Co-Design of an Active Suspension Using Simultaneous Dynamic Optimization

James T. Allison[1]
Mem. ASME
Assistant Professor
University of Illinois at Urbana-Champaign,
Urbana, IL 61801
e-mail: jtalliso@illinois.edu

Tinghao Guo
Graduate Student
University of Illinois at Urbana-Champaign,
Urbana, IL 61801
e-mail: guo32@illinois.edu

Zhi Han
Mem. ASME
Senior Software Developer
MathWorks, Inc.,
Natick, MA 01760
e-mail: zhi.han@mathworks.com

Design of physical systems and associated control systems are coupled tasks; design methods that manage this interaction explicitly can produce system-optimal designs, whereas conventional sequential processes may not. Here, we explore a new technique for combined physical and control system design (co-design) based on a simultaneous dynamic optimization approach known as direct transcription, which transforms infinite-dimensional control design problems into finite-dimensional nonlinear programming problems. While direct transcription problem dimension is often large, sparse problem structures and fine-grained parallelism (among other advantageous properties) can be exploited to yield computationally efficient implementations. Extension of direct transcription to co-design gives rise to new problem structures and new challenges. Here, we illustrate direct transcription for co-design using a new automotive active suspension design example developed specifically for testing co-design methods. This example builds on prior active suspension problems by incorporating a more realistic physical design component that includes independent design variables and a broad set of physical design constraints, while maintaining linearity of the associated differential equations. A simultaneous co-design approach was implemented using direct transcription, and numerical results were compared with conventional sequential optimization. The simultaneous optimization approach achieves better performance than sequential design across a range of design studies. The dynamics of the active system were analyzed with varied level of control authority to investigate how dynamic systems should be designed differently when active control is introduced. [DOI: 10.1115/1.4027335]

1 Introduction

Often dynamic systems developed by engineers employ electronic control. Successful design of controlled systems is becoming increasingly important as their ubiquity and complexity rises. In conventional design processes, the physical system is designed first, followed by control system design [1–4]. This sequential approach does not fully account for coupling between physical artifact and control system design, and produces suboptimal results [5,6]. More effective multidisciplinary design methods that manage the artifact-control coupling explicitly are being developed and are often termed *co-design* methods.

Engineers often employ simulation to predict dynamic system performance and support design decisions. Simulation can be used with numerical optimization algorithms to aid effective exploration of design alternatives. Typically, a nested approach is used: the optimization algorithm proposes a candidate design, which is then tested via simulation, and the results are then used by the optimization algorithm in finding a new candidate design. Alternative approaches that perform simulation and optimization simultaneously are explored here, specifically a family of methods known as direct transcription (DT).

DT has been employed successfully for control design and parameter estimation across a wide range of applications from chemical process design [7,8] to aerospace trajectory optimization [9–11], to walking dynamics [12], and to epidemiology [13] but has not yet been applied to co-design. In this article, we introduce an extension of DT for co-design, explore the associated advantages and challenges, and demonstrate this approach using a newly developed active suspension system example. DT is demonstrated to be an effective approach for co-design when accounting for more realistic physical system design considerations. An initial set of studies is also performed that comprise a first step toward understanding how the physical aspects of a system should be designed differently when transitioning to actively controlled systems.

1.1 Direct Transcription.

In this section, the essential background for direct transcription, including its application to optimal control, will be provided to facilitate an understanding of its extension to co-design (i.e., integrated physical and control system design). Here, it is assumed that the system to be designed can be modeled using a system of continuous differential equations

$$\dot{\boldsymbol{\xi}} = \mathbf{f_d}(\boldsymbol{\xi}(t), \mathbf{u}(t), t) \tag{1}$$

$$\mathbf{0} = \mathbf{f_a}(\boldsymbol{\xi}(t), \mathbf{u}(t), t) \tag{2}$$

$$\mathbf{0} = \mathbf{f_b}(\boldsymbol{\xi}(t_\mathrm{F}), \mathbf{u}(t_\mathrm{F}), t_\mathrm{F}) \tag{3}$$

Equation (1) defines the time derivatives of a system model, where $\boldsymbol{\xi}(t)$ (length n_s) is the vector of state variables, $\mathbf{u}(t)$ (length n_c) is the control input, and $\mathbf{f_d}(\cdot)$ (length n_s) is the time derivative function. Equation (2) is the algebraic or path constraint, and if present, the model is a system of differential-algebraic equations (DAEs) [14]. The system may also have a boundary condition enforced at the simulation end time t_F (Eq. (3)). A simulation of the system based on these equations will produce the state variable trajectories that describe the state of the system through time. If dynamic response is important to system performance, these state trajectories are essential for predicting system utility.

If the model consists only of Eq. (1) with initial conditions $\boldsymbol{\xi}(t_0) = \boldsymbol{\xi}_0$, it can be discretized in time solved using a numerical

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received February 25, 2013; final manuscript received March 23, 2014; published online June 2, 2014. Assoc. Editor: Shinji Nishiwaki.

method for simulating ordinary differential equations (ODEs), such as the first-order Runge–Kutta method (forward Euler method)

$$\boldsymbol{\xi}_i = \boldsymbol{\xi}_{i-1} + h_i \mathbf{f_d}(\boldsymbol{\xi}_{i-1}, \mathbf{u}(t_{i-1}), t_{i-1}) \tag{4}$$

Here, $h_i = t_i - t_{i-1}$ is the integration step size at time t_i and $i = 1, 2, \ldots, n_t$ is the time step index. Higher-order Runge–Kutta methods improve accuracy for a given step size, as do implicit methods that employ an inner iterative method to solve for $\boldsymbol{\xi}_i$ at each time step.

The open-loop optimal control design problem can be written as

$$\min_{\boldsymbol{\xi}(t), \mathbf{u}(t), t_F} \quad J = \phi(\boldsymbol{\xi}(t), \mathbf{u}(t), t_F) \tag{5}$$

subject to Eqs. (1)–(3), where J is the response of a cost function. This problem is called open loop because the control input is specified directly and is independent of state, whereas in a feedback control system the control input depends on system state. The cost function used here is based on a standard form used in optimal control because of its ability to quantify the impact of time-dependent system behavior on overall system performance. If the primary utility of a system is to perform a task (or set of tasks) where dynamic performance is important, this type of cost function can capture the impact of both physical system and control system design changes on system performance.

If the cost function takes the standard form

$$J = \eta(\dot{\boldsymbol{\xi}}(t_F), \boldsymbol{\xi}(t_F), \mathbf{u}(t_F), t) + \int_{t_0}^{t_F} L(\dot{\boldsymbol{\xi}}(t), \boldsymbol{\xi}(t), \mathbf{u}(t), t) dt \tag{6}$$

it is known as a Bolza objective, where $\eta(\cdot)$ is known as the terminal or Mayer term, and $L(\cdot)$ is the running cost or Lagrange term. For example, Eq. (38) utilizes a Lagrange term to quantify handling and comfort characteristics of the vehicle in the case study presented in this article. Here, we posit that this form of objective function can be used as a single system-wide objective function that applies to both physical and control system design considerations.

The final simulation time t_F that appears in Eqs. (5) and (6) often is an important quantity, particularly if the cost function involves a Mayer term that depends on the terminal state. Final time might also be treated as a fixed parameter. When t_F is an optimization variable as in Eq. (5), and when a discretized solution implementation is used, the number of time steps can be fixed, and the time each step h_i represents is scaled according to the value of t_F.

The problem formulation given in Eq. (5) is based on open-loop control, i.e., feedback is not utilized (control input $\mathbf{u}(t)$ is independent of state). One advantage of using open-loop optimal control methods for early-stage design exploration is that the assumption of a specific control architecture is not required, thus, supporting the exploration of upper system performance limits. Once a particular control architecture is selected, performance and exploration may be restricted. Feedback control (i.e., control input depends on observed state) can be accommodated by replacing $\mathbf{u}(t)$ with control design variables $\mathbf{x_c}$ that parametrize a feedback control system. Time-independent parameters $\mathbf{p}$ can also be added to the set of optimization variables for parameter estimation problems

$$\min_{\boldsymbol{\xi}(t), \mathbf{x_c}, \mathbf{p}} \quad J = \phi(\boldsymbol{\xi}(t), \mathbf{x_c}, \mathbf{p}) \tag{7}$$

Optimal control problems may be solved using techniques based on the calculus of variations, including classical optimal control methods such as those for optimal linear quadratic regulators [15]. These variational or "indirect" methods take an "optimize-then-discrete" approach, where numerical methods are employed to solve the DAEs that arise from applying optimality conditions [16]. The classical approach, however, requires derivation of system Hamiltonian derivatives, so its applicability is limited. A common alternative approach eliminates $\boldsymbol{\xi}(t)$ from the optimization variable set by solving for state variable values using a simulation nested within the optimization problem

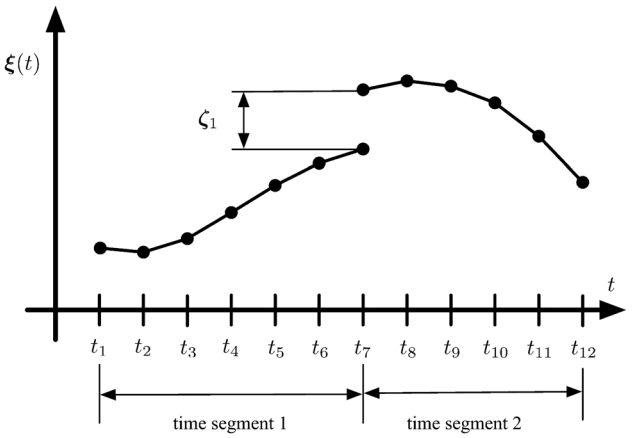

Fig. 1 Illustration of continuity defect ζ_1 between time segments 1 and 2 using the multiple shooting method

$$\min_{\mathbf{x_c}} \quad J = \phi(\Xi, \mathbf{x_c}) \tag{8}$$

where Ξ is a discretized representation of the state variables (the ith row of Ξ is $\boldsymbol{\xi}_i = \boldsymbol{\xi}(t_i)$, $i = 1, 2, \ldots, n_t$, and n_t is the number of time steps), and Ξ is solved for using forward simulation for every candidate control design $\mathbf{x_c}$. This nested approach enables use of finite-dimensional optimization methods, such as sequential quadratic programming (SQP) [17], and is implemented in commercial software [18–20], but has limitations. Discretization and variable step sizes yield a cost function that is not smooth or arithmetically consistent [11], i.e., the set of arithmetic operations used to compute cost changes with $\mathbf{x_c}$. Nonsmoothness can be managed by increasing optimization finite difference step sizes or by using gradient-free methods, but this approach may still suffer from numerical sensitivity, particularly for highly nonlinear or stiff dynamic systems. Numerical sensitivity can be reduced by partitioning the simulation into n_T time segments using a technique know as multiple shooting (MS). Each time segment contains multiple time steps. The state at the interfaces between time segments ($\mathbf{Y}$) is controlled directly by the optimization algorithm, and state continuity is enforced at convergence by defect constraints $\boldsymbol{\zeta}(\Xi, \mathbf{Y})$

$$\begin{aligned} &\min_{\mathbf{x_c}, \mathbf{Y}} \quad J = \phi(\Xi, \mathbf{x_c}) \\ &\text{s.t.} \quad \boldsymbol{\zeta}_i(\Xi, \mathbf{Y}) = \mathbf{0}, \quad i = 1, 2, \ldots n_T - 1 \end{aligned} \tag{9}$$

Row j of $\mathbf{Y}$ is the state at the beginning of time segment $j+1$. Within each time segment, the system is simulated, and defect constraints require the state at the end of time segment j to match the jth row of $\mathbf{Y}$. To clarify, in Fig. 1, the value for ξ_7 calculated using simulation over time segment 1 does not match the value for ξ_7 specified by $\mathbf{Y}$ that serves as the initial conditions for the simulation over time segment 2. The size of the gap at this interface is the value of the corresponding defect constraint ζ_1, and this gap is driven to zero by the optimization algorithm.

Because $\mathbf{Y}$ is specified by the optimization algorithm, the n_T simulations are independent and can be executed using coarse-

grained parallel computing. Shorter simulation segments reduce numerical sensitivity.

In the limit, as $n_T \to n_t - 1$, the simulation for each time segment collapses to a single difference equation. The set of defect constraints replace simulation completely, and $\mathbf{Y}$ expands into the full state matrix Ξ. This is DT; the infinite-dimensional optimal control problem is transcribed directly to a finite-dimension nonlinear programming problem

$$\begin{aligned} \min_{\Xi,\mathbf{x_c},t_F} \quad & J = \phi(\Xi, \mathbf{x_c}, t_F) \\ \text{s.t.} \quad & \zeta_i(\Xi, \mathbf{x_c}, t_F) = \mathbf{0} \\ & \hat{\mathbf{f}}_{\mathbf{a}i}(\Xi, \mathbf{x_c}, t_F) = \mathbf{0} \\ \text{where} \quad & i = 2, 3, \ldots, n_t - 1, n_t \end{aligned} \tag{10}$$

A defect constraint is defined for each time step, and path constraints are discretized ($\hat{\mathbf{f}}_{\mathbf{a}i}(\cdot)$). In some cases, it may be desirable for implementation convenience to include a defect constraint $\zeta_1(\cdot)$ for initial conditions and allow the first row of Ξ to represent system state at t_1 instead of t_2. The differential equations from the original optimal control problem are discretized and converted to a system of algebraic equations. This conversion requires the use of a collocation method, such as the trapezoidal method. This will be explained in detail shortly. Differential equations are also converted to algebraic equations via discretization when applying forward simulation (e.g., a Runge–Kutta method); in the special case of forward simulation, the resulting system of algebraic equations can be solved using successive substitution as the solver steps through time. In DT, the algebraic equations are instead solved simultaneously by the optimization algorithm, enabling the use of efficient collocation methods that do not support a forward simulation solution. In other words, the defect constraints in DT are analogous to the algebraic equations that arise when using forward simulation, but the type of algebraic equations that can be used with DT is much more general than with forward simulation. If the DT defect constraints are satisfied, then the underlying physics of the system are satisfied approximately within the accuracy of the collocation method and the system model.

The formulation defined in Eq. (10) allows for variable t_F, and $\mathbf{x_c}$ can be replaced easily with discretized $\mathbf{u}(t)$ to solve open-loop optimal control problems. As $\max(h_i) \to 0$, the optimality conditions for the DT problem converge to the optimality conditions for the infinite-dimension optimal control problem given in Eq. (5).

While the dimension of the DT problem is large (Ξ has $n_t \cdot n_s$ elements), its problem structure presents important advantages. At every optimization iteration, Ξ is specified completely, so each of the $n_t - 1$ defect equations and path constraints are independent, enabling fine-grained parallel computing. The DT problem can be implemented such that the problem structure is nearly diagonal (completely diagonal in some cases), enabling sparse finite difference gradient evaluations with as few as three perturbations regardless of problem dimension [11,21]. The DT problem is arithmetically consistent, enhancing numerical properties. Often analytical derivatives can be derived, further improving solution speed and accuracy. Variable step and order techniques for DT are available to provide error control [9,11,22,23].

Direct transcription is especially useful for problems with path constraints as it avoids another level of nesting within simulation to solve algebraic constraints; it is capable of handling higher-index DAEs [22,24]. Inequality path constraints may be added directly, a problem element that often is impossible to solve using any other means [11]. The ability to manage nonlinear inequality constraints is a primary motivation for extending DT to co-design, as these constraints normally arise when realistic physical design considerations are included in co-design problems [25], and classical optimal control methods cannot manage general inequality constraints.

If Euler's method, defined in Eq. (4), is used to convert state equations to algebraic equations, the defect constraint functions become

$$\zeta_i(\Xi, \mathbf{x_c}) = \xi_i - \xi_{i-1} - h_i \mathbf{f_d}(\xi_{i-1}, \mathbf{x_c}, t_{i-1})$$

Implicit collocation methods for general nonlinear systems can be used with DT without requiring inner iterations to solve for the implicit state values. For example, the defect constraint functions for the implicit trapezoidal method are

$$\zeta_i(\Xi, \mathbf{x_c}) = \xi_i - \xi_{i-1} - \frac{h_i}{2}(\mathbf{f_d}(\xi_{i-1}, \mathbf{x_c}, t_{i-1}) + \mathbf{f_d}(\xi_i, \mathbf{x_c}, t_i))$$

ξ_i is known *a priori* (it is a row of Ξ), so defect equations may be evaluated without inner iteration as is required with forward simulation using implicit methods. This concept extends to high-order implicit methods such as Gauss-Legendre, Radau, and Lobatto collocation [16]. While these methods are very accurate, they typically are impractical for forward simulation. DT, however, specifies Ξ completely at each optimization iteration. The global (simultaneous) solution of DT defect constraints makes possible the use of higher-order implicit methods without the need for inner iteration loops [8,26,27]. Using higher-order methods reduces the number of time steps needed to maintain the required precision, consequently reducing DT problem dimension.

Helpful insights can be gained by considering the different paths through the combined state and design solution space traced by the conventional nested (Eq. (8)) and DT methods. Figure 2 is an abstraction of the design and state subspaces, each of which may have high cardinality.

The nested approach (often referred to as single shooting (SS)) begins with an initial design point $\mathbf{x}_0$, which might include physical system design variables $\mathbf{x_p}$ in addition to $\mathbf{x_c}$ if a co-design problem is being solved. Forward simulation is then used to solve for state values Ξ_0 that are consistent with physics. The optimization algorithm chooses a new point $\mathbf{x}_1$ with the objective of reducing J, and the process repeats until convergence to $\mathbf{x}_*, \Xi_*$. The nested method can move only in one subspace at a time, whereas DT can move in both simultaneously, tracing a more direct path to the solution. The ability to move in multiple subspaces simultaneously like this has been shown to enable identification of superior optimization solutions in some cases [28].

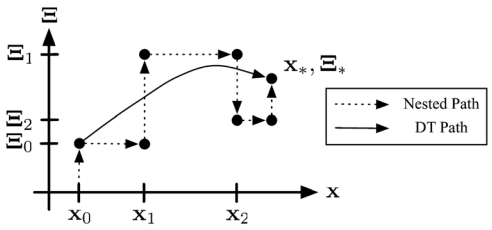

Fig. 2 Conceptual solution trajectories through design and state subspaces

1.2 Relationship to Multidisciplinary Design Optimization. In this section, the relationship between the three formulations for optimal control discussed above (nested or SS, MS, and DT) and single-level formulations for multidisciplinary design optimization (MDO) [28,29] is clarified. The nested solution approach in Eq. (8) can be viewed as a special case of the multidisciplinary feasible (MDF) formulation, where a complete system analysis is performed at every optimization iteration. In other words, the system is consistent with physics (i.e., feasible) during the entire optimization process because the differential equations are satisfied (approximately) at each optimization step. As with MDF, optimization variables for the nested approach involve only independent design variables, and the analysis task is consolidated into a single monolithic process.

Table 1 Relationship between direct transcription and MDO

	MDF/nested	IDF/MS	AAO/DT
Solution process	Entire analysis feasible at every optimization iteration	Individual analysis components feasible at every optimization iteration	Analysis feasible only at optimization convergence
Optimization variables	Design variables	Design and interface variables	Design and state variables
Analysis type	Consolidated analysis	Partially distributed analysis	Fully distributed analysis

The multiple shooting approach defined in Eq. (9) is a special case of the individual feasible formulation (IDF), where the analysis problem is partitioned into several smaller analysis tasks. Feasibility is maintained within analysis subproblems via simulation, but feasibility across subproblems is only guaranteed at optimization convergence by defect constraints. In addition to independent design variables, the set of optimization variables includes interface (coupling) variables $\mathbf{Y}$ that quantify interactions between analysis subproblems. Defect constraints ensure consistency across subproblems at convergence and correspond to auxiliary constraints in IDF. As a distributed analysis approach, multiple shooting supports coarse-grained parallel computing.

Direct transcription (Eq. (10)) is a special case of the all-at-once (AAO) formulation, where the optimization algorithm manages both design and all analysis tasks. Analysis equations are embedded in optimization equality constraints, and state variables are part of the optimization variable set. The defect constraints in DT are independent, enabling fine-grained parallel computing. These relationships are summarized in Table 1.

2 Direct Transcription for Co-Design

Co-design methods seek to identify system-optimal designs for controlled engineering systems by considering simultaneously physical artifact (plant) and control system design. Conventional sequential (plant then control) design processes produce suboptimal results because the two design domains are coupled. The sequential process is, in essence, a single iteration of block-coordinate descent [25,30], which even if iterated does not always guarantee system optimality.

It is the experience of the authors, through numerous interactions with engineers across the automotive and aerospace industries, that control engineers often struggle to meet system requirements when using sequential design due to the increasing complexity and demands associated with modern engineering systems. Additionally, many new system capabilities are possible only when plant and control design are considered together, including applications in areas such as automotive suspension systems [31], biomimetic robotic locomotion [32,33], controlled compliant mechanisms [34,35], and structural design [36,37].

New approaches are needed that preserve design freedom of both plant and control design domains throughout the process. Reyer et al. provided an early review of such strategies [2]. Fathy proposed a co-design technique where the control design problem is nested within the plant design problem, producing system-optimal results [38]. Allison and Nazari introduced a co-design method with distinct plant and control design subproblems that are guided toward system optimality using augmented Lagrangian coordination [39].

A fundamental challenge of solving co-design problems is managing a heterogeneous set of design variables; control variables either depend directly on time (e.g., $\mathbf{u}(t)$ in Eq. (5)) or are proxies for time dependent variables, while plant design variables are time-independent. Fathy's nested co-design approach manages this by employing established optimal control methods to solve the inner control problem and nonlinear programming algorithms to solve the outer plant design problem. Please note that Fathy's nested co-design method is different from the nested single-shooting method discussed above; the former nests optimal control within plant optimization, whereas the latter nests simulation within parameterized control design optimization. Allison and Nazari introduced an alternative approach where an extension of optimal control theory was used to account for time-independent linking variables in the control design subproblem [39]. Please see Ref. [25] for a more comprehensive review and classification of co-design methods.

Allison and Herber identified challenges that arise when treating physical design aspects of co-design problems at a practical level of sophistication [25]. Most previous co-design studies simplified physical design to the point where no physical system design constraints were employed beyond simple variable bounds, and dependent variables $\mathbf{p}$ were treated as independent optimization variables. This physical system design simplification has had several critical effects. First, classical optimal control methods may be applied to optimal control problems without inequality constraints, leading to the conclusion that these methods are suitable for co-design problems. Realistic co-design problems, however, have inequality constraints (e.g., stress or fatigue constraints), and alternative solution strategies must be investigated. The second effect of this simplification is that plant-control design coupling appears to be unidirectional. The assumption of unidirectionality has had a profound impact on recent developments in co-design theory (e.g., Refs. [39] and [40]). These recent advancements apply only to unidirectional co-design problems. Realistic treatment of plant design requires the inclusion of nonlinear inequality constraints, and these constraints often depend on state trajectories (e.g., fatigue depends on cyclic loading). Since state trajectories depend on control design, and physical constraints and design decisions depend on state trajectories, physical design, therefore, depends on control design, and co-design problems that treat physical design in a realistic way by nature have bidirectional coupling. As a result, recent developments in co-design that are based on assumptions of unidirectional coupling do not apply to co-design problems with more complete treatment of physical system design. The third effect of plant design simplification stems from the use of dependent variables as optimization variables. Not only does this influence problem structure in a profound way, but optimizing with respect to dependent variables can result in problem formulations that are not well-posed [41].

These three effects of deeper plant design treatment motivate the exploration of solution methods for optimal control that accommodate nonlinear inequality constraints, account for bidirectional coupling, and cope with problem structures associated with independent design variables. In this article, we present an extension of DT as a feasible strategy for solving co-design problems with deeper treatment of physical system design than in previous work. The suitability of this extension will be demonstrated using a case study of an active automotive suspension, and it will be shown that DT supports efficient solution of co-design problems. Using DT in co-design is just one element of a larger effort to move toward more balanced co-design strategies that support the consideration of practical physical design issues. For example, Deshmukh and Allison introduced a new method for dynamic system optimization, derivative function surrogate modeling, that has reduced the computational expense of co-design problems based on high-fidelity physical system models by up to an order of magnitude [42].

As described earlier in this section, a core challenge in co-design is the concurrent management of time-dependent and time-independent variables. While DT has been applied successfully to

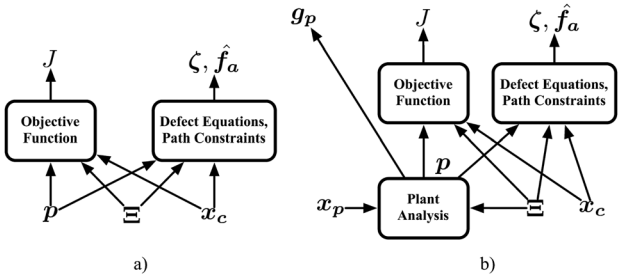

Fig. 3 Analysis structure of (*a*) DT and (*b*) DT for co-design

parameter estimation problems with time-independent variables [8,11,43], the structure of parameter estimation and co-design problems are very different. Problem structure can be analyzed using graph-theoretic techniques [44]. Figure 3 illustrates a simplified directed-graph view of the standard DT problem and DT augmented to handle co-design problems. The plant (physical system) analysis depends on both plant design variables and state variable values, and calculates plant design constraints $\mathbf{g_p}(\cdot)$ and intermediate variables ($\mathbf{p}$) used in computing the objective function and dynamic constraints. This problem structure accounts for bidirectional coupling between plant and control design (i.e., control design depends on plant design, and vice versa), which is a more general problem representation than previous co-design studies that account only for unidirectional coupling (control design depends on plant design). A detailed examination of DT co-design problem structure based on the active suspension case study is provided in Sec. 3 (please refer to Fig. 11).

Dependence of plant analysis on Ξ and influence on J, $\boldsymbol{\zeta}$, and $\mathbf{f_a}$ complicates DT implementation of co-design problems. This may preclude the use of analytical derivatives and impacts the density of the Jacobian matrix (complicating the application of large-scale optimization methods).

The DT formulation for co-design used here is

$$\begin{aligned} \min_{\Xi,\mathbf{x_c},\mathbf{x_p},t_F} \quad & J = \phi(\Xi, \mathbf{x_c}, \mathbf{x_p}, t_F) \\ \text{s.t.} \quad & \mathbf{g_p}(\Xi, \mathbf{x_p}) \le \mathbf{0} \\ & \boldsymbol{\zeta}_i(\Xi, \mathbf{x_c}, \mathbf{x_p}, t_F) = \mathbf{0} \\ & \mathbf{f_{ai}}(\Xi, \mathbf{x_c}, \mathbf{x_p}, t_F) = \mathbf{0} \\ \text{where} \quad & i = 1, 2, \ldots, n_t - 1, n_t \end{aligned} \tag{11}$$

Figure 4 illustrates possible constraint Jacobian structures (algebraic constraints omitted). Shaded areas indicate dependence of constraints on variables. Here, $\mathbf{x_c}$ is assumed to be a discretization of $\mathbf{u}(t)$. In both cases (a) and (b), defect constraints each depend only on a limited number of state and control variables. The diagonal structure for the defect constraints arises when state and control variables are ordered chronologically. This sparse diagonal structure allows the use of sparse finite differences to evaluate derivatives efficiently for those Jacobian elements.

Incorporating the plant design variables and constraints associated with co-design problems increases overall problem density. In general, plant and defect constraints may both depend on any plant design variable (indicated by the shading in the second columns in Fig. 4). In some problems, plant constraint dependence on state variables may be dense (e.g., upper shaded rectangle in Fig. 4(*a*), third column), increasing solution difficulty. Note that plant constraints do not depend directly on final time or control design in DT. Control or algebraic (path) constraints, if included, normally depend on both state and control design variables.

Clever problem formulation can improve problem sparsity, allowing the use of sparse finite differences to calculate additional elements of the constraint Jacobian. For example, consider the

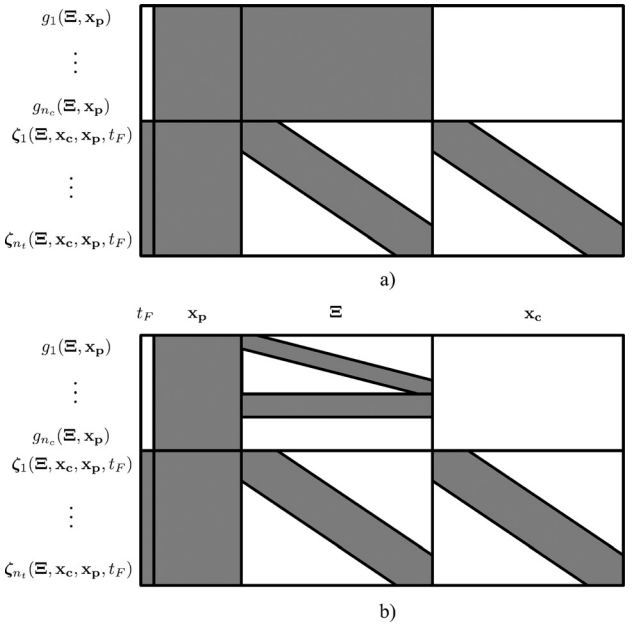

Fig. 4 DT Jacobian structure: (*a*) dense and (*b*) sparse plant constraints

simple path constraint $g(\Xi) = \max(\Xi_{\cdot j}) - \xi_{j\text{allow}} \le 0$, i.e., the jth state must remain less than $\xi_{j\text{allow}}$ at every time step. This is a very dense nonlinear constraint; it depends on all values of state j. We could instead formulate a linear constraint for each time step: $g_i(\Xi_{ij}) = \Xi_{ij} - \xi_{j\text{allow}} \le 0$, $i = 1, 2, \ldots, n_t$, resulting in a large number of linear constraints with a diagonal sparsity pattern. Even with the increased problem dimension, solution may be easier due to linear constraint sparsity. This type of reformulation will be demonstrated using the case study.

Figure 4(*b*) illustrates a DT co-design problem, where this type of sparsity is exploited for some of the plant constraints (diagonal pattern at the top of the third column). Other plant constraints may still indeed depend on states at every time step (the dense area below the diagonal area), and still other plant constraints may be completely static and independent of state values (the blank area). Even if a co-design problem is very dense, DT may prove advantageous due to its ability to manage singular optimal control problems, inequality path constraints, and support massive parallelization [11]. Problem structure details will be explicated further in the active suspension case study.

3 Active Suspension Case Study

In this section, we introduce a new model for an active suspension system—in fully reproducible detail—that includes a model of important physical system design considerations in addition to a dynamic model of the suspension. Effort was made to maintain linearity of system dynamics to preserve the usefulness of this model in other studies that are limited to linear time-invariant systems. Complete MATLAB® code for this case study is available at the URL in Ref. [45].

Consider the quarter-car model of a vehicle suspension illustrated in Fig. 5. The sprung (m_s) and unsprung (m_{us}) mass vertical positions are given by z_s and z_{us}, respectively. The system is excited by variations in road elevation z_0 as the vehicle travels at speed v. Two types of road inputs are used to test the suspension: a ramp input and a rough road input. Here we explore both the sequential and simultaneous approaches for designing this system using direct transcription.

The passive dynamic response of this system can be characterized by the following system of linear differential equations:

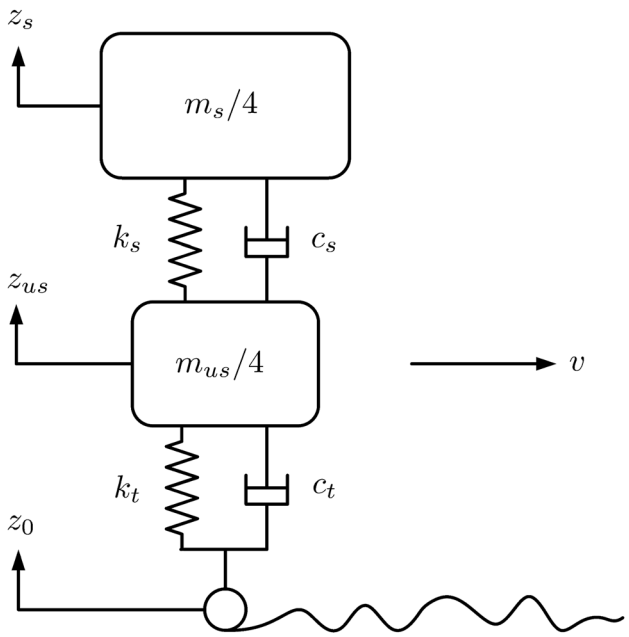

Fig. 5 Quarter-car vehicle suspension model

$$\dot{\boldsymbol{\xi}} = \mathbf{A}\boldsymbol{\xi} + \begin{bmatrix} -1 \\ \dfrac{4c_t}{m_{us}} \\ 0 \\ 0 \end{bmatrix} \dot{z}_0, \quad \text{where} \quad \boldsymbol{\xi} = \begin{bmatrix} z_{us} - z_0 \\ \dot{z}_{us} \\ z_s - z_{us} \\ \dot{z}_s \end{bmatrix} \quad \text{and}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\dfrac{4k_t}{m_{us}} & -\dfrac{4(c_s + c_t)}{m_{us}} & \dfrac{4k_s}{m_{us}} & \dfrac{4c_s}{m_{us}} \\ 0 & -1 & 0 & 1 \\ 0 & \dfrac{4c_s}{m_s} & -\dfrac{4k_s}{m_s} & -\dfrac{4c_s}{m_s} \end{bmatrix} \tag{12}$$

The tire and sprung mass spring stiffnesses are k_t (232.5×10^3 N/m) and k_s, respectively, and the tire and sprung mass damping rates are c_t and c_s, respectively. Here we assume $c_t = 0$. This canonical model has been used as an example in numerous design studies [46–49], including the design of active control systems where an additional control input term $\mathbf{B}u$ is appended to Eq. (12) [50–52]. This control input represents an actuator force between the sprung and unsprung masses. The parameters k_s and c_s have often been treated as independent design variables in past versions of this co-design problem [49,50,53] but are in fact dependent on geometric design and are subject to stress, fatigue, packaging, thermal, and other constraints. Here we introduce an extension to the basic quarter-car model that treats k_s and c_s as dependent variables and incorporates a plant model that computes stiffness and damping coefficients as a function of independent geometric spring and damper design variables. This changes co-design problem structure and is an important step toward co-design approaches that treat plant design with more realistic detail. The detailed spring and damper models are presented below, followed by a demonstration of active suspension co-design using DT.

3.1 Spring Design. The vehicle suspension in this model utilizes a helical compression spring with squared and ground ends (Fig. 6). The suspension has a "coil-over" configuration; the coil

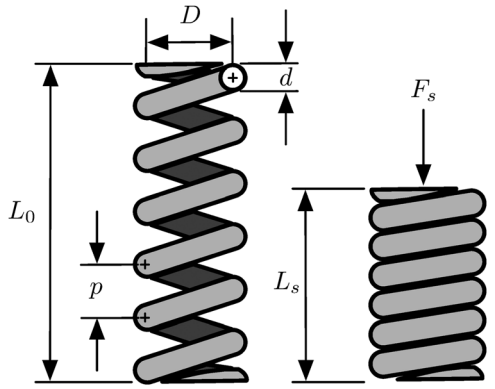

Fig. 6 Helical compression spring with squared ground ends

spring surrounds the damper and they are coaxial. The spring model presented here is derived from Ref. [54]. See also Refs. [55] and [56] for alternative spring design optimization formulations. The independent spring design variables here are the helix diameter D, wire diameter d, spring pitch p, and the number of active coils N_a (relaxed to a continuous variable). These four variables are part of the complete plant design vector $\mathbf{x_p}$. The formula for stiffness and a collection of spring design constraints are presented below.

The free length of the spring is $L_0 = pN_a + 2d$, and the solid height is $L_s = d(N_a + Q - 1)$, where $Q = 1.75$ for squared and ground ends. F_s is the axial force at the solid height, and the spring constant is

$$k_s = \frac{d^4 G}{8D^3 N_a \left(1 + \frac{1}{2C^2}\right)} \tag{13}$$

where G is the shear modulus (ASTM A401, $G = 77.2$ GPa) and $C = D/d$ is the spring index. Springs with $C < 4$ are difficult to form, and springs with $C > 12$ can tangle. These requirements provide our first two plant design constraints

$$g_1(\mathbf{x_p}) = 4 - C \leq 0 \tag{14}$$

$$g_2(\mathbf{x_p}) = C - 12 \leq 0 \tag{15}$$

The following constraint prevents buckling:

$$g_3(\mathbf{x_p}) = L_0 - 5.26D \leq 0 \tag{16}$$

The uncompressed spring must fit within the specified pocket length ($L_{0\max} = 0.40$ m) for the vehicle

$$g_4(\mathbf{x_p}) = L_0 - L_{0\max} \leq 0 \tag{17}$$

The outer spring diameter must not exceed $D_{o\max} = 0.25$ m to prevent interference with vehicle components

$$g_5(\mathbf{x_p}) = d + D - D_{o\max} \leq 0 \tag{18}$$

The spring inner diameter must be large enough to fit around the damper with at least $\delta_{dc} = 9.0$ mm clearance

$$g_6(\mathbf{x_p}) = d - D + D_p + 2(\delta_{dc} + t_d) \leq 0 \tag{19}$$

where D_p is the damper piston diameter and $t_d = 2.0$ mm is the damper wall thickness.

While suspension rattlespace (permissible peak-to-peak displacement [46]) is often treated as an objective function to avoid inequality constraints [47,50], it is more natural to formulate it as a constraint based on vehicle geometry. Here, peak suspension

displacement δ_{max} is calculated using a vehicle simulation with a ramp input (road grade of 25% at a speed of $v_1 = 10$ m/s) to check for maximum rattlespace violation

$$g_7(\mathbf{x_p}, \Xi) = \delta_{max} - L_0 + L_s + L_B + \delta_g \leq 0 \tag{20}$$

This test requires specification of damper design, discussed in Sec. 3.2. L_B is bumpstop thickness (0.02 m), and $\delta_g = m_s g/4k_s$ is the static suspension deflection ($g = 9.81$ m/s^2).

Spring shear stress τ at maximum deflection must not exceed shear yield stress S_{sy}. The following model of shear stress incorporates the Bergsträsser augmentation factor:

$$\tau = \left(\frac{4C+2}{4C-3}\right)\frac{8F_s D}{\pi d^3},$$

where the axial force at the spring solid height is calculated using $F_s = k_s(L_0 - L_s)$. The scaled stress constraint is

$$g_8(\mathbf{x_p}) = (\tau n_d - S_{sy})/S_{sy} \leq 0 \tag{21}$$

and a design factor of $n_d = 1.2$ is used. The shear yield strength is assumed proportional to the ultimate tensile strength for the spring material: $S_{sy} = 0.65 S_{ut}$, where $S_{ut} = A \times 10^6/(d^m)$, $A = 1974$ MPa mm^m, and $m = 0.108$ (note here d is measured in mm). To ensure spring linearity and validity of Eq. (12), the following constraint must be satisfied:

$$g_9(\mathbf{x_p}, \Xi) = 0.15 + 1 - \frac{L_0 - L_s}{\delta_g + 1.1\delta_{maxf}} \leq 0 \tag{22}$$

where $\delta_{max\,f}$ is the maximum spring deflection during a simulation of the quarter-car over a rough road (International Roughness Index (IRI) = 7.37 [49,57]) at a forward velocity of $v_2 = 20$ m/s. Under these rough road conditions, the maximum axial spring force is $F_{max} = k_s(\delta_{maxf} + \delta_g)$, and the minimum force is $F_{min} = k_s(\delta_g - \delta_{maxf})$. The mean axial force and force amplitude are $F_m = (F_{max} + F_{min})/2$ and $F_a = (F_{max} - F_{min})/2$, respectively. The mean shear stress and amplitude are

$$\tau_m = \left(\frac{4C+2}{4C-3}\right)\frac{8F_m D}{\pi d^3}, \quad \tau_a = \left(\frac{4C+2}{4C-3}\right)\frac{8F_a D}{\pi d^3}$$

Soderberg fatigue criterion with Zimmerli data is applied here

$$g_{10}(\mathbf{x_p}, \Xi) = \frac{\tau_a}{S_{e2}} + \frac{\tau_m}{S_{sy}} - 1 \leq 0, \quad \text{where} \quad S_{e2} = 0.24 S_{ut}/n_d \tag{23}$$

$$g_{11}(\mathbf{x_p}, \Xi) = \frac{\tau_a n_d - 241 \times 10^6}{241 \times 10^6} \leq 0 \tag{24}$$

Spring design parameters are summarized in Table 2.

3.2 Damper Design. Equation (12) assumes linear damping, i.e., the damping force F_D is proportional to damper piston velocity $\dot{\xi}_3$. Real suspension dampers are highly nonlinear due to both

Table 2 Spring design parameters

$m_s/4$	325 kg	$m_{us}/4$	65 kg
k_t	232.5×10^3 N/m	c_t	0
Q	1.75	G	77.2 GPa
L_{0max}	0.40 m	D_{omax}	0.25 m
δ_{dc}	9 mm	t_d	2.0 mm
l_B	0.02 m	n_d	1.2
A	1974 MPa mm^m	m	0.108
v_1	10 m/s	v_2	20 m/s

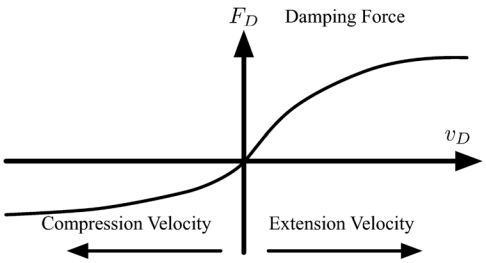

Fig. 7 Typical nonlinear damping curve

design intent and practical physical limitations (Fig. 7). Here an approach for constructing a linear damper is explored to improve the utility of this model as a test problem.

Figure 8 illustrates a single-tube telescopic damper. On jounce (compression), hydraulic fluid flows from the compression chamber to the extension chamber through the compression valve. On rebound (extension), fluid flows through the extension valve in the reverse direction. The pressurized gas chamber, separated from the hydraulic fluid by a floating piston, compensates for volume change from rod movement. The effect of the foot valve is neglected, and the piston valves are assumed to be spring-biased spool valves, as shown in Fig. 9. As the damper is compressed, the compression valve opens and fluid flows through ports in the side of the spool valve. Other valve types exist and are in common use, including disc, rod, and shim valves [58].

3.2.1 Damping Properties. The dependence of exposed port area (A_v) on valve lift (x_v) is a key design element that influences damping curve shape. Under standard assumptions, the damper will be linear if either $A_v \propto \sqrt{x_v}$ or $x_v \propto \sqrt{F_v}$, where F_v is the valve spring axial force. The latter relationship requires a nonlinear valve spring. The former relationship (used here) requires that the valve port shape is designed such that A_v increases proportionately with $\sqrt{x_v}$. More precisely, the ideal valve port area function is

$$\tilde{A}_v(x_v) = C_2 C_0 \sqrt{x_v} \tag{25}$$

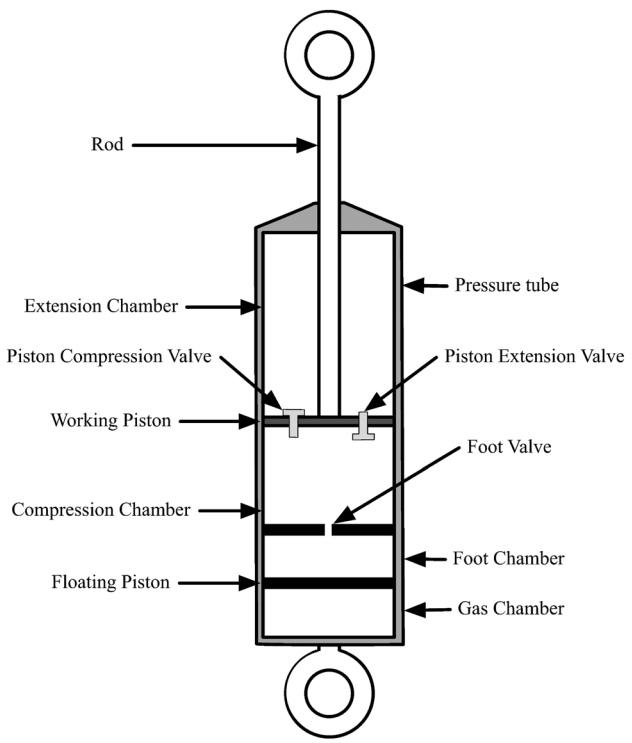

Fig. 8 Sectional view of a single-tube telescopic damper

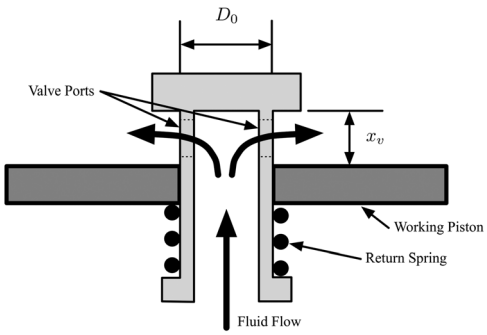

Fig. 9 Sectional view of the piston compression valve

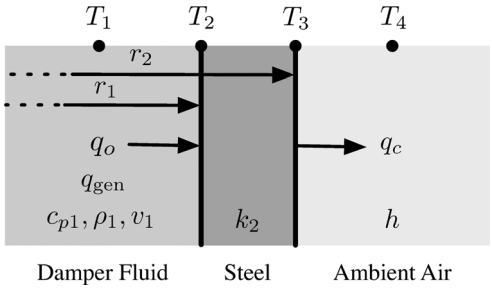

Fig. 10 Damper heat transfer model

where C_2 is the damper valve coefficient, $C_0 = \pi D_0$ is the outer circumference of the spool valve, and D_0 is the valve diameter. Port shape can be characterized by the arc length exposed by the ports at the top of the working piston as a function of valve lift: $C_e(x_v)$. The ideal valve port area function can be rewritten as

$$\tilde{A}_v(x_v) = \int_0^{x_v} C_e(\tau)d\tau \tag{26}$$

Solving Eqs. (25) and (26) for $C_e(x_v)$ reveals that linear damping requires the following exposed arc function:

$$\tilde{C}_e(x_v) = \frac{d}{dx_v}\tilde{A}_v(x_v) = \frac{1}{2}C_2C_0x_v^{-1/2} \tag{27}$$

Unfortunately, as $x_v \to 0$, $\tilde{C}_e(x_v) \to \infty$, which is physically unrealizable; $C_e(x_v)$ must remain less than the spool valve outer circumference C_0. Several candidate functions for approximating $\tilde{C}_e(x_v)$ were evaluated using a least squares approach with the requirement $C_e(x_v) \le \eta C_0$ for all $x_v \ge 0$. The coefficient $0 < \eta < 1$ defines the upper limit of the proportion of C_0 that may be exposed ($\eta = 0.9$ here). The following exposed arc function was selected:

$$C_e(x_v) = a_1\left(x_v + a_1^2\right)^{-1/2} \tag{28}$$

where $a_1 = C_3(x_m - C_3)^{-1/2}/2$, $x_m = A_0P_{\text{allow}}/k_v$ is the maximum valve lift at the maximum allowed damper pressure $P_{\text{allow}} = 4.75 \cdot 10^6$ Pa, $k_v = 7500$ N/m is the spool valve spring constant, $A_0 = \pi D_0^2/4\text{m}^2$ is the spool valve frontal area, and

$$C_3 = \frac{C_2D_0}{2\eta}\sqrt{\frac{\pi P_{\text{allow}}}{k_v}}$$

The damper valve coefficient is defined here as $C_2 = \eta A_f\sqrt{x_m}$, where A_f is an area factor that can be used to tune port shape ($0 < A_f < 1$). Here, $A_f = 0.1$. Decreasing A_f brings $C_e(x_v)$ closer to ideal for linear damping, but increases the valve diameter required to deliver a particular suspension damping coefficient c_s, which is calculated using

$$c_s = \frac{D_p^4}{8C_dC_2D_0^2}\sqrt{\frac{\pi k_v\rho_1}{2}} \tag{29}$$

where D_p is the working piston diameter, ρ_1 is the damper fluid density (850 kg/m^3), and C_d is the discharge coefficient ($\approx$0.7 for spool valves).

Damper stroke (total available axial displacement of the working piston) D_s is chosen here as an independent design variable. The complete plant design vector can now be defined

$$\mathbf{x_p} = [d, D, p, N_a, D_0, D_p, D_s]$$

3.2.2 Thermal Properties. Fluid temperature increase due to energy dissipation is an important consideration in damper design as it can induce damper fade and influences useful damper life. Heat generation in the damper is calculated based on the damping coefficient and suspension stroke velocity, i.e., $q_{\text{gen}} = c_s\dot{\xi}_3^2$. The same rough road profile is used for thermal tests that were specified for the spring fatigue calculations.

Figure 10 illustrates the thermal model for the damper, similar to the model found in Ref. [59], except here constant viscosity is assumed. As the damper piston moves, heat is generated in the damper fluid, which has heat capacity $c_{p1} = 2500$ J/kgK, density $\rho_1 = 850$ kg/m^3, and volume v_f. The fluid volume is $v_f = \pi D_p^2(D_s + \ell_{d1} + \ell_{d3})/4$, where $\ell_{d1} = 0.02$ m and $\ell_{d3} = 0.02$ m represent the space required for damper valving and casting extensions, respectively. Heat is conducted through the steel damper tube (with conduction coefficient $k_2 = 60.5$ W/mK) to the atmosphere with a constant temperature $T_4 = 300$ K. The convection coefficient between the damper and the atmosphere is $h = 50$ W/m^2 K. Heat capacity in the steel shell is small compared to that of the damper fluid (c_{p1}) and is assumed to be zero here, so $T_2 = T_3$.

Heat flow from the damper fluid through the shell is

$$q_0 = q_{\text{gen}} - \rho_1v_fc_{p1}\frac{\partial T_1}{\partial t} = \frac{2\pi L_2k_2}{\ln r_2/r_1}(T_1 - T_3) = hA_4(T_3 - T_4)$$

where r_1 and r_2 are the inner and outer shell radii, $L_2 = D_s + \ell_{d1} + \ell_{d3}$ is the shell height, and $A_4 = 2\pi r_2L_2$ is the external surface area of the shell. This DAE can be converted to a single ordinary differential equation via substitution. We can rewrite this system as

$$q_{\text{gen}} - b_1\dot{T}_1 = b_2(T_1 - T_3) = b_3(T_3 - T_4)$$

where the appropriate constants are replaced with b_1, b_2, and b_3 for convenience. Choosing T_1 as the state variable, the resulting ODE is

$$\dot{T}_1 = -\frac{b_2b_3}{b_1b_2 + b_1b_3}T_1 + \frac{b_2b_3}{b_1b_2 + b_1b_3}T_4 + \frac{q_{\text{gen}}}{b_1} \tag{30}$$

which can be simulated using the time history of ξ_3 as input to predict the resulting damper fluid temperature trajectory $T_1(t)$, which is treated as a fifth state variable (ξ_5) in this case study.

3.2.3 Damper Constraints

$$g_{12}(\mathbf{x_p}) = L_0 - L_s - D_s \le 0 \tag{31}$$

ensures adequate damper range of motion

$$g_{13}(\mathbf{x_p}) = 2D_s + \ell_{d1} + \ell_{d2} - L_{0\max} \le 0 \tag{32}$$

requires the damper to fit within the pocket length $L_{0\max} = 0.4$ m; $\ell_{d1} = 0.02$ m and $\ell_{d2} = 0.04$ m quantify the space required for

Table 3 Damper design parameters

η	0.9	P_{allow}	4.75×10^6 Pa
k_v	7500 N/m	A_f	0.1
ρ_1	850 kg/m^3	c_d	0.7
c_{p1}	2500 J/kgK	k_2	60.5 W/mK
T_4	300 K	h	50 W/m^2 K
ℓ_{d1}	0.02 m	ℓ_{d2}	0.04 m
ℓ_{d3}	0.02 m	$T_{1\text{allow}}$	390 K
$\dot{\xi}_{3\text{allow}}$	5.0 m/s	$x_{v\text{allow}}$	0.03 m

damper components above and below the working piston range, respectively. The damper fluid temperature constraint prevents fade and seal damage

$$g_{14}(\mathbf{x_p}, \Xi) = T_{1\max} - T_{1\text{allow}} \leq 0 \tag{33}$$

where $T_{1\max}$ is the maximum damper temperature achieved across all vehicle simulations and $T_{1\text{allow}} = 390$ K. This constraint was never active in our suspension system design studies but may become active given different design requirements or operating conditions.

$$g_{15}(\mathbf{x_p}, \Xi) = P_{\max} - P_{\text{allow}} \leq 0 \tag{34}$$

ensures the maximum damper pressure achieved during simulation ($P_{\max} = 4c_s\dot{\xi}_{3\max}/(\pi D_p^2)$) does not exceed the seal maximum pressure ($P_{\text{allow}} = 4.75 \times 10^6$ Pa).

$$g_{16}(\mathbf{x_p}, \Xi) = \dot{\xi}_{3\max} - \dot{\xi}_{3\text{allow}} \leq 0 \tag{35}$$

protects damper components from excessive velocity ($\dot{\xi}_{3\text{allow}} = 5.0$ m/s). The following is a clearance requirement for the maximum amount of spool valve lift, where $x_{v\max} = A_0 P_{\max}/k_v$ and $x_{v\text{allow}} = 0.03$ m:

$$g_{17}(\mathbf{x_p}, \Xi) = x_{v\max} - x_{v\text{allow}} \leq 0 \tag{36}$$

Damper design parameters are summarized in Table 3.

3.3 Problem Structure. Figure 11 illustrates the plant design constraint Jacobian structure for the DT implementation of the active suspension co-design problem. Black squares indicate linear dependence of a constraint on a variable, gray squares indicate nonlinear dependence, and squares with diagonal stripes indicate nonlinear relationships that can be made linear through a reformulation. This is a detailed subset of the problem structure diagram (first introduced in Fig. 4) that illustrates specifically the dependence of plant constraints on design and state variables. More precisely, Fig. 11 corresponds to the two blocks in the first row and second and third columns of Fig. 4(*a*). The dependence relationships expressed in Fig. 11 can be used to construct part of the directed graph problem structure representation introduced in Fig. 3 (specifically, the dependence of plant constraints on plant design and state variables). The rest of the directed graph can be constructed based on the model presented earlier in this section. This problem has no equality path constraints. The objective function (defined in Sec. 3.4) depends directly on state and control design variables.

The DT co-design problem structure illustrated in Fig. 3 can be verified by exploring the dependence relationships present in the DT co-design formulation of this case study. Plant analysis, including the calculation of plant constraints $\mathbf{g_p}(\cdot)$ and dynamic system model parameters $\mathbf{p}$, such as stiffness k_s and damping c_s rates, depends on plant design $\mathbf{x_p}$ (e.g., see Eqs. (13) and (29)). Many plant constraints depend also on discretized state trajectories Ξ. For example, stress and fatigue constraints depend on the oscillatory behavior of the suspension. Defect constraints $\boldsymbol{\zeta}(\cdot)$

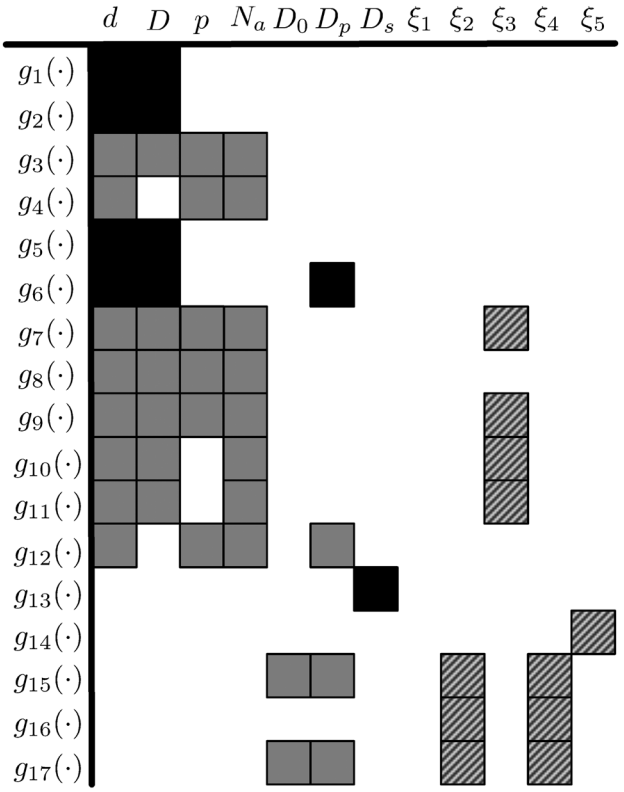

Fig. 11 Structure of the plant constraint Jacobian

depend on state trajectories and control design. Here, control design is represented by the discretized control trajectory $\mathbf{x_c}$. The objective function (given in Eq. (38)) depends on both state and control trajectories. It is influenced by plant design indirectly since system dynamics depend on plant design.

The constraints $g_1(\cdot)$–$g_{12}(\cdot)$ have dense dependence on $\mathbf{x_p}$. Plant constraints 7, 9–11, and 14–17 are influenced indirectly by plant design variables since they depend on state, and defect constraints enforce relationships between state and plant design variables. Defect constraint dependence on $\mathbf{x_p}$ and $\boldsymbol{\xi}$ is clear from the presence of these variables in the system matrix $\mathbf{A}$ of Eq. (12). Columns corresponding to control input are omitted from Fig. 11 as plant design constraints do not depend directly on $\mathbf{x_c}$. Plant constraints, however, may be influenced indirectly by control design through defect constraints (e.g., the fatigue constraint depends on state trajectories, and state trajectories depend on control design).

Figure 11 is a condensed representation of the plant constraint Jacobian structure. The actual Jacobian includes n_t columns for each state variable (one for every time step). Note that nonlinear dependence of a constraint on a variable results in a nonconstant Jacobian entry (with respect to changes in that variable); a linear dependence of a constraint on a variable corresponds to a constant Jacobian entry, and no dependence of a constraint on a variable results in a zero Jacobian entry (corresponding to white space in Fig. 11). Note that while constraints $g_1(\cdot)$ and $g_2(\cdot)$ appear to be nonlinear in Eqs. (14) and (15) since $C = D/d$, they can be made linear through simple algebraic manipulation.

In this example, all plant constraints that depend on state variables involve finding the maximum value of the associated state(s). This is a nonlinear relationship. We can remove this nonlinearity by replacing each constraint that depends on state with n_t constraints—one for every time step. For example, in constraint 16, we need to ensure that the maximum velocity $\dot{\xi}_3$ does not exceed $\dot{\xi}_{3\text{allow}}$. Instead of calculating $\dot{\xi}_{3\max}$ (a nonlinear operation), we can compare $\dot{\xi}_3 = \xi_4 - \xi_2$ to $\dot{\xi}_{3\text{allow}}$ at every time step. Using this approach, constraints 7, 9–11, and 14–17 can be made linear (with

respect to states). This increases problem size, but these constraints have a sparse diagonal structure, and this reformulation can result in more efficient solution when using algorithms that can exploit the increased linearity within the problem. This is an example of a reformulation that can produce the sparse diagonal structure shown in the first row and third column of Fig. 4(*b*) (i.e., dependence of plant constraints on state).

The defect constraint Jacobian for this case study, which is not shown here due to size, has dense dependence on $\mathbf{x_p}$, and a sparse diagonal dependence on Ξ and $\mathbf{x_c}$, similar to the structure of the second row illustrated in Fig. 4(*a*).

3.4 Active Suspension Co-Design. The control input $u(t)$ to the quarter-car active suspension system is an active force between the sprung and unsprung masses. This force could be realized via an actuator such as an electromagnetic linear motor [60,61], but actuator details will not be considered here and we will assume that an arbitrary force trajectory $\mathbf{u}(t)$ can be achieved, in some cases within maximum force limits. The control design variable $\mathbf{x_c}$ is a time-discretization of this actuation force trajectory, i.e.,

$$\mathbf{x_c} = [u_1, u_2, \ldots, u_{n_t}]^T$$

where $u_i = u(t_i)$. The active suspension state-space model with the road disturbance and control input terms is

$$\dot{\boldsymbol{\xi}} = \mathbf{A}\boldsymbol{\xi} + \begin{bmatrix} -1 \\ \dfrac{4c_t}{m_{us}} \\ 0 \\ 0 \end{bmatrix} \dot{z}_0 + \begin{bmatrix} 0 \\ -\dfrac{1}{m_{us}} \\ 0 \\ \dfrac{1}{m_s} \end{bmatrix} u \tag{37}$$

The system objective function incorporates handling, comfort, and control cost using a Lagrange term

$$J = \int_0^{t_F} (r_1(z_{us} - z_0)^2 + r_2 \ddot{z}_s^2 + r_3 u^2) dt \tag{38}$$

where the weights $r_1 = 10^5$, $r_2 = 0.5$, and $r_3 = 10^{-5}$ ensure each component of the objective is approximately the same magnitude. The objective depends on end time t_F, which is treated as a fixed parameter here. As described above, this design problem incorporates two road profile inputs: ramp and rough road profiles. To balance the contribution of each road profile to the overall objective function, the following weighted summation was used: $J = 1 \times 10^{-2}_{\text{ramp}} + J_{\text{rough}}$, where J_{ramp} and J_{rough} are the values of the above integral for the ramp and rough road inputs, respectively.

The plant design variable bounds $\underline{\mathbf{x}}_\mathbf{p} \le \mathbf{x_p} \le \bar{\mathbf{x}}_\mathbf{p}$ used here are

$$\underline{\mathbf{x}}_\mathbf{p} = [0.005, 0.05, 0.02, 3, 0.003, 0.03, 0.1]$$
$$\bar{\mathbf{x}}_\mathbf{p} = [0.02, 0.4, 0.5, 16, 0.012, 0.08, 0.3]$$

The simultaneous solution approach was implemented using a large-scale interior-point algorithm from the MATLAB® optimization toolbox. The sequential approach employed a SQP algorithm for plant optimization and then an interior-point algorithm for DT control design optimization. Problem structure was exploited by utilizing the sparse Jacobian functionality of these algorithms. An initial simulation of the open-loop system was also performed to determine appropriate time steps h_i to use in solving Eq. (11).

3.4.1 Sequential and Simultaneous Optimization for Co-Design. The active suspension system design problem was solved using both the sequential and simultaneous design approaches. When using the sequential approach, we can either assume a fixed control system design as we design the plant, or ignore the control system and design the suspension as a passive system. Allison and Herber classify plant design options for the sequential approach in greater detail and identify five specific strategies for plant design [25]. Case 1 is used here, where the plant design is based on a passive system (no active control, $u = 0$), and the original system objective function is used. In many previous studies, a different objective is used for plant design. In Allison's classification system, using a different objective function would correspond to case 2 if a passive dynamic model is used for plant design or case 3 if a static plant model is used.

In the sequential approach, the optimal control problem was solved using direct transcription after solving the plant design optimization problem. The resulting solution is

$$\mathbf{x_{seq}}* = [0.0147, 0.150, 0.0522, 4.05, 0.0070, 0.0422, 0.170] \tag{39}$$

The objective function values for both approaches are reported in Table 4, including the individual contributions to this value from both the ramp and rough road simulations. The solution obtained using the DT simultaneous approach (Eq. (11)) with the trapezoidal collocation method is

$$\mathbf{x_{DT}}* = [0.0097, 0.0620, 0.0201, 15.3, 0.0061, 0.0303, 0.170] \tag{40}$$

The computation time and required number of function evaluations are reported in Table 5 based on solution using a workstation with a 64 bit six-core, 12-thread i7–3930 K processor with 32 GB of DDR3 1600 MHz RAM running Ubuntu 12.04.

The simultaneous approach generates a superior design (almost 20% lower objective function value). While the sequential approach required less computation time, its result is not system-optimal. In addition, a simplified DT implementation was used in this case study (i.e., low-order trapezoidal collocation). Extension to high-order implicit collocation and further investigation of capitalizing on sparsity patterns within the new DT co-design problem structure may lead to substantial efficiency improvements for simultaneous solution. For example, distributed optimization methods for co-design, such as those presented in Ref. [39], may support the use of sparse optimization methods with DT while producing system-optimal designs. This is a subject for future work.

Additional insights can be obtained by inspecting the dynamic response corresponding to these two solutions (Figs. 12 and 13). The sprung mass response is shown along with the control force for both the ramp and rough road inputs. One immediate observation is the significantly higher control force magnitude required by the simultaneous approach (≈2400 N compared to ≈1400 N for sequential). In the sequential approach, the plant design is first optimized as a passive system, so the lower control effort is expected. The damping rate for the sequential result ($c_s = 2580$

Table 4 Objective function values and components

	Sequential	Simultaneous
J_{ramp}	52.4	55.8
J_{rough}	2.09	1.57
$J = 0.01J_{\text{ramp}} + J_{\text{rough}}$	2.63	2.12

Table 5 Computation time and function evaluation

	Sequential	Simultaneous
Computation time	4.12×10^3 s	5.81×10^4 s
Function evaluations	773	12,927

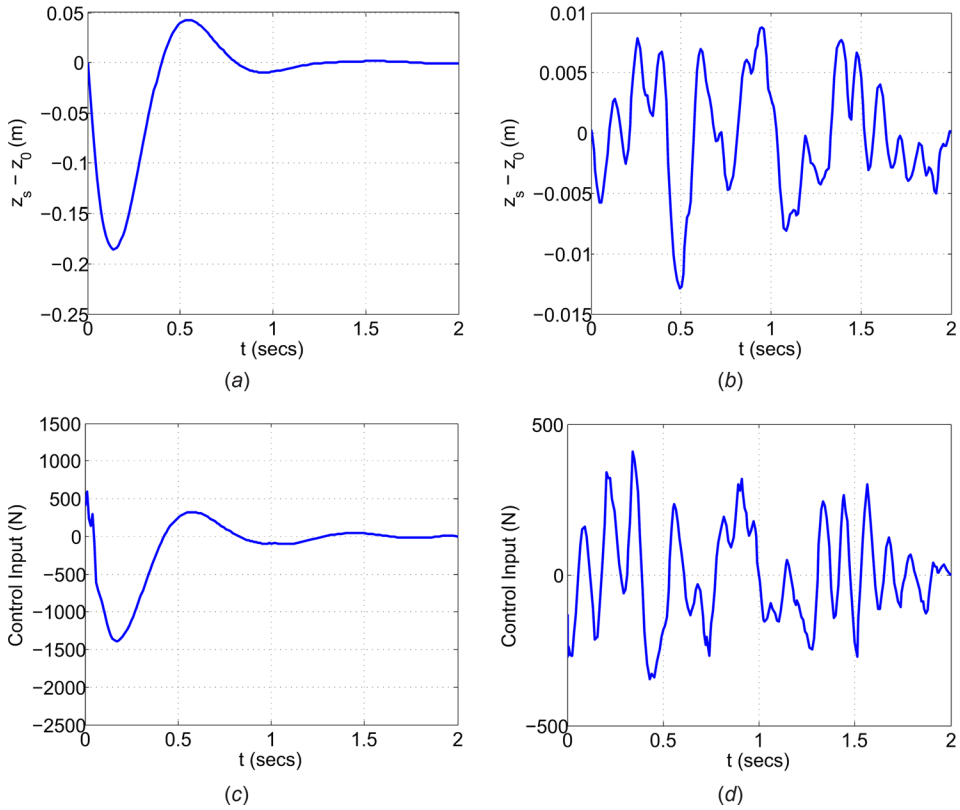

Fig. 12 System responses for sequential optimization. (a) Sprung mass response to ramp input. (b) Sprung mass response to rough road input. (c) Control force (ramp input). (d) Control force (rough road input)

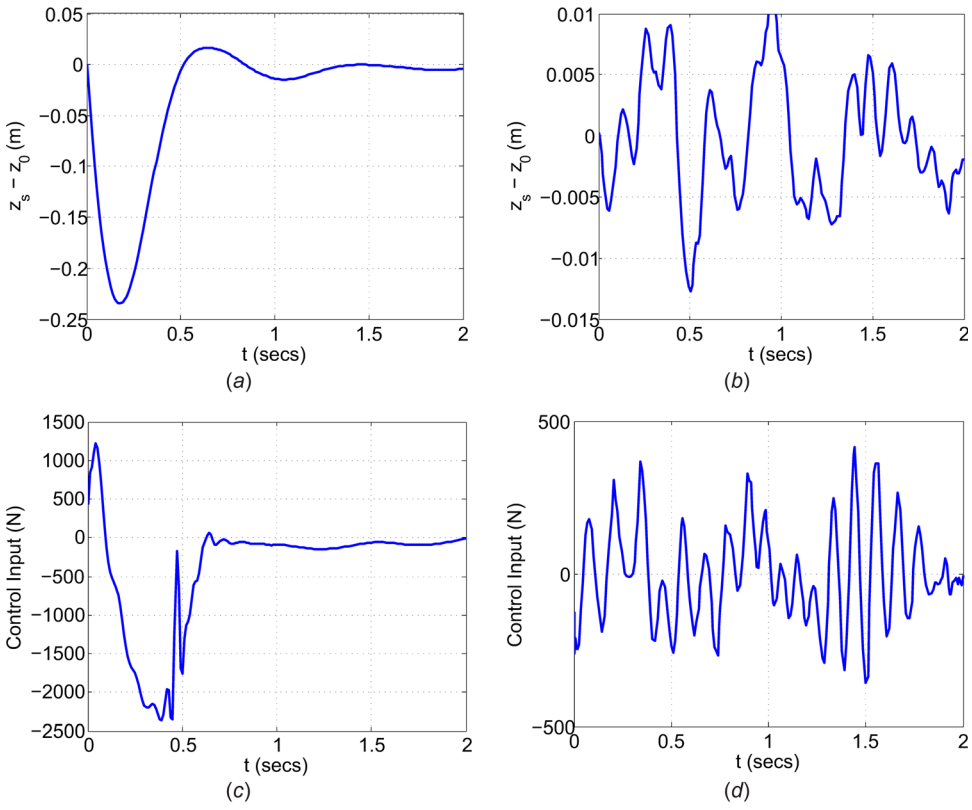

Fig. 13 System responses for simultaneous optimization. (a) Sprung mass response to ramp input. (b) Sprung mass response to rough road input. (c) Control force (ramp input). (d) Control force (rough road input)

Table 6 Comparison of optimal stiffness and damping rates

	Sequential	Simultaneous
k_s	3.30×10^4 N/m	2.36×10^4 N/m
c_s	2.58×10^3 Ns/m	1.03×10^3 Ns/m

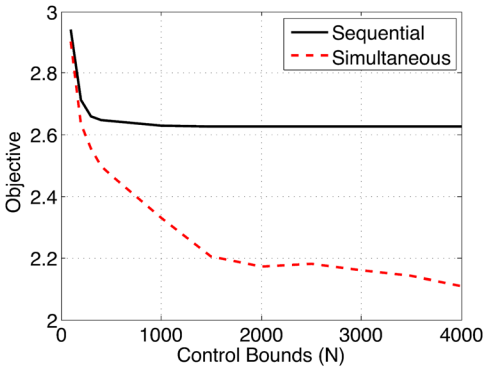

Fig. 14 Pareto sets illustrating the tradeoff between maximum control force and system objective function value (both sequential and simultaneous solution approaches)

Ns/m) is higher than for the simultaneous result, which is also congruent with the lower control effort required for the sequential design. See Table 6 for a comparison of damping and stiffness rates. The nested co-design method was implemented as well. While nested co-design can produce system-optimal results, it was very computationally inefficient for this case study, and the results are not reported here. The simultaneous co-design strategy with DT proved to be the most efficient method for identifying a system-optimal solution.

3.4.2 Transition From Passive to Active Dynamics. An important issue that we can begin to address here is how active systems should be designed differently than passive systems. This is a core question in the study of co-design, and improved understanding of the answer is becoming increasingly important as active dynamic systems become more ubiquitous. One strategy for investigating fundamental principles of active system design is to examine how system-optimal designs change as the role of active control is gradually increased for a particular system. The role of active control in the active suspension case study can be modified by imposing explicit bounds on maximum control force. Co-design results with bounds near zero will be close to the passive system results, and the bounds can then be increased continuously to observe how design and dynamic properties change as active control plays an increasingly important role.

This study was performed using multiobjective optimization. The resulting Pareto sets are presented in Fig. 14, which illustrates the tradeoff between maximum control force u_{max} and dynamic system performance (as measured by the combined objective function J defined in Eq. (38)). Results are displayed for both the sequential and simultaneous solution methods.

For both the sequential and simultaneous methods, reducing u_{max} increases the smallest attainable value for J. For all values of u_{max} tested in this study, the simultaneous result is superior to the sequential result. Reducing u_{max} results in a smaller difference between the sequential and simultaneous method minimum objective function values. This is expected since tight bounds restrict the potential influence of active control as well as the potential synergistic coupling between plant and control system design.

As u_{max} is increased, the objective for the simultaneous approach decreases (approximately) monotonically. The objective obtained using the sequential approach, however, decreases with increasing u_{max} only until $u_{max} \approx 1000$ N. Beyond this, no improvement can be made. To explain this observation, notice that the optimal plant design is independent of control in sequential design and is therefore, not influenced by u_{max}. In other words, when using sequential design, the plant design result is always the same (see Eq. (39) for this value). While some improvement to system performance can be made by adding active control to the optimal passive system result, Fig. 14 illustrates a clear limit to the improvement that can be achieved via control design improvements alone. Further improvement will require plant design changes that are synergistic with active control (i.e., co-design). Co-design helps engineers create passive system dynamics that combine in an ideal manner with active control [62,25] to produce the best possible system performance.

Each point on the Pareto sets in Fig. 14 represents a complete system design (plant and control). These designs can be explored in more detail to reveal additional insights about the transition from passive to active system design. Table 7 lists the plant design and dependent suspension parameters for a subset of points in the Pareto set. Each column corresponds to a different control force bound u_{max} and its associated optimal design. Only co-design results are listed (recall that $\mathbf{x}_{\mathbf{p}*}$ in sequential design is constant).

Stiffness and damping decrease with larger u_{max}. Internal forces from passive elements must be reduced to accommodate a stronger control system if higher-performance dynamics that are fundamentally distinct from the passive system are to be realized.

While examining plant design does provide important insights, it does not paint a complete picture of system dynamics. Deeper analysis of changes in system dynamics is required. Modal analysis was performed on each of the designs in the Pareto set corresponding to simultaneous design [63]. The results of this analysis are summarized in Table 8.

The suspension has two mechanical degrees of freedom (DOFs). The unsprung mass position z_{us} is the first DOF, and the sprung mass position z_s is the second. As a two-DOF system, the quarter car suspension model has two vibration mode shapes (patterns of oscillation), $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$. Each mode has a corresponding natural frequency ω_i; both masses will oscillate at frequency ω_i when mode shape $\boldsymbol{\psi}_i$ is excited. Vibration modes are a property of passive dynamics (i.e., free vibrations) that are independent of control design, but the passive dynamic properties reported here reflect a system that has been optimized to work with active control, providing insights into how the dynamic properties of physical systems should be designed differently for active control.

Table 7 Optimal plant designs for a range of maximum control force values

u_{max} (N)	100	200	300	400	500	1000	1500	2000	2500	3000	4000
d (m)	0.0117	0.0117	0.0115	0.0113	0.0113	0.0108	0.0105	0.0104	0.0101	0. 098	0.0097
D (m)	0.0752	0.0743	0.0743	0.0733	0.0727	0.0711	0.0675	0.0662	0.0631	0.0639	0.062
p (m)	0.0239	0.0236	0.0237	0.0234	0.0232	0.0228	0.0217	0.0213	0.0203	0.0207	0.0201
N_a	12.73	13.04	12.72	12.89	13.04	13.04	13.97	14.36	15.31	14.47	15.28
D_0 (m)	0.007	0.0069	0.0069	0.0069	0.0068	0.0068	0.066	0.0065	0.0061	0.0064	0.0062
D_p (m)	0.0415	0.0406	0.0408	0.04	0.0394	0.0382	0.035	0.0338	0.0311	0.0321	0.0300
D_s (m)	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
k_s (10^4 N/m)	3.30	3.32	3.15	3.10	3.07	2.82	2.68	2.64	2.53	2.35	2.36
c_s (10^4 Ns/m)	2.47	2.34	2.33	2.22	2.15	1.87	1.48	1.34	1.17	1.15	0.97

Table 8 Modal analysis of simultaneous design results for a range of maximum control force values

u_{max} (N)	100	200	300	400	500	1000	1500	2000	2500	3000	4000
$\psi_1(\times 10^{-3})$	$\begin{bmatrix}123.8\\-3.2\end{bmatrix}$	$\begin{bmatrix}123.8\\-3.2\end{bmatrix}$	$\begin{bmatrix}123.8\\-3.2\end{bmatrix}$	$\begin{bmatrix}123.9\\-3.0\end{bmatrix}$	$\begin{bmatrix}123.9\\-3.0\end{bmatrix}$	$\begin{bmatrix}123.9\\-2.9\end{bmatrix}$	$\begin{bmatrix}123.9\\-2.7\end{bmatrix}$	$\begin{bmatrix}123.9\\-2.6\end{bmatrix}$	$\begin{bmatrix}123.9\\-2.5\end{bmatrix}$	$\begin{bmatrix}123.9\\-2.3\end{bmatrix}$	$\begin{bmatrix}123.9\\-2.3\end{bmatrix}$
ω_1 (rad/s)	64.02	64.04	63.83	63.75	63.72	63.40	63.23	63.18	63.03	62.81	62.82
ν_1	0.36	0.35	0.34	0.32	0.31	0.27	0.21	0.19	0.16	0.16	0.14
$\psi_2(\times 10^{-3})$	$\begin{bmatrix}7.1\\5.54\end{bmatrix}$	$\begin{bmatrix}6.8\\5.54\end{bmatrix}$	$\begin{bmatrix}6.6\\5.54\end{bmatrix}$	$\begin{bmatrix}6.6\\5.54\end{bmatrix}$	$\begin{bmatrix}6.1\\5.54\end{bmatrix}$	$\begin{bmatrix}5.8\\5.54\end{bmatrix}$	$\begin{bmatrix}5.7\\5.54\end{bmatrix}$	$\begin{bmatrix}5.5\\5.54\end{bmatrix}$	$\begin{bmatrix}5.2\\5.54\end{bmatrix}$	$\begin{bmatrix}5.2\\5.54\end{bmatrix}$	$\begin{bmatrix}5.2\\5.54\end{bmatrix}$
ω_2(rad/s)	9.42	9.44	9.24	9.16	9.12	8.79	8.59	8.53	8.37	8.1	8.11
ν_2	0.38	0.36	0.37	0.35	0.34	0.31	0.25	0.23	0.21	0.21	0.18
$\mathrm{Re}(\lambda_{1,2})$	−19.72	−18.75	−18.65	−17.71	−17.21	−15.04	−11.83	−10.27	−9.32	−9.15	−7.70
$\mathrm{Re}(\lambda_{3,4})$	−3.04	−2.86	−2.90	−2.75	−2.67	−2.37	−1.86	−1.69	−1.47	−1.47	−1.23
$Im(\lambda_{1,2})$	58.86	59.41	59.21	59.6	59.8	60.4	61.39	61.66	61.89	61.70	62.03
$Im(\lambda_{3,4})$	9.23	9.28	9.04	8.98	8.95	8.63	8.49	8.44	8.30	8.02	8.06

The mode shapes and natural frequencies in Table 8 were calculated based on a mass-normalized stiffness matrix, i.e., $\tilde{\mathbf{K}} = \mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$, where $\mathbf{M}$ and $\mathbf{K}$ are the 2×2 mass and stiffness matrices from the second-order system of differential equations that is equivalent to the first-order system given in Eq. (37).

Table 8 shows that natural frequency for both modes decreases with increasing control authority. This is congruent with the decreasing stiffness and damping apparent in Table 7. Mode shapes indicate how a system resonates. While stiffness changes substantially as the role of control is increased, both suspension mode shapes exhibit relatively little change. Sprung mass amplitude in $\boldsymbol{\psi}_1$ does decrease some with increasing u_{max}, and unsprung mass amplitude in $\boldsymbol{\psi}_2$ decreases with increasing u_{max}.

Modal damping values cannot be computed exactly from system matrices because this system does not exhibit proportional damping. An approximate modal damping value ν_i was calculated for each design by identifying a proportionally damped system (i.e., a system where the damping matrix is $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$, where α and β are constants) that is as close as possible to the actual system in a least-squares sense. This provides an approximate indication of how close to critical damping ($\nu_i = 1$, fastest possible response without overshoot) each design is. The modal damping values in Table 8 move farther away critically damped with increasing control effort, indicating that these designs rely more on active control to manage overshoot than on passive elements.

Finally, the eigenvalues for the system matrix $\mathbf{A}$ from Eq. (37) were computed (Table 8). Both real and imaginary parts decrease in magnitude as u_{max} increases, indicating that the passive system dynamics slow down and play a smaller role in system behavior as the control force bound is increased. Additional analysis methods for studying the dynamics of active and passive vehicle suspension systems can be found in Refs. [53], [64], and [65]. An important observation here is that passive forces are not just reduced, but reduced in a way that complements control system dynamics.

While this example is a relatively simple system that exhibited simple trends in the parametric study on control force, it illustrates the type of investigation that could help reveal how the physical aspects of actively controlled systems should be designed differently than for passive systems. These results motivate more extensive studies based on carefully selected co-design problems—from a range of application domains—to seek fundamental design principles for active dynamic systems. For example, test problems might be designed such that off-diagonal terms in the mass matrix can be varied by adjusting plant design to investigate changes in optimal levels of dynamic coupling. Nonlinear test problems may provide complementary insights. This work also highlights the importance of co-design investigations with deeper treatment of physical system design. The case study presented here is one step toward more realistic plant design models, but still utilizes a low-fidelity model that involves only continuous design variables. This creates a fundamental limitation on plant design exploration. In addition to use of high-fidelity models, co-design studies aimed at deeper understanding of fundamental plant design principles for active systems should incorporate system architecture design exploration [25,66–69].

4 Conclusion

Direct transcription was reviewed, and an extension of DT to co-design problems was introduced and demonstrated using an active suspension design problem. The case study built upon existing active suspension design problems by incorporating detailed physical system design considerations and physically independent design variables. The system model maintains linearity of the differential equations, making it suitable for a range of other co-design studies. While DT results in large problem dimension, it is arithmetically consistent, enables fine-grained parallelism and the use of higher-order implicit methods, and can be applied to challenging design problems that are otherwise unsolvable (e.g., inequality path constraints, open-loop instabilities). The case study illustrates that DT co-design has advantages over the sequential design approach. DT offers a path forward for solving especially difficult co-design problems, including those with realistic plant inequality constraints, bidirectional coupling, and co-design problems involving singular optimal control. The large size and density of the DT optimization problem remains a challenge for its applicability in co-design. Creative formulations can help improve problem structure, but in some cases problem density will make efficient DT solution difficult, at least with currently known techniques. Exploration of this new problem structure and development of more efficient DT co-design solution implementations are opportunities for further contribution to the design of modern engineering systems.

References

[1] Friedland, B., 1996, *Advanced Control System Design*, Prentice-Hall, Upper Saddle River, NJ.

[2] Reyer, J. A., Fathy, H. K., Papalambros, P. Y., and Ulsoy, A. G., 2001, "Comparison of Combined Embodiment Design and Control Optimization Strategies Using Optimality Conditions," The Proceedings of the 2001 ASME Design Engineering Technical Conferences, Pittsburgh, PA.

[3] Roos, F., 2007, "Towards a Methodology for Integrated Design of Mechatronic Servo Systems," Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden.

[4] Li, Q., Zhang, W., and Chen, L., 2001, "Design for Control—A Concurrent Engineering Approach for Mechatronic Systems Design," IEEE/ASME Trans. Mechatronics, **6**(2), pp. 161–169.

[5] Fathy, H. K., Reyer, J. A., Papalambros, P. Y., and Ulsoy, A. G., 2001, "On the Coupling Between the Plant and Controller Optimization Problems," The Proceedings of the 2001 American Control Conference, Arlington, VA.

[6] Peters, D. L., Papalambros, P. Y., and Ulsoy, A. G., 2009, "On Measures of Coupling Between the Artifact and Controller Optimal Design Problems," The

Proceedings of the 2009 ASME Design Engineering Technical Conferences, San Diego, CA.
[7] Cervantes, A., and Biegler, L. T., 1999, "Optimization Strategies for Dynamic Systems," Encycl. Optim., **4**, pp. 216–227.
[8] Biegler, L. T., 2007, "An Overview of Simultaneous Strategies for Dynamic Optimization," Chem. Eng. Process.: Process Intensif., **46**(11), pp. 1043–1053.
[9] Hargraves, C. R., and Paris, S. W., 1987, "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," J. Guid. Control Dynam., **10**(4), pp. 338–342.
[10] Ozimek, M. T., Grebow, D. J., and Howell, K. C., 2008, "Solar Sails and Lunar South Pole Coverage," The Proceedings of the 2008 AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, HI.
[11] Betts, J. T., 2010, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM, Philadelphia, PA.
[12] Xiang, Y., Arora, J. S., and K. Abdel-Malek, 2010, "Physics-Based Modeling and Simulation of Human Walking: A Review of Optimization-Based and Other Approaches," Struct. Multidiscip. Optim., **42**(1), pp. 1–23.
[13] Jung, E., Lenhart, S., and Feng, Z., 2002, "Optimal Control of Treatments in a Two-Strain Tuberculosis Model," Discrete Contin. Dynam. Syst. Series B, **2**(4), pp. 476–482. Available at: http://www.math.utk.edu/~lenhart/docs/tbfinal.pdf.
[14] Brenan, K. E., Campbell, S. L., and Petzold, L. R., 1996, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM, Philadelphia, PA.
[15] Athans, M., and Falb, P., 1966, *Optimal Control: An Introduction to the Theory and Its Applications*, McGraw-Hill, New York.
[16] Biegler, L. T., 2010, *Nonlinear Programming Concepts, Algorithms, and Applications to Chemical Processes*, SIAM, Philadelphia, PA.
[17] Powell, D. M. J., 1978, "Algorithms for Nonlinear Constraints That Use Lagrangian Functions," Math. Programming, **14**(1), pp. 224–248.
[18] MathWorks, Inc., 2013, "Simulink Design Optimization Product Reference," http://www.mathworks.com/products/sl-design-optimization/, retrieved Feb. 19.
[19] Coleman, T. F., and More, J. J., 1983, "Estimation of Sparse Jacobian Matrices and Graph Coloring Problems," SIAM J. Numer. Anal., **20**(1), pp. 187–209.
[20] Rutquist, P. E., and Edvall, M. M., 2010, PROPT-MATLAB Optimal Control Software, Tomlab Optimization, Inc., Pullman, WA.
[21] Sun, E. T., and Stadtherr, M. A., 1988, "On Sparse Finite-Difference Schemes Applied to Chemical Process Engineering Problems," Comput. Chem. Eng., **12**(8), pp. 849–851.
[22] Vasantharajan, S., and Biegler, L. T., 1990, "Simultaneous Strategies for Optimization of Differential-Algebraic Systems with Enforcement of Error Criteria," Comput. Chem. Eng., **14**(10), pp. 1083–1100.
[23] Rao, A. V., Benson, D. A., Darby, C., Patterson, M. A., Francolin, C., Sanders, I., and Huntington, G. T., 2010, "Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," ACM Trans. Math. Software, **37**(2), p. 22.
[24] Laird, C. D., Wong, A. V., and Akesson, J., 2011, "Parallel Solution of Large-Scale Dynamic Optimization Problems," The Proceedings of the 21st European Symposium on Computer Aided Process Engineering—ESCAPE 21, Sithonia, Chalkidiki, Greece.
[25] Allison, J. T., and Herber, D. R., 2014, "Multidisciplinary Design Optimization of Dynamic Engineering Systems," AIAA J., **52**(4), pp. 691–710.
[26] Herman, A. L., and Conway, B. A., 1996, "Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules," J. Guid. Control Dynam., **19**(3), pp. 592–599.
[27] Williams, P., 2009, "Hermite-Legendre-Gauss-Lobatto Direct Transcription in Trajectory Optimization," J. Guid. Control Dynam., **32**(4), pp. 1392–1395.
[28] Allison, J. T., Kokkolaras, M., and Papalambros, P. Y., 2007, "On Selecting Single-Level Formulations for Complex System Design Optimization," ASME J. Mech. Des., **129**(9), pp. 898–906.
[29] Cramer, E. J., Dennis, J. E., Jr., Frank, P. D., Lewis, R. M., and Shubin, G. R., 1994, "Problem Formulation for Multidisciplinary Optimization," SIAM J. Optim., **4**(4), pp. 754–776.
[30] Bertsekas, D. P., and Tsitsiklis, J. N., 1997, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Belmont, MA.
[31] Smith, M. C., and Walker, G. W., 2005, "Interconnected Vehicle Suspension," Proc. Inst. Mech. Eng., Part D, **219**(3), pp. 295–307.
[32] Schroer, R. T., Boggess, M. J., Bachmann, R. J., Quinn, R. D., and Ritzmann, R. E., 2004, "Comparing Cockroach and Whegs Robot Body Motions," The Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, IEEE.
[33] Lee, J., Lamperski, A., Schmitt, J., and Cowan, N., 2006, "Task-Level Control of the Lateral Leg Spring Model of Cockroach Locomotion," Fast Motions Biomech. Rob., **340**, pp. 167–188.
[34] Nishigaki, H., and Kawashima, K., 1998, "Motion Control and Shape Optimization of a Suitlike Flexible Arm," Struct. Optim., **15**(3–4), pp. 163–171.
[35] Trease, B. P., 2008, "Topology Synthesis of Compliant Systems with Embedded Actuators and Sensors," Ph.D. dissertation, University of Michigan, Ann Arbor, MI.
[36] Carmichael, D. G., 1990, "Structural Optimization and System Dynamics," Struct. Multidiscip. Optim., **2**(2), pp. 105–108.
[37] Fares, M. E., Youssif, Y. G., and Hafiz, M. A., 2005, "Multiobjective Design and Control Optimization for Minimum Thermal Postbuckling Dynamic Response and Maximum Buckling Temperature of Composite Laminates," Struct. Multidiscip. Optim., **30**(2), pp. 89–100.
[38] Fathy, H. K., 2003, "Combined Plant and Control Optimization: Theory, Strategies and Applications," Ph.D. dissertation, University of Michigan, Ann Arbor, MI.
[39] Allison, J. T., and Nazari, S., 2010, "Combined Plant and Controller Design Using Decomposition-Based Design Optimization and the Minimum Principle," The Proceedings of the 2010 ASME Design Engineering Technical Conferences, Montreal, Quebec, Canada.
[40] Peters, D. L., Papalambros, P. Y., and Ulsoy, A. G., 2013, "Sequential Co-Design of an Artifact and Its Controller via Control Proxy Functions," Mechatronics, **23**(4), pp. 409–418.
[41] Papalambros, P. Y., and Wilde, D., 2000, *Principles of Optimal Design: Modeling and Computation*, 2nd ed., Cambridge University Press, Cambridge, UK.
[42] Deshmukh, A., and Allison, J. T., 2013, "Design of Nonlinear Dynamic Systems Using Surrogate Models of Derivative Functions," The Proceedings of the 2013 ASME Design Engineering Technical Conferences, Portland, OR.
[43] Williams, P., and Trivailoa, P., 2005, "Optimal Parameter Estimation of Dynamical Systems Using Direct Transcription Methods," Inverse Probl. Sci. Eng., **13**(4), pp. 377–409.
[44] Reyer, J. A., and Papalambros, P. Y., 2000, "An Investigation into Modeling and Solution Strategies for Optimal Design and Control," The Proceedings of the 2000 ASME Design Engineering Technical Conferences, Baltimore, MD.
[45] Allison, J. T., 2013, "Co-Design of an Active Automotive Suspension Using Direct Transcription," http://www.mathworks.us/matlabcentral/fileexchange/40504
[46] Kasturi, P., and Dupont, P., 1998, "Constrained Optimal Control of Vibration Dampers," J. Sound Vib., **215**(3), pp. 499–509.
[47] Gobbi, M., 2001, "Analytical Description and Optimization of the Dynamic Behaviour of Passively Suspended Road Vehicles," J. Sound Vib., **245**(3), pp. 457–481.
[48] He, Y., and McPhee, J., 2005, "Multidisciplinary Design Optimization of Mechatronic Vehicle With Active Suspensions," J. Sound Vib., **283**(1), pp. 217–241.
[49] Allison, J. T., 2008, "Optimal Partitioning and Coordination Decisions in Decomposition-Based Design Optimization," Ph.D. dissertation, University of Michigan, Ann Arbor, MI.
[50] Fathy, H. K., Papalambros, P. Y., Ulsoy, A. G., and Hrovat, D., 2003, "Nested Plant/Controller Optimization With Application to Combined Passive/Active Automotive Suspensions," The Proceedings of the 2003 American Control Conference, Denver, CO, IEEE.
[51] Bourmistrova, A., Storey, I., and Subic, A., 2005, "Multiobjective Optimisation of Active and Semi-Active Suspension Systems With Application of Evolutionary Algorithm," The Proceedings of the 2005 International Conference on Modelling and Simulation, Melbourne, Australia.
[52] Alyaqout, S. F., Papalambros, P. Y., and Ulsoy, A. G., 2007, "Combined Design and Robust Control of a Vehicle Passive/Active Suspension," The Proceedings of the 2007 European Control Conference, Kos, Greece.
[53] Verros, G., Natsiavas, S., and Papadimitriou, C., 2005, "Design Optimization of Quarter-Car Models With Passive and Semi-Active Suspensions Under Random Road Excitation," J. Vib. Control, **11**(5), pp. 581–606.
[54] Shigley, J., Mischke, C., and Budynas, R., 2003, *Mechanical Engineering Design*, McGraw-Hill, New York.
[55] Azarm, S., 1982, "An Interactive Design Procedure for Optimization of Helical Compression Springs," University of Michigan, Technical Report No. UM-MEAM-82-7.
[56] Stoicescu, A., 2009, "On the Optimal Design of Helical Springs of an Automobile Suspension," UPB Sci. Bull., Series D **71**(1), pp. 81–94. Available at: http://scientificbulletin.upb.ro/rev_docs_arhiva/full1052.pdf.
[57] Sayers, M. W., and Karamihas, S. M., 1998, *The Little Book of Profiling*, University of Michigan Transportation Research Institute, Ann Arbor, MI.
[58] Dixon, J. C., 2007, *The Shock Absorber Handbook*, 2nd ed., John Wiley & Sons Ltd., Chichester, UK.
[59] Lion, A., and Loose, S., 2002, "A Thermomechanically Coupled Model for Automotive Shock Absorbers: Theory, Experiments and Vehicle Simulations on Test Tracks," Vehicle System Dynamics, **37**(4), pp. 241–261.
[60] Bose, A. G., 1990, "Wheel Assembly Suspending," US Patent No. 4,960,290, October.
[61] Jones, W. D., 2005, "Easy Ride: Bose Corp. Uses Speaker Technology to Give Cars Adaptive Suspension," IEEE Spectrum **42**(5), pp. 12–14.
[62] Allison, J. T., 2013, "Engineering System Co-Design with Limited Plant Redesign," Eng. Optim., **46**(2), pp. 200–217.
[63] Inman, D. J., 2014, *Engineering Vibrations*, 4th ed. Pearson Education, Inc., Upper Saddle River, NJ.
[64] Wang, F. C., 2001, "Design and Synthesis of Active and Passive Vehicle Suspensions," Ph.D. dissertation, University of Cambridge, Cambridge, UK.
[65] Thite, A. N., 2012, "Development of a Refined Quarter Car Model for the Analysis of Discomfort Due to Vibration," Adv. Acoust. Vib., **2012**, p. 863061.
[66] Allison, J. T., Khetan, A., and Lohan, D. J., 2013, "Managing Variable-Dimension Structural Optimization Problems Using Generative Algorithms," The Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization (WCSMO), Orlando, FL.
[67] Guo, T., and Allison, J. T., 2013, "On the Use of MPCCs in Combined Topological and Parametric Design of Genetic Regulatory Circuits," The Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization (WCSMO), Orlando, FL.
[68] Clune, J., and Lipson, H., 2011, "Evolving 3D Objects with a Generative Encoding Inspired by Developmental Biology," The Proceedings of the 2011 European Conference on Artificial Life, Paris, France.
[69] Campbell, M. I., Rai, R., and Kurtoglu, T., 2012, "A Stochastic Tree-Search Algorithm for Generative Grammars," ASME J. Comput. Inform. Sci. Eng., **12**(3), p. 031006.