Design of Nonlinear Dynamic Systems Using Surrogate Models of Derivative Functions

Conference Paper · August 2013

DOI: 10.1115/DETC2013-12262

2 authors:

Anand Deshmukh
Dr. Babasaheb Ambedkar Marathwada University

James Allison
University of Illinois, Urbana-Champaign

Proceedings of IDETC/CIE 2013
the ASME 2013 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
August 4–7, 2013, Portland, Oregon, USA

DETC2013–12262

DESIGN OF NONLINEAR DYNAMIC SYSTEMS USING SURROGATE MODELS OF DERIVATIVE FUNCTIONS

Anand P. Deshmukh
University of Illinois at Urbana-Champaign
Urbana, IL 61801
Email: adeshmu2@illinois.edu

James T. Allison*
University of Illinois at Urbana-Champaign
Urbana, IL, 61801
Email: jtalliso@illinois.edu

ABSTRACT

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 article, 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 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 inner loop solves the system optimization problem for a given derivative function surrogate model, and the outer loop updates the surrogate model based on optimization results. This solution approach presents unique challenges. For example, 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. This method is demonstrated using two simple design examples, followed by a wind turbine design problem. In the last example, system dynamics are modeled using a linear state-dependent model where updating the system matrix based on state and design variable changes is computationally expensive.

1 Introduction

Optimal design of nonlinear (or linear state-dependent) dynamic systems is a challenging task, due in part to the computationally expensive system simulations involved. Moreover, nonlinear dynamic system design problems often exhibit tight coupling between physical system (plant) and control system design, requiring the use of integrated plant and control design methods to arrive at system-optimal solutions [1]. Integrated dynamic system design methods are often referred to as 'co-design' methods. In co-design formulations, plant and control design are considered simultaneously to solve an integrated optimization problem [2, 3]. Applying co-design to a nonlinear dynamic system (e.g., a multibody dynamic system) often incurs significant computational expense [4].

Many solutions have been proposed for reducing the computational expense of dynamic system design, including the use of surrogate models. The most widely-used surrogate modeling approach for simulation-based dynamic system design treats the simulation as a 'black-box', where a surrogate model is constructed based on inputs to and outputs from the simulation. That is, if $\mathbf{u}(t)$ are inputs to a simulation, and if $\mathbf{y}(t)$ are the corresponding outputs, a set of $\{\mathbf{u}(t), \mathbf{y}(t)\}$ pairs is generated via

*Address all correspondence to this author.

simulation. This data is then used to construct a nonlinear surrogate model with established data fitting techniques such as neural networks, radial basis networks, echo state networks, or wavelet networks [5,6]. The resulting surrogate model is computationally inexpensive to evaluate, supporting the use of optimization in dynamic system design. Challenges to consider when using surrogate modeling for dynamic system design include the expense of the initial sampling required to build the surrogate model, as well as the accuracy of the surrogate model at points between sample points. Applications of this general strategy have included multibody dynamic system optimization [4, 7], nonlinear dynamic biological systems [8], aircraft wing design with uncertain parameters [9], and vehicle ride comfort [10].

Surrogate modeling approaches used so far employ generalized functions to approximate the input–output mapping $\mathbf{u}(t) \longrightarrow \mathbf{y}(t)$. This approach does not exploit the special properties of a dynamic system, such as the continuous evolution of state trajectories, to improve model accuracy between sample points. The method proposed in this article retains the use of simulation to maintain the unique characteristics of dynamic systems, enhancing accuracy between sample points and reducing surrogate model complexity.

An important relationship exists between surrogate modeling of dynamic systems and the well-established field of reduced-order modeling (ROM). A reduced-order model of a dynamic system approximates a higher-fidelity dynamic model using fewer states. Model reduction reduces computational expense, making dynamic system design more tractable. The reduced-order approach is based on projecting the dynamical system onto sub-spaces consisting of basis elements that contain characteristics of the expected solution. This approach helps models retain dynamic properties that correlate with the physical characteristics of the systems they approximate [11], whereas the conventional surrogate modeling approaches described above use generalized approximation functions not based on system physics.

Reduced-order modeling often is applied to high-order models of systems governed by partial differential equations (PDEs), as these models are especially expensive to simulate [11, 12]. Several classes of model reduction methods have been investigated, including those based on proper orthogonal decomposition (POD) [13], Krylov subspace methods [14], and the block Arnoldi algorithm [15]. Modal analysis has been used extensively for understanding the dynamic behavior of structures [16]. These model reduction methods, however, are largely restricted to linear systems. Notable application of reduced-order modeling to linear dynamical systems include microelectromechanical systems [15] and fluid systems [12, 17, 18]. The ability to simulate nonlinear systems is vital for progress in dynamic system design. While some model reduction methods have been extended to nonlinear systems (e.g., POD for nonlinear structural dynamics [19]), the use of model reduction as a general tool for design of nonlinear systems is limited.

Both ROM methods and conventional surrogate modeling have been used in the design of dynamic systems, but each approach has limitations. Surrogate models of black-box simulations use generalized approximation functions that do not leverage the intrinsic properties of dynamic systems, reducing the potential model accuracy for a given number of samples, and obstructing insights about the underlying system. While model reduction approaches retain simplified dynamic system properties, the underlying dynamics may be oversimplified in some cases [14], and higher-fidelity models are still required at later design stages [20]. In addition, reduced-order modeling applies largely only to linear dynamic systems, and ROMs often do not accommodate the direct modification of independent physical system design variables. As a result, reduced-order modeling is not suitable for the co-design of nonlinear dynamic systems, which is the focal-point of this article.

In this paper we propose a unique design process that addresses these above issues by approximating only the state derivatives of a nonlinear dynamic system using surrogate models, rather than treating the whole system simulation as a black-box. This provides direct access to approximated system dynamics, and preserves the dynamic nature of the system model to improve approximation accuracy and efficiency. This method applies to nonlinear derivative functions. In addition, derivative function surrogates are constructed that depend both on state and design variables, enabling the solution of nonlinear co-design problems.

2 Design Process

The process presented here is similar to previous studies (see [21, 22, 23] for reviews) in that it is a method for obtaining optimal design solutions through the use of surrogate mathematical models that approximate physics-based system models. 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 [24]. The computational expense of obtaining samples required to build the surrogate model must be accounted for when determining whether a surrogate modeling approach is a good choice for a particular problem [25].

Successful surrogate modeling methods support the rapid identification of an accurate optimum design point with a minimum number of high-fidelity function evaluations [26]. Accuracy can be preserved by using a trust region approach [27, 28], 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) [29, 30]. 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 [31], and extension of surrogate modeling to multi-objective optimization problems where high accuracy is maintained in regions near the Pareto front [32].

While in many cases surrogate modeling has been applied to a single engineering discipline at a time [32] (e.g. structural design [33], multibody dynamic systems [34], design based on aerodynamics and aero-acoustics [17]), it can be extended to multidisciplinary problems [35]. Co-design problems are multidisciplinary design optimization problems that involve the coupled physical and control system design disciplines [36]. 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 article introduces one possible approach for tackling this challenge associated with co-design problems.

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

$$\begin{aligned} \min_{\mathbf{x_p},\mathbf{u}(t)} \quad & J = \int_0^{t_F} L(\boldsymbol{\xi}(t),\mathbf{x_p},\mathbf{u}(t))dt \\ \text{s.t.} \quad & \mathbf{g}(\boldsymbol{\xi}(t),\mathbf{x_p}) \le \mathbf{0} \\ & \mathbf{h}(\boldsymbol{\xi}(t),\mathbf{x_p}) = \mathbf{0} \\ & \dot{\boldsymbol{\xi}}(t) = \mathbf{f}(\boldsymbol{\xi}(t),\mathbf{x_p}) + \mathbf{B}\mathbf{u}(t). \end{aligned} \tag{1}$$

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 $\mathbf{x_p}$ and $\mathbf{u}(t)$, respectively, and $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$ are the inequality and equality design constraints, respectively. Note that design constraints depend indirectly on control design since $\mathbf{u}(t)$ influences state trajectories $\boldsymbol{\xi}(t)$. This problem structure allows for bi-directional plant-control design coupling [37]. This formulation admits nonlinear system dynamics, i.e. the state derivatives $\dot{\boldsymbol{\xi}}(t)$ are nonlinear functions of states and physical design. The scope of this article is limited to systems that depend linearly on control $\mathbf{u}(t)$.

The core contribution of this paper is centered on efficient approximation methods for the derivative function $\mathbf{f}(\cdot)$. We seek to construct a surrogate model $\hat{\mathbf{f}}(\cdot)$ of $\mathbf{f}(\cdot)$ based on sampling in both the state and design spaces. Equation (2) illustrates an approximate system dynamics model based on the surrogate model $\hat{\mathbf{f}}(\cdot)$:

$$\dot{\boldsymbol{\xi}}(t) \approx \hat{\mathbf{f}}(\boldsymbol{\xi}(t),\mathbf{x_p}) + \mathbf{B}\mathbf{u}(t) \tag{2}$$

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

$$\begin{aligned} \min_{\mathbf{x_p},\mathbf{u}(t)} \quad & J = \int L(\boldsymbol{\xi}(t),\mathbf{x_p},\mathbf{u}(t))dt \\ \text{s.t.} \quad & \mathbf{g}(\boldsymbol{\xi}(t),\mathbf{x_p}) \le \mathbf{0} \\ & \mathbf{h}(\boldsymbol{\xi}(t),\mathbf{x_p}) = \mathbf{0} \\ & \dot{\boldsymbol{\xi}}(t) \approx \hat{\mathbf{f}}(\boldsymbol{\xi}(t),\mathbf{x_p}) + \mathbf{B}\mathbf{u}(t) \end{aligned} \tag{3}$$

The design method proposed here consists of an inner loop that solves Prob. (3), and an outer loop that iteratively enhances the surrogate model. The method consists of the following five steps:

1. Define the sampling domain in state space and design space
2. Sample test points in the combined state and design spaces
3. Build and validate the state derivative surrogate model
4. Solve the co-design problem
5. Check accuracy and convergence requirements, repeat steps 1–4 until requirements are satisfied

This iterative process is illustrated in Fig. 1, and described in detail in the following subsections.

2.1 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) [23].

2.2 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

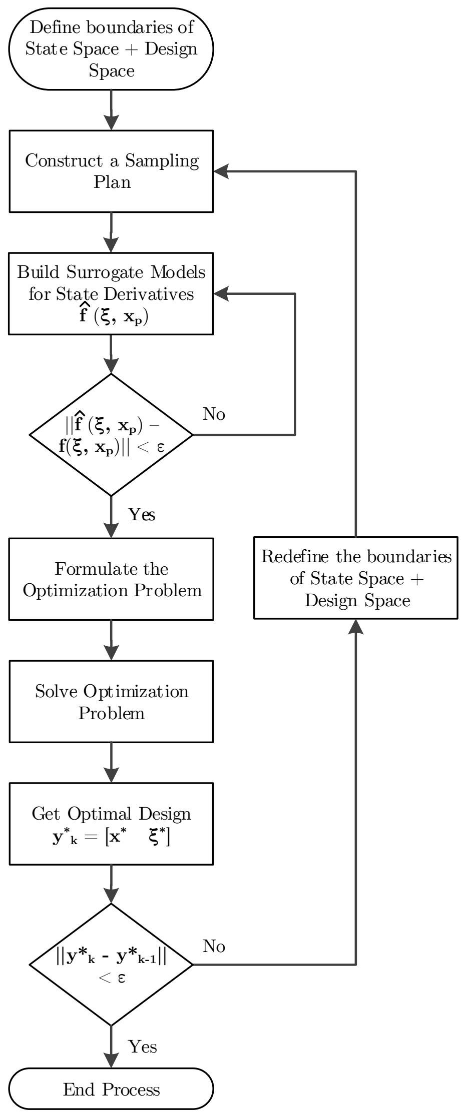

Figure 1. Surrogate modeling based optimization process for dynamic systems

jth state derivative ($j = 1, 2, 3, \ldots, n$). Figure 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 function that governs system dynamics: $\mathbf{f}(\boldsymbol{\xi}(t), \mathbf{x_p})$. A training point consists of values for both state and design variables, i.e., $\mathbf{y}_i = [\boldsymbol{\xi}, \mathbf{x_p}]$, and the output of the derivative function is vector-valued, so each training point produces multiple (n) observed outputs. Once the observed outputs ($\mathbf{f}(\mathbf{y}_i)$) are obtained, the input–output pairs may be used

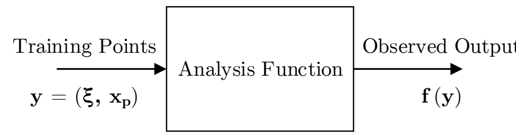

Figure 2. Evaluation of sample training points

to 'train' the surrogate model for each of the state derivatives. The surrogate model used here employs Radial Basis Functions (RBFs) [38]. For each of the n state derivatives, we write the interpolation condition using p training points:

$$f_j(\mathbf{y}_i) = \sum_{k=1}^{p} w_k{}^j \psi(\|\mathbf{y}_i - \mathbf{c}_k\|), \quad \text{for} \quad \begin{array}{c} i = 1,\, 2,\, \ldots,\, p, \\ j = 1,\, 2,\, \ldots,\, n \end{array}. \tag{4}$$

where $\psi(\cdot)$ is the radial basis function, $w_k{}^j$ are unknown weighting coefficients, n is the number of states, and $\mathbf{c}_k$ is the k^{th} basis function center. The specific RBF used here is the thin plate spline function [23]: $\psi(r) = r^2 \ln r$, where r is the Euclidean distance between the training point and function center: $r = \|\mathbf{y}_i - \mathbf{c}_k\|$. The objective in constructing the surrogate model is to find the coefficients $w_k{}^j$. This can be done by solving the following equation for $\mathbf{w}^j = [w_1{}^j,\, w_2{}^j,\, \ldots,\, w_p{}^j]^T$:

$$\boldsymbol{\psi} \mathbf{w}^j = \mathbf{f}_j, \tag{5}$$

where $\boldsymbol{\psi}$ is the 'Gram matrix' [23]: $\boldsymbol{\psi}_{i,k} = \psi(\|\mathbf{y}_i - \mathbf{c}_k\|)$ for $i,k = 1,\, 2,\, \ldots\, p$, and $\mathbf{f}_j = [f_j(\mathbf{y}_1),\, f_j(\mathbf{y}_2),\, \ldots,\, f_j(\mathbf{y}_p)]^T$ is the vector of observed outputs for jth 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., $\mathbf{c}_i = \mathbf{y}_i$. This simplification provides reasonably accurate results for the case studies presented in this article.

2.3 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. 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) [39, 40], 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) [41, 42]. 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 ($\mathbf{y}_{\mathbf{s}i}$) is defined as the 2-norm of difference between the actual state derivative vector $\mathbf{f}(\mathbf{y}_{\mathbf{s}i})$ and prediction of state derivative vector $\hat{\mathbf{f}}(\mathbf{y}_{\mathbf{s}i})$:

$$SNE = \sum_{i=1}^{n_s} \|\hat{\mathbf{f}}(\mathbf{y}_{\mathbf{s}i}) - \mathbf{f}(\mathbf{y}_{\mathbf{s}i})\|, \tag{6}$$

where n_s is the number of test points. As illustrated in Fig 1, model error is checked before proceeding with co-design solution. Alternative approaches may include validation checks after co-design solution.

2.4 Co-Design Formulation

The primary motivation for the surrogate modeling approach presented here is the efficient solution of co-design problems that involve nonlinear dynamic systems with computationally expensive derivative functions. The formulation of two important co-design techniques is reviewed here.

2.4.1 Nested Co-Design Allison and Herber [37] identified the nested co-design formulation as a special case of the Multidisciplinary Design Feasible (MDF) formulation. This formulation incorporates two loops: an outer loop optimizes the plant design, and an inner loop solves the optimal control problem for each plant design tested by the outer loop. The outer loop formulation is:

$$\begin{aligned} \min_{\mathbf{x_p}} \quad & \phi_*(\mathbf{x_p}) \\ \text{s.t.} \quad & \mathbf{g_p}(\mathbf{x_p}) \leq \mathbf{0}, \\ & \mathbf{h_p}(\mathbf{x_p}) = \mathbf{0}, \end{aligned} \tag{7}$$

where $\mathbf{x_p}$ is the plant design vector, $\mathbf{g_p}(\cdot)$ and $\mathbf{h_p}(\cdot)$ are the plant design constraints, and $\phi_*(\cdot)$ is an optimal value function that depends only on $\mathbf{x_p}$. 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 design $\mathbf{u}_*(t)$ and returns the corresponding objective function value. The inner-loop formulation is:

$$\begin{aligned} \min_{\mathbf{u}(t)} \quad & \int L(\boldsymbol{\xi}(t), \mathbf{x_p}, \mathbf{u}(t)) \\ \text{s.t.} \quad & \mathbf{g_p}(\boldsymbol{\xi}(t), \mathbf{x_p}, \mathbf{u}(t)) \leq \mathbf{0} \\ & \mathbf{h_p}(\boldsymbol{\xi}(t), \mathbf{x_p}, \mathbf{u}(t)) = \mathbf{0} \\ & \dot{\boldsymbol{\xi}}(t) - \mathbf{f}(\boldsymbol{\xi}(t), \mathbf{x_p}) - \mathbf{Bu}(t) = \mathbf{0}. \end{aligned} \tag{8}$$

The plant design $\mathbf{x_p}$ is specified by the outer loop and is held fixed during the inner loop solution. Plant design constraints $\mathbf{g_p}(\cdot)$ and $\mathbf{h_p}(\cdot)$are imposed in both loops to ensure system-level design feasibility. The presence of inequality path constraints in the inner loop problem motivates the use of 'discretize–then–optimize' optimal control methods that can manage these constraints (e.g., direct transcription, which is discussed later in this section).

2.4.2 Simultaneous Co-Design An alternative co-design formulation solves for the plant and control design variables simultaneously:

$$\begin{aligned} \min_{\mathbf{x_p}, \mathbf{u}(t)} \quad & J = \int L(\boldsymbol{\xi}(t), \mathbf{x_p}, \mathbf{u}(t)) dt \\ \text{s.t.} \quad & \mathbf{g}(\boldsymbol{\xi}(t), \mathbf{x_p}) \leq \mathbf{0} \\ & \mathbf{h}(\boldsymbol{\xi}(t), \mathbf{x_p}) = \mathbf{0} \\ & \dot{\boldsymbol{\xi}}(t) = \mathbf{f}(\boldsymbol{\xi}(t), \mathbf{x_p}) + \mathbf{Bu}(t) \end{aligned} \tag{9}$$

As with nested co-design, this simultaneous formulation accounts for all dynamic system interactions and plant-control design coupling, resulting in a system-optimal design that is often significantly better than conventional sequential (plant design followed by control design) methods. This simultaneous co-design approach is used in the investigation presented here.

2.5 Direct Transcription

Conventional optimal control methods based on Pontryagin's Maximum Principle [43] 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 converse approach: the optimal control problem is discretized first, and the resulting nonlinear program (NLP) is solved using a standard NLP algorithm [44, 45]. 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 [46]. A DT co-design formulation based on this work, using the trapezoidal method, follows:

$$\begin{aligned} \min_{\mathbf{y}=\mathbf{x_p},\Xi,\mathbf{U}} \quad & \sum_{i=1}^{n_t-1} L(\mathbf{x_p},\boldsymbol{\xi}_i,\mathbf{u}_i)h_i \\ \text{subject to:} \quad & \boldsymbol{\zeta}(\mathbf{x_p},\mathbf{U},\Xi)=\mathbf{0} \\ & \mathbf{g_p}(\mathbf{x_p},\Xi)\leq\mathbf{0}, \end{aligned} \tag{10}$$

where, n_t is the number of time steps, $\Xi = \left[\boldsymbol{\xi}_1^{\mathrm{T}},\, \boldsymbol{\xi}_2^{\mathrm{T}},\, \cdots\, \boldsymbol{\xi}_{n_t}^{\mathrm{T}}\right]^{\mathrm{T}}$ is the matrix of discretized state variables (row i, or $\boldsymbol{\xi}_i$, corresponds to the state vector at time t_i), $\boldsymbol{\zeta}(\cdot)$ are the defect constraint functions imposed to ensure that Ξ satisfies system state equations, $\mathbf{U} = \left[\mathbf{u}_1^{\mathrm{T}},\, \mathbf{u}_2^{\mathrm{T}},\, \cdots,\, \mathbf{u}_n^{\mathrm{T}}\right]^{\mathrm{T}}$ is the control input matrix, and h_i is the ith time step size. This DT-based co-design approach was used to solve Prob. (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 $\boldsymbol{\varphi}(\cdot)$ is used in the calculation of the defect constraints $\boldsymbol{\zeta}(\cdot)$. The NLP defined in Prob. (10) was solved using the *fmincon* algorithm in MATLAB®. An important advantage of DT to emphasize here is its parallel nature; all defect constraints are independent, enabling massively parallel implementations.

3 Numerical Results

The surrogate modeling design method is demonstrated here 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 using FAST, a high-fidelity (and computationally expensive) model developed by the U.S. National Renewable Energy Laboratory [47]. 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.

Table 1. First example: Optimal plant design vector

Variable	Non-Surrogate Approach	Surrogate Approach
a	0.6706	0.7085
b	-1.1249	-1.4603

Example 1: 2D Nonlinear System

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

$$\begin{aligned} \min_{[a,b,u]} \quad & J=(a^2+b^2)+\int_{t=0}^{10}({\xi_1}^2(t)+u^2(t))dt \\ \text{s.t.} \quad & \dot{\xi}_1 = -a\xi_1 + b^3\xi_2(t) \\ & \dot{\xi}_2 = b\xi_1 - 2a^3\xi_2(t) - {\xi_1}^2(t) + {\xi_1}^3(t) + u(t). \end{aligned} \tag{11}$$

This example is based on a problem presented in Ref. [48], and was extended here to correspond to a co-design problem formulation where optimization is performed with respect both to control input and time-independent variables. Here $\mathbf{x_p} = [a,\, b]^T$ are plant design variables, and $\boldsymbol{\xi}(t) = [\xi_1(t),\, \xi_2(t)]^T$ are the state variables. Figure 3 illustrates the solution of this example based on actual system dynamics. Compare this to the results given in Fig. 4 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 1, 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 Ref. [48], extended to a co-design formulation:

$$\begin{aligned} \min_{[a,b,u]} \quad & J=(a^2+b^2)+\int_{t=0}^{10}({\xi_1}^2(t)+{\xi_2}^2(t)+u^2(t))dt \\ \text{s.t.} \quad & \dot{\xi}_1(t) = -0.84\xi_1(t) - a\xi_2(t) - b\xi_1(t)\xi_2(t) \\ & \dot{\xi}_2(t) = 0.54\xi_1(t) + a\xi_2(t) + b\xi_1(t)\xi_2(t) + u(t). \end{aligned} \tag{12}$$

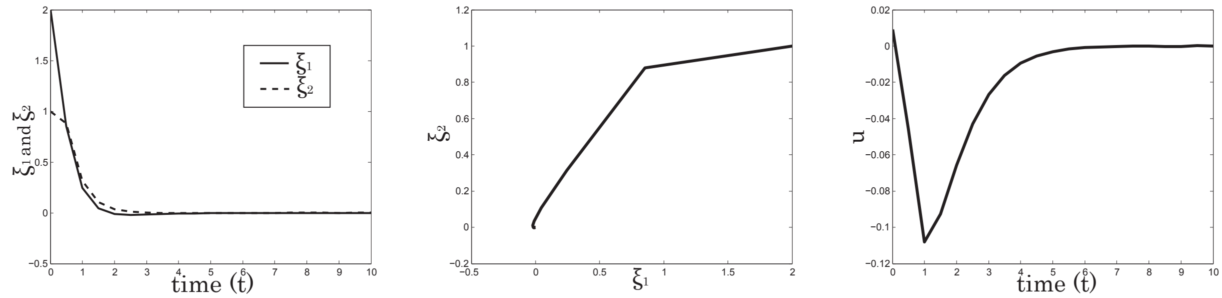

Figure 3. Example 1 solution with non surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control.

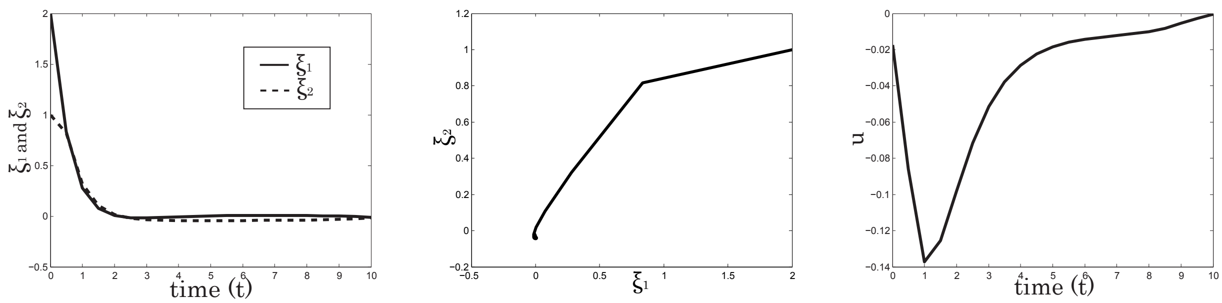

Figure 4. Example 1 solution with surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control.

As in the first example, $\mathbf{x_p} = [a,\ b]^T$ are the plant design variables, and $\boldsymbol{\xi}(t) = [\xi_1(t),\ \xi_2(t)]^T$ are the system state variables. Figure 5 shows the solution of above co-design problem using actual system dynamics, and Fig. 6 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 2. This example required 240 original derivative function evaluations for surrogate modeling approach, which is again far fewer than the 9,517 evaluations required for the direct approach.

Table 2. Second example: Optimal plant design vector

Variable	Non-Surrogate Approach	Surrogate Approach
a	0.4616	0.4947
b	-0.2504	-0.2279

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 [49].

$$\begin{aligned}
\min_{\mathbf{x}=[\mathbf{x_p},\tau(t)]^T} \quad & w_1 m_s(\mathbf{x_p}) + w_2 \int_0^{t_f} (\tau(t) - \tau_{opt}(\boldsymbol{\xi}, t))^2 dt \\
\text{s.t.} \quad & \mathbf{x_p} \geq 0 \\
& \|\boldsymbol{\xi}_1\|_\infty - \xi_{1max} \leq 0 \\
& \|\boldsymbol{\xi}_2\|_\infty - \xi_{2max} \leq 0 \\
& P_m(\mathrm{v}_{rated}) - P_{rated} \geq 0 \\
& \dot{\boldsymbol{\xi}}(t) = \mathbf{f}(\boldsymbol{\xi}(t), \mathbf{x_p}) + \mathbf{B}\mathbf{u}(t)
\end{aligned} \tag{13}$$

The objective here is to maximize the power extraction and minimize structural mass ($m_s(\mathbf{x_p})$) for given input wind conditions, while satisfying the structural deflection constraints on states ξ_1 and ξ_2 (i.e., the tower aft-fore bending and blade out-of-plane bending values, respectively). The constraint $P_m(\mathrm{v}_{rated}) - P_{rated} \geq 0$ ensures that the wind turbine generates the full rated power when wind is blowing at rated speed v_{rated}. The power capture maximization can be achieved by minimizing the deviation of rotor control torque $\tau(t)$ from the optimal torque $\tau_{opt}(\boldsymbol{\xi}, t)$ required to track the locus of maximum power coefficient [50]. The state space model of this system is highly nonlinear in nature, and is based on the state derivative calculations available through FAST [47]. 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 when using the original derivative function directly.

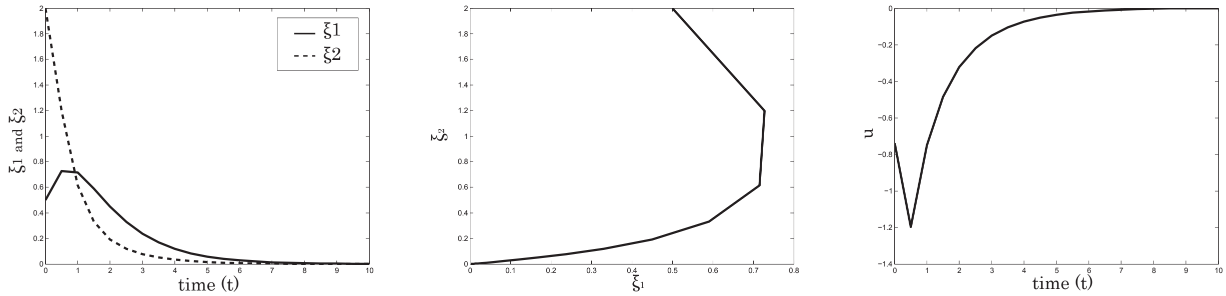

Figure 5. Example 2 Solution with non surrogate approach (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control.

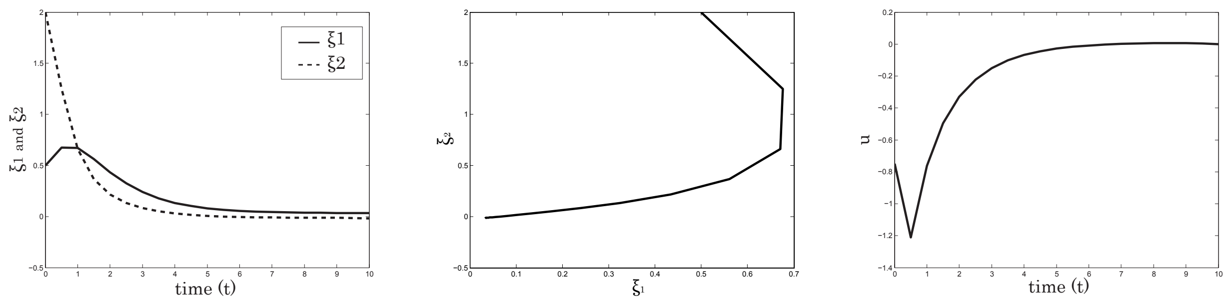

Figure 6. Example 2 Solution with surrogate approach: (a) Optimal state trajectories (b) Optimal phase plot (c) Optimal control.

Here, $w_1 > 0$ and $w_2 > 0$ are the weights on structural design (mass) and control design ($\int_0^{t_f} L(\cdot)dt$) 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 [51]. This wind profile, shown in Fig. 7(a), was obtained for a 24 hour duration averaged over 7 days. The optimal plant design vector for this problem is listed in Table 3, and the optimal torque trajectories are illustrated in Fig. 7. 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.

Table 3. Optimal plant design vector

Variable	Standard Approach	Surrogate Approach
Blade Radius	56.93 m	57.91 m
Tower outer diameter	5.00 m	5.00 m
Tower wall thickness	0.03 m	0.03 m
Tower height	70.00 m	70.00 m
Blade hub radius	0.96 m	0.95 m
Objective Function	1.95×10^{10}	1.98×10^{6}

Table 4. Solution characteristics

Parameter	Standard Approach	Surrogate Approach
No. of original derivative evaluations	25160	2800
Solution Time	419 mins	124 mins

4 Discussion

The work reported in this article is an important component of advancing the field of multidisciplinary dynamic system design optimization (MDSDO) [37], which often involves computationally expensive dynamic system simulations. Here we demonstrated a new way of using surrogate modeling methods 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. In

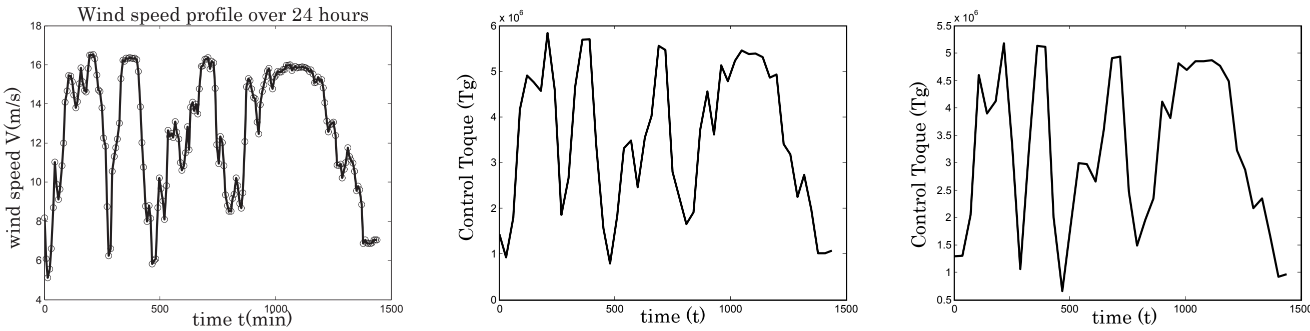

Figure 7. Example 3 Solution: (a) Input wind speed (b) Optimal torque trajectory using non surrogate approach (c) Optimal torque trajectory using surrogate approach

the near term we plan to investigate improved validation methods (including validation domain description), efficient resampling techniques, and extension to fully nonlinear systems, i.e., $\dot{\boldsymbol{\xi}}(t) = \mathbf{f}(\boldsymbol{\xi}(t), \mathbf{x_p}, \mathbf{u}(t))$

5 Conclusion

In this article we 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.

REFERENCES

[1] Roos, F., 2007. "Towards a methodology for integrated design of mechatronic servo systems". Phd thesis, Royal Institute of Technology.

[2] Li, Q., Zhang, W., and Chen, L., 2001. "Design for control - a concurrent engineering approach for mechatronic systems design". *IEEE/ASME Transaction on Mechatronics,* **6**(2), pp. 161–169.

[3] Fathy, H., Reyer, J., Papalambros, P., and Ulsoy, A. G., 2001. "On the coupling between the plant and controller optimization problems". In 2001 American Control Conference, IEEE, pp. 1864–1869.

[4] Eberhard, P., Dignath, F., and Kubler, L., 2003. "Parallel evolutionary optimization of multibody systems with application to railway dynamics". *Multibody System Dynamics,* **9**, pp. 143–164.

[5] Sjoberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P., Hjalamarsson, H., and Juditsky, A., 1995. "Nonlinear black-box modeling in system identification: a unified overview". *Automatica,* **31**(12), pp. 1691–1724.

[6] Pan, Y., and Wang, J., 2012. "Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks". *IEEE Transactions on Industrial Electronics,* **59**(8), pp. 3089–3101.

[7] Bestle, D., and Eberhard, P., 1992. "Analyzing and optimizing multibody systems". *Mech. Struct. & Mach,* **20**(1), pp. 67–92.

[8] Rodriguez-Fernandez, M., Egea, J., and Banga, J., 2006. "Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems". *BMC Bioinformatics,* **7**(483).

[9] Chowdhury, R., and Adhikari, S., 2012. "Fuzzy parametric uncertainty analysis of linear dynamical systems: A surrogate modeling approach". *Mechanical Systems and Signal Processing*(32), pp. 7–12.

[10] Goncalves, J., and Ambrosio, J., 2003. "Optimization of vehicle suspension systems for improved comfort of road vehicles using flexible multibody dynamics". *Struct Multidisc Optim*(34), pp. 113–131.

[11] Hinze, M., and S.Volkwein, 2010. "Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control". *Lecture Notes in Computational Science and Engineering,* **45**, pp. 261–306.

[12] Lang, Y., Malacina, A., Biegler, L., Munteanu, S., Madsen, J., and

Zitney, S., 2009. "Reduced order model based on principal component analysis for process simulation and optimization". *Energy & Fuels*(23), pp. 1695–1706.

[13] Agarwal, A., Biegler, L., and Zitney, S., 2009. "Simulation and optimization of pressure swing adsorption systems using reduced-order modeling". *Ind. Eng. Chem. Res.*(48), pp. 2327–2343.

[14] Bai, Z., 2002. "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems". *Applied Numerical Mathematics*(43), pp. 9–44.

[15] Han, J., Rudnyi, E., and Korvink, J., 2005. "Efficient optimization of transient dynamic problems in mems devices using model order reduction". *J. Micromech. Microeng.,* **15**, pp. 822–832.

[16] Hurty, W., 1965. "Dynamic analysis of structural systems using component modes". *AIAA Journal,* **3**(4), pp. 678–685.

[17] Jouhaud, J.-C., Sagaut, P., and Montagnac, M., 2007. "A surrogate-model based multidisciplinary shape optimization method with application to a 2d subsonic airfoil". *Computers & Fluids*(36), pp. 520–529.

[18] Akhtar, I., Borggaard, J., and Burns, J., 2010. "High Performance Computing for Energy Efficient Buildings". In Proceedings of the 8th International Conference on Frontiers of Information Technology, ACM.

[19] Kerschen, G., Golinval, J., Vakakis, A., and Bergman, L., 2005. "The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview". *Nolinear Dynamics*(41), pp. 147–169.

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

[21] Queipo, N., Haftka, R., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., 2005. "Surrogate-based analysis and optimization". *Progress in Aerospace Sciences*(41), pp. 1–28.

[22] Wang, G., and Shan, S., 2007. "Review of Metamodeling Techniques in Support of Engineering Design Optimization". *Journal of Mechanical Design,* **129**(4), pp. 370–380.

[23] Forrester, A., Sobester, A., and Keane, A., 2008. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons Ltd., UK.

[24] Venter, G., Haftka, R., and Starnes, J., 1998. "Construction of Response Surface Approximations for Design Optimization". *AIAA Journal,* **36**, pp. 2242–2249.

[25] Crary, S., 2002. "Design of Computer Experiments for Metamodel Generation". *Analog Integrated Circuits and Signal Processing,* **32**, pp. 7–16.

[26] Rasmussen, J., 1998. "Nonlinear Programming by Cumulative Approximation Refinement". *Structural and Multidisciplinary Optimization,* **15**, pp. 1–7.

[27] Rodríguez, J. F., Renaud, J., and Watson, L., 1998. "Trust Region Augmented Lagrangian Methods for Sequential Response Surface Approximation and Optimization". *Journal of Mechanical Design,* **120**(1), pp. 58–66.

[28] Alexandrov, N., Dennis, J., Lewis, R., and Torczon, V., 1998. "A Trust-Region Framework for Managing the Use of Approximation Models in Optimization". *Structural and Multidisciplinary Optimization,* **15**, pp. 16–23.

[29] Trosset, M., and Torczon, V., 1997. Numerical Optimization Using Computer Experiments. Tech. rep., NASA ICASE Report No. 97-38.

[30] Booker, A., Dennis, J., Frank, P., Serafini, D., Torczon, V., and Trosset, M., 1999. "A Rigorous Framework for Optimization of Expensive Functions by Surrogates". *Structural and Multidisciplinary Optimization,* **17**, pp. 1–13.

[31] Goel, T., Haftka, R., Shyy, W., and Queipo, N., 2007. "Ensemble of surrogates". *Struct Multidisc Optim*(33), pp. 199–216.

[32] Messac, A., and Mullur, A., 2008. "A computationally efficient metamodeling approach for expensive multiobjective optimization". *Optim Eng*(9), pp. 37–67.

[33] Sakata, S., Ashida, F., and Zako, M., 2008. "Approximate structural optimization using kriging method and digital modeling technique considering noise in sampling data". *Computers & Structures*(86), pp. 1477–1485.

[34] Lin, Y., Haftka, R., Queipo, N., and Fregly, B., 2010. "Surrogate articular contact models for computationally efcient multibody dynamic simulations". *Medical Engineering & Physics*(32), pp. 584–594.

[35] Simpson, T., Booker, A., Ghosh, D., Giunta, A., Koch, P., and Yang, R.-J., 2004. "Approximation Methods in Multidisciplinary Analysis and Optimization: a Panel Discussion". *Structural and Multidisciplinary Optimization,* **27**, pp. 302–313.

[36] Allison, J. T., Guo, T., and Han, Z., in review. "Co-design of an active suspension using simultaneous dynamic optimization". *Journal of Mechanical Design*.

[37] Allison, J., and Herber, D., 2013. "Multidisciplinary design optimization for dynamic engineering systems". *AIAA Journal*. conditionally accepted.

[38] Mullur, A., and Messac, A., 2005. "Extended radial basis functions: More flexible and effective metamodeling". *AIAA Journal,* **43**(6), pp. 1306–1315.

[39] Tax, D., and Duin, R., 1999. "Support Vector Domain Description". *Pattern Recognition Letters,* **20**(11), pp. 1191–1199.

[40] Alexander, M. J., Allison, J. T., Papalambros, P. Y., and Gorsich, D. J., 2011. "Constraint management of reduced representation variables in decomposition-based design optimization". *Journal of Mechanical Design,* **133**(10), p. 101014.

[41] Jin, R., Chen, W., and Simpson, T., 2001. "Comparative studies of metamodelling techniques under multiple modelling criteria". *Structural and Multidisciplinary Optimization,* **23**(1), pp. 1–13.

[42] Hussain, M., Barton, R., and Joshi, S., 2002. "Metamodeling: Radial basis functions versus polynomials". *European Journal of Operational Research,* **138**(1), pp. 142–154.

[43] Pontryagin, L. S., 1962. *The Mathematical Theory of Optimal Processes*. Interscience.

[44] Betts, J. T., 2010. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, Philadelphia.

[45] Biegler, L. T., 2010. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM.

[46] Allison, J., and Han, Z., 2011. "Co-Design of an Active Suspension Using Simultaneous Dynamic Optimization". In the Proceedings of the 2011 ASME Design Engineering Technical Conference.

[47] Jonkman, J., and Buhl, M., 2005. "Fast user's guide". *Technical*

Report NREL/EL-500-38230, August.
[48] Matallana, L., Blanco, A., and Bandoni, J., 2011. "Nonlinear dynamic systems design based on the optimization of the domain of attraction". *Mathematical and Computer Modelling*(53), pp. 731–745.
[49] Deshmukh, A., and Allison, J., 2013. "Simultaneous structural and control system design for horizontal axis wind turbines". In Proceedings of 9th Multidisciplinary Design Optimization Specialist Conference, AIAA.
[50] Thiringer, T., and Linders, J., 1993. "Control by variable rotor speed of a fixed-pitch wind turbine operating in a wide speed range". *IEEE Transactions on Energy Conversion,* **8**(3), pp. 520–526.
[51] Brower, M., 2009. "Development of eastern regional wind resource and wind plant output datasets". *Subcontract Report NREL/SR-550-46764*, December.