogate model based derivative function technique to co-design problems. In codesign problems the plant and control design are solved together. In that case, the surrogate model approximates derivative functions that depend on both design and state variables. As a result, the method must not only ensure accuracy of the surrogate model near the optimal design point in the design space, but also the accuracy of the model in the state space near the state trajectory that corresponds to the optimal design.

The surrogate modeling process presented here is similar to previous studies (see [9, 6, 7] for reviews) in that it is a method for obtaining optimal design solutions through the use of surrogate mathematical models that approximate physics-based models of an actual system. The optimization algorithm does not operate directly on the high-fidelity physics-based model, but rather
on the approximate surrogate model. This approach is particularly useful in cases where the original high-fidelity system model is computationally expensive. Operating on the surrogate model can speed up the optimization process significantly as the surrogate model requires much less time to evaluate than the original model, and surrogate models often can help smooth out numerical noise present in the original model [8]. The computational expense of obtaining samples required to build the surrogate model must be accounted for when evaluating whether a surrogate modeling approach is a good choice for a particular problem [10].

Successful surrogate modeling methods support the rapid identification of an accurate optimum design point with a minimum number of high-fidelity function evaluations [11]. Accuracy can be preserved by using a trust region approach [12, 13], and the number of high-fidelity model evaluations can be reduced by using an adaptive resampling method that focuses on improving accuracy only in regions of strategic interest (e.g., near the optimum) [14, 15]. A significant number of developments have been made in the area of black-box surrogate modeling, including the use of a family of surrogate models where the best (or weighted average) surrogate model is used as required [16], and extension of surrogate modeling to multi-objective optimization problems where high accuracy is maintained in regions near the Pareto front [17].

While in many cases surrogate modeling has been applied to a single engineering discipline at a time [17] (e.g., structural design [18], multibody dynamic systems [19], design based on aerodynamics and aero-acoustics [20], etc.), it can be extended to multidisciplinary problems [21]. Co-design problems are multidisciplinary design optimization problems that involve the coupled physical and control system design disciplines [22]. This introduces additional complexity to the surrogate modeling problem, as accuracy must be provided not only in the design space in the neighborhood of the optimum design point, but also in the state space in the neighborhood of the state trajectory that corresponds to the optimum design point. The latter requirement is more difficult because we are concerned about accuracy in a region near an entire path as opposed to a single point. This thesis work introduces one possible approach for tackling this challenge associated with co-design problems.

Consider a general co-design optimization problem formulation that in-
volves the simultaneous optimization of physical system and control system
designs:

$$\min_{x_p,u(t)} \quad J = \int_{0}^{t_E} L(\xi(t), x_p, u(t)) dt$$

subject to

$$g(\xi(t), x_p) \leq 0$$

$$h(\xi(t), x_p) = 0$$

$$\dot{\xi}(t) = f(\xi(t), x_p) + Bu(t).$$

Here $J$ is a cost function that represents the overall system design objective,
where the integrand $L(\cdot)$ is the Lagrangian. The plant and control design
variables are $x_p$ and $u(t)$, respectively, and $g(\cdot)$ and $h(\cdot)$ are the inequality
and equality design constraints, respectively. Note that design constraints
depend indirectly on control design since $u(t)$ influences state trajectories
$\xi(t)$. This problem structure allows for bi-directional plant-control design
coupling [23]. This formulation admits nonlinear system dynamics, i.e. the
state derivatives $\dot{\xi}(t)$ are nonlinear functions of states and physical design.
The scope of this work is limited to systems that depend linearly on control
$u(t)$.

The core contribution of this work is centered on efficient approximation
methods for the derivative function $f(\cdot)$. We seek to construct a surrogate
model $\hat{f}(\cdot)$ of $f(\cdot)$ based on sampling in both the state and design spaces.
Equation (4.2) illustrates an approximate system dynamics model based on
the surrogate model $\hat{f}(\cdot)$:

$$\dot{\xi}(t) \approx \hat{f}(\xi(t), x_p) + Bu(t)$$

where $\hat{f}(\xi(t), x_p) \approx f(\xi(t), x_p)$. The co-design problem based on this surro-
gate model is:

$$\min_{x_p,u(t)} \quad J = \int_{0}^{t_E} L(\xi(t), x_p, u(t)) dt$$

subject to

$$g(\xi(t), x_p) \leq 0$$

$$h(\xi(t), x_p) = 0$$

$$\dot{\xi}(t) \approx \hat{f}(\xi(t), x_p) + Bu(t)$$

The design method proposed here consists of an inner loop that solves Prob. (4.3),
and an outer loop that iteratively enhances the surrogate model. The method
consists of the following five steps:
• Define the sampling domain in state space and design space
• Sample test points in the combined state and design spaces
• Build and validate the state derivative surrogate model
• Solve the co-design problem
• Check accuracy and convergence requirements, repeat steps 1–4 until requirements are satisfied

This iterative process is illustrated in Fig. 4.1 and described in detail in the following subsections.

4.1 Constructing the Sampling Plan

The process starts with a definition of the modeling domain, i.e., the regions within the state and design spaces where the surrogate model will be constructed, and the regions from which samples will be obtained. Here the modeling domain is defined using simple bounds on the state and design spaces that are estimates of the maximum and minimum values that the plant design and state variables will attain. Sample points are chosen from within the modeling domain using Latin Hypercube Sampling (LHS) [7].

4.2 Surrogate Model Construction

The sample points obtained via LHS in the previous step are used as training points to construct the surrogate model. For every training point defined, a corresponding output point must be obtained by evaluating the analysis function (the original model to be approximated) using the training point as input. The observed output points are functions of the training points \( \mathbf{y} \), i.e., \( \mathbf{f}(\mathbf{y}) \). Here \( \mathbf{f}(\mathbf{y}_i) = [f_1(\mathbf{y}_i), f_2(\mathbf{y}_i), f_3(\mathbf{y}_i), \ldots, f_n(\mathbf{y}_i)]^T \) is the output vector of observed derivatives for the training point \( \mathbf{y}_i \), where \( n \) is the number of states and each entry \( f_j \) corresponds to the \( j \)th state derivative \( (j = 1, 2, 3, \ldots, n) \). Figure 4.2 illustrates this relationship.

Here the analysis function is neither a design objective nor constraint function as is normally the case, but is the computationally-expensive derivative
Figure 4.1: Surrogate modeling based optimization process for dynamic systems
Figure 4.2: Evaluation of sample training points

function that governs system dynamics: \( f(\xi(t), x_p) \). A training point consists of values for both state and design variables, i.e., \( y_i = [\xi, x_p] \), and the output of the derivative function is vector-valued, so each training point produces multiple \((n)\) observed outputs. Once the observed outputs \((f(y_i))\) are obtained, the input–output pairs may be used to ‘train’ the surrogate model for each of the state derivatives, separately. The surrogate model used here employs Radial Basis Functions (RBFs) \([24]\). For each of the \(n\) state derivatives, we write the interpolation condition using \(p\) training points:

\[
    f_j(y_i) = \sum_{k=1}^{p} w_{kj}^i \psi(\|y_i - c_k\|), \quad \text{for} \quad i = 1, 2, \ldots, p, \quad j = 1, 2, \ldots, n.
\]

where \(\psi(\cdot)\) is the radial basis function, \(w_{kj}^i\) are unknown weighting coefficients, \(n\) is the number of states, and \(c_k\) is the \(k^{th}\) basis function center. The specific RBF used here is the thin plate spline function \([7]\): \(\psi(r) = r^2 \ln r\), where \(r\) is the Euclidean distance between the training point and function center: \(r = \|y_i - c_k\|\). The objective in constructing the surrogate model is to find the coefficients \(w_{kj}^i\). This can be done by solving the following equation for \(w_j^i = [w_{1j}^i, w_{2j}^i, \ldots, w_{pj}^i]^T:\)

\[
    \psi w_j^i = f_j,
\]

where \(\psi\) is the ‘Gram matrix’ \([7]\): \(\psi_{i,k} = \psi(\|y_i - c_k\|)\) for \(i, k = 1, 2, \ldots p\), and \(f_j = [f_j(y_1), f_j(y_2), \ldots, f_j(y_p)]^T\) is the vector of observed outputs for \(j^{th}\) state derivative for \(p\) training points.

Unique values for coefficients may be found since the Gram matrix is square. Problem complexity is reduced further here by assuming that the RBF centers coincide with training points, i.e., \(c_i = y_i\). This simplification provides reasonably accurate results for the case studies presented in this thesis work.
4.3 Model Validation

Surrogate models used in optimization must be accurate in regions of interest. Adaptive surrogate modeling methods gradually enhance accuracy in the region of the approximate optimum, and at convergence the highest desired level of accuracy need only be achieved right at the optimum. Ensuring model accuracy in regions far from the optimum incurs unnecessary computational expense. Using surrogate models of derivative functions adds some complexity to the task of validating model accuracy. In addition to checking accuracy in regions near the optimal design point, accuracy must also be assured in the neighborhood of the state trajectory that corresponds to this point. A validation domain must be defined in both the state and design spaces where the surrogate model is required to be accurate within a specified tolerance.

In the iterative solution process outlined in Fig. 4.1, the boundaries of the modeling domain from which training samples are obtained are updated with each outer loop iteration. The inner loop (co-design) solution is used to determine new bounds on the modeling domain for the next iteration. For simplicity here, the validation domain is assumed to be equivalent to the modeling domain, although more sophisticated approaches may be taken where the validation domain is much smaller than the modeling domain. Techniques, such as support vector domain description (SVDD) [25, 26], could be used to construct non-convex boundaries around the state trajectories to define a tighter validation domain.

Several different error metrics for surrogate model validation have been investigated. One of the most widely used metrics is the root mean square error (RMSE) [27, 28]. RMSE is suitable for scalar-valued functions, but the derivative functions of interest here are vector-valued. The sum of normed errors (SNE) is used here to accommodate the vector-valued analysis function. The error for each test point \((y_{s_i})\) is defined as the 2-norm of difference between the actual state derivative vector \(f(y_{s_i})\) and prediction of state derivative vector \(\hat{f}(y_{s_i})\):

\[
SNE = \sum_{i=1}^{n_s} \|\hat{f}(y_{s_i}) - f(y_{s_i})\|, \quad (4.6)
\]

where \(n_s\) is the number of test points. As illustrated in Fig 4.1, model error
is checked before proceeding with co-design solution. Alternative approaches may include validation checks after co-design solution.

4.4 Direct Transcription

Conventional optimal control methods based on Pontryagin’s Maximum Principle [3] take an ‘optimize–then–discretize’ approach, where optimality conditions are applied to generate a closed-form solution (possible only in limited cases), or a boundary value problem that can then be discretized and solved for the optimal control trajectories. Direct Transcription (DT) takes the inverse approach: the optimal control problem is discretized first, and the resulting nonlinear program (NLP) is solved using a standard NLP algorithm [29, 30]. DT is a ‘discretize–then–optimize’ approach that transcribes an infinite-dimensional optimal control problem into a large sparse finite dimensional NLP. State and control trajectories trajectories are discretized over a finite number of time intervals, and these discretized representations are part of the set of optimization variables. The differential constraint that governs system dynamics is replaced by a finite set of algebraic defect constraints. These defect constraints can be formed using any standard numerical collocation method, such as implicit Runge-Kutta (IRK) methods or Gaussian quadrature. The trapezoidal method, an IRK method, is used in the implementations here.

Allison and Han introduced an extension of DT for co-design problems [22]. A DT co-design formulation based on this work, using the trapezoidal method, follows:

$$\min_{\gamma=x_p, \Xi, U} \sum_{i=1}^{n_t-1} L(x_p, \xi_i, u_i) h_i$$

subject to:

$$\zeta(x_p, U, \Xi) = 0$$

$$g_p(x_p, \Xi) \leq 0,$$

where, \(n_t\) is the number of time steps, \(\Xi = [\xi_1, \xi_2, \cdots, \xi_n]\) is the matrix of discretized state variables (row \(i\) corresponds to the state at time \(t_i\)), \(\zeta(\cdot)\) are the defect constraint functions imposed to ensure that \(\Xi\) satisfies system state equations, \(U = [u_1, u_2, \cdots, u_n]\) is control input matrix, and \(h_i\) is the \(i\)th time step size. This DT-based co-design approach was used to solve Prob. 4.3—the inner loop problem in the surrogate modeling method—
for several numerical examples that are presented in the next section. The surrogate derivative function \( \hat{f}(\cdot) \) is used in the calculation of the defect constraints \( \zeta(\cdot) \). The NLP defined in Prob. (4.7) was solved using the \textit{fmincon} algorithm in MATLAB\textsuperscript{®}. An important advantage of DT to emphasize here is its parallel nature; all defect constraints are independent, enabling massively parallel implementations.

With this background on co–design formulations and surrogate modeling based design optimization of dynamic systems, we then move on to solving a wind turbine co–design problem.
Wind energy is proving to be a promising energy source to complement conventional energy systems in meeting global energy demands, and is currently one of the fastest growing renewable energy sources. Modern wind turbines are large, flexible structures operating in uncertain environments. Power capture and economic value increase with turbine size, leading to steadily increased turbine size over the last three decades. Along with larger size comes intensified structural loads, presenting challenges in mechanical system design. One of these challenges is the dynamic deflection of structural components. These passive system dynamics interact with the active control of wind turbine energy generation. Because of this interaction, addressing the physical and control system design of these devices in a comprehensive manner is vital to ensuring maximum energy extraction, system reliability, and other critical metrics. A large portion of existing work has aimed to increase energy production through optimal control system design (through some combination of rotor speed and pitch control). This strategy treats physical system design as a fixed entity, overlooking potential gains. Further performance increases can be realized through a broader systems approach where physical and control system design are tackled simultaneously. This approach, known as co-design, can capitalize on the synergy that exists between passive system dynamics and active control to increase performance further. In this work a new method for wind turbine design is presented that produces system-optimal results by accounting for the coupling between plant system and control system design. A case study is presented that demonstrates significant performance improvements over conventional sequential design approaches.
5.1 Wind System Optimization

The operating regimes for wind turbine systems have traditionally been categorized in three operational zones: Zone 1: below cut-in wind speed (i.e., speeds below the minimum required to produce useful power), Zone 2: between cut-in and cut-out wind speeds (cut-out speed is the speed at which turbine operation must be modified to prevent damage), and Zone 3: above the cut-out wind speed. Wind turbines are designed to provide optimum power at the rated wind speed, which is in Zone 2. Efforts aimed at improving dynamic turbine performance in Zone 2 are often focused on increasing power extraction using some combination of rotor torque and speed control, as well as blade pitch control[31]. Addressing control system design without considering the potential synergy of simultaneous physical design modifications does not lead to the best possible system performance. This article presents an investigation that compares conventional sequential design (i.e., performing control system design after physical system design is complete) methods applied to horizontal-axis wind turbines to integrated design methods that consider control and physical system design together (e.g., co-design). Results indicate a significant performance increase between the sub-optimal sequential design result and the system-optimal co-design result.

One widely-used model for wind turbine rotor power, $P_m$, is given by[32]:

$$P_m = \frac{1}{2} C_p(\lambda, \beta) \rho \pi R^2 v^3$$  \hspace{1cm} (5.1)

where $C_p$ is the power coefficient, which is a non linear function of blade tip speed ratio ($\lambda$) and blade pitch angle ($\beta$). $\rho$ is the air density, $R$ is blade length, and $v$ is the wind speed (assumed here to be uniform over the entire swept rotor area). For a given physical turbine design and wind speed in Zone 2, the power capture maximization problem reduces to tracking the optimal power coefficient ($C_{p_{opt}}$) by controlling the blade-tip speed ratio and blade pitch angle.

Thiringer and Linders[32] presented an early investigation of power capture maximization via rotor speed control (in turn controlling the tip speed ratio $\lambda$) in a variable-speed, fixed-pitch machine configuration. They controlled the rotor speed trajectory to track the maximum power coefficient over the range of operating wind speeds in Zone 2. They also implemented the control
strategy on a physical wind conversion system and obtained results at two different geographic sites. Experimental results showed significant improvement in power capture compared to a fixed rotor speed approach. Narayana and Putrus[33] extended this idea by using artificial neural networks[34]. They used a Nonlinear Autoregressive Moving Average (NARMA) neural network model to provide the sensorless prediction of wind speed. By using this wind speed prediction, they tracked optimal rotor speed for varying wind speeds. Dang et al.[35, 36] used Model Predictive Control (MPC) for maximum power capture of wind power below rated wind speed. Maximum power was obtained by regulating shaft speed to track the optimal trajectories closely. Burnham[37] proposed a novel way of optimal power tracking by manipulations on the generator side. He showed that by adding a variable external resistance to the rotor of an induction generator used in a wind turbine, it is possible to manipulate the torque-speed curve and control the output power. Kusiak et al.[38] presented an intelligent wind turbine control system based on models integrating the following three approaches: data mining, model predictive control, and evolutionary computation. They proposed a multi-objective model involving five different weighted objectives. These weights were adjusted in response to the variable wind conditions and operational requirements.

The above studies focus on rotor speed or torque control. An alternative approach is to use blade pitch control. For example, Namik and Stol[39] proposed a method based on individual blade pitch control that improved power output performance for onshore and offshore wind turbines. They showed that using individualized blade pitch control enhances wind disturbance rejection, helps reduce structural tower loads, and improves power capture.

Tower mass correlates strongly with structural system cost. The need to reduce mass and cost competes with the need to construct taller towers to improve energy capture. Increasing height while targeting lower mass designs results in lighter-weight towers with significant elastic compliance. This increased compliance intensifies the risk of aeroelastic instabilities, adding to design and reliability challenges, and hindering efforts to improve energy capture[40]. The coupling between structural dynamics and control of the turbine and generator, which is stronger for taller towers, motivates the development design approaches that simultaneously address structural and control system design to account for (an even capitalize on) control-structure inter-
action. This work presents a design study that accounts for flexibility using a finite-element model, and control-structure interaction using a simultaneous structure-control design approach.

It is clear from the literature that most efforts have been focused on individual wind turbine design disciplines or objectives, such as optimal control for power production\cite{41}, control for load alleviation or structural design\cite{42, 43, 44}, optimization for strength or system weight\cite{45, 46}, and blade design for improved efficiency\cite{47, 48}. There appears to be a strong need to solve integrated aeroservoelastic problems in wind energy domain, as identified by Jonkman\cite{49}, to obtain system-optimal designs. A tighter integration between physical system (plant) and control system design must be established at a much earlier phase in design process, accounting for the coupling between plant and control system design. With this underlying motivation, we propose a simultaneous approach for plant and control system co-design to generate system optimal solutions to power extraction problem that outperform single-discipline design results. A co-design optimization formulation is proposed, and results are compared to those generated using convention sequential design. The co-design formulation well-suited for this problem, in the sense that it opens up design possibilities not accessible when treating plant and control design separately.

5.2 Performance Characteristics of Wind Turbine

In this section we look at some of the underlying physics of wind turbine operation and formulate the co-design problem for wind turbine design. The wind that is incident on the turbine rotor (consisting of blades and shaft) generates a torque on the rotor shaft. This torque drives the generator that produces the output electric power. The rotor torque, as a function of wind speed, $v$, is modeled as:

$$T_r = \frac{1}{2} C_q(\lambda, \beta) \rho \pi R^3 v^2$$

where, $C_q(\lambda, \beta)$ is the torque coefficient. Normally $C_q(\cdot) < 1$ due to aerodynamic losses. The torque coefficient and power coefficient $C_p(\lambda, \beta)$ are
related by:

\[ C_q(\lambda, \beta) = \frac{C_p(\lambda, \beta)}{\lambda} \quad (5.3) \]

where \( \lambda \) is the tip speed ratio of the blade, defined as the ratio of tip tangential speed \( (\omega R) \) to wind speed \( (v) \). This ratio is of particular significance as it is a key factor that governs the optimal power extraction from wind turbines.

\[ \lambda = \frac{\omega R}{v} \quad (5.4) \]

Our objective here is to maximize power production, defined in Eqn. (5.1), with respect to plant and control design variables, subject to plant design constraints. Using the above model, this is equivalent to maximizing the power coefficient \( C_p(\lambda, \beta) \) across the range of expected wind speeds. The values for power coefficient are typically obtained by performing an analysis using Blade Element Theory (BEM) [50]. The power coefficient curves are different for each wind turbine. An example power coefficient curve is illustrated in Fig. 5.1. This curve was obtained using the following empirical relationship [32]:

\[ C_p = 0.5 \left( \frac{116}{\lambda_i} - 0.4\beta - 5 \right) e^{\frac{21}{\lambda_i}}, \quad \text{where} \quad \lambda_i = \left( \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \right)^{-1} \quad (5.5) \]

is an intermediate value of the tip speed ratio.

Power production maximization requires that we: 1) Physically design the turbine so that it produces the maximum possible power while satisfying the structural constraints, and 2) Employ an optimal control law that ensures the maximum power coefficient is achieved (and hence maximum power production). The wind turbine considered here is a variable speed, fixed pitch machine. Since \( \beta \) is constant, the turbine should be operated in a way such that the optimum tip-speed ratio \( (\lambda) \) is maintained, which in turn assures maximum power production. This can be achieved by controlling rotor speed via a rotor torque resistance provided by the generator. Consider the following dynamics:

\[ J_r \ddot{\omega}_r = T_r - K_r \omega_r - T_g \quad (5.6) \]

where, \( J_r \) is the rotor inertia, \( \omega_r \) is the rotor speed, \( K_r \) is the torsional stiffness of the rotor, and \( T_g \) is the generator torque as seen on the low speed side.
Figure 5.1: Variation of power coefficient ($C_p$)

(LSS) of the rotor, and acts as a resistance torque on the rotor. This torque $T_g$ is the control input to the system that can be used to regulate the optimal tip speed ratio. The generator torque on the LSS side is related to torque on high speed side (HSS), by the gear ratio:

$$\frac{T_g}{T_{g\,hss}} = \eta_g$$  \hspace{1cm} (5.7)

where the rotor and blades are defined as being on the LSS, and generator side is defined as HSS since the generator shaft rotates at higher speed than the rotor.

A small subset of the possible plant design variables were chosen for this design formulation based on several factors, including impact on power production and influence on system dynamics (e.g., variables that influence component stiffness, inertia, and damping). The plant design vector used here
Figure 5.2: Variation of power coefficient ($C_p$), for a fixed $\beta = 0.2$

Figure 5.3: Sketch of wind turbine system with considered modes of vibration
is:

\[ x_p = \begin{bmatrix} R, & R_h, & H_t \end{bmatrix}^T, \]

where \( R \) is the blade length, \( R_h \) is the blade hub distance from the rotor axis, and \( H_t \) is the tower height from the ground. Please refer to Fig. (5.3) for a graphical definition of these design variables.

As introduced in Eqn. (5.6), the control input is: \( T_g(t) \), generator torque. In this study open-loop control control is assumed, so the control design variable is the control input trajectory \( T_g(t) \). With the analysis model and design variable definitions in place, the design optimization formulations can now be presented. Here the three different formulations are presented for comparison: sequential design, nested co-design, and simultaneous co-design.

### 5.3 Sequential Design Formulation

In the sequential design formulation we consider the the structural design problem first, followed by the control design. The structural design problem is formulated to maximize the peak power capture \( P_{\text{max}} \) from wind turbine rotor by adjusting the plant design variables \( x_p \) only. The plant design optimization formulation is:

\[
\min_{x_p} -P_{\text{max}}(\xi(t), x_p)
\]

s.t. \[
\|\delta_b(t)\|_{\infty} - \delta_{b_{\text{max}}} \leq 0 \\
\|\delta_{t1}(t)\|_{\infty} - \delta_{t1_{\text{max}}} \leq 0 \\
\|\delta_{t2}(t)\|_{\infty} - \delta_{t2_{\text{max}}} \leq 0 \\
\dot{\xi}(t) = f(\xi(t), x_p, u_e(t)) \\
0 < x_l \leq x_p \leq x_u,
\]

where, \( P_{\text{max}}(\cdot) \) is the peak power output from the wind turbine for given input wind profile. \( \dot{\xi}(t) = f(\xi(t), x_p, u_e(t)) \) is the nonlinear structural dynamic model of the wind turbine provided by FAST\[51\]. FAST is an open-source software developed by National Renewable Energy Laboratory (NREL) for wind turbine design and analysis. FAST includes a finite-element model that approximates structural dynamics. The dynamic model is a nonlinear function of plant design variables and system states. \( u_e(t) \) consists of exogenous
inputs, such as thrust force and rotor torque that are input to the system due to incoming wind, which are distinct from control inputs. \( \delta_b(t) \) is the flap-wise deflection of the blade tip, \( \delta_{t1}(t) \) is the side-to-side deflection of the tower top, and \( \delta_{t2}(t) \) is the fore-to-aft tower top deflection. These deflections are constrained by upper bounds \( \delta_b^{\text{max}} \), \( \delta_{t1}^{\text{max}} \), and \( \delta_{t2}^{\text{max}} \), respectively. We employ these deflection constraints as surrogates for preventing fatigue failure. These deflections can be obtained by running the forward simulation of system dynamic model in FAST. \( x_l \) and \( x_u \) are the lower and upper bounds on the plant design variables.

Once the plant design is optimized, we then proceed onto solving the optimal control problem based on the optimal plant design \( x_{p_\star} \). The optimal control problem is formulated to track the optimal tip speed ratio trajectory \( \lambda_\star(t) \) for a given input wind speed by regulating the resistance torque \( (T_g(t)) \):

\[
\begin{align*}
\min_{T_g(t)} \int_{t_0}^{t_f} (\lambda(t) - \lambda_\star(t))^2 \\
\text{s.t. } \dot{\omega}_r(t) &= -\left( \frac{K_r}{J_r} \right) \omega_r(t) + \frac{1}{J_r} T_r(t) - \frac{1}{J_r} T_g(t).
\end{align*}
\]  

(5.9)

In sequential system design approach, the plant design problem given by Prob. (5.8) is solved first, and then optimal control problem given by Prob. (5.9) is solved while holding the plant design fixed at the value produced by solving Prob. (5.8). Note that this sequential approach only accounts partially for control-structure interaction. It does not fully capitalize on synergy between plant and control design because the plant design problem is not informed by control design needs. The sequential approach could be iterated to help address this issue, but iterated sequential approaches are typically inefficient and may exhibit convergence problems\[2\]. Peters et al. introduced the concept of control proxy functions that help inform plant design of control design needs using only a single pass of the sequential approach\[52\]. In very limited cases the control proxy function approach is capable of producing system-optimal designs. The following two formulations, nested and simultaneous co-design, are able to produce system-optimal designs for a much more general set of system design problems.
5.4 Nested Co-Design Formulation

As described in Section 2.2, the nested co-design formulation is solved as a two-loop problem. The outer loop solves the problem with respect to plant design variables only. For every function call in the outer loop, an inner-loop problem must be solved. More specifically, the outer loop provides the inner loop with a candidate plant design, and the inner loop problem is to solve the optimal control problem for that particular plant design. The following formulation details the outer-loop formulation for the wind turbine design problem:

\[
\begin{align*}
\min_{x_p} & \quad -P_{\text{max}}(x_p) \\
\text{s.t.} & \quad \| \delta_b(t) \|_\infty - \delta_{b,\text{max}} \leq 0 \\
& \quad \| \delta_{t1}(t) \|_\infty - \delta_{t1,\text{max}} \leq 0 \\
& \quad \| \delta_{t2}(t) \|_\infty - \delta_{t2,\text{max}} \leq 0 \\
& \quad \dot{x}(t) = f(x(t), x_p, u_e(t)) \\
& \quad 0 < x_l \leq x_p \leq x_u.
\end{align*}
\]

\(P_{\text{max}}(\cdot)\) is an optimal value function; it returns the best possible power capture for a given plant design, which is calculated by solving the optimal control problem. As noted earlier, the plant design constraints are imposed in both the inner and outer loops to system design feasibility. The inner-loop formulation is:

\[
\begin{align*}
\min_{T_g(t)} & \quad \int_0^{T_f} (\lambda(t) - \lambda_*(t))^2 \\
\text{s.t.} & \quad \| \delta_b(t) \|_\infty - \delta_{b,\text{max}} \leq 0 \\
& \quad \| \delta_{t1}(t) \|_\infty - \delta_{t1,\text{max}} \leq 0 \\
& \quad \| \delta_{t2}(t) \|_\infty - \delta_{t2,\text{max}} \leq 0 \\
& \quad \dot{\omega}_r(t) = -\left( \frac{K_r}{J_r} \right) \omega_r(t) + \frac{1}{J_r} T_r(t) - \frac{1}{J_r} T_g(t)
\end{align*}
\]
5.5 Simultaneous Co-Design Formulation

The simultaneous co-design formulation for power capture maximization is:

\[
\min_{x_{p}, T_g(t)} - P_{\text{max}}(\xi(t), x_{p}, T_g(t)) \\
\text{s.t.} \\
\|\delta_b(t)\|_{\infty} - \delta_b^{\text{max}} \leq 0 \\
\|\delta_{t1}(t)\|_{\infty} - \delta_{t1}^{\text{max}} \leq 0 \\
\|\delta_{t2}(t)\|_{\infty} - \delta_{t2}^{\text{max}} \leq 0 \\
\dot{\xi}(t) = f(\xi(t), x_{p}, u_e(t)) \\
\dot{\omega}_r(t) = -\left(\frac{K_r}{J_r}\right)\omega_r(t) + \frac{1}{J_r}T_r(t) - \frac{1}{J_r}T_g(t) \\
0 < x_l \leq x_{p} \leq x_u,
\]

Problem (5.12) is solved using a discretize-then-optimize approach, such as Direct Transcription[5]. This involves discretizing in time the control and state trajectories, as well as the system dynamics constraints. We seek to solve the Prob. (5.12) by simultaneously considering plant and control design. The solution provides several important insights based on optimal plant and control design results, improving our understanding of how to improve wind turbine design. Comparison of the nested and simultaneous solution approaches with the sequential approach is presented in the results section.

5.6 Problem Setup

The wind turbine power production maximization problem for all the formulations was solved using FAST[51] as the analysis function. The FAST code models the wind turbine as a combination of rigid and flexible bodies. The rigid bodies are the earth, nacelle, hub, and optional tip brakes (point masses). The flexible bodies include blades, tower, and drive shaft. The model connects these bodies with several degrees of freedom (DOFs). These include tower bending, blade bending, nacelle yaw, rotor teeter, rotor speed, and drive shaft torsional flexibility. The flexible tower has two modes each in the fore-aft and side-to-side directions. The flexible blades have two flap-wise modes and one edgewise mode per blade. One can turn these DOFs on or
off individually in the analysis by simply setting a switch in the input data file.

An example of the FAST input file is shown in fig. 5.6.

5.7 Interfacing FAST with MATLAB

The FAST code was interfaced with MATLAB for easier integration with optimization routines. FAST needs a *.fst file as an input (refer fig. 5.6). This *.fst file contains the geometry parameters such as blade length, tower radius etc., as well as the configuration parameters such as analysis mode and simulation time, generator configuration etc. The optimization algorithm in MATLAB passes on design variables at every iteration, which are then written to the *.fst file. The FAST exe is then run for each new input file using MATLAB system command. The typical output message generated by FAST is shown in Fig. 5.5.
Figure 5.5: Output window of FAST.exe called through MATLAB system command
Figure 5.6: Input file for FAST executable
Chapter 6

Results and Discussion

In this chapter, two case studies are reported. First case study is about the co-design solution for wind turbine design optimization, wherein the observed results are really encouraging. The integrated plant and control design approach is shown to yield much better performance in terms of power production. Second case study is about the application of theory developed in chapter 4, to the general co-design problems involving non-linear dynamics. The optimization method of non-linear dynamic systems using surrogate models of derivative functions is demonstrated by two simple 2-dimensional examples. Finally the method is extended to the wind turbine co-design problem and it is shown to be helpful in significantly reducing the computational expense.

6.1 Wind Turbine Co-Design

In this section the power output maximization problem for wind turbine is solved for each of the three formulations described in section 3, and the results are reported. Fig. 6.1 shows the input wind profile for which the system was designed. Fig. 6.2 shows the comparison of power output obtained for each of the formulations. The nested and simultaneous approaches both result in the same trajectory, with a peak power of 10.84 kW, which is approximately a 7% increase over the sequential approach (10.14 kW of peak power output). The 7% power increase is very significant when designing higher-capacity wind turbines. This increase in power output can be attributed to the fact that co-design formulations were able to harness the strong interdependence of power output on plant design variables as well as the optimal control torque, to arrive at truly system optimal solution.

The plant design variables dictate the size of wind turbine, which has
Table 6.1: Optimal plant design vector for each of the formulations

<table>
<thead>
<tr>
<th>Variable</th>
<th>Sequential</th>
<th>Nested</th>
<th>Simultaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$, m</td>
<td>3.75</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>$R_h$, m</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>$H_t$, m</td>
<td>30.00</td>
<td>35.95</td>
<td>35.94</td>
</tr>
<tr>
<td>Max Power, kW</td>
<td>10.14</td>
<td>10.84</td>
<td>10.83</td>
</tr>
</tbody>
</table>

direct impact on power production capability, but bigger turbines are more likely to violate structural constraints. When we solve the co-design problem, the control torque (resistance torque) applied on the rotor, not only helps in maintaining optimal tip speed ratio, but also results in rotor speed reduction. This reduced rotor speed helps in reducing the structural deflections and allows the plant design optimization (outer loop in case of nested formulation) to change plant design so that peak power is optimized. The exploitation of this synergy is only possible if we consider the co-design approach.

Fig. 6.4 illustrates the trajectories of optimal tip-speed ratio for each of the formulations, allowing comparison and extraction of additional insights. It can be observed that in case of sequential design, the plant optimization problem was solved first and optimal control problem was employed to find the optimal tip-speed ratio trajectory for an already optimized plant. The optimal control problem had no flexibility in adjusting plant design to further increase peak power, whereas co-design (both nested and simultaneous) has the ability to: 1) maintain optimal tip-speed ratio and 2) modify the plant design at the same time, helping the optimization algorithm to identify a much better result in the sense of absolute output power.

Fig. 6.3 shows the blade-tip deflection in the flap-wise mode. In all the three formulations blade-tip deflection constraint was observed to be active. Finally, the optimal plant design vector obtained for each of the formulations is listed in Table 6.1 and bounds imposed on the plant design vector and constraints are listed in Table 6.2.
Figure 6.1: Input wind profile

Figure 6.2: Power output for each of the formulations
Figure 6.3: Blade flap-wise tip deflection

Figure 6.4: Optimal tip-speed ratio over time
Table 6.2: Bounds on constraints and design variables for wind turbine co-design problem

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_b^\text{max}$</td>
<td>0.09 m</td>
<td></td>
</tr>
<tr>
<td>$\delta_{\xi_1}^\text{max}$</td>
<td>0.01 m</td>
<td></td>
</tr>
<tr>
<td>$\delta_{\xi_2}^\text{max}$</td>
<td>0.01 m</td>
<td></td>
</tr>
<tr>
<td>$\mathbf{x}_l$</td>
<td>[2,0.1,30]T</td>
<td>m</td>
</tr>
<tr>
<td>$\mathbf{x}_u$</td>
<td>[8,0.5,50]T</td>
<td>m</td>
</tr>
</tbody>
</table>

6.2 Surrogate Modeling based Design Optimization

In this section, the surrogate modeling design method is demonstrated using three example problems that involve nonlinear dynamics. The first two examples are simple analytical problems with a known solution, enabling a demonstration of method validity. The third example is a wind turbine co-design problem, where the state derivatives are calculated by a computationally expensive analysis function FAST [53]. In all three cases a surrogate model is developed for the derivative function with dependence on both state and design variables, enabling efficient solution of the co-design problem.

Example 1: 2D Nonlinear System

Consider the following co-design problem that is based on a second-order non-linear dynamic system:

$$\begin{align*}
\min_{[a,b,u]} & \quad J = (a^2 + b^2) + \int_{t=0}^{10} (\xi_1^2(t) + u^2(t))dt \\
\text{s.t.} & \quad \dot{\xi}_1 = -a\xi_1 + b^3\xi_2(t) \\
& \quad \dot{\xi}_2 = b\xi_1 - 2a^3\xi_2(t) - \xi_1^2(t) + \xi_1^3(t) + u(t).
\end{align*}$$

(6.1)

This example is based on a problem presented in ref. [54], and was extended to co-design problem formulation. Here $\mathbf{x}_p = [a, b]^T$ are plant design variables, and $\mathbf{\xi}(t) = [\xi_1(t), \xi_2(t)]^T$ are the state variables. Figure 6.5 illustrates the solution of this example based on actual system dynamics. Compare
Table 6.3: Example 1: Optimal plant design vector

<table>
<thead>
<tr>
<th>Variable</th>
<th>Non-Surrogate Approach</th>
<th>Surrogate Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.6706</td>
<td>0.7085</td>
</tr>
<tr>
<td>b</td>
<td>-1.1249</td>
<td>-1.4603</td>
</tr>
</tbody>
</table>

this to the results given in Fig. 6.6 that are based on the surrogate modeling solution process. They are identical within the specified surrogate model tolerance limit (0.001) and are qualitatively similar. The solution to this co-design problem also involves the optimal plant design, described in Table 6.3, for both the surrogate and non-surrogate (optimization with original dynamics) design approaches. Since we approximate state derivatives using surrogate models, the number of original state derivative function evaluations needed to reach optimal solution were 160 (far fewer than the 792 evaluations required without surrogate modeling). This highlights the value of the surrogate modeling approach in cases where derivative function evaluations dominate solution expense.

Example 2: 2D Nonlinear System

The second illustrative example presented here is also based on a nonlinear dynamic system presented in [54], extended to a co-design formulation:

\[
\begin{align*}
\min_{[a, b, u]} \quad & J = (a^2 + b^2) + \int_{t=0}^{10} (\xi_1'(t)^2 + \xi_2'(t)^2 + u(t)^2) \, dt \\
\text{s.t.} \quad & \dot{\xi}_1(t) = -0.84\xi_1(t) - a\xi_2(t) - b\xi_1(t)\xi_2(t) \\
& \dot{\xi}_2(t) = 0.54\xi_1(t) + a\xi_2(t) + b\xi_1(t)\xi_2(t) + u(t).
\end{align*}
\]

(6.2)

As in the first example, \( x_p = [a, b]^T \) are the plant design variables and \( \xi(t) = [\xi_1(t), \xi_2(t)]^T \) are the system state variables. Figure 6.7 shows the solution of above co-design problem using actual system dynamics, and Fig. 6.8 illustrates the solution based on surrogate modeling. As with the first example, the solution obtained using surrogate modeling is identical to the solution of the original problem within the provided model tolerance limit. The optimal plant design results are presented in Table 6.4. This example required 240 original derivative function evaluations for surrogate modeling approach, which is again far fewer than the 9,517 evaluations required for
Figure 6.5: Example 1 solution with non surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control.
Figure 6.6: Example 1 solution with surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control.
Table 6.4: Example 2: Optimal plant design vector

<table>
<thead>
<tr>
<th>Variable</th>
<th>Non-Surrogate Approach</th>
<th>Surrogate Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.4616</td>
<td>0.4947</td>
</tr>
<tr>
<td>b</td>
<td>-0.2504</td>
<td>-0.2279</td>
</tr>
</tbody>
</table>

the direct approach.

Example 3: Wind Turbine Design

The surrogate modeling based co-design process was also used to solve the simultaneous structural and control design problem for a horizontal axis wind turbine [55].

\[
\begin{align*}
\min_{x=[x_p, \tau(t)]^T} & \quad w_1 m_s(x_p) + w_2 \int_0^{t_f} (\tau(t) - \tau_{opt}(t))^2 dt \\
\text{s.t.} & \quad x_p \geq 0 \\
& \quad \|\xi_1\|_{\infty} - \xi_{1\text{max}} \leq 0 \\
& \quad \|\xi_2\|_{\infty} - \xi_{2\text{max}} \leq 0 \\
& \quad P_m(v_{\text{rated}}) - P_{\text{rated}} \geq 0 \\
& \quad \dot{\xi} = f(\xi, x_p) + Bu
\end{align*}
\]

The objective here is to maximize the power extraction from the lightest (minimal mass \(m_s(\cdot)\)) wind turbine for given input wind conditions, while satisfying the structural deflection constraints on states \(\xi_1\) and \(\xi_2\) (i.e., the tower aft-fore bending and blade out-of-plane bending values, respectively). The constraint \(P_m(v_{\text{rated}}) - P_{\text{rated}} \geq 0\) ensures that the wind turbine generates the full rated power when wind is blowing at rated speed \(v_{\text{rated}}\). The power capture maximization can be achieved by minimizing the deviation of rotor control torque \(\tau(t)\) from the optimal torque \(\tau_{opt}(t)\) required to track the locus of maximum power coefficient [56]. The state space model of this system is highly nonlinear in nature, and is based on the state derivative calculations available through FAST [53]. It is important to note that these derivative function evaluations require seconds to evaluate, meaning that simulation based on direct derivative function evaluation is much slower than real time, making co-design problems impractical to solve without surrogate modeling.

Here, \(w_1 > 0\) and \(w_2 > 0\) are the weights on structural design (mass)
Figure 6.7: Example 2 Solution with non surrogate approach (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control.
Figure 6.8: Example 2 Solution with surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control.
Table 6.5: Optimal plant design vector

<table>
<thead>
<tr>
<th>Variable</th>
<th>Non Surrogate Approach</th>
<th>Surrogate Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blade Radius</td>
<td>56.93 m</td>
<td>57.91 m</td>
</tr>
<tr>
<td>Tower outer diameter</td>
<td>5.00 m</td>
<td>5.00 m</td>
</tr>
<tr>
<td>Tower wall thickness</td>
<td>0.03 m</td>
<td>0.03 m</td>
</tr>
<tr>
<td>Tower height</td>
<td>70.00 m</td>
<td>70.00 m</td>
</tr>
<tr>
<td>Blade hub radius</td>
<td>0.96 m</td>
<td>0.95 m</td>
</tr>
<tr>
<td>Objective Function</td>
<td>$1.95 \times 10^{10}$</td>
<td>$1.98 \times 10^6$</td>
</tr>
</tbody>
</table>

Table 6.6: Solution characteristics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Non Surrogate Approach</th>
<th>Surrogate Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of original derivative evaluations</td>
<td>25160</td>
<td>2800</td>
</tr>
<tr>
<td>Solution Time</td>
<td>419 mins</td>
<td>124 mins</td>
</tr>
</tbody>
</table>

and control design ($\int_0^T L(\cdot) dt$) objective function terms, respectively. The control design term approximates power production. This design problem was solved using wind profiles obtained at SITE-05730, Indiana, USA. This wind profile, shown in Fig. 6.9(a), was obtained for a 24 hour duration averaged over 7 days. The optimal plant design vector for this problem is listed in Table 6.5, and the optimal torque trajectories are illustrated in Fig. 6.9. Solution statistics are provided in Table 4. The number of original derivative function evaluations and overall solution time are both reduced significantly when using the surrogate modeling based approach, indicating that this method is a promising approach for the design of nonlinear dynamic systems.

6.3 Discussion

This thesis presented a novel approach for optimizing wind turbine design using a simultaneous co-design method to achieve system optimal solutions. Solution of this problem provides significant insight via exploration of design alternatives that are overlooked when using conventional sequential design. The co-design approach presented here (nested & simultaneous) is a balanced formulation in which the deeper treatment is provided to plant model. This
Figure 6.9: Example 3 Solution: (a) Input wind speed (b) Optimal torque trajectory using non surrogate approach (c) Optimal torque trajectory using surrogate approach
type of balanced approach to co-design of engineering systems is a promising step towards enabling more meaningful solutions to multidisciplinary optimization problems for dynamic systems [2]. This approach is in contrast to most of the optimal control formulations in research community which consider a simplistic plant model for control implementation. The exploitation of the synergy between plant and control design to get system optimal solutions was demonstrated through this work.

This work also proposed a novel and efficient approach for solving co-design problems that involve nonlinear dynamic systems. Previous studies have incorporated surrogate modeling in the solution of nonlinear dynamic system design problems by constructing a surrogate model based on the entire simulation, treating the system analysis as a black-box. In the new approach presented here, surrogate models are constructed only for derivative functions (often the most computationally intensive component). This approach also has the advantage of capitalizing on the intrinsic properties of dynamic systems by retaining the use of simulation. Surrogate modeling of dynamic systems introduces several interesting challenges, including how to construct and validate surrogate models that must be accurate within the region of a trajectory instead of a point. We have demonstrated the potential of this new method in solving computationally–overwhelming nonlinear dynamic system design problems, many of which right now are impractical to solve using established methods if high-fidelity models of complete system dynamics are to be employed. This article also illustrated the use of direct transcription in solving co-design problems, an emerging area of MDSDO. Three example problems were used to demonstrate how to efficiently utilize surrogate models of derivative functions in co-design problems. Two were simple analytical problems, whereas the third was a high-fidelity wind turbine design problem that would be impractical to solve using conventional techniques.

This work is an important component of advancing the field of multidisciplinary dynamic system design optimization (MDSDO) [23], which often involves computationally expensive dynamic system simulations. Here a new way of using surrogate modeling methods was demonstrated, that capitalizes on the unique properties of dynamic systems to enable efficient solution. Often the derivative function calculations dominate computational expense for high-fidelity models of dynamic systems, and the method introduced here can reduce dramatically the number of expensive original derivative calculations,
accounting for all sample points required for surrogate model construction and validation. This preliminary work opens the door to a wide range of further research topics. The near term plan for the future work includes, investigation of improved validation methods (including validation domain description), efficient resampling techniques, and extension to fully nonlinear systems, i.e., $\dot{\xi}(t) = f(\xi(t), x_p, u(t))$, wherein the control is not assumed to be affine ($Bu(t)$) in the system dynamics.
References


