Co-Design of an Active Suspension Using Simultaneous Dynamic Optimization

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

\[ \ddot{\xi} = f_a(\xi(t), u(t), t) \quad (1) \]

\[ \theta = f_a(\xi(t), u(t), t) \quad (2) \]

\[ \theta = f_b(\xi(t), u(t), t, t) \quad (3) \]

Equation (1) defines the time derivatives of a system model, where \( \xi(t) \) (length \( n_\xi \)) is the vector of state variables, \( u(t) \) (length \( n_u \)) is the control input, and \( f_a(\cdot) \) (length \( n_a \)) 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 \( \xi(t_0) = \xi_0 \), it can be discretized in time solved using a numerical
method for simulating ordinary differential equations (ODEs), such as the first-order Runge–Kutta method (forward Euler method)

$$\xi_i = \xi_{i-1} + h \dot{f}_d(t_{i-1}, \mathbf{u}(t_{i-1}), t_{i-1})$$

(4)

Here, $h = t_i - t_{i-1}$ is the integration step size at time $t_i$ and $i = 1, 2, \ldots, N$ 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 $\xi_i$ at each time step.

The open-loop optimal control problem can be written as

$$\min_{\xi(t), \mathbf{u}(t), t_f} J = \phi(\xi(t), \mathbf{u}(t), t_f)$$

(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

$$J = \eta(\xi(t_f), \xi(t_0), \mathbf{u}(t_0), t)$$

$$+ \int_{t_0}^{t_f} L(\xi(t), \xi(t), \mathbf{u}(t), t) dt$$

(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

$$\min_{\xi(t), \mathbf{x}_c, \mathbf{p}} J = \phi(\xi(t), \mathbf{x}_c, \mathbf{p})$$

(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 $\xi$ from the optimization variable set by solving for state variable values using a simulation nested within the optimization problem

$$\min_{\mathbf{x}_c} J = \phi(\mathbf{X}, \mathbf{x}_c)$$

(8)

where $\mathbf{X}$ is a discretized representation of the state variables (the $i$th row of $\mathbf{X}$ is $\xi_i = \xi(t_i)$, $i = 1, 2, \ldots, N_t$, and $N_t$ is the number of time steps), and $\mathbf{X}$ 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 known as multiple shooting (MS). Each time segment contains multiple time steps. The state at the interfaces between time segments (Y) is controlled directly by the optimization algorithm, and state continuity is enforced at convergence by defect constraints $\zeta(\mathbf{X}, \mathbf{Y})$

$$\min_{\mathbf{x}_c, \mathbf{Y}} J = \phi(\mathbf{X}, \mathbf{x}_c)$$

s.t. $\zeta(\mathbf{X}, \mathbf{Y}) = 0, \quad i = 1, 2, \ldots, N_t - 1$

(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 $j$th row of $\mathbf{Y}$. To clarify, in Fig. 1, the value for $\zeta_j$ calculated using simulation over time segment 1 does not match the value for $\zeta_j$ 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 $\zeta_j$, 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 \( n_T \rightarrow n_t - 1 \), the simulation for each time segment collapses to a single difference equation. The set of defect constraints replace simulation completely, and \( Y \) expands into the full state matrix \( \Xi \). This is DT; the infinite-dimensional optimal control problem is transcribed directly to a finite-dimension nonlinear programming problem

\[
\begin{align*}
\min_{\Xi, \mathbf{x}} & \quad J = \phi(\Xi, \mathbf{x}, t_f) \\
\text{s.t.} & \quad \zeta_i(\Xi, \mathbf{x}, t_p) = 0 \\
& \quad f_u(\Xi, \mathbf{x}, t_p) = 0 \\
& \quad \text{where } i = 2, 3, \ldots, n_t - 1, n_t
\end{align*}
\]

A defect constraint is defined for each time step, and path constraints are discretized (\( \zeta_i(\cdot) \)). In some cases, it may be desirable for implementation convenience to include a defect constraint \( \zeta_i(\cdot) \) for initial conditions and allow the first row of \( \Xi \) 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 \( t_p \), and \( \mathbf{x}_t \) can be replaced easily with discretized \( \mathbf{u}(t) \) to solve open-loop optimal control problems. As \( \max(x_i) = 0 \), 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 (\( \Xi \) has \( n_t \cdot n_s \) elements), its problem structure presents important advantages. At every optimization iteration, \( \Xi \) 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 as 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

\[
\zeta_i(\Xi, \mathbf{x}_t) = \zeta_i - f_u(\Xi, \mathbf{x}_t, t_{i-1}) - h_i f_a(\Xi, \mathbf{x}_t, t_{i-1})
\]

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

\[
\zeta_i(\Xi, \mathbf{x}_t) = \zeta_i - f_u(\Xi, \mathbf{x}_t, t_{i-1}) - \frac{h_i}{2} [f_u(\Xi, \mathbf{x}_t, t_{i-1}) + f_u(\Xi, \mathbf{x}_t, t_i)]
\]

\( \zeta \) is known a priori (it is a row of \( \Xi \)), so defect equations may be evaluated without inner iteration as is required with forward simulation using implicit methods. This concept extends to higher-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 \( \Xi \) 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 \( \mathbf{x}_d \) in addition to \( \mathbf{x}_t \) if a co-design problem is being solved. Forward simulation is then used to solve for state values \( \mathbf{x}_d \) that are consistent with physics. The optimization algorithm chooses a new point \( \mathbf{x}_t \) with the objective of reducing \( J \), and the process repeats until convergence to \( \mathbf{x}_d, \Xi, \mathbf{x}_t \). 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](image_url)

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.
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 analysis subproblems. Defect constraints in IDF are part of the optimization variable set. The defect constraints in DT are independent, enabling fine-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 (plant then control) design processes produce suboptimal results because the two design domains are coupled. 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 advances 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

### Table 1 Relationship between direct transcription and MDO

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

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 $p$ 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 advances 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
The passive dynamic response of this system can be characterized by its natural frequencies and mode shapes. The natural frequencies are the eigenvalues of the system matrix, and the mode shapes are the eigenvectors corresponding to these eigenvalues. These quantities provide insight into the system's vibrational modes and can be used to predict its response to various inputs.

The system's natural modes are the solutions to the characteristic equation, which is obtained by setting the determinant of the matrix (M - λI) to zero, where λ represents the natural frequency and I is the identity matrix. The characteristic equation is given by:

\[ \det(M - \lambda I) = 0 \]

By solving this equation, we obtain the natural frequencies λ of the system. The mode shapes are then calculated by finding the eigenvectors corresponding to each eigenvalue.

In summary, the analysis of the quarter-car model's passive dynamic behavior involves calculating the natural frequencies and mode shapes using the characteristic equation and eigenvector calculations. These results are crucial for understanding the system's vibrational characteristics and predicting its response to external forces.

**References**


**Figures**

- Figure 1: Schematic diagram of a quarter-car model showing the suspension components and their connections.
- Figure 2: Graphical representation of the characteristic equation and its roots.
- Figure 3: Natural frequencies and mode shapes for the quarter-car model versus different values of suspension stiffness and damping.
- Figure 4: Time-domain response of the quarter-car model to a step input with varying damping ratios.
- Figure 5: Frequency response of the quarter-car model to a harmonic input with different natural frequencies.

**Tables**

- Table 1: Comparison of natural frequencies and mode shapes for different suspension stiffness and damping values.
- Table 2: Summary of key findings and conclusions from the dynamic analysis of the quarter-car model.
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 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 $N_s$ (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 design constraints are presented below.

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

$$k_s = \frac{d^4G}{8D^3N_s(1 + \frac{1}{\pi^2})}$$  \hspace{1cm} (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

$$g_1(x_p) = 4 - C \leq 0$$  \hspace{1cm} (14)
$$g_2(x_p) = C - 12 \leq 0$$  \hspace{1cm} (15)

The following constraint prevents buckling:

$$g_3(x_p) = L_0 - 5.26D \leq 0$$  \hspace{1cm} (16)

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

$$g_4(x_p) = L_0 - L_{0\text{max}} \leq 0$$  \hspace{1cm} (17)

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

$$g_5(x_p) = d + D - D_{\text{omax}} \leq 0$$  \hspace{1cm} (18)

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

$$g_6(x_p) = d - D + D_p + 2(\delta_d + t_d) \leq 0$$  \hspace{1cm} (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 $\delta_{\text{max}}$ is calculated using a vehicle simulation with a ramp input (road grade of 25% at a speed of $v_1 = 10 \text{ m/s}$) to check for maximum rattlespace violation

$$ g_{10}(x_p, \Xi) = \delta_{\text{max}} - L_0 + L_s + L_h + \delta_b \leq 0 \quad (20) $$

This test requires specification of damper design, discussed in Sec. 3.2. $L_0$ is bumpstop thickness (0.02 m), and $\delta_b = m_s g/4 k_s$ is the static suspension deflection ($g = 9.81 \text{ m/s}^2$).

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

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

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

$$ g_{11}(x_p, \Xi) = \frac{\tau_s}{S_{sy}} \leq 0 \quad (21) $$

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 $S_{ut} = A 	imes 10^6/(d^2)$, $A = 1974 \text{ MPa mm}^2$, 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:

$$ g_{10}(x_p, \Xi) = 0.15 + 1 - \frac{L_0 - L_s}{\delta_s + 1.1 \delta_{\text{max}}} \leq 0 \quad (22) $$

where $\delta_{\text{max}}$ 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 $v_2 = 20 \text{ m/s}$. Under these rough road conditions, the maximum axial spring force is $F_{\text{max}} = k_s (\delta_{\text{max}} + \delta_b)$, and the minimum force is $F_{\text{min}} = k_s (\delta_b - \delta_{\text{max}})$. The mean axial force and force amplitude are $F_m = (F_{\text{max}} + F_{\text{min}})/2$ and $F_A = (F_{\text{max}} - F_{\text{min}})/2$, respectively. The mean shear stress and amplitude are

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

Soderberg fatigue criterion with Zimmerli data is applied here

$$ g_{10}(x_p, \Xi) = \frac{\tau_s}{S_{sy}} - 1 \leq 0, \text{ where } S_{sy} = 0.24 S_{ut}/n_d \quad (23) $$

$$ g_{11}(x_p, \Xi) = \frac{\tau_s n_d}{241 \times 10^6} \leq 0 \quad (24) $$

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 $\dot{\xi}_s$. Real suspension dampers are highly nonlinear due to both 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

$$ A_v(x_v) = C_2 C_0 \sqrt{x_v} \quad (25) $$

### Table 2: Spring design parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_s/4$</td>
<td>325 kg</td>
</tr>
<tr>
<td>$k_s$</td>
<td>232.5 $\times 10^5$ N/m</td>
</tr>
<tr>
<td>$Q$</td>
<td>1.75</td>
</tr>
<tr>
<td>$L_{s\text{max}}$</td>
<td>0.40 m</td>
</tr>
<tr>
<td>$l_0$</td>
<td>9 mm</td>
</tr>
<tr>
<td>$\delta_{s\text{c}}$</td>
<td>0.02 m</td>
</tr>
<tr>
<td>$A$</td>
<td>1974 MPa mm$^2$</td>
</tr>
<tr>
<td>$v_2$</td>
<td>10 m/s</td>
</tr>
</tbody>
</table>

---

Journal of Mechanical Design

AUGUST 2014, Vol. 136 / 081003-7
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

\[
\tilde{A}_v(x_v) = \int_0^{x_v} C_e(\tau) \, d\tau
\]

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

\[
\tilde{C}_e(x_v) = \frac{d}{dx_v} \tilde{A}_v(x_v) = \frac{1}{2} C_2 D_0 x_v^{1/2}
\]

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 \( C_e(x_v) \leq \pi C_0 \) for all \( x_v \geq 0 \). 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:

\[
C_e(x_v) = a_1 (x_v + a_2^2)^{-1/2}
\]

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

\[
C_3 = \frac{C_2 D_0}{2 \eta} \sqrt{\frac{\pi P_{\text{allow}}}{k_v}}
\]

The damper valve coefficient is defined here as \( C_2 = \eta A_1 \sqrt{x_m} \), where \( A_1 \) is an area that can be used to tune port shape (\( 0 < A_1 < 1 \)). Here, \( A_1 = 0.1 \). Increasing \( A_1 \) 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

\[
c_s = \frac{D_p^4}{8 \pi C_2 C_0 D_0} \sqrt{\frac{\pi k_v \rho_1}{2}}
\]

where \( D_p \) is the working piston diameter, \( \rho_1 \) is the damper fluid density (850 kg/m³), and \( C_d \) is the discharge coefficient (≈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

\[
x_p = [d, D, p, N_s, D_0, D_p, D_s]
\]

### 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., \( q_{\text{gen}} = C_e \xi \dot{\xi} \). 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 \( c_p = 2500 \text{ J/kgK} \), density \( \rho_1 = 850 \) kg/m³, and volume \( v_1 \). The fluid volume is \( v_1 = \pi D_p^2 (D_s + \varepsilon_1 + \varepsilon_3) / 4 \), where \( \varepsilon_1 = 0.02 \) m and \( \varepsilon_3 = 0.02 \) m represent the space required for damper valve and casting extensions, respectively. Heat is conducted through the steel damper tube (with conduction coefficient \( k_1 = 60.5 \) W/mK) to the atmosphere with a constant temperature \( T_0 = 300 \) K. The convection coefficient between the damper and the atmosphere is \( h = 50 \) W/m² K. Heat capacity in the steel shell is small compared to that of the damper fluid (\( c_1 \)) and is assumed to be zero here, so \( T_3 = T_4 \).

Heat flow from the damper fluid through the shell is

\[
q_0 = q_{\text{gen}} - \rho_1 v_1 c_p \frac{dT_1}{dr} = \frac{2n_2 k_3}{\ln r_2 / r_1} (T_1 - T_3) = hA_k (T_3 - T_4)
\]

where \( r_1 \) and \( r_2 \) are the inner and outer shell radii, \( L_2 = D_3 + \varepsilon_1 + \varepsilon_3 \) is the shell height, and \( A_k = 2 \pi r_L L_2 \) 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

\[
q_{\text{gen}} - b_1 T_1 = b_2 (T_1 - T_3) = b_3 (T_3 - T_4)
\]

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

\[
\dot{T}_1 = -\frac{b_2 b_3}{b_1 b_2 + b_1 b_3} T_1 + \frac{b_2 b_1}{b_1 b_2 + b_1 b_3} T_3 + \frac{q_{\text{gen}}}{b_1}
\]

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

### 3.2.3 Damper Constraints

\[ g_{12}(x_p) = L_0 - L_s - D_s \leq 0 \] (31)

ensures adequate damper range of motion

\[ g_{13}(x_p) = 2D_s + \varepsilon_1 + \varepsilon_2 - L_{0\text{max}} \leq 0 \] (32)

requires the damper to fit within the pocket length \( L_{0\text{max}} = 0.4 \) m; \( \varepsilon_1 = 0.02 \) m and \( \varepsilon_2 = 0.04 \) m quantify the size required for...
Table 3 Damper design parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>η</td>
<td>0.9</td>
</tr>
<tr>
<td>k_v</td>
<td>7500 N/m</td>
</tr>
<tr>
<td>ρ_1</td>
<td>850 kg/m^3</td>
</tr>
<tr>
<td>c_p</td>
<td>2500 J/kgK</td>
</tr>
<tr>
<td>T_3</td>
<td>300 K</td>
</tr>
<tr>
<td>x_41</td>
<td>0.02 m</td>
</tr>
<tr>
<td>x_42</td>
<td>0.04 m</td>
</tr>
<tr>
<td>v_3allow</td>
<td>5.0 m/s</td>
</tr>
<tr>
<td>p_3allow</td>
<td>4.75 x 10^6 Pa</td>
</tr>
</tbody>
</table>

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 plant constraint structure (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 g_p (·) and dynamic system model parameters p, such as stiffness k_v and damping c_p rates, depends on plant design 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 ξ (·) depend on state trajectories and control design. Here, control design is represented by the discretized control trajectory 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 (·)-g_12 (·) have dense dependence on 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 x_p and ti is clear from the presence of these variables in the system matrix A of Eq. (12). Columns corresponding to control input are omitted from Fig. 11 as plant design constraints do not depend directly on 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 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 (·) and g_2 (·) 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 columns for each state variable (one for every time step). For example, in constraint 16, we need to find the maximum velocity g (ξ) does not exceed g (ξ) where g (ξ) appears to be a nonlinear operation, we can compare g (ξ) = ξ_3 - ξ_2 to g (ξ) 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 \( x_p \) and a sparse diagonal dependence on \( \Xi \) and \( x_h \), 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 \( u(t) \) can be achieved, in some cases within maximum force limits. The control design variable \( u_p \) is a time-discretization of this actuation force trajectory, i.e.,

\[
\begin{align*}
0 & \\
& u_c \quad \text{where } u_c(u(t)).
\end{align*}
\]

The active suspension state-space model with the road disturbance and control input terms is

\[
\begin{align*}
\dot{x} = A x + \begin{bmatrix} -1 \\ \frac{4c_1}{m_{ru}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{m_c} \end{bmatrix} z_0 + \begin{bmatrix} 0 \\ -\frac{1}{m_{pu}} \\ \frac{1}{m_{pu}} \\ \frac{1}{m_s} \end{bmatrix} u
\end{align*}
\] (37)

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

\[
J = \int_0^{t_f} \left( r_1(z_0 - z)^2 + r_2 z_0^2 + r_3 u^2 \right) dt
\] (38)

where the weights \( r_1 = 10^3 \), \( r_2 = 0.5 \), 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:

\[
J = J_{\text{ramp}} + J_{\text{rough}}, \quad \text{where } J_{\text{ramp}} \text{ and } J_{\text{rough}} \text{ are the values of the above integral for the ramp and rough road inputs, respectively.}
\]

The plant design variable bounds \( x_p \leq x_p \leq x_p \) used here are

\[
\begin{align*}
x_p &= [0.005, 0.05, 0.02, 3, 0.003, 0.03, 0.1] \\
x_p &= [0.02, 0.4, 0.5, 16, 0.012, 0.08, 0.3]
\end{align*}
\]

The simultaneous solution approach was implemented using a large-scale interior-point algorithm from the MatLAB\textsuperscript{\textregistered} 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 \( h \) 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.
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).
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 understood by imposing explicit bounds on maximum control force. Co-design results with bounds near zero will be close to the passive dynamic properties reported here (i.e., free vibrations) that are independent of active control play at a system that has been optimized to work 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 x_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_u 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), \( \psi_1 \) and \( \psi_2 \). Each mode has a corresponding natural frequency \( \omega_n \); both masses will oscillate at frequency \( \omega_0 \) when mode shape \( \psi_1 \) 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 dynamics properties of physical systems should be designed differently for active control.

**Table 6 Comparison of optimal stiffness and damping rates**

<table>
<thead>
<tr>
<th></th>
<th>Sequential</th>
<th>Simultaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_c ) (N/m)</td>
<td>3.30 ( \times 10^4 )</td>
<td>2.36 ( \times 10^4 )</td>
</tr>
<tr>
<td>( c_s ) (Ns/m)</td>
<td>2.58 ( \times 10^3 )</td>
<td>1.03 ( \times 10^3 )</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>( u_{\text{max}} ) (N)</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
<th>2500</th>
<th>3000</th>
<th>4000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d ) (m)</td>
<td>0.0117</td>
<td>0.0117</td>
<td>0.0115</td>
<td>0.0113</td>
<td>0.0113</td>
<td>0.0108</td>
<td>0.0105</td>
<td>0.0104</td>
<td>0.0101</td>
<td>0.0098</td>
<td>0.0097</td>
</tr>
<tr>
<td>( D ) (m)</td>
<td>0.0752</td>
<td>0.0743</td>
<td>0.0743</td>
<td>0.0733</td>
<td>0.0727</td>
<td>0.0711</td>
<td>0.0675</td>
<td>0.0662</td>
<td>0.0631</td>
<td>0.0639</td>
<td>0.062</td>
</tr>
<tr>
<td>( p ) (m)</td>
<td>0.0239</td>
<td>0.0236</td>
<td>0.0237</td>
<td>0.0234</td>
<td>0.0232</td>
<td>0.0228</td>
<td>0.0217</td>
<td>0.0213</td>
<td>0.0203</td>
<td>0.0207</td>
<td>0.0201</td>
</tr>
<tr>
<td>( D_0 ) (m)</td>
<td>0.007</td>
<td>0.0069</td>
<td>0.0069</td>
<td>0.0069</td>
<td>0.0069</td>
<td>0.0068</td>
<td>0.0066</td>
<td>0.0065</td>
<td>0.0061</td>
<td>0.0064</td>
<td>0.0062</td>
</tr>
<tr>
<td>( D_s ) (m)</td>
<td>0.0415</td>
<td>0.0406</td>
<td>0.0408</td>
<td>0.04</td>
<td>0.0394</td>
<td>0.0382</td>
<td>0.035</td>
<td>0.0338</td>
<td>0.0311</td>
<td>0.0321</td>
<td>0.030</td>
</tr>
<tr>
<td>( d_c ) (10^4) N/m</td>
<td>3.30</td>
<td>3.32</td>
<td>3.15</td>
<td>3.10</td>
<td>3.07</td>
<td>2.82</td>
<td>2.68</td>
<td>2.64</td>
<td>2.53</td>
<td>2.35</td>
<td>2.36</td>
</tr>
<tr>
<td>( c_s ) (10^4) N/m</td>
<td>2.47</td>
<td>2.34</td>
<td>2.33</td>
<td>2.22</td>
<td>2.15</td>
<td>1.87</td>
<td>1.48</td>
<td>1.34</td>
<td>1.17</td>
<td>1.15</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Fig. 14 Pareto sets illustrating the tradeoff between maximum control force and system objective function value (both sequential and simultaneous solution approaches)
The mode shapes and natural frequencies in Table 8 were calculated based on a mass-normalized stiffness matrix, i.e., $K = M^{-1/2}KM^{-1/2}$, where $M$ and $K$ are the $2 \times 2$ 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 responds. While stiffness changes substantially as the control level is increased, both suspension mode shapes exhibit relatively little change. Sprung mass amplitude in $\psi_1$ does decrease some with increasing $u_{\text{max}}$ and unsprung mass amplitude in $\psi_2$ decreases with increasing $u_{\text{max}}$.

Modal damping values cannot be computed exactly from system matrices because this system does not exhibit proportional damping. An approximate modal damping value $\nu_i$ was calculated for each design by identifying a proportionally damped system (i.e., a system where the damping matrix is $\kappa_i$), for each design by identifying a proportionally damped system (i.e., a system where the damping matrix is $\kappa_i$). The modal damping for 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 $\Lambda$ from Eq. (37) were computed (Table 8). Both real and imaginary parts decrease in magnitude as $u_{\text{max}}$ 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 part 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 computationally 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
