

NESTED AND SIMULTANEOUS SOLUTION STRATEGIES FOR GENERAL COMBINED PLANT AND CONTROL DESIGN PROBLEMS

Daniel R. Herber*, James T. Allison

University of Illinois at Urbana-Champaign
Industrial & Enterprise Systems Engineering
Urbana, IL 61801

Email: {[herber1](mailto:herber1@illinois.edu), [jtalliso](mailto:jtalliso@illinois.edu)}@illinois.edu

Abstract

In this article, general combined plant and control design or co-design problems are examined. Previous work in co-design theory imposed restrictions on the type of problems that could be posed. This article lifts many of those restrictions. The problem formulations and optimality conditions for both the simultaneous and nested solution strategies are given. Due to a number of challenges associated with the optimality conditions, practical solution considerations are discussed with a focus on the motivating reasons for using direct transcription in co-design. This article highlights some of the key concepts in general co-design including general coupling, the differences between the feasible regions for each strategy, general boundary conditions, inequality path constraints, system-level objectives, and the complexity of the closed-form solutions. Three co-design test problems are provided. A number of research directions are proposed to further co-design theory including tailored solution methods for reducing total computational expense, better comparisons between the two solution strategies, and more realistic test problems.

1 Introduction

Many dynamic engineering system design problems contain two groupings of the design variables: plant (or artifact) and control. For instance, consider the system-level design of a robotic manipulator. The plant design may comprise the geometric properties of the links and the control design may be embodied by the joint torque time trajectories for a specific task [1]. Many authors have shown the benefit of a combined strategy rather than a sequential approach [2–6]. With a sequential approach, the plant is optimized initially, followed by the controller [6, 7]. In this article, we

*Corresponding author.

focus on two solution strategies appropriate for combined plant and controller design or co-design: *simultaneous* and *nested*.

The simultaneous solution strategy optimizes both the plant and control variables in same optimization formulation. With the nested strategy, an outer optimization loop optimizes the plant design, and an inner optimization loop identifies the optimal control for each plant design tested by the outer loop [6, 7]. These two strategies are selected for a few reasons beyond them being the most studied and used in the literature.

First, the simultaneous strategy is the most fundamental representation of an integrated design problem [8]. On the other hand, the nested strategy is a specific reorganization of the optimization problem based on the plant and control disciplines. Multidisciplinary design optimization (MDO) is a field of research that investigates design methods for systems with multiple disciplines [7, 8]. However, if we are limited to single-system problems and implementations that do not partition the system across trajectories, there are a limited number of appropriate MDO methods suitable for co-design. One reason for this is that many MDO formulations were developed in a way that does not explicitly address the dynamic nature of general co-design problems [7]. This has led to partitioning schemes without full consideration of the coupling in the system. Allison and Nazari developed an augmented Lagrangian coordination method for co-design, but only accounted for unidirectional (plant) coupling [9]. The nested approach naturally handles single-system problems, bidirectional coupling, and the various trajectories in the problem. Another important motivating reason for reorganizing the original problem is to employ specialized optimization algorithms to solve specific subproblems [9, 10]. A number of efficient solution methods for specific problem forms can be utilized with the nested strategy. However, as it will be discussed throughout this article, both of these strategies have their drawbacks. Many of these issues are well known for simultaneous strategy, motivating the fields of MDO and distributed optimization [8]. The nested strategy is not amenable to coarse-grained parallelism and can be computationally intensive [9]. Additionally, it can have potential feasibility issues.

Here we consider co-design problems that are well-posed as deterministic dynamic optimization (DO) problems [6, 7, 11]. Co-design theory has focused primarily on specific DO formulations; as a result, only limited types of co-design problems fit into the existing frameworks. With a specific problem structure, a variety of algorithms have been developed to provide numerical solutions, and detailed analysis of the coupling and partitioning has been investigated [6, 12–18]. There are even some promising tailored iterative sequential methods for solving specific co-design problems [19]. Such works have made considerable progress towards addressing specific challenges found in certain co-design problems. However, there has been less attention towards the general co-design problem (i.e., a co-design formulation with no restrictions) perhaps due to the lack of efficient solution techniques and continuing legacy design practices that treat certain problem elements as naturally separate [7].

Recently, numerical solution methods have been utilized to solve co-design problems that are not captured by previous problem definitions [7, 11]. Nontraditional problem formulation elements include system-level objective functions, general inequality path constraints, and general boundary conditions [1–5, 20–22]. With these nontraditional problem elements, much of the previous work in co-design theory does not apply. An essential element of engineering design optimization is an appropriate problem formulation. A thorough analysis of the general co-design formulation is needed to better allow designers to leverage the advantages intrinsic to the co-design methodology.

One of the numerical solution methods recently applied to co-design problems is known as

direct transcription (DT), which approximates the infinite-dimensional DO problem with a finite nonlinear program (NLP) that can be solved with standard NLP solvers [3, 23–25]. DT has a number of favorable properties that support the efficient generation of solutions to general co-design problems [3, 7, 11].

In this article, we will explore the general co-design problem with a focus on the simultaneous and nested solution strategies. Section 2 provides the formulations for the simultaneous and nested co-design solution strategies and Sec. 3 outlines the necessary conditions for optimality for both. Section 4 discusses some practical solution considerations relevant to the two strategies. Section 5 presents a number of test problems and finally, Sec. 6 offers the conclusions.

2 Problem Formulation

Here we present the two co-design strategies. First, we state a few assumptions. Initially, the time horizon $t_0 \leq t \leq t_f$ is assumed to be fixed. Section 3.4 discusses the inclusion of the horizon boundaries in the co-design problem formulations. Next, only general inequality constraints will directly appear in the formulations for conciseness of the formulations. Equality constraints are captured by two inequality constraints (i.e., $f(x) = 0$ is equivalent to $f(x) \leq 0$ and $-f(x) \leq 0$).

2.1 Simultaneous Formulation

The general simultaneous co-design problem formulation contains all relevant objectives and constraints in a single optimization formulation. The DO problem formulation is:

$$\min_{\mathbf{x}_p, \mathbf{x}_c} \Psi(\mathbf{x}_p, \mathbf{x}_c) = \int_{t_0}^{t_f} \mathcal{L}(t, \boldsymbol{\xi}, \mathbf{x}_c, \mathbf{x}_p) dt + \dots \quad (1a)$$

$$\mathcal{M}(\boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f), \mathbf{x}_c, \mathbf{x}_p) \quad (1b)$$

$$\text{subject to: } \dot{\boldsymbol{\xi}} - \mathbf{f}(t, \boldsymbol{\xi}, \mathbf{x}_c, \mathbf{x}_p) = \mathbf{0} \quad (1c)$$

$$\mathbf{C}(t, \boldsymbol{\xi}, \mathbf{x}_c, \mathbf{x}_p) \leq \mathbf{0} \quad (1d)$$

$$\boldsymbol{\phi}(\boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f), \mathbf{x}_c, \mathbf{x}_p) \leq \mathbf{0} \quad (1e)$$

where \mathbf{x}_p are the plant variables, \mathbf{x}_c are the control variables, t is the time continuum defined between t_0 and t_f , $\boldsymbol{\xi}$ are the states, Ψ is the objective function in Bolza form, \mathcal{L} is the Lagrange (running cost) term, \mathcal{M} is the Mayer (terminal cost) term, Eqn. (1c) enforces the dynamics modeled as a first-order ordinary differential equation, Eqn. (1d) enforces any time-varying path constraints, and Eqn. (1e) enforces any time-independent constraints. See Ref. [11] for a general discussion of this co-design formulation and the problem elements.

2.2 Nested Formulation

The alternative solution strategy to the simultaneous formulation in Problem (1) is a nested one where an outer optimization loop optimizes the plant design, and an inner optimization loop identifies the optimal control for each plant design tested by the outer loop. Figure 1 illustrates the two

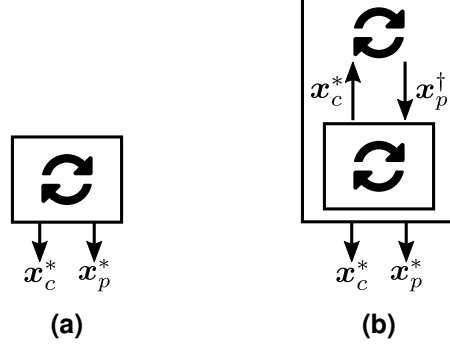


FIGURE 1: Two co-design solution strategies where \curvearrowright indicates an optimization problem: **(a)** simultaneous strategy; **(b)** nested strategy.

strategies. This is a two-level optimization problem (a subclass of bilevel optimization) [26–28]. The outer-loop problem is defined as:

$$\min_{x_p} \psi(x_p) \quad (2a)$$

$$\text{subject to: } \left. \begin{array}{l} \phi_o(x_p) \leq 0 \\ F(x_p) \leq 0 \end{array} \right\} = g_o(x_p) \leq 0 \quad (2b)$$

$$(2c)$$

where ϕ_o are the constraints in ϕ that only depend on the plant design, g_o is the collection of the outer-loop constraints, and $\{\psi(x_p), F(x_p)\}$ are two new problem formulation elements that will be outlined after the following inner-loop formulation:

$$\min_{x_c} \Psi(x_p^\dagger, x_c) \quad (3a)$$

$$\text{subject to: } \left. \begin{array}{l} \dot{\xi} - f(t, \xi, x_c, x_p^\dagger) = 0 \\ C(t, \xi, x_c, x_p^\dagger) \leq 0 \\ \phi_i(\xi(t_0), \xi(t_f), x_c, x_p^\dagger) \leq 0 \end{array} \right\} = g_i(\cdot) \leq 0 \quad (3b)$$

$$(3c)$$

$$(3d)$$

where x_p^\dagger is a candidate plant design, ϕ_i are the constraints of ϕ not in ϕ_o , and g_i is the collection of the inner-loop constraints. With both formulations presented, some additional concepts relevant to two-level optimization problems need to be discussed.

The simultaneous method's feasible set (or constraint region) is defined as:

$$\Omega = \{(x_p, x_c) : g_o(x_p) \leq 0 \text{ and } g_i(x_p, x_c) \leq 0\} \quad (4)$$

For a given x_p , the inner-loop feasible set is defined by:

$$\Lambda = \{x_c : g_i(x_p, x_c) \leq 0\} \quad (5)$$

and then set of potential inner-loop optimal control designs is:

$$M = \{x_c : x_c \in \arg \min \{\Psi(x_p, x_c) : x_c \in \Lambda\}\} \quad (6)$$

For a given \mathbf{x}_p , M may be empty for some values of its argument, i.e., no feasible control design exists for the given \mathbf{x}_p . If $\mathbf{x}_c \in M$, then the optimal objective function value from the inner loop is:

$$\psi(\mathbf{x}_p) = \Psi(\mathbf{x}_p, \mathbf{x}_c) \quad (7)$$

which is used as the objective function of the outer loop, Eqn. (2a). Finally, the feasible set of the outer loop (known as the induced region) is:

$$I = \{(\mathbf{x}_p, \mathbf{x}_c) : (\mathbf{x}_p, \mathbf{x}_c) \in \Omega \text{ and } \mathbf{x}_c \in M\} \quad (8)$$

and we note that Ω and I are not the same feasible region. This is the motivation behind the addition of Eqn. (2c). $F(\mathbf{x}_p)$ is an additional constraint that may be added to ensure that for all candidate \mathbf{x}_p^\dagger , Λ^\dagger (i.e., Λ with \mathbf{x}_p^\dagger) is nonempty, and is termed the outer-loop feasibility constraint. We distinguish between \mathbf{x}_p and \mathbf{x}_p^\dagger because not all points in Λ will be evaluated when using the nested solution strategy, only a specific set of candidate plant designs. The nested strategy then limits the feasible directions when compared to the simultaneous strategy. The two formulations are only equivalent if Ω remains unchanged. This constraint will be discussed more in the following discussion section.

2.3 Formulation Discussion

Here we discuss the specific problem formulation elements of both approaches, focusing on comparisons to the current co-design literature.

2.3.1 Control Design Variables

While plant variables will always be time-independent, control variables may be either time-independent or infinite-dimensional trajectories (with respect to time). We partition the control variables into two sets as:

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{p} \\ \mathbf{u} \end{bmatrix} \quad (9)$$

where \mathbf{p} are the control parameters and \mathbf{u} are the open-loop control (OLC) variables. The control parameters are static (time-independent) control variables (e.g., gains). The OLC variables are infinite-dimensional trajectories through time (e.g., joint torque time profile for a robotic manipulator). Both types of control design variables are found in the literature, although both types are not typically in the same design problem. Even if there are no OLC variables, the general co-design problem is still an infinite-dimensional problem due to the dynamic and path constraints. One very special case is when the optimal OLC can be defined with static gains such as with infinite-horizon linear-quadratic regulator (LQR) [29–31] discussed later in Sec. 4.2.2. See Refs. [2, 5, 15, 16, 19, 30, 31] for examples with \mathbf{p} and Refs. [3, 4, 20–22] for examples with \mathbf{u} .

2.3.2 Objective Function

Even though the objective function is partitioned in Eqns. (1a) and (1b), converting between the Lagrange and Mayer terms is possible [11, 29]. Both forms are present to allow for more natural formulations of the problem (and to be consistent with the literature).

A common form of the co-design objective function is:

$$\min_{\mathbf{x}_p, \mathbf{x}_c} \Psi = w_p \Psi_p(\mathbf{x}_p) + w_c \Psi_c(\mathbf{x}_p, \mathbf{x}_c) \quad (10)$$

where $\{\Psi_p, \Psi_c\}$ are the plant and control objective functions and $\{w_p, w_c\}$ are objective weights [6]. While many co-design problems are appropriately partitioned with control and plant specific objective function terms [2, 12–14, 16–18, 31], others necessitate the general objective form in Eqns. (1a)–(1b) [1, 2, 4, 20, 22]. These general objective functions are defined at the system level as they do not necessarily need to be partitioned into control and plant specific terms. Frequently, these system-level objectives may include one term or terms that are typically only classified as “control” objective terms.

2.3.3 Path and Boundary Constraints

Equations (1d) and (1e) could be further generalized into a single general constraint form, but the distinction is important for both the optimality conditions and the numerical approaches used to find solutions. The difference is primarily based on time dependence. Therefore, path and boundary constraints could be more appropriately named time-dependent and time-independent constraints.

Boundary constraints are common to many existing co-design formulations [6, 7, 11]. Traditional plant-only constraints such geometry or mass constraints are time-independent constraints [3, 17, 18, 31]. Many co-design formulations explicitly require initial condition constraints, but some co-design problems have unknown initial conditions or periodic constraints [20]. Another set of boundary constraints is the kinematic relationships in robotics [32]. These constraints might require an algebraic variable that depends on the states (e.g., the end position of the robotic manipulator depends on the state joint angles and geometric plant variables).

More recently, general path constraints have been included in co-design problems and are infinite-dimensional constraints [1, 3, 7, 11]. Many traditional engineering constraints can be formulated naturally as path constraints. States or outputs often need to be constrained between allowable bounds such as temperature, position, force, pressure, deflection, stress, power, etc. These examples are typically inequality path constraints but equality path constraints are also possible, such as an automobile following a prescribed drive cycle or ensuring the steady-state optimal tip-speed ratio is followed by a horizontal axis wind turbine [4]. The inclusion of path constraints is critical to the usefulness of many co-design studies. Fathy et al. included mixed control-plant constraints:

$$C(t, \mathbf{u}(t), \mathbf{x}_p) \leq 0 \quad (11)$$

but mixed state-control-plant constraints as in Eqn. (1d) are needed to represent many of the engineering constraints mentioned above [6].

2.3.4 Outer-Loop Feasibility Constraint

For the two solution strategies to be successful at finding equivalent system optimal designs, Λ^\dagger must be nonempty for every candidate \mathbf{x}_p^\dagger [6]. Equivalence here means that given an arbitrary starting point using the nested solution strategy, a converged point is only a minimizer with respect to the simultaneous strategy’s optimality conditions, if and only if Λ^\dagger is nonempty for every

considered \mathbf{x}_p^\dagger . Note, this does not imply that the two strategies will necessarily converge to the same system optima given identical starting conditions. Since the nested strategy limits the search space/directions relative to the simultaneous strategy, the converged points may be different in non-convex problems. Introducing an appropriate outer-loop feasibility constraint is one technique to ensure this property is satisfied.

Consider again the design of a robotic manipulator. For certain geometric configurations, the kinematics restrictions of the manipulator may be such that it is unable to reach the prescribed target. To avoid a candidate \mathbf{x}_p that has no feasible control solution, we can add a reachability constraint to $\mathbf{F}(\mathbf{x}_p)$ [1,32]. However, even with this additional constraint, mixed state-control-plant constraints may be present that cannot be satisfied with the given plant design, such as stress or deflections [1]. This example helps illustrate that for general co-design problems, these constraints can be quite difficult (or impossible) to determine.

An important type feasibility constraint is the classic controllability property applicable to linear systems theory [6]. Unfortunately, controllability is not sufficient to guarantee the nonemptiness of Λ^\dagger for general co-design problems. For example, if control bounds are provided, then it is not guaranteed that we can move between arbitrary initial and final positions in finite time.

Many gradient-based optimization schemes can fail when encountering undefined objective function values, especially if it is challenging to determine suitable feasible values/directions. One benefit of including an appropriate constraint $\mathbf{F}(\cdot)$ is that the feasible directions are now better known in the outer loop. Many optimizers can try to manage such undefined objective function values, so it is important to consider the feasibility properties of the co-design problem when selecting a suitable optimization algorithm for the outer loop. For example, global search strategies or non-gradient methods could be used. Good (i.e., feasible or near-feasible) starting points can help avoid these issues.

It is also important to note that this outer-loop feasibility constraint may not strictly be required to find the system-optimal solution using the simultaneous strategy. Another strategy is to add a feasibility constraint that restricts Ω such that the desired property holds. This is not ideal since there will be no guarantee that the nested strategy will produce the system-level optimum, but might be a necessity for certain co-design problems. The implications of this will be discussed in the following sections.

3 Necessary Conditions for Optimality

In this section, we describe the necessary conditions for optimality for the two general co-design strategies from the previous section. Sufficient conditions will not be discussed directly, but are the Hamiltonian minimization conditions [29,33,34] or second-order conditions on the Lagrangian for finite-dimensional problems [35]. If a problem is singular, there may be additional sufficient conditions needed [33, 34]. We also assume that the simultaneous co-design problem is well-posed, i.e., there exists a solution. See Refs. [29,33,35] for discussions on constraint qualification, regularity, differentiability, and other relevant properties for a well-posed problem. We begin with the conditions for the OLC design only.

3.1 Open-Loop Control Design

The optimal OLC design problem formulation is:

$$\min_u \Psi = \int_{t_0}^{t_f} \mathcal{L}(t, \xi, \mathbf{u}) dt + \mathcal{M}(\xi(t_0), \xi(t_f)) \quad (12a)$$

$$\text{subject to: } \dot{\xi} - \mathbf{f}(t, \xi, \mathbf{u}) = \mathbf{0} \quad (12b)$$

$$\mathbf{C}(t, \xi, \mathbf{u}) \leq \mathbf{0} \quad (12c)$$

$$\phi(\xi(t_0), \xi(t_f)) \leq \mathbf{0} \quad (12d)$$

which is a subformulation of Problem (1). Similar to the Lagrangian in finite-dimensional optimization [29, 35], the infinite-dimensional constraints can be adjoined to the Lagrange term with time-varying Lagrange multipliers, creating the Hamiltonian of the problem:

$$H = \mathcal{L} + \boldsymbol{\lambda}^\top \mathbf{f} + \boldsymbol{\mu}^\top \mathbf{C} \quad (13)$$

where $\boldsymbol{\lambda}(t)$ are the costates (multipliers for the state dynamics) and $\boldsymbol{\mu}(t)$ are the multipliers for \mathbf{C} .

With this control-only formulation, we can directly apply Pontryagin's minimum principle (PMP) with path constraints [29, 33, 36] to arrive at the following necessary conditions for optimality (note, \square^* is used to denote an optimal value):

$$\dot{\boldsymbol{\lambda}}^* = - \left[\frac{\partial H}{\partial \xi} \right]^* \quad (14a)$$

$$\mathbf{0} = \left[\frac{\partial H}{\partial \mathbf{u}} \right]^* \quad (14b)$$

$$\mathbf{0} = [\boldsymbol{\mu}^\top \mathbf{C}]^*, \quad \mathbf{0} = [\boldsymbol{\nu}^\top \phi]^* \quad (14c)$$

$$\boldsymbol{\mu}^* \geq \mathbf{0}, \quad \boldsymbol{\nu}^* \geq \mathbf{0} \quad (14d)$$

$$\mathbf{0} = \left[\boldsymbol{\lambda} + \frac{\partial \mathcal{M}}{\partial \xi} + \boldsymbol{\nu}^\top \frac{\partial \phi}{\partial \xi} \right]_{t_0}^*, \quad \mathbf{0} = \left[\boldsymbol{\lambda} - \frac{\partial \mathcal{M}}{\partial \xi} - \boldsymbol{\nu}^\top \frac{\partial \phi}{\partial \xi} \right]_{t_f}^* \quad (14e)$$

where $\boldsymbol{\nu}$ are the Lagrange multipliers for ϕ , Eqn. (14a) is the costate dynamics, Eqn. (14b) is the control stationarity condition, Eqns. (14c) are the complementary slackness conditions, Eqns. (14d) are the dual feasibility conditions, and Eqns. (14e) are the initial and final time transversality conditions. The conditions in Eqn. (14) are in addition to the constraints in Eqns. (12b)–(12d). The necessary conditions for both strategies can now be derived.

3.2 Co-Design—Simultaneous Strategy

Here we will derive the simultaneous co-design optimality conditions only using PMP. Others have derived the optimality conditions for similar co-design formulations, but only for very restricted problem types with significant simplifying assumptions (e.g., unidirectional coupling), and not for the much more general class of co-design problem formulations addressed here. See Ref. [37] for early work outside the co-design context. Fathy et al. utilized a combination of the Karush-Kuhn-Tucker (KKT) conditions and PMP [6].

Consider the following augmented state vector:

$$\Theta = \begin{bmatrix} \xi \\ \mathbf{x}_p \end{bmatrix}, \quad \dot{\Theta} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \quad (15)$$

with the replacement of all plant variables with $\mathbf{x}_p(t_0)$ and, therefore, no dependence on $\mathbf{x}_p(t_f)$. Now consider if unconstrained, the choice of initial and final states are, in effect, additional decision variables that can be modified to satisfy the optimality conditions.

The ξ -dependent terms in Eqn. (14) are Eqns. (14a) and (14e). Applying these conditions to the “state” plant variables, we have the following additional conditions:

$$-\dot{\lambda}_p^* = \left[\frac{\partial H}{\partial \mathbf{x}_p} \right]^* = \left[\frac{\partial \mathcal{L}}{\partial \mathbf{x}_p} + \lambda^\top \frac{\partial \mathbf{f}}{\partial \mathbf{x}_p} + \lambda_p^\top \frac{\partial \dot{\mathbf{x}}_p}{\partial \mathbf{x}_p} + \mu^\top \frac{\partial \mathbf{C}}{\partial \mathbf{x}_p} \right]^* \quad (16a)$$

$$\mathbf{0} = \left[\lambda_p + \frac{\partial \mathcal{M}}{\partial \mathbf{x}_p} + \nu^\top \frac{\partial \phi}{\partial \mathbf{x}_p} \right]_{t_0}^* \quad (16b)$$

$$\mathbf{0} = \left[\lambda_p - \frac{\partial \mathcal{M}}{\partial \mathbf{x}_p} - \nu^\top \frac{\partial \phi}{\partial \mathbf{x}_p} \right]_{t_f}^* = \lambda_p^*(t_f) \quad (16c)$$

Now using the first fundamental theorem of calculus, we can combine these equations into a single condition:

$$\lambda_p^*(t_f) = \lambda_p^*(t_0) + \int_{t_0}^{t_f} \dot{\lambda}_p^* dt \quad (17a)$$

$$\mathbf{0} = \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}_p} + \nu^\top \frac{\partial \phi}{\partial \mathbf{x}_p} \right]_{t_0}^* + \int_{t_0}^{t_f} \left[\frac{\partial \mathcal{L}}{\partial \mathbf{x}_p} + \lambda^\top \frac{\partial \mathbf{f}}{\partial \mathbf{x}_p} + \mu^\top \frac{\partial \mathbf{C}}{\partial \mathbf{x}_p} \right]^* dt \quad (17b)$$

This condition is analogous to the condition found in other derivations of the optimality conditions [6, 37]. Note that the other conditions in Eqn. (14) remain unchanged. Therefore, the simultaneous co-design optimality conditions are a combination of Eqn. (14) and Eqn. (17b):

$$\dot{\lambda}^* = - \left[\frac{\partial H}{\partial \xi} \right]^*, \quad \mathbf{0} = \left[\frac{\partial H}{\partial \mathbf{u}} \right]^* \quad (18a)$$

$$\mathbf{0} = [\mu^\top \mathbf{C}]^*, \quad \mathbf{0} = [\nu^\top \phi]^*, \quad \mu^* \geq \mathbf{0}, \quad \nu^* \geq \mathbf{0} \quad (18b)$$

$$\mathbf{0} = \left[\lambda + \frac{\partial \mathcal{M}}{\partial \xi} + \nu^\top \frac{\partial \phi}{\partial \xi} \right]_{t_0}^*, \quad \mathbf{0} = \left[\lambda - \frac{\partial \mathcal{M}}{\partial \xi} - \nu^\top \frac{\partial \phi}{\partial \xi} \right]_{t_f}^* \quad (18c)$$

$$\mathbf{0} = \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}_p} + \nu^\top \frac{\partial \phi}{\partial \mathbf{x}_p} \right]_{t_0}^* + \int_{t_0}^{t_f} \left[\frac{\partial \mathcal{L}}{\partial \mathbf{x}_p} + \lambda^\top \frac{\partial \mathbf{f}}{\partial \mathbf{x}_p} + \mu^\top \frac{\partial \mathbf{C}}{\partial \mathbf{x}_p} \right]^* dt \quad (18d)$$

3.3 Co-Design—Nested Strategy

We now derive the necessary conditions for the nested strategy where the outer-loop problem is defined in Problem (2) and the inner loop in Problem (3). The inner-loop optimality conditions

are defined in Eqn. (14) with only OLC variables. If the inner loop contains \mathbf{p} , then the necessary conditions are analogous to the simultaneous conditions in Eqn. (18) with \mathbf{p} replacing \mathbf{x}_p .

The outer-loop problem is finite-dimensional optimization since neither the dynamics nor path constraints are present. Therefore, we can directly utilize the KKT conditions [35]. The Lagrangian for the outer-loop problem is:

$$L = \psi + \chi^\top \phi_o + \eta^\top \mathbf{F} \quad (19)$$

We first express the stationarity condition:

$$\mathbf{0} = \frac{dL}{d\mathbf{x}_p} = \frac{d\psi}{d\mathbf{x}_p} + \chi^\top \frac{d\phi_o}{d\mathbf{x}_p} + \eta^\top \frac{d\mathbf{F}}{d\mathbf{x}_p} \quad (20a)$$

$$= \int_{t_0}^{t_f} \frac{d\mathcal{L}}{d\mathbf{x}_p} dt + \frac{dM}{d\mathbf{x}_p} + \chi^\top \frac{d\phi_o}{d\mathbf{x}_p} + \eta^\top \frac{d\mathbf{F}}{d\mathbf{x}_p} \quad (20b)$$

Now the total derivative terms can be explicated [38]:

$$\frac{d\mathcal{L}}{d\mathbf{x}_p} = \frac{\partial \mathcal{L}}{\partial \mathbf{x}_p} + \frac{\partial \mathcal{L}}{\partial \xi^*} \frac{d\xi^*}{d\mathbf{x}_p} + \frac{\partial \mathcal{L}}{\partial \mathbf{u}^*} \frac{d\mathbf{u}^*}{d\mathbf{x}_p} + \frac{\partial \mathcal{L}}{\partial \mathbf{p}^*} \frac{d\mathbf{p}^*}{d\mathbf{x}_p} \quad (21a)$$

$$\frac{dM}{d\mathbf{x}_p} = \frac{\partial M}{\partial \mathbf{x}_p} + \left[\frac{\partial M}{\partial \xi^*} \frac{d\xi^*}{d\mathbf{x}_p} \right]_{t_0} + \left[\frac{\partial M}{\partial \xi^*} \frac{d\xi^*}{d\mathbf{x}_p} \right]_{t_f} + \frac{\partial M}{\partial \mathbf{p}^*} \frac{d\mathbf{p}^*}{d\mathbf{x}_p} \quad (21b)$$

Here we see a number of coupled terms.

The complete set of necessary conditions for the outer loop are then:

$$\mathbf{0} = \frac{d\Psi}{d\mathbf{x}_p} + \chi^\top \frac{d\phi_o}{d\mathbf{x}_p} + \eta^\top \frac{d\mathbf{F}}{d\mathbf{x}_p} \quad (22a)$$

$$\chi \geq \mathbf{0}, \quad \eta \geq \mathbf{0}, \quad \chi^\top \phi_o = \mathbf{0}, \quad \eta^\top \mathbf{F} = \mathbf{0} \quad (22b)$$

which are a set of KKT conditions. The conditions in Eqn. (22) are in addition to the constraints Eqns. (2b) and (2c).

3.4 Free Time

The initial and final time values can be optimization variables as well [6, 11, 29, 33]. Optimal control problems frequently contain these terms and therefore, appropriate optimality conditions are provided. Subsequently, these additional optimization variables are typically included in the inner-loop problem of the nested co-design strategy [6]. However, some of the numerical methods used jointly with certain co-design strategies lose certain properties when the time horizon is not fixed (e.g., the specific finite-horizon inner-loop problems that will be discussed in Sec. 4.2.1 that can be approximated as a convex quadratic program can no longer be approximated in this manner when considering a variable horizon.). In these situations, it may be more beneficial to place these additional variables in the outer loop to maintain such properties.

The simultaneous co-design strategy will need one additional equation if t_f is free:

$$\mathbf{0} = \left[\frac{\partial M}{\partial t_f} + \nu^\top \frac{\partial \phi}{\partial t_f} \right]^* + \left[\mathcal{L} + \lambda^\top \mathbf{f} + \mu^\top \mathbf{C} \right]_{t_f}^* \quad (23)$$

This is an additional transversality condition with the others in Eqn. (14e) [29, 33].

For a nested approach with t_f free in the outer loop, the additional stationary condition $dL/dt_f = 0$ would include:

$$\frac{d\mathcal{L}}{dt_f} = \frac{\partial\mathcal{L}}{\partial t_f} + \frac{\partial\mathcal{L}}{\partial\xi^*} \frac{d\xi^*}{dt_f} + \frac{\partial\mathcal{L}}{\partial u^*} \frac{du^*}{dt_f} + \frac{\partial\mathcal{L}}{\partial p^*} \frac{dp^*}{dt_f} \quad (24a)$$

$$\frac{d\mathcal{M}}{dt_f} = \frac{\partial\mathcal{M}}{\partial t_f} + \left[\frac{\partial\mathcal{M}}{\partial\xi^*} \frac{d\xi^*}{dt_f} \right]_{t_0} + \left[\frac{\partial\mathcal{M}}{\partial\xi^*} \frac{d\xi^*}{dt_f} \right]_{t_f} + \frac{\partial\mathcal{M}}{\partial p^*} \frac{dp^*}{dt_f} \quad (24b)$$

This completes the necessary conditions for optimality for the various problem formulations and solution strategies. Now some of the computational and practical aspects will be discussed.

4 Practical Solution Considerations

Obtaining solutions only using the optimality conditions in the previous section can be challenging. Practical formulation limitations, robustness, and computational efficiency also need to be considered when seeking solutions to co-design problems.

4.1 Obtaining Approximate Solutions

In the previous section, finding an analytic solution that satisfies the optimality conditions is called an indirect method [23, 24]. Although this approach can lead to important insights into the structure of the solution, it can be quite challenging (or impossible) to solve these equations analytically. If complicated black box functions or table interpolation is used, analytic forms for their derivatives do not even exist. Therefore, numerical methods are often employed that provide approximate solutions to the original problem.

Numeric indirect methods derive explicitly the optimality conditions and then use discretization to form a boundary value problem (BVP) [24]. If inequality path constraints are present, multiple BVPs will need to be linked for each arc that a path constraint is active since a new set of differential algebraic equations will need to be solved. The number of constrained subarcs and the sequence of constrained/unconstrained arcs, however, are unknown a priori, so it is quite difficult or even impossible to construct the correct BVP.

Another issue with numeric indirect methods is standard solution procedures are not robust. An initial guess for the costates must be given, but these are not physical quantities, so there typically is no good way of providing a reasonable estimate [23]. Furthermore, even with a reasonable guess, the numerical solution of the costate dynamics in Eqn. (14a) can be very ill-conditioned [34].

An alternative is direct methods [23, 24]. Instead of stating the optimality conditions, the control and/or state are parametrized using function approximation and the objective function is approximated using numerical quadrature. This creates a discrete, finite-dimensional problem that then is optimized using large-scale NLP solvers [24].

The first class of direct methods is the sequential (or shooting) methods, which only parametrize the control. Given initial conditions and a set of control parameters, the differential-algebraic equation (DAE) model is solved through conventional DAE solvers (forward simulation) such as a Runge-Kutta method. Due to the use of conventional DAE solvers, this approach has the advantage of easily finding feasible solutions to the state equations but needs to perform a full simulation

for each perturbation in the optimization algorithm. Repeated numerical integration of the DAE model, however, does not guarantee convergence with open-loop unstable systems [24] and the resulting solution can be very sensitive to the choice of control. This strategy is the easiest to construct out of all the direct methods since reliable and efficient codes for DAE and NLP solvers are naturally linked [24].

Sequential approaches typically produce low-accuracy solutions and are computationally inefficient [23–25, 39]. Many of these issues are due to the inability to handle path and boundary constraints efficiently. Since the states are calculated through a forward simulation, we must approximate numerically how local control perturbations will affect the global state trajectory. This leads to simultaneous approaches that forgo the nested analysis of sequential methods for a large set of constraints¹.

4.1.1 Direct Transcription

The simultaneous approach, also as known as direct transcription (DT), parametrizes both the state and control trajectories. Most DT methods are represented by the following NLP formulation:

$$\min_{U, \Xi, x_p} \sum_{k=0}^{N_t} w_k \mathcal{L}(t_k, \xi(t_k), u(t_k), x_p) + \mathcal{M}(\xi(t_0), \xi(t_f), x_p) \quad (25a)$$

$$\text{subject to: } \zeta(t, \Xi, U, x_p) = 0 \quad (25b)$$

$$C(t, \Xi, U, x_p) \leq 0 \quad (25c)$$

$$\phi(\xi(t_0), \xi(t_f), x_p) \leq 0 \quad (25d)$$

where N_t+1 is the number of discrete time points and $\{t, \Xi, U\}$ are the discretized forms of the time, states, and controls. The Lagrange term is approximated with numerical quadrature with weights w_k . The dynamic constraint is now enforced through a large number of equality constraints ζ , termed defect constraints [23–25]. Also, path constraints are now no more complicated than the dynamic constraints, a key advantage over other solution methods [24, 40].

This new large NLP formulation has a specific structure and sparsity pattern that can be exploited in NLP solvers to reduce total computational effort [24]. Simultaneous approaches have been shown to have good convergence properties and handle unstable DAEs [39, 41]. Finally, these approaches have specific advantages for singular control problems and high-index path constraints [41]. This collection of desirable properties makes DT a strong candidate for finding solutions to general co-design problems.

General overviews of DT theory are available in Refs. [3, 11, 23–25, 41]. Two common direct transcription schemes include 1) approximating the defect constraints using the trapezoidal rule and composite trapezoidal quadrature [11, 23, 24]; and 2) approximating the defect constraints with Lagrange interpolating polynomials on a Legendre-Gauss-Lobatto time grid (pseudospectral methods) and Gaussian quadrature [25, 39, 40]. Please see Refs. [3, 7, 11] for further discussion on

¹The sequential and simultaneous methods in this section are entirely different than the sequential design method mentioned in Sec. 1 and the simultaneous co-design method [3, 7, 24]. The methods in this section find approximate solutions to DO problems. The design methods are alternative solution methods for problems with plant and control design variables. For example, one could pose a simultaneous co-design problem and solve it with a sequential (or shooting) method for dynamic optimization.

using DT to find solutions to co-design problems. In these co-design studies, DT was demonstrated to be a suitable method for finding solutions to co-design problems that are more general than problems solved using earlier theory and methods.

4.2 Specific Forms

In this article, we have presented the most general form of the co-design problem: both a nonlinear plant outer-loop and nonlinear control inner-loop. However, specific forms of the control subproblem have been utilized heavily in a variety of co-design studies. The primary motivation for these forms is the efficient generation of solutions for the inner-loop subproblem. These two forms are linear-quadratic dynamic optimization (LQDO) and the linear-quadratic regulator (LQR).

4.2.1 Linear-Quadratic Dynamic Optimization

Consider the following specific form of the control subproblem in Problem (12):

$$\min_u \int_{t_0}^{t_f} \left(\begin{bmatrix} \xi \\ u \end{bmatrix}^\top \begin{bmatrix} Q & N \\ N & R \end{bmatrix} \begin{bmatrix} \xi \\ u \end{bmatrix} + \begin{bmatrix} q \\ r \end{bmatrix}^\top \begin{bmatrix} \xi \\ u \end{bmatrix} \right) dt + \dots \quad (26a)$$

$$\begin{bmatrix} \xi^\top M \xi + m^\top \xi \end{bmatrix}_{t_0} + \begin{bmatrix} \xi^\top S \xi + s^\top \xi \end{bmatrix}_{t_f} \quad (26b)$$

$$\text{subject to: } \dot{\xi} - (A\xi + Bu + d) = 0 \quad (26c)$$

$$C_1 \xi + C_2 u - C_3 \leq 0 \quad (26d)$$

$$\phi_1 \xi(t_0) + \phi_2 \xi(t_f) - \phi_3 \leq 0 \quad (26e)$$

where t_f is finite and any of the matrices (with the exception of $\{\phi, M, m, S, s\}$) can be time-varying. This particular LQDO problem can be approximated as a finite-dimensional quadratic program (QP) using a DT method (see Ref. [42]) for a more general LQDO problem form). Both single-step and pseudospectral DT methods can generate sparse QPs for this problem form [11, 24]. Under certain conditions, the QP is convex and efficient to solve for even large systems and the global optimal solution for u is guaranteed.

This form has been utilized in a number of nested co-design studies [11, 22]. The simultaneous strategy cannot be applied with this form since most (if not all) co-design dynamic constraints have a nonlinear dependence on the optimization variables [6]. For example consider the following bilinear dynamic constraint: $\dot{\xi} = -k\xi$, where k is a plant optimization variable. This dynamic constraint would need to be approximated by quadratic constraints.

4.2.2 Infinite-Horizon Linear-Quadratic Regulator

Consider the following special form of the control-only subproblem: Problem (26) containing only time-invariant matrices $\{Q \geq 0, R > 0, A, B\}$, no path constraints, simple initial value constraints ($\xi(t_0) = \xi_0$), and is infinite-horizon ($t_f \rightarrow \infty$). The optimal OLC can then be readily computed with the solution of the algebraic Riccati equation (ARE):

$$PA + A^\top P - PBR^{-1}B^\top P + Q = 0 \quad (27)$$

where P is the unique, positive definite solution and u is:

$$u^* = -R^{-1}B^T P^* \xi^* \quad (28)$$

This problem form is known as the infinite-horizon, continuous-time LQR [29–31], and is quite heavily utilized in co-design studies [2, 15, 16, 30, 31]. Solutions of this form can be computed efficiently and have a number of favorable properties such as a closed-form solution that can be used to create an equivalent simultaneous formulation [6]. There are additional forms (e.g., finite-time and time-varying versions) of this type of problem [29, 34] but the computation cost quickly becomes comparable with highly structured QPs using DT solution methods. However, this problem structure is simply unsatisfactory in addressing the requirements for some co-design problems (see Sec. 2.3). For example, Xu and Carrillo have stated that finite-horizon formulations are more appropriate for practical implementation than infinite-horizon versions [43].

4.3 Comparing the Strategies

A natural question is what solution technique should be used? This section will discuss some of the research around this question.

4.3.1 Computational Expense

Here we parameterize the solution time for each method with:

$$T_s = N_s \bar{T}_s \quad (\text{simultaneous}) \quad (29a)$$

$$T_n = N_n \bar{T}_n = N_n (\bar{T}_o + \bar{N}_i \bar{T}_i) \quad (\text{nested}) \quad (29b)$$

where N is the number of optimization algorithm iterations needed for convergence, \bar{T} is an average time, \square_s is simultaneous strategy, and \square_n is nested strategy.

Obviously, if $N_n > N_s$ and $\bar{T}_o > \bar{T}_s$, then the simultaneous approach is superior, but this is unlikely as the nested strategy reduces the number of variables and constraints in the outer-loop problem. This highlights the computational goal of the nested approach. Ideally we seek a nested strategy that significantly reduces both the number of iterations and time required during each iteration in the outer loop, while keeping the time required to solve the nested subproblem small. This parameterization facilitates an experimental approach for determining which method is appropriate. The designer could perform some tests on specific steps in the solution method such as determining the average inner-loop cost or initial iteration costs. Frequently, the solutions to multiple co-design problems are desired (such as changing different problem elements or parameter values) or a particular class of problems is being studied. If such solution experience exists, then knowledge from previous similar problems can be used to better determine which method should be selected.

Methods for reducing total computational expense for the general co-design problem with respect to either solution strategy are limited (other than favoring DT over other methods). One commonly-used approach is to utilize parallel computing when solving the outer-loop problem when multiple variations on x_p need to be evaluated, e.g., parallel finite differences or genetic algorithms. This will increase the coordination costs but can greatly reduce the overall computational expense (ideally $\bar{T}_n \rightarrow \bar{T}_n/N_p$, where N_p is the number of parallel resources). Another

suggestion from Ref. [44] is to treat plant variables in the simultaneous approach with DT exactly as in Eqn. (15), where there is a plant variable at each time point and they are all constrained to be equal (which keeps Jacobian of the constraints and objective function sparse). The development of guidelines and methods to help address the computational cost for different classes of co-design problems is left as future work.

4.3.2 Literature-Based Recommendations

Reyer et al. stated that the size and complexity of the simultaneous problem formulation could make this strategy impractical [13]. A number of authors have agreed with this statement, i.e., the nested strategy was better suited for their particular problem [4,22,30,45]. In the case of Refs. [30,45], the studies utilize the LQR for an extremely efficient inner loop, i.e., $\bar{T}_o + \bar{N}_i \bar{T}_i$ was quite small. In Ref. [4], there was an extreme computational expense associated with changes in the plant variables (T_i was large), hence minimizing the number of function calls was paramount. In Ref. [22], there was both a high computational expense for modifying the plant design and the inner-loop could be solved with a QP.

There are a limited set of studies that claim the opposite. In Ref. [3], the authors state that the simultaneous strategy is better, but the inner-loop was not solved with a QP even though it was in the Problem (26) class, so it is hard to fairly compare the strategies. However, it is still hard to make general recommendations since it is not very common for co-design study to compare the tradeoffs between the simultaneous and nested strategies for their particular design problem. In most design studies, only one strategy is needed if it is producing the desired results.

There are some circumstances where the simultaneous strategy might be preferred. For some problems, it might be the only viable method. For example, it may be impossible to separate the domains if black-boxes are used. Also, consider the discussion of an empty Ω for a particular candidate plant design. In this situation, the nested strategy could struggle or fail to find the optimal solution. Another point is that the nested strategy does not follow the steepest descent direction in the simultaneous formulation (since the control variables are optimal for the control subproblem). A simultaneous approach may better utilize the coupling between the plant and control design variables for faster convergence.

It remains future work to determine if there are conditions such that a well-behaved simultaneous co-design formulation becomes more challenging to solve with the nested strategy.

5 Test Problems

In this section, three co-design test problems (abbreviated TP) are defined which highlight some of the general co-design theory concepts². Please see also Refs. [1–5,20–22] for examples of complex co-design problems that utilize nontraditional problem elements in the general co-design problem class.

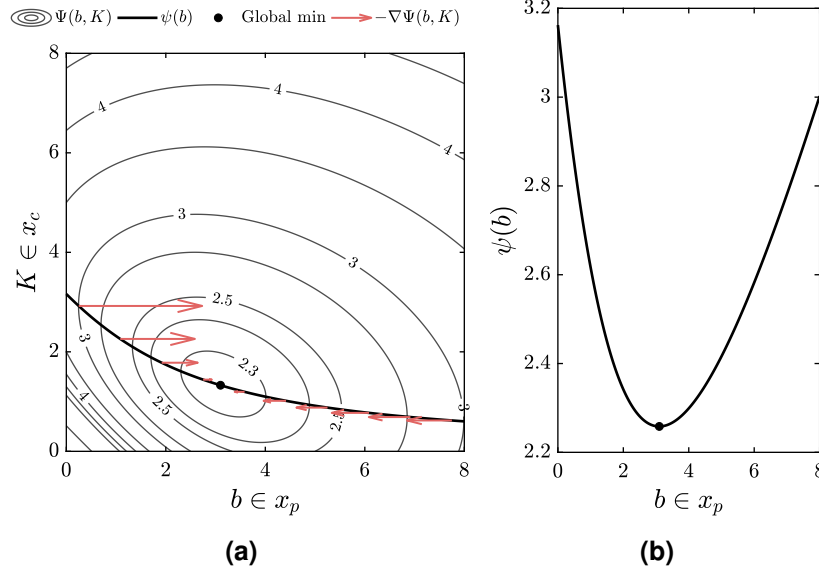


FIGURE 2: Scalar plant, scalar control problem (TP1) results with $q = 10$, $r = 1$, $w_c = 1$, and $w_p = 0.3$: **(a)** $\Psi(b, K)$; **(b)** $\psi(b)$.

5.1 Test Problem 1: Scalar Plant, Scalar Control

The first TP belongs to many early co-design formulations [6, 16] including infinite time horizon, optimization of control gains, no path constraints, and separate control and plant objectives as in Eqn. (10). The co-design problem is stated as:

$$\min_{b, K} \frac{w_c}{\xi_0^2} \int_0^\infty (q\xi^2 + ru^2) dt + w_p b \quad (30a)$$

$$\text{subject to: } \dot{\xi} = -b\xi + u \quad (30b)$$

$$\phi_1 := \xi(0) - \xi_0 = 0 \quad (30c)$$

$$\phi_2 := -b \leq 0, \phi_3 := -K \leq 0 \quad (30d)$$

where $u = -K\xi$, $b \in x_p$, and $K \in x_c$. The results are shown in Fig. 2 for one particular set of values of the problem parameters.

For this problem, the ARE defined in Eqn. (27) is a single quadratic equation: $0 = q - 2bP - P^2/r$. The solution to this equation defines the gains of a full-state feedback optimal control law and therefore can be used to obtain the potential inner-loop optimal control designs, M , in a nested formulation. This is visualized in Fig. 2a as the solid black curve. We note that this is a subspace of Ω , or simultaneous feasible set. Comparing Fig. 2a and Fig. 2b, we can visualize directly the difference between $\Psi(x_p, x_c)$ and $\psi(x_p)$. We also note that the gradient of the nested solution trajectory in Fig. 2a is always orthogonal to the K -axis since it is the optimal gain value for a given b .

The coupling between the plant and control design variables is also present, noting the ‘‘tilt’’ of the level sets. Therefore, a sequential method would not be guaranteed to arrive at the global optima (but an iterative sequential strategy would in TP1) [6]. In TP1, from all starting conditions,

²The MATLAB codes for these test problems is available in Ref. [46].

both the simultaneous and nested solution strategies arrive at the global optimum. Although this is an extremely simple co-design problem, it does help visualize a number of important concepts.

5.2 Test Problem 2: Co-Design Transfer

Consider the following simple co-design problem that seeks to move a second-order system from an arbitrary initial state to rest while minimizing control effort:

$$\min_{k, u(t)} \int_0^{t_f} u^2 dt \quad (31a)$$

$$\text{subject to: } \dot{\xi} = \begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (31b)$$

$$\phi_1 := \xi_1(0) - x_0 = 0, \quad \phi_2 := \xi_2(0) - v_0 = 0 \quad (31c)$$

$$\phi_3 := \xi_1(t_f) = 0, \quad \phi_4 := \xi_2(t_f) = 0 \quad (31d)$$

where $k \in \mathbf{x}_p$ and $u(t) \in \mathbf{x}_c$. The solution for $u^*(t, k)$ (i.e., the nested strategy) can be obtained by scaling the problem [47] into an equivalent DO problem in Ref. [34, pp. 166–167]:

$$u^*(t, k) = -\frac{2k}{kt_f^2 - \sin^2(\sqrt{k}t_f)} \left(c_1(t, k)x_0 + c_2(t, k)\frac{v_0}{\sqrt{k}} \right) \quad (32a)$$

$$c_1(t, k) = \sin(\sqrt{k}(t_f - t)) \sin(\sqrt{k}t_f) - \sqrt{k}t_f \sin(\sqrt{k}t) \quad (32b)$$

$$c_2(t, k) = -\cos(\sqrt{k}(t_f - t)) \sin(\sqrt{k}t_f) + \sqrt{k}t_f \cos(\sqrt{k}t) \quad (32c)$$

If $k = 0$, then the solution above is not valid and u^* is linear with respect to t . The original objective function can now be computed analytically by integrating the square of Eqn. (32). With this closed-form expression for the objective function, an optimality condition for k can be derived using Eqn. (20).

The results for this TP are shown in Fig. 3 for various values of the problem parameters. In Fig. 3a, there are a large number of local solutions of $\psi(k)$ and a clear global minimum at $k^* = 3.554$. In Fig. 3b, for the different values of the problem parameters, there is a single global minimum at $k^* = 6.924$. Finally in Fig. 3c, there is a global minimum with $k^* = 0$, i.e., the plant solution is degenerate.

From these results, it is clear that caution should be used when claiming that a globally-optimal co-design solution is found. More accurately expressed, global search algorithms could improve the confidence of finding the true optimal solution such as a multistart approach or genetic algorithms (as well as handling the feasibility issues discussed in Sec. 2.3.4) [35].

The control solution in Eqn. (32) demonstrates the complicated nature of the analytical solutions for even the simplest of co-design problems. Furthermore, this test problem demonstrates that the optimal solutions are sensitive to the problem parameters. The optimal value of k is different in all three cases, but a single value for k would need to be selected (as it is a plant variable) if the system needs to operate at each of considered scenarios. This problem may be a particularly useful TP as it is a well-posed co-design problem with a single system-level objective, general boundary conditions, and a closed-form OLC solution.

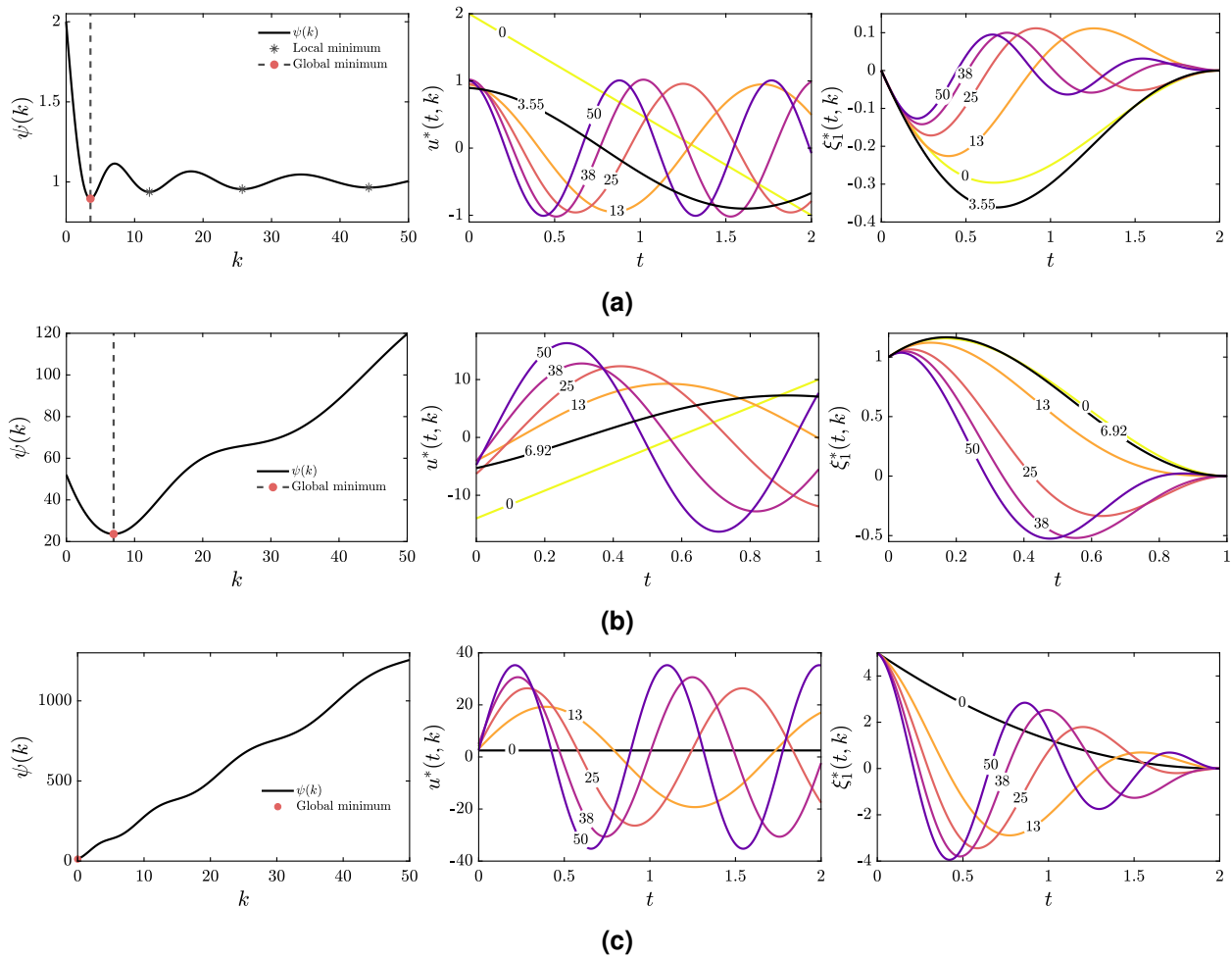


FIGURE 3: Co-design transfer problem (TP2) results for various values of the problem parameters and different values of k marked: **(a)** results demonstrating a large number of local solutions with $t_f = 2$, $x_0 = 0$, and $v_0 = -1$; **(b)** results demonstrating single global minimum with $t_f = 1$, $x_0 = 1$, and $v_0 = 2$; **(c)** results demonstrating degenerate plant solution with $t_f = 2$, $x_0 = 5$, and $v_0 = -5$.

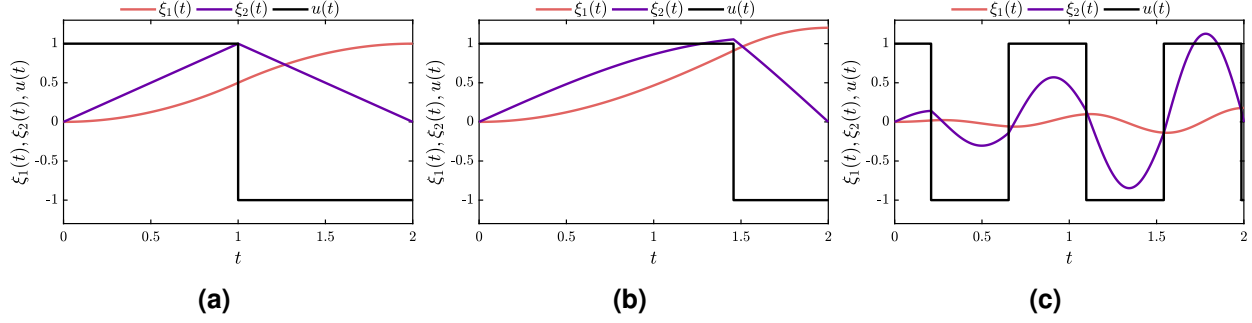


FIGURE 4: Simple SASA test problem (TP3) results with $J = 1$, $t_f = 2$, and $u_{\max} = 1$: **(a)** $k = 0$; **(b)** $k^* = 0.8543$; **(c)** $k = 50$.

5.3 Test Problem 3: Simple SASA

The final TP is a co-design problem that was used directly in a detailed co-design study. The co-design study focused on developing a novel strain-actuated solar array (SASA) system for spacecraft pointing control and jitter reduction [22] (please see Ref. [47] for a detailed discussion of the connections to the original study). Both the geometric properties of the solar array and OLC voltages along the array were the design variables. To help provide some insight into the results of the original design study, a much simpler, but still representative, co-design problem was proposed:

$$\min_{k, u(t)} -\xi_1(t_f) \quad (33a)$$

$$\text{subject to: } \dot{\boldsymbol{\xi}} = \begin{bmatrix} 0 & 1 \\ -k/J & 0 \end{bmatrix} \boldsymbol{\xi} + \begin{bmatrix} 0 \\ 1/J \end{bmatrix} u \quad (33b)$$

$$\phi_1 := \xi_1(0) = 0, \quad \phi_2 := \xi_2(0) = 0 \quad (33c)$$

$$\phi_3 := \xi_2(t_f) = 0 \quad (33d)$$

$$C_1 := u - u_{\max} \leq 0, \quad C_2 := -u - u_{\max} \leq 0 \quad (33e)$$

where $k \in \boldsymbol{x}_p$ and $u(t) \in \boldsymbol{x}_c$. The results for this problem are shown in Fig. 4 for various values of k including $k^* = 0.8543$.

The OLC exhibits bang-bang behavior [29, 34], i.e., u is at either the minimum or maximum value. Using the control stationarity condition in Eqn. (14b), we can show directly that the Hamiltonian is minimized if u exhibits bang-bang behavior, but determining the locations of the switching is quite challenging. In a nested strategy, the number and location of the switches vary for different values of k (see Fig. 4c containing five switches while the optimal co-design solution only contains one switch in Fig. 4b). This is a direct example of the discussion in Sec. 4.1 motivating the using of DT in co-design (see Sec. 4.1.1). The nested co-design solutions were found using DT [11, 23, 24].

We will also use this TP to compare the computational expense associated with using various co-design solution strategies. If the nested strategy is used, the obligatory inner-loop problem is an LQDO problem (cf. Problem (26)). Therefore, extremely efficient means for solving the inner-loop are available [42]. Accordingly, we can test three strategies: simultaneous, nested, and nested solving the inner-loop problem with a QP. A naive NLP DT implementation, mostly default opti-

TABLE 1: Simple SASA test problem (TP3) computational results for various strategies (all times are in seconds).

accuracy	method	k^*	$\theta^*(t_f)$	$T_{s/n}$	$N_{s/n}$	$\bar{T}_{s/n}$	\bar{T}_o	\bar{N}_i	\bar{T}_i
	closed-form	0.8543	1.2031	–	–	–	–	–	–
low ($n_t = 10$)	simultaneous	0.8407	1.1808	0.06	29	0.002	–	–	–
	nested	0.8497	1.1797	4.18	16	0.261	0.002	4.81	0.054
	nested (QP)	0.8407	1.1808	0.16	15	0.011	0.002	4.67	0.002
medium ($n_t = 100$)	simultaneous	0.8690	1.2029	6.54	173	0.038	–	–	–
	nested	0.8682	1.2029	257.95	8	32.244	0.004	6.00	5.373
	nested (QP)	0.8691	1.2029	1.22	13	0.094	0.004	7.23	0.012
high ($n_t = 1,000$)	simultaneous	0.8543	1.2028	11464.33	1342	8.543	–	–	–
	nested (QP)	0.8535	1.2031	17.63	17	1.037	0.009	7.29	0.141

mization algorithm tolerances, and zero-valued initial guesses were used³. The results are shown in Table 1 using the notation from Sec. 4.3.1. We note that these results are not indicative of the most efficient implementations, but are meant to serve as illustrative results that highlight some of the potential differences that may be observed when using the different strategies, especially when an efficient inner-loop solution strategy exists. These results will vary from problem to problem and for different implementations of the solution strategies; therefore, similar tests should be performed unless existing implementation knowledge for a similar class of problems is available.

As expected, the naive nested strategy was always many orders of magnitude slower than the nested (QP) strategy. For the low accuracy approximation of the original DO problem using DT, we see that the simultaneous strategy is faster than the nested (QP) strategy, although both total solve times are quite small. For the medium and high accuracy cases, the nested (QP) approach was significantly better than the simultaneous strategy. In all the cases, the simultaneous and nested (QP) strategies converged to similar solutions, demonstrating equivalence of these two strategies since none of the issues discussed in Sec. 2.2 are present. Therefore, the primary result is under these testing conditions, the nested (QP) strategy facilitates faster, more accurate solutions to the co-design problem, but only because of efficient methods for solving the inner loop.

TP3 is an attractive TP as the closed-form solution for the simultaneous strategy can be computed to a high degree of accuracy, and it exhibits challenging behavior associated with inequality path constraints.

6 Conclusion

In this article, general combined plant and controller design (or co-design) problems were examined. A large portion of existing co-design theory has focused on specific DO formulations, leaving many open questions for co-design studies that do not fit the previous definitions.

There are two basic co-design solution strategies: simultaneous and nested. The problem formulations for both strategies were presented and a discussion of the formulation elements was provided. The nested strategy was presented as two-level optimization problem including a characterization of the differences in the feasibility region between the two strategies. This motivated

³Full details of these tests can be seen in Ref. [46].

the outer-loop feasibility constraint as one approach for ensuring that for every candidate plant design, the control subproblem is well-posed.

The natural next step was the presentation of the optimality conditions for both methods. For the simultaneous strategy, the optimality conditions were derived using only Pontryagin's minimum principle and an augmented state vector. Due to a number of challenges associated with the optimality conditions, practical solution considerations were discussed with a focus on the motivating reasons for using DT in co-design (e.g., the activity of inequality path constraints). Finally, three test problems were presented. These problems were fairly simple but highlighted a number of key concepts including coupling, the difference between the feasible regions for each strategy, general boundary conditions, inequality path constraints, system-level objectives, and complexity of the closed-form solutions.

This work seeks to provide a foundation for additional advances in general co-design theory. The outer-loop feasibility constraint warrants further investigation. Better comparisons between the two strategies are needed, especially for co-design problems without a specific form of the inner loop. Developing methods for reducing total computational expense is also an important area such as the suggestion in Ref. [44] for treating plant variables like dummy states (or control) or alternative general strategies such decentralized optimization [9, 48]. Furthermore, there is a need to develop solution strategies that specifically address practical implementation challenges, such as neuro-dynamic programming [43]. In addition, analyzing the implications of uncertainty in various problem elements (with respect to both the plant and control [49, 50]) is also an important topic to investigate further, and advancements in this area would increase practical utility in the solution of many relevant co-design problems. Finally, development of test problems that are more realistic could help answer some of the remaining open questions.

Acknowledgments

The authors would like to thank Anand Deshmukh and Danny Lohan from the University of Illinois at Urbana-Champaign for their insightful discussions on some of the topics in this article.

References

- [1] Allison, J. T., 2013. "Plant-Limited Co-Design of an Energy-Efficient Counterbalanced Robotic Manipulator". *ASME Journal of Mechanical Design*, **135**(10), Aug., p. 101003. doi: [10.1115/1.4024978](https://doi.org/10.1115/1.4024978)
- [2] 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". In American Control Conference, pp. 3375–3380. doi: [10.1109/ACC.2003.1244053](https://doi.org/10.1109/ACC.2003.1244053)
- [3] Allison, J. T., Guo, T., and Han, Z., 2014. "Co-Design of an Active Suspension Using Simultaneous Dynamic Optimization". *ASME Journal of Mechanical Design*, **136**(8), June, p. 081003. doi: [10.1115/1.4027335](https://doi.org/10.1115/1.4027335)
- [4] Deshmukh, A. P., and Allison, J. T., 2016. "Multidisciplinary Dynamic Optimization of Horizontal Axis Wind Turbine Design". *Structural and Multidisciplinary Optimization*, **53**(1), Aug., pp. 15–27. doi: [10.1007/s00158-015-1308-y](https://doi.org/10.1007/s00158-015-1308-y)
- [5] Yan, H.-S., and Yan, G.-J., 2009. "Integrated Control and Mechanism Design for the Variable Input-speed Servo Four-bar Linkages". *Mechatronics*, **19**(2), Mar., pp. 274–285. doi: [10.1016/j.mechatronics.2008.07.008](https://doi.org/10.1016/j.mechatronics.2008.07.008)

- [6] Fathy, H., Reyer, J., Papalambros, P., and Ulsov, A., 2001. “On the Coupling between the Plant and Controller Optimization Problems”. In American Control Conference, pp. 1864–1869. doi: [10.1109/ACC.2001.946008](https://doi.org/10.1109/ACC.2001.946008)
- [7] Allison, J. T., and Herber, D. R., 2014. “Multidisciplinary Design Optimization of Dynamic Engineering Systems”. *AIAA Journal*, **52**(4), Apr., pp. 691–710. doi: [10.2514/1.J052182](https://doi.org/10.2514/1.J052182)
- [8] Martins, J. R. R. A., and Lambe, A. B., 2013. “Multidisciplinary Design Optimization: A Survey of Architectures”. *AIAA Journal*, **51**(9), Sept., pp. 2049–2075. doi: [10.2514/1.j051895](https://doi.org/10.2514/1.j051895)
- [9] Allison, J. T., and Nazari, S., 2010. “Combined Plant and Controller Design Using Decomposition-Based Design Optimization and the Minimum Principle”. In ASME International Design Engineering Technical Conferences, pp. 765–774. doi: [10.1115/detc2010-28887](https://doi.org/10.1115/detc2010-28887)
- [10] Kusiak, A., and Larson, N., 1995. “Decomposition and Representation Methods in Mechanical Design”. *ASME Journal of Mechanical Design*, **117**(B), June, pp. 17–24. doi: [10.1115/1.2836453](https://doi.org/10.1115/1.2836453)
- [11] Herber, D. R., 2014. “Dynamic System Design Optimization of Wave Energy Converters Utilizing Direct Transcription”. M.S. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, May. <http://hdl.handle.net/2142/49463>
- [12] Frischknecht, B. D., Peters, D. L., and Papalambros, P. Y., 2011. “Pareto Set Analysis: Local Measures of Objective Coupling in Multiobjective Design Optimization”. *Structural and Multidisciplinary Optimization*, **43**(5), Nov., pp. 617–630. doi: [10.1007/s00158-010-0599-2](https://doi.org/10.1007/s00158-010-0599-2)
- [13] 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”. In ASME Design Engineering Technical Conference, no. DETC2001/DAC-21119.
- [14] Hale, A. L., Lisowski, R. J., and Dahl, W. E., 1985. “Optimal Simultaneous Structural and Control Design of Maneuvering Flexible Spacecraft”. *AIAA Journal of Guidance, Control, and Dynamics*, **8**(1), Jan., pp. 86–93. doi: [10.2514/3.19939](https://doi.org/10.2514/3.19939)
- [15] Eastep, F., Khot, N., and Grandhi, R., 1987. “Improving the Active Vibrational Control of Large Space Structures through Structural Modifications”. *Acta Astronautica*, **15**(6-7), June, pp. 383–389. doi: [10.1016/0094-5765\(87\)90174-3](https://doi.org/10.1016/0094-5765(87)90174-3)
- [16] Sunar, M., and Rao, S. S., 1993. “Simultaneous Passive and Active Control Design of Structures Using Multiobjective Optimization Strategies”. *Computers & Structures*, **48**(5), Sept., pp. 913–924. doi: [10.1016/0045-7949\(93\)90513-D](https://doi.org/10.1016/0045-7949(93)90513-D)
- [17] Peters, D. L., Papalambros, P. Y., and Ulsoy, A. G., 2011. “Control Proxy Functions for Sequential Design and Control Optimization”. *ASME Journal of Mechanical Design*, **133**(9), Sept., p. 091007. doi: [10.1115/1.4004792](https://doi.org/10.1115/1.4004792)
- [18] Peters, D. L., Papalambros, P. Y., and Ulsoy, A. G., 2009. “On Measures of Coupling Between the Artifact and Controller Optimal Design Problems”. In ASME International Design Engineering Technical Conferences, no. DETC2009-86868, pp. 1363–1372. doi: [10.1115/DETC2009-86868](https://doi.org/10.1115/DETC2009-86868)
- [19] Wang, Y.-S., and Wang, Y., 2015. “A Gradient-Based Approach for Optimal Plant Controller Co-Design”. In American Control Conference. doi: [10.1109/ACC.2015.7171833](https://doi.org/10.1109/ACC.2015.7171833)
- [20] Herber, D. R., and Allison, J. T., 2013. “Wave Energy Extraction Maximization in Irregular Ocean Waves Using Pseudospectral Methods”. In ASME International Design Engineering Technical Conferences, no. DETC2013-12600, p. V03AT03A018. doi: [10.1115/DETC2013-12600](https://doi.org/10.1115/DETC2013-12600)
- [21] Maraniello, S., and Palacios, R., 2016. “Optimal Vibration Control and Co-design of Very Flexible Actuated Structures”. *Journal of Sound and Vibration*, **377**, Sept., pp. 1–21. doi: [10.1016/j.jsv.2016.05.018](https://doi.org/10.1016/j.jsv.2016.05.018)
- [22] Chilan, C. M., Herber, D. R., Nakka, Y. K., Chung, S.-J., Allison, J. T., Aldrich, J. B., and Alvarez-Salazar, O. S., 2017. “Co-Design of Strain-Actuated Solar Arrays for Spacecraft Precision Pointing and Jitter Reduction”. *AIAA Journal*, **55**(9), Sept., pp. 3180–3195. doi: [10.2514/1.J055748](https://doi.org/10.2514/1.J055748)

- [23] Betts, J. T., 2010. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM. doi: [10.1137/1.9780898718577](https://doi.org/10.1137/1.9780898718577)
- [24] Biegler, L. T., 2010. *Nonlinear Programming*. SIAM. doi: [10.1137/1.9780898719383](https://doi.org/10.1137/1.9780898719383)
- [25] Rao, A. V., 2010. “A Survey of Numerical Methods for Optimal Control”. *Advances in the Astronautical Sciences*, **135**(1), pp. 497–528.
- [26] Colson, B., Marcotte, P., and Savard, G., 2007. “An Overview of Bilevel Optimization”. *Annals of Operations Research*, **153**(1), Apr., pp. 235–256. doi: [10.1007/s10479-007-0176-2](https://doi.org/10.1007/s10479-007-0176-2)
- [27] Vicente, L. N., and Calamai, P. H., 1994. “Bilevel and Multilevel Programming: A Bibliography Review”. *Journal of Global Optimization*, **5**(3), Oct., pp. 291–306. doi: [10.1007/BF01096458](https://doi.org/10.1007/BF01096458)
- [28] Tanino, T., and Ogawa, T., 1984. “An Algorithm for Solving Two-level Convex Optimization Problems”. *International Journal of Systems Science*, **15**(2), Feb., pp. 163–174. doi: [10.1080/00207728408926552](https://doi.org/10.1080/00207728408926552)
- [29] Liberzon, D., 2012. *Calculus of Variations and Optimal Control Theory*. Princeton University Press.
- [30] Belvin, W. K., and Park, K. C., 1990. “Structural Tailoring and Feedback Control Synthesis: An Interdisciplinary Approach”. *AIAA Journal of Guidance, Control, and Dynamics*, **13**(3), May, pp. 424–429. doi: [10.2514/3.25354](https://doi.org/10.2514/3.25354)
- [31] Rao, S. S., 1988. “Combined Structural and Control Optimization of Flexible Structures”. *Engineering Optimization*, **13**(1), Jan., pp. 1–16. doi: [10.1080/03052158808940943](https://doi.org/10.1080/03052158808940943)
- [32] Spong, M. W., Hutchinson, S., and Vidyasagar, M., 2005. *Robot Modeling and Control*, 1st ed. Wiley.
- [33] Chachuat, B., 2007. *Nonlinear and Dynamic Optimization: From Theory to Practice*, IC-32: winter semester 2006/2007 ed. Automatic Control Laboratory.
- [34] Bryson Jr., A. E., and Ho, Y.-C., 1975. *Applied Optimal Control*, revised ed. CRC Press.
- [35] Papalambros, P. Y., and Wilde, D. J., 2017. *Principles of Optimal Design*, 3rd ed. Cambridge University Press.
- [36] Pontryagin, L. S., 1962. *The Mathematical Theory of Optimal Processes*. Interscience.
- [37] Doležal, J., 1981. “On Optimal Control Problems with General Boundary Conditions”. *Journal of Optimization Theory and Applications*, **35**(1), Sept., pp. 143–148. doi: [10.1007/BF00934709](https://doi.org/10.1007/BF00934709)
- [38] Weisstein, E. W. Total Derivative. From MathWorld—A Wolfram Web Resource. Accessed on Dec 6, 2016. <http://mathworld.wolfram.com/TotalDerivative.html>
- [39] Garg, D., 2011. “Advances in Global Pseudospectral Methods for Optimal Control”. Ph.D. Dissertation, University of Florida, Gainesville, FL, USA, Aug.
- [40] Herber, D. R., 2015. Basic Implementation of Multiple-Interval Pseudospectral Methods to Solve Optimal Control Problems. Technical Report UIUC-ESDL-2015-01, Engineering System Design Lab, June. <http://hdl.handle.net/2142/77888>
- [41] Biegler, L. T., 2007. “An Overview of Simultaneous Strategies for Dynamic Optimization”. *Chemical Engineering and Processing: Process Intensification*, **46**(11), Nov., pp. 1043–1053. doi: [10.1016/j.cep.2006.06.021](https://doi.org/10.1016/j.cep.2006.06.021)
- [42] Herber, D. R., 2017. “Advances in Combined Architecture, Plant, and Control Design”. Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Dec. <https://www.ideals.illinois.edu/handle/2142/99394>
- [43] Xu, H., and Carrillo, L. R. G., 2017. “Near Optimal Control and Network Co-design for Uncertain Networked Control System with Constraints”. In American Control Conference. doi: [10.23919/ACC.2017.7963302](https://doi.org/10.23919/ACC.2017.7963302)
- [44] Kelly, M. P., 2015. Transcription Methods for Trajectory Optimization. Tutorial, Cornell University, Feb. <https://arxiv.org/abs/1707.00284>
- [45] Onoda, J., and Haftka, R. T., 1987. “An Approach to Structure/Control Simultaneous Optimization for Largeflexible Spacecraft”. *AIAA Journal*, **25**(8), Aug., pp. 1133–1138. doi: [10.2514/3.9754](https://doi.org/10.2514/3.9754)

- [46] Herber, D. R. Co-Design Examples Repository. [Online]. Available: <https://github.com/danielrherber/co-design-examples-repository>.
- [47] Herber, D. R., and Allison, J. T., 2017. “Unified Scaling of Dynamic Optimization Design Formulations”. In ASME International Design Engineering Technical Conferences, no. DETC2017-67676, p. V02AT03A003. doi: [10.1115/DETC2017-67676](https://doi.org/10.1115/DETC2017-67676)
- [48] Liu, T., Azarm, S., and Chopra, N., 2017. “On Decentralized Optimization for a Class of Multi-subsystem Codesign Problems”. *ASME Journal of Mechanical Design*, **139**(12), Oct., p. 121404. doi: [10.1115/1.4037893](https://doi.org/10.1115/1.4037893)
- [49] Bertsimas, D., Brown, D. B., and Caramanis, C., 2011. “Theory and Applications of Robust Optimization”. *SIAM Review*, **53**(3), Jan., pp. 464–501. doi: [10.1137/080734510](https://doi.org/10.1137/080734510)
- [50] Zhou, K., Doyle, J. C., and Glover, K., 1996. *Robust and Optimal Control*. Prentice Hall.