Towards a Fair Comparison between the Nested and Simultaneous Control Co-Design Methods using an Active Suspension Case Study

Athul K. Sundarrajan[1†] and Daniel R. Herber[2†]

***Abstract*— This paper tackles perhaps the two most common control co-design coordination strategies: simultaneous analysis and design and the nested control problem formulation. Many practical insights into the two strategies are presented using the literature and comprehensive numerical results from a detailed and challenging CCD problem of an active vehicle suspension. The study conducted attempts to provide a fair comparison and discussion between the control co-design coordination implementations. The results indicate a substantial reduction in computational costs over the existing implementations and conclusions on method selection contrary to common assumptions in the literature. However, additional work is needed to provide a robust set of CCD implementation guidelines.**

I. Introduction

Along with the increasing demand for advanced controlled and multidisciplinary engineering systems, there is also a need for superior performance and novel solutions that address the complex requirements of these integrated systems [1]. Control co-design (CCD) is the term for a class of integrated design methods that concurrently treat the dynamic system's physical and control aspects, overcoming some of the limitations of traditional sequential and siloed approaches [1], [2]. Because of the broad applicability and promise of improved integrated system performance [3], CCD has been adopted by researchers in several areas including automotive [4], [5], thermal management systems [6], spacecraft [7], wave energy [8], and wind energy [9], [10].

There are various solution strategies for CCD that have been studied, but only relatively few studies have made detailed comparisons. Without thorough investigations utilizing state-of-the-art methods and modern design problems, the existing domain-specific suggestions might not apply to today's problems and methods. Such broad and up-to-date investigations are critical as CCD is being used to solve increasingly complex and large-scale system design problems. Such studies could elucidate needed early insights into the appropriate choice of strategy and implementation techniques. The goal of this article is to make a contribution towards this end for a subset of popular CCD coordination and solution strategies on a complex CCD problem. While guidelines applicable to broad classes of CCD problems will not be made, the existing literature-based comparisons and a detailed case study will better demonstrate the state-of-the-art understanding and how a thorough investigation can be conducted to yield the desired implementation insights.

† Colorado State University, Department of Systems Engineering, Fort Collins, CO 80523

1 Graduate Student, athul.sundarrajan@colostate.edu

2 Assistant Professor, daniel.herber@colostate.edu, Corresp. Author

The rest of the paper is organized as following: Sec. II presents an overview of the two common CCD coordination strategies; Sec. III reviews the some of the CCD solution methods; Sec. IV discusses the trade-offs between these two strategies; and Secs. V and VI discuss the usage of the strategies on an CCD case study of an active vehicle suspension. Finally, Sec. VII presents the conclusions.

II. Two Control Co-Design Coordination Strategies

Here we are considering coordination strategies for CCD problems that can be represented as a deterministic nonlinear dynamic optimization problem (NLDO) [4], [8]. Because of the multidisciplinary nature of these problems, different multidisciplinary design optimization (MDO) architectures can be considered [1], [4], [11]. However, suppose we are limited to single-system problems and architectures that do not partition the system across trajectories. In that case, there are a limited number of appropriate MDO methods suitable for CCD [1], [12]. Because of these reasons and more, the two leading coordination strategies for CCD are 1) simultaneous analysis and design or *simultaneous*, and 2) the nested control problem formulation or simply *nested* [13].

A. Simultaneous Formulation

In the simultaneous CCD formulation, a single optimization problem is put forth, and the optimizer simultaneously analyzes and designs the plant, state, and control variables, as shown in Fig. 1a [4], [14]. The general NLDO problem form is[1]:

$$\min_{\boldsymbol{x}=[\boldsymbol{\xi},\boldsymbol{x}_c,\boldsymbol{x}_p]} \ o = \int_{t_0}^{t_f} \ell\left(t,\boldsymbol{\xi},\boldsymbol{x}_c,\boldsymbol{x}_p\right) dt + m\left(\boldsymbol{\xi}_0,\boldsymbol{\xi}_f,\boldsymbol{x}_c,\boldsymbol{x}_p\right) \tag{1a}$$

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

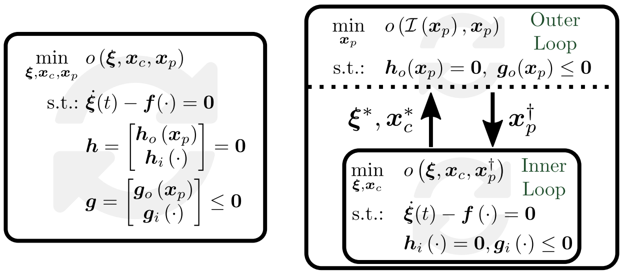
$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_o\left(\boldsymbol{x}_p\right) \\ \boldsymbol{h}_i\left(t,\boldsymbol{\xi},\boldsymbol{x}_c,\boldsymbol{x}_p,\boldsymbol{\xi}_0,\boldsymbol{\xi}_f\right) \end{bmatrix} = \boldsymbol{0} \tag{1c}$$

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}_o\left(\boldsymbol{x}_p\right) \\ \boldsymbol{g}_i\left(t,\boldsymbol{\xi},\boldsymbol{x}_c,\boldsymbol{x}_p,\boldsymbol{\xi}_0,\boldsymbol{\xi}_f\right) \end{bmatrix} \le \boldsymbol{0} \tag{1d}$$

$$\text{where: } \boldsymbol{\xi}_0 = \boldsymbol{\xi}(t_0),\ \boldsymbol{\xi}_f = \boldsymbol{\xi}(t_f) \tag{1e}$$

where $t \in [t_0, t_f]$ is the fixed time horizon, $\{\boldsymbol{\xi},\boldsymbol{x}_c,\boldsymbol{x}_p\}$ are the collections of the selected states, control design variables, and plant design variables, respectively. The objective function $o(\cdot)$ is composed of the Lagrange term $\ell(\cdot)$ and

[1]For simplicity, a free time horizon, multiple phases, general state differential equations, and differentiating between path and boundary constraints are omitted from the formulation.



(a) Simultaneous. (b) Nested.

Fig. 1: Two common CCD coordination strategies.

Mayer term $m(\cdot)$ [15]. The first-order differential equation $\dot{\boldsymbol{\xi}}(t) - \boldsymbol{f}(\cdot) = \mathbf{0}$ represents the system dynamics. The equality constraints $\boldsymbol{h}(\cdot)$ are partitioned into two sets $\{\boldsymbol{h}_o(\cdot), \boldsymbol{h}_i(\cdot)\}$ where $\boldsymbol{h}_o(\cdot)$ depends only on $\boldsymbol{x}_p$. A similar partitioned form is used for the inequality constraints $\boldsymbol{g}(\cdot)$.

B. Nested Formulation

The nested CCD formulation is an intentional reorganization of the simultaneous CCD problem in Prob (1) as a two-level optimization problem with an outer and inner problem hierarchy, as shown in Fig. 1b. For the outer-loop problem, we solve only with respect to the plant design variables:

$$\min_{\boldsymbol{x}_p} \quad o\left(\mathcal{I}\left(\boldsymbol{x}_p\right), \boldsymbol{x}_p\right) \tag{2a}$$

$$\text{subject to:} \quad \boldsymbol{h}_o(\boldsymbol{x}_p) = \mathbf{0} \tag{2b}$$

$$\boldsymbol{g}_o(\boldsymbol{x}_p) \leq \mathbf{0} \tag{2c}$$

$$\text{where:} \quad \mathcal{I}\left(\boldsymbol{x}_p\right) = \arg\min_{\boldsymbol{\xi},\boldsymbol{x}_c} (\text{Prob. (3) with } \boldsymbol{x}_p) \tag{2d}$$

where $\mathcal{I}\left(\boldsymbol{x}_p\right)$ is the solution to the following inner-loop problem:

$$\min_{\boldsymbol{\xi},\boldsymbol{x}_c} \quad o\left(\boldsymbol{\xi}, \boldsymbol{x}_c, \boldsymbol{x}_p^\dagger\right) \tag{3a}$$

$$\text{subject to:} \quad \dot{\boldsymbol{\xi}}(t) - \boldsymbol{f}\left(t, \boldsymbol{\xi}, \boldsymbol{x}_c, \boldsymbol{x}_p^\dagger\right) = \mathbf{0} \tag{3b}$$

$$\boldsymbol{h}_i\left(t, \boldsymbol{\xi}, \boldsymbol{x}_c, \boldsymbol{x}_p^\dagger, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f\right) = \mathbf{0} \tag{3c}$$

$$\boldsymbol{g}_i\left(t, \boldsymbol{\xi}, \boldsymbol{x}_c, \boldsymbol{x}_p^\dagger, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f\right) \leq \mathbf{0} \tag{3d}$$

where $\boldsymbol{x}_p^\dagger$ is the candidate plant design from the outer loop. This nested strategy was termed nested control problem formulation because Prob. (3) is now in the standard optimal control form where the optimal states and controls are desired. In Refs. [12], [13], it is shown that the solution to Prob. (2) is mathematically equivalent to the simultaneous form in Prob. (1) under the condition that a solution exists for the inner-loop problem for every considered $\boldsymbol{x}_p$ by the outer loop.

III. Solution Methods for the Infinite-Dimensional Problems

In order to start a discussion comparing the two strategies, it is first important to understand the practical methods with which the different optimization problems will be solved. Problems (1) and (3) are infinite-dimensional dynamic optimization problems because the constraints, such as the dynamics in Eq. (1b), need to be satisfied at all uncountable points in the time horizon. For some problems, optimality conditions can be useful in determining a solution. For example, a frequent inner-loop problem form is the infinite-horizon linear–quadratic regulator (LQR) where solutions can be efficiently determined [2], [13], [16], [17]. However, it is frequently necessary to apply some form of numerical approximation to obtain solutions (for a more complete discussion, see Refs. [12], [15], [18]).

The optimality conditions of the dynamic optimization problem can be discretized in time and numerically solved for as a boundary value problem [15], [19]. Discretization of the open-loop control trajectories and use of an integrator for evaluating the differential equations leads to the common shooting approach (i.e., simulation-in-the-loop). However, both of these approaches can have significant drawbacks. Another class of methods that has a number of favorable properties and has been quite successful in finding solutions to a number of CCD problems [4], [5], [7]–[9] is direct transcription (DT). Because of its central role in many CCD studies in supporting bi-directional coupling, early-stage control design, and the identification of system performance limits, it will now be discussed further.

A. Direct Transcription

DT is a family of methods that transcribe the original infinite-dimensional problem to a finite-dimensional nonlinear program (NLP) [1], [8], [15], [18], [20]. The time horizon is discretized into a finite number of points n_t, known as the mesh $\boldsymbol{t}$. The original optimization variables for all the trajectories (i.e., the states and open-loop controls) are remodeled as many additional scalar variables for each point in $\boldsymbol{t}$. Then the original NLDO problem is converted into the following NLP that approximates the original solution:

$$\min_{\mathbf{x}=[\boldsymbol{U},\boldsymbol{\Xi},\boldsymbol{x}_p]} \quad v\left(\boldsymbol{t}, \boldsymbol{\Xi}, \boldsymbol{X}_c, \boldsymbol{x}_p, \boldsymbol{\Xi}_1, \boldsymbol{\Xi}_{n_t}\right) \tag{4a}$$

$$\text{subject to:} \quad \boldsymbol{\zeta}\left(\boldsymbol{t}, \boldsymbol{\Xi}, \boldsymbol{X}_c, \boldsymbol{x}_p\right) = \mathbf{0} \tag{4b}$$

$$\boldsymbol{h}\left(\boldsymbol{t}, \boldsymbol{\Xi}, \boldsymbol{X}_c, \boldsymbol{x}_p, \boldsymbol{\Xi}_1, \boldsymbol{\Xi}_{n_t}\right) = \mathbf{0} \tag{4c}$$

$$\boldsymbol{g}\left(\boldsymbol{t}, \boldsymbol{\Xi}, \boldsymbol{X}_c, \boldsymbol{x}_p, \boldsymbol{\Xi}_1, \boldsymbol{\Xi}_{n_t}\right) \leq \mathbf{0} \tag{4d}$$

where $\boldsymbol{\Xi}$ is a matrix of discretized state variables (so $\boldsymbol{\Xi}_1 \equiv \boldsymbol{\xi}_0$ and $\boldsymbol{\Xi}_{n_t} \equiv \boldsymbol{\xi}_f$), $\boldsymbol{X}_c$ is a matrix of discretized control variables[2], $\boldsymbol{\zeta}(\cdot)$ are the defect constraints used to approximate the dynamic constraints in Eq. (1b), and $v(\cdot)$ is an approximation of the original objective function using a selected numerical quadrature scheme for the Lagrange term [1], [20]. Please see Refs. [15] for implementation details, and Refs. [4], [8] for detailed discussions of DT in the context of CCD.

Although the NLP in Prob (4) can be quite large with many constraints, there are many properties and methods for efficiency finding solutions [4], [15], [18], [19]. The NLP that is constructed has a specific sparse structure that can be leveraged. Additionally, because of the large problem

[2]For static control variables, such as gains, only a single value is needed per control design variable, similar to the plant design variables. For conciseness, we use $\boldsymbol{X}_c$ assuming replication of the static control design variables as needed.

size, gradient-based methods are frequently used to solve the problem requiring the use symbolic, complex-step, and other real-valued difference derivative methods [15], [21].

Additionally, linear-quadratic dynamic optimization (LQDO) is where the objective only contains quadratic terms, and the constraints are linear in Prob. (1). Using a DT method on an LQDO problem leads to Prob. (4) being a (convex) quadratic program (QP), which can be solved efficiently with tailored optimization algorithms [20], [22]. While no realistic CCD problem is an LQDO problem [2], [12], there are many studies where for fixed plant design, the remainder is an LQDO problem. This leads to the LQDO-amendable CCD problem class, defined in Ref. [23], and has been used in combination with with the nested coordination strategy in several CCD studies [7], [8], [23].

IV. Comparing the Strategies

With both the considered CCD coordination strategies and solution methods described, we can now start comparing the nested and simultaneous strategies.

A. Literature-Based Discussions

In many studies, there is a brief statement on the coordination selection (likely supported by work not directly shown). In Refs. [7], [9], [16], [17], the authors state that the nested approach is better suited for their problem, while Refs. [4], [5], [24] assert the simultaneous approach. For the studies that selected the nested approach, the main motivating reasons can be summarized as the impractical size and complexity of the simultaneous formulation [25], ability to use tailored inner-loop methods (e.g., LQR and LQDO) [7], [16], [17], [23], and reduction of calls to computationally-expensive plant models [7], [9]. In Ref. [4], the simultaneous approach was selected because the nested implementation used was very computationally inefficient. However, the CCD problem in that study (and the one considered in this article) is an LQDO-amenable CCD problem but was not treated as such in Ref. [4], among other implementation improvements that would reduce computational expense.

Several studies have shown quantitative comparisons between a simultaneous and nested implementation of the same CCD problem. In Ref. [9], the two strategies converged to the same solution (as expected if all equivalency assumptions are met). The main computational complexity comparison was in the form of function evaluations of $\boldsymbol{f}$, which was a costly black-box function. The nested approach had about half of the number of function calls. However, without also comparing computation times, it may have been the case that the simultaneous approach had a lower CPU time because there is more involved with the two coordination strategies than function calls (although in this study, it may indeed have been the dominating factor).

In Ref. [26], the simultaneous approach produced a lower objective function value, even using a QP for the nested inner-loop problem. However, the implementation details and computation times are lacking, so it is challenging to generalize the outcomes. Similar statements can be made for Ref. [14] (and the simultaneous and nested approaches are much better than the other considered strategies). In Ref. [12], comparisons were made on a very simple CCD problem, and results showed that the nested approach was much better for large n_t. However, the simultaneous implementation was extremely inefficient, not considering sparsity or accurate derivative computations. In Ref. [22], a moderately-complex LQDO-amendable CCD problem was considered using both approaches. It was shown that there are several factors that impact which method is superior. However, the conclusions were relatively limited scope because the sensitivity to the convergence tolerances and other means for reducing runtime were not explored.

B. Potential Trade-offs

We now summarize the potential trade-offs between the coordination strategies as found in the CCD and MDO literature.

1) Advantages (+) of the simultaneous strategy:

- Can potentially find the solution quickly by letting the optimizer explore regions that are infeasible [11].
- Naturally handles bidirectional coupling between the plant and control design variables [4], [12].
- Supports fine-grained parallelization [4].
- Better supports advanced derivative methods (e.g., complete problem analytic derivatives and the complex-step method).
- Only a single problem to construct and manage.

2) Advantages (+) of the nested strategy:

- Each subproblem's structure is simplified and size reduced from the original simultaneous formulation (e.g., dynamics now considered with fixed $\boldsymbol{x}_p$).
- Tailored optimization algorithms (and tolerances) can be used in the different subproblems that can leverage the simplified subproblem structure (e.g., QP or LQR) or outer-loop global search [7], [12], [16], [17], [23].
- If the inner loop is always feasible, this approach naturally handles bidirectional coupling between the plant and control design variables [12].
- Results from intermediate iterations are feasible.
- In some cases, potentially fewer function calls for certain elements (e.g., if $\boldsymbol{f}(\cdot)$ is a linear dynamic system) [7], [9], [23].

3) Disadvantages (−) of the simultaneous strategy:

- Large problem size [11].
- Requires many function calls to potentially expensive problem elements such as $\boldsymbol{f}(\cdot)$ [7], [9], [23].
- Results from intermediate iterations are not guaranteed to be feasible [11].

4) Disadvantages (−) of the nested strategy:

- Issues when the inner-loop problem is not defined at a particular $\boldsymbol{x}_p$ [12].
- Challenging to compute accurate outer-loop derivatives (e.g., accuracy of the subproblem impacts accuracy of outer-loop derivative information [11], and analytical derivatives sometimes impossible to construct).
- Only supports coarse-grained parallelization [4].

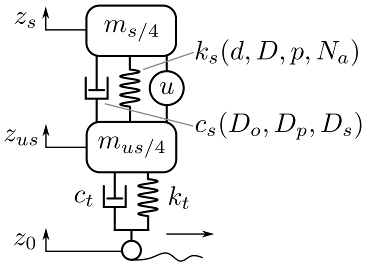


Fig. 2: Active suspension CCD problem.

C. Towards a Fair Comparison

From the discussion above and existing CCD literature, there does not seem to be a consensus or even good, purposeful guidelines on when to use either strategy. This shortcoming may be due to CCD problems and design goals diversity, but clear case studies could be quite insightful.

One of the main challenges in making a fair comparison is the fact that different optimization and analysis architectures are typically used when implementing the two coordination strategies on the same problem. Furthermore, if ineffective implementations of some parts of a coordination/solution method are used, then the comparison may be biased. For example, in Ref. [12], no fair comparisons were made because the simultaneous did not leverage the problem's sparsity. While complete optimization of the implementation strategy may not be useful in every case, a good case study should put forth the effort to compare multiple implementations. More universal metrics such as the number of function calls can be useful [1], [9], but often, the function calls are significantly different between the coordination strategies, including the use of different models and code vectorization, obscuring totals (and a similar case can be made for iterations).

In Ref. [12], runtime benchmarking was used to compare the strategies, but this approach may have issues with generalizability due to its dependence on computer architecture and the environment [27]. However, this approach does try to normalize the utility of the two strategies where lowered measured runtime cost (for the same outcome) is desirable. Yet, solution quality is another dimension that could be argued. Generally, better solution quality implies lower objective function value but also can include closer satisfaction of the constraints and more alternative feasible solutions. Additionally, it can be argued that in (early-stage) CCD problems, the optimal plant design variables may only need to be determined to a few decimal places. In contrast, state variables need many more to model the system dynamics accurately and not have accumulated errors. Therefore, the commonplace solution quality vs. computational expense trade-off should be considered.

The results presented in Sec. VI will be towards making better comparisons considering the many factors enumerated in this section.

V. Active Suspension Problem

Active vehicle suspension CCD problems have been used to show the efficacy of the CCD methodology and compare the trade-offs between different strategies [1], [13], [23], [26]. The considered quarter-car suspension system and components are illustrated in Fig. 2 [1], [26]. The system consists of two masses (sprung mass $m_{s/4}$ and unsprung mass $m_{us/4}$), and the suspension between them is consists of a force actuator $u(t)$, a linear spring $k_s(\boldsymbol{x}_p)$, and a linear damper $c_s(\boldsymbol{x}_p)$. The remainder of the system consists of a linear tire spring k_t and damper c_t and road input $z_0(t)$.

A. System Dynamics

There are four states in the system:

$$\boldsymbol{\xi}(t) = \begin{bmatrix} z_{us} - z_0 & \dot{z}_{us} & z_s - z_{us} & \dot{z}_s \end{bmatrix}^T \tag{5}$$

where $z_{us} - z_0$ is the displacement between $m_{us/4}$ and z_0, $\dot{z}_{us}$ is the velocity of the $m_{us/4}$, $z_s - z_{us}$ is the relative displacement between the masses, and $\dot{z}_s$ is the velocity of $m_{s/4}$. The initial conditions for the states are assumed to be zero, i.e., $\boldsymbol{\xi}(0) = \mathbf{0}$. The differential equation is:

$$\dot{\boldsymbol{\xi}}(t) = \boldsymbol{A}(\boldsymbol{x}_p)\boldsymbol{\xi}(t) + \boldsymbol{B}u(t) + \boldsymbol{E}\dot{z}_0(t) \tag{6a}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_t(\boldsymbol{x}_p)}{m_{us/4}} & \frac{-[c_s(\boldsymbol{x}_p)+c_t]}{m_{us/4}} & \frac{k_s(\boldsymbol{x}_p)}{m_{us/4}} & \frac{c_s(\boldsymbol{x}_p)}{m_{us/4}} \\ 0 & -1 & 0 & 1 \\ 0 & \frac{c_s(\boldsymbol{x}_p)}{m_{s/4}} & \frac{-k_s(\boldsymbol{x}_p)}{m_{s/4}} & \frac{-c_s(\boldsymbol{x}_p)}{m_{s/4}} \end{bmatrix} \tag{6b}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ \frac{-1}{m_{us/4}} \\ 0 \\ \frac{1}{m_{s/4}} \end{bmatrix}, \quad \boldsymbol{E} = \begin{bmatrix} -1 \\ \frac{c_t}{m_{us/4}} \\ 0 \\ 0 \end{bmatrix} \tag{6c}$$

B. System Objective

The performance measure for a design load case is a combination of quadratic penalties on handling $(z_{us} - z_0)$, passenger comfort $\ddot{z}_s$, and control effort u:

$$o = \int_{t_0}^{t_f} \left[w_1 \xi_1^2 + w_2 [\dot{\xi}_4(t, \boldsymbol{\xi}, u, \boldsymbol{x}_p)]^2 + w_3 u^2 \right] dt \tag{7}$$

with $w_1 = 10^5$, $w_2 = 0.5$, and $w_3 = 10^{-5}$ from Ref. [4].

Here we consider two load cases: 1) a ramp input conditions (road grade at 25% speed of 10 m/s), 2) rough road profile using IRI737 data [4]. The two are combined using a weighted sum as follows:

$$\min_{\boldsymbol{\xi}, u, \boldsymbol{x}_p} \; 10^{-2} o(\boldsymbol{\xi}_{\text{ramp}}, u_{\text{ramp}}, \boldsymbol{x}_p) + o(\boldsymbol{\xi}_{\text{rough}}, u_{\text{rough}}, \boldsymbol{x}_p) \tag{8}$$

where $\boldsymbol{x}_p$ is naturally shared between the two cases.

C. Spring Design

The spring physical design variables are the wire diameter $d \in [0.005, 0.02]$ [m], helix diameter $D \in [0.05, 0.4]$ [m], pitch $p \in [0.02, 0.5]$ [m], and number of active coils $N_a \in [3, 16]$. The spring constant k_s is computed by:

$$k_s(\boldsymbol{x}_p) = \frac{d^4 G}{8 D^3 N_a \left[1 + \frac{d^2}{2D^2}\right]} \tag{9}$$

where G is the shear modulus. The quantity $L_0 = pN_a + 2d$ is the free length of the spring. $L_s = d(N_a + Q - 1)$ represents the solid height of the spring with $Q = 1.75$. $C = D/d$ is the spring index. The following constraints are included for manufacturability, antitangling, buckling, minimum pocket length, and minimum pocket width:

$$g_{o,1}(\boldsymbol{x}_p) = 4 - C \leq 0 \tag{10}$$

$$g_{o,2}(\boldsymbol{x}_p) = C - 12 \leq 0 \tag{11}$$

$$g_{o,3}(\boldsymbol{x}_p) = L_0 - 5.26D \leq 0 \tag{12}$$

$$g_{o,4}(\boldsymbol{x}_p) = L_0 - 0.40 \leq 0 \tag{13}$$

$$g_{o,5}(\boldsymbol{x}_p) = d + D - 0.25 \leq 0 \tag{14}$$

The axial force is $F_s = k_s(L_0 - L_s)$, and the shear yield stress is related to the ultimate shear stress by $S_{sy} = 0.65 S_{ut}$, where $S_{ut} = 1974 d^{-0.108} \times 10^6$. The shear stress is:

$$\tau(F) = \left(\frac{4C+2}{4C-3}\right)\frac{8FD}{\pi d^3} \tag{15}$$

Now, we enforce that the maximum shear stress must not be higher than S_{sy}:

$$g_{o,6}(\boldsymbol{x}_p) = 1.2\tau(F_s) - S_{sy} \leq 0 \tag{16}$$

There are also constraints that depend on the dynamic nature of the states. A rattlespace constraint depends on the maximum displacement of the spring:

$$g_{i,1}(\boldsymbol{x}_p, \boldsymbol{\xi}) = \max_t |\xi_3(t)| - L_0 + L_s + 0.02 + \delta_g \leq 0 \tag{17}$$

where $\delta_g = m_{s/4} g / k_s$ is the static suspension deflection with $g = 9.81$ m/s^2. To ensure the spring linearity assumption, we include:

$$g_{i,2}(\boldsymbol{x}_p, \boldsymbol{\xi}) = 0.15 + 1 - \frac{L_0 - L_s}{\delta_g + 1.1\xi_3(t)} \leq 0 \tag{18}$$

The maximum and minimum axial forces are $F_{\max} = k_s(\max_t|\xi_3(t)| + \delta_g)$ and $F_{\min} = k_s(\delta_g - \max_t|\xi_3(t)|)$, respectively. Then, the mean axial force and force amplitude are defined by $F_m = (F_{\max} + F_{\min})/2$ and $F_a = (F_{\max} - F_{\min})/2$, respectively. Soderberg fatigue criterion and Zimmerli limit are considered with:

$$g_{i,3}(\boldsymbol{x}_p, \boldsymbol{\xi}) = \frac{1.2\tau(F_a)}{0.24 S_{ut}} + \frac{\tau(F_m)}{S_{sy}} - 1 \leq 0 \tag{19}$$

$$g_{i,4}(\boldsymbol{x}_p, \boldsymbol{\xi}) = \frac{1.2\tau(F_a)}{241 \times 10^6} - 1 \leq 0 \tag{20}$$

D. Damper Design

There are three parameters for damper design including the valve diameter $D_o \in [0.003, 0.012]$ [m], working piston diameter $D_p \in [0.03, 0.08]$ [m], and damper stroke $D_s \in [0.1, 0.3]$ [m]. The damper constant c_s is:

$$c_s(\boldsymbol{x}_p) = \frac{D_p^4}{8 C_d C_2(D_o) D_o^2}\sqrt{\frac{\pi k_v \rho_1}{2}} \tag{21}$$

where C_d, $C_2(D_o)$, ρ_1, and k_v are parameters in Ref. [4]. The damper must fit inside the spring, thus constraints on its size and range of motion are included:

$$g_{o,7}(\boldsymbol{x}_p) = d - D + D_p + 0.022 \leq 0 \tag{22}$$

$$g_{o,8}(\boldsymbol{x}_p) = 2D_s - 0.394 \leq 0 \tag{23}$$

$$g_{o,9}(\boldsymbol{x}_p) = L_0 - L_s - D_s \leq 0 \tag{24}$$

The dynamic constraints depend on the suspension's relative velocity $\dot{\xi}_3 = \xi_4 - \xi_2$ and are meant to limit the heat generated by the working fluid. These constraints restrict the maximum pressure on the damper fluid, the maximum allowable velocity, and the spool valve lift, respectively:

$$g_{i,5}(\boldsymbol{x}_p, \boldsymbol{\xi}) = \frac{4 c_s(D_o)\max_t|\dot{\xi}_3(t)|}{\pi D_p^2} - 4.75 \times 10^6 \leq 0 \tag{25}$$

$$g_{i,6}(\boldsymbol{x}_p, \boldsymbol{\xi}) = \max_t |\dot{\xi}_3(t)| - 5 \leq 0 \tag{26}$$

$$g_{i,7}(\boldsymbol{x}_p, \boldsymbol{\xi}) = \frac{4\pi D_o^2 c_s(D_o)\max_t|\dot{\xi}_3(t)|}{4 k_v \pi D_p^2} - 0.03 \leq 0 \tag{27}$$

E. Analysis of the CCD Problem

With the entire problem defined, it can be helpful to analyze the CCD problem structure. To this end, a dependency matrix visualization template is provided in Fig. 3 where rows are problem elements from Prob. (1), and the columns are for design variable dependence. A similar analysis was performed in Refs. [7] without the detail provided here. Additionally, this is similar to the Jacobian matrix shown in Ref. [4] with a few key differences. First, the dependencies are categorized as linear, quadratic, and nonlinear, and a specialized qualifier is added for the case when the plant design is fixed (denoted $\boldsymbol{x}_p^\dagger$). For example, the optimization variable dependence for $\boldsymbol{f}(\cdot)$ in Eq. (6) is only linear if the plant design is fixed. By filling in all the appropriate entries, one can determine if their CCD problem is an LQDO-amendable CCD problem by merely verifying the complexity is lower than the level shown in Fig. 3. It also facilitates the partitioning of the inner- and outer-loop constraints. Because of the definitionally disallowed X region, any constraints that do not only depend on $\boldsymbol{x}_p$ must be in the inner-loop.

For this case study, the dependency matrix is shown in Fig. 4. All $\boldsymbol{g}_i(\cdot)$ can be transformed into a linear state form, potentially needing two constraints. Comparing with Fig. 3, this is an LQDO-amendable CCD problem. There are additional uses for this representation. Regions II and IV provide insights into when the inner-loop problem needs to be solved. Here D_s doesn't change the inner-loop problem. Furthermore, dominated inner-loop constraints are observed in Eqs. (17)–(20) because they have the same row properties with only a single linear state variable. Therefore, only one can be active at a time, and this constraint can be easily determined. The same can be said for Eqs. (25)–(27), again reducing three path constraints to only one. Finally, it helps determine in Region V which plant constraints are linear (if the entire row is linear).

This discussion highlights the potential advantage of the nested strategy to structure and simplify the subproblem. Effective nested and simultaneous implementations should take advantage of all of this information.

VI. Results

In this section, we solve the LQDO-amenable CCD problem defined in Sec. V using a variety of coordination strategies, computational methods, and tunable parameters.

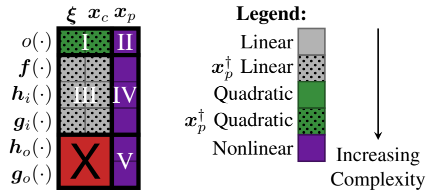


Fig. 3: Allowable dependency matrix form for an LQDO-amendable CCD problem.

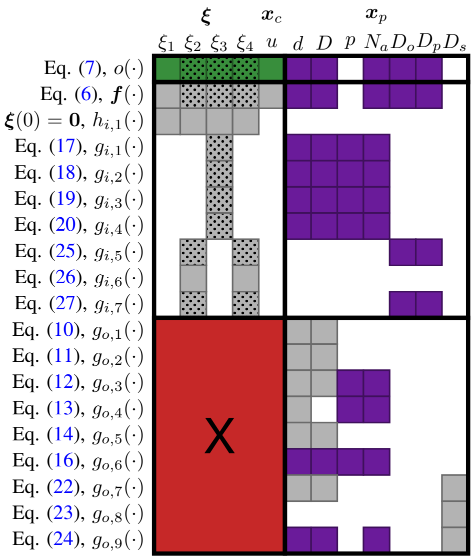


Fig. 4: Dependency matrix for suspension CCD problem.

The case study is available at and solved using the free and open-source *DTQP* software tool (based on commit `33c2f24`) [28]. The computer architecture used for the results was a desktop workstation with an AMD 3970X CPU at 3.7 GHz, 128 GB 3200 MHz RAM, *Matlab* 2020a update 5, and *Windows* 10 build 17763.1432.

A. Complete Solution

Before we begin comparing different instances of the CCD solution strategies, we first want to present a single complete and accurate solution to the CCD problem. The optimal trajectories for select states and controls under the ramp and rough road load cases are shown in Fig. 5. The optimal objective function value o^* was 2.0677, and the ramp and rough road constituents were 56.340 and 1.504, respectively. These performance values were obtain using both the simultaneous (*S*) and nested (*N*) CCD strategies using $n_t = 5000$ and small tolerances to ensure high quality solutions. Interestingly, the plant designs from the two implementations were slightly different as noted in Fig. 5. However, the key intermediate variable values found using *S* of $k_s^* = 2.366 \times 10^4$ N/m and $c_s^* = 839.8$ Ns/m were within 2% of *N* so the optimal state and control trajectories were nearly indistinguishable. In Ref. [4], a solution was demonstrated with $o^* = 2.12$ using a much coarser mesh, but the overall the values of k_s^* and c_s^* and the optimal trajectories were generally similar.

B. Solution Implementation Variations

There are various key decisions to be made when attempting to find a solution to a CCD problem numerically. In this section, we will explore several implementation variations to gain insights into best practices.

The *S* architecture used *fmincon* with the built-in interior point algorithm where all derivatives are provided utilizing the sparsity pattern for DT problems. The variations tested were composed of permutations of the following:

- Three different values of $n_t = [200, 600, 2000]$
- Three optimality tolerance values $[10^{-3}, 10^{-5}, 10^{-7}]$
- Three feasibility tolerance values $[10^{-4}, 10^{-8}, 10^{-12}]$
- Four derivative methods (symbolic, complex-step differentiation, real-central finite difference (FD), and real-forward FD)

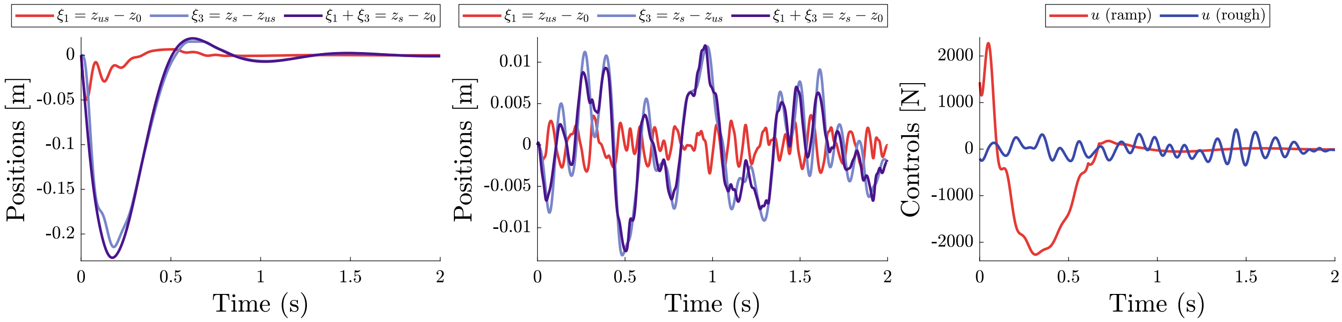
The *N* architecture used *fmincon* with the built-in interior point algorithm for the outer loop and *quadprog* for the inner-loop QP. Parallel computing was utilized in the outer loop for FD. The variations tested were composed of permutations of the following:

- Three different values of $n_t = [200, 600, 2000]$
- Three outer-loop optimality tolerances values $[10^{-1}, 10^{-3}, 10^{-5}]$
- Three inner-loop optimality and feasibility tolerance values $[10^{-4}, 10^{-8}, 10^{-12}]$
- Two outer-loop derivative methods (real-central and real-forward FD)
- If a hybrid approach was used where a genetic algorithm was first run for one iteration (for global search and to find a point with a feasible inner loop). Otherwise a starting point with a feasible inner loop was used.

1) Simultaneous vs. Nested: All the variations are shown in Fig. 6 comparing the relative error of the objective value found above and the optimization runtime on the prescribed computer architecture. In Fig. 6a, the results indicate *S* is the superior approach when using enough points for an accurate time discretization, i.e., $n_t \geq 600$.

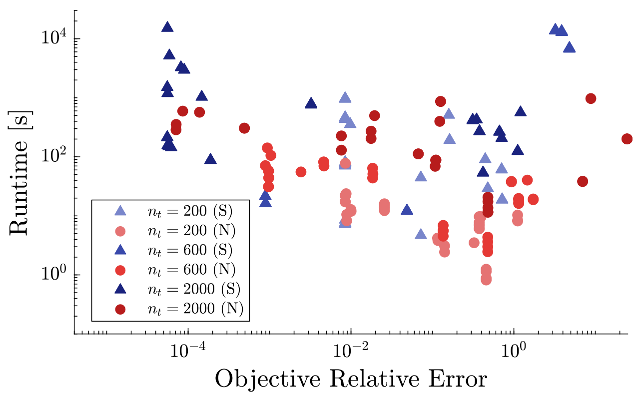
However, it is important to look at the specific implementation that achieved this result. Figure 6b labels the points by the derivative method used. Now we can see why *S* was superior; it was using symbolic derivatives! Accurate derivative information is shown to be a critical factor for efficient *S* implementations. The implementations using the complex-step method, which is nearly as accurate as symbolic derivatives (second-order derivatives are less accurate), was on average 10× slower. Therefore, these results indicate that *N* is superior when symbolic derivatives are unavailable and potentially by an order of magnitude or more in runtime.

2) Additional Simultaneous Insights: The real-central and real-forward FD methods were about 5 to 10× slower than the complex-step implementations. This result also explains why the reported runtimes in Ref. [1] where so much higher than the values reported here, which used the low-order derivative method with a large step size. Not shown in the figure, several instances of *S* using the low-order derivative

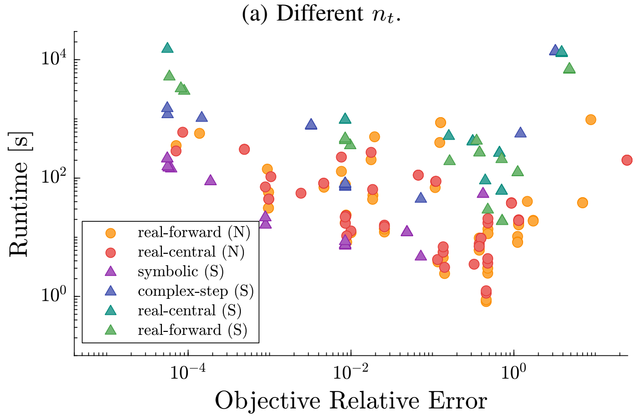


(a) Optimal states for ramp case. (b) Optimal states for rough road case. (c) Optimal controls.

Fig. 5: Optimal trajectories with $n_t = 5000$ and $\boldsymbol{x}_p^* = [0.0200\ 0.1820\ 0.0335\ 10.7308\ 0.0097\ 0.0405\ 0.1700]$ using simultaneous and $\boldsymbol{x}_p^* = [0.0166\ 0.1686\ 0.0396\ 6.5015\ 0.0094\ 0.0395\ 0.1700]$ using nested.

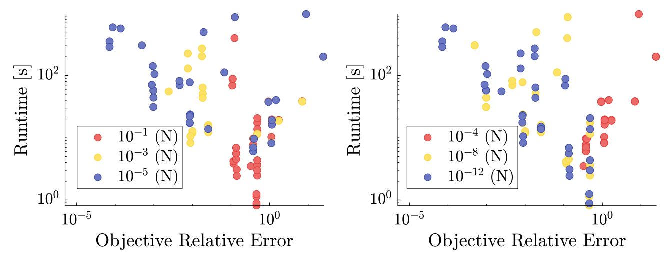


(a) Different n_t.



(b) Different derivative methods.

Fig. 6: Run time vs. relative objective function error for various solution implementations.



(a) Outer-loop problem optimality tolerance. (b) Inner-loop problem feasibility and optimality tolerance.

Fig. 7: Tolerance study results for the nested strategy.

methods did not converge within a large iteration limit. Many models do not readily support symbolic derivatives or complex numbers, so one would need to use another derivative strategy shown to be quite inefficient, even when requesting low accuracy solutions (recalling that many variations on the tolerances were tested). Solutions with $O(10^{-2}, 10^{-3}, 10^{-4})$ accuracy were obtained in (7, 16, 144) s, respectively.

3) Additional Nested Insights: Figure 7 shows only the N implementations highlighting the different inner and outer loop tolerances. In Fig. 7a, the outer-loop tolerance shows clear trade-off between accuracy and runtime. Loss in outer-loop accuracy is commonplace when the inner-loop tolerance is not small enough (cf. Fig. 7b). Balancing these tolerances can permit efficient CCD solution implementations that achieve the desired accuracy with extra computational cost. Solutions with $O(10^{-2}, 10^{-3}, 10^{-4})$ accuracy were obtained in (8, 31, 287) s, respectively.

A fundamental assumption for equivalence between the nested and simultaneous formulations is consistent inner-loop feasibility. However, uniform random sampling of plant designs within the simple upper and lower bounds indicate 44% have infeasible inner loops, and approximately 0.033% are infeasible with respect to all constraints! The starting point used in Ref. [4] is one of those infeasible points, so a simple nested implementation would not converge. In this case, infeasible inner loops are when some of the additional inequalities are not well-posed. In fact, the inner-loop QP is immediately declared infeasible by `quadprog`.

A possible deterrent against this issue is a hybrid optimization scheme. In this study, the hybrid scheme consisted of a genetic algorithm and then a derivative-based optimizer. Additionally, implementing the five linear $\boldsymbol{g}_o(\cdot)$ constraints appropriately (so not as nonlinear constraints) was quite helpful in finding better populations and search directions because many optimization algorithms can directly avoid linear constraint infeasibility regions. Also, including appropriate outer-loop feasibility constraints, as suggested in Ref. [12], could help guide the outer-loop towards a feasible inner-loop region and would take the form of positive bound coefficients in $g_{i,1}(\cdot)$ through $g_{i,4}(\cdot)$ in this case.

As the complexity of CCD design problems increases, this issue could become more prevalent. Even with the observed runtime savings and other potential benefits, the nested strategy may simply not be robust enough for problems with challenging feasible regions.

VII. Conclusion

In this article, the simultaneous analysis and design and the nested control problem formulation CCD strategies were presented and compared. The varied results of the case study showed that the simultaneous strategy using symbolic derivatives was generally superior while the nested strategy was preferable when any other derivative method was used with the simultaneous strategy. However, special care is needed to handle potentially infeasible inner loops.

It is important to highlight that this case study uses an LQDO-amendable CCD problem so the structure of the inner-loop problem permits efficient solution methods. For other general CCD problems, the observed trends might be different. In general, these results indicate the potential advantages of capitalizing on problem structure, selecting the most effective implementation, and understanding the goals of a particular CCD study. While this particular CCD problem had many features that could be explored and exploited, each individual CCD problem may have different challenges, and therefore, different implementation study outcomes. However, this study highlights the need for more fair comparisons between the strategies and optimization implementations because there can be significant differences than when using naïve implementations.

Furthermore, it is imperative to understand the trade-offs and goals of a particular CCD study (e.g., desired accuracy) and use sound judgment to make informed decisions on how to solve a chosen CCD problem. Future work will seek to understand these trade-offs and best practices in a more diverse set of CCD problem scenarios moving towards a robust set of CCD implementation guidelines. Detailed theoretical and numerical investigations and approachable and effective CCD tools are critical to increasing the applicability and impact of the CCD methodology.

References

[1] J. T. Allison and D. R. Herber, “Multidisciplinary design optimization of dynamic engineering systems,” *AIAA J.*, vol. 52, no. 4, pp. 691–710, Apr. 2014, doi: 10.2514/1.j052182

[2] H. K. Fathy, J. A. Reyer, P. Y. Papalambros, and A. G. Ulsoy, “On the coupling between the plant and controller optimization problems,” in *American Control Conference*, 2001, doi: 10.1109/acc.2001.946008

[3] M. Garcia-Sanz, “Control co-design: an engineering game changer,” *Adv. Control Appl.*, vol. 1, no. 1, Oct. 2019, doi: 10.1002/adc2.18

[4] J. T. Allison, T. Guo, and Z. Han, “Co-design of an active suspension using simultaneous dynamic optimization,” *J. Mech. Des.*, vol. 136, no. 8, Jun. 2014, doi: 10.1115/1.4027335

[5] S. Azad, M. Behtash, A. Houshmand, and M. J. Alexander-Ramos, “PHEV powertrain co-design with vehicle performance considerations using MDSDO,” *Struct. Multidiscip. Optim.*, vol. 60, no. 3, pp. 1155–1169, Apr. 2019, doi: 10.1007/s00158-019-02264-0

[6] A. L. Nash and N. Jain, “Hierarchical control co-design using a model fidelity-based decomposition framework,” *J. Mech. Des.*, vol. 143, no. 1, Aug. 2020, doi: 10.1115/1.4047691

[7] C. M. Chilan, D. R. Herber, Y. K. Nakka, S.-J. Chung, J. T. Allison, J. B. Aldrich, and O. S. Alvarez-Salazar, “Co-design of strain-actuated solar arrays for spacecraft precision pointing and jitter reduction,” *AIAA J.*, vol. 55, no. 9, pp. 3180–3195, Sep. 2017, doi: 10.2514/1.J055748

[8] D. R. Herber, “Dynamic system design optimization of wave energy converters utilizing direct transcription,” M.S. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, May 2014.

[9] A. P. Deshmukh and J. T. Allison, “Multidisciplinary dynamic optimization of horizontal axis wind turbine design,” *Struct. Multidiscip. Optim.*, vol. 53, no. 1, pp. 15–27, Aug. 2015, doi: 10.1007/s00158-015-1308-y

[10] J. Jonkman, A. Wright, G. Barter, M. Hall, J. T. Allison, and D. R. Herber, “Functional requirements for the WEIS toolset to enable controls co-design of floating offshore wind turbines,” in *International Offshore Wind Technical Conference*, no. IOWTC2020-3533, Feb. 2021.

[11] J. R. R. A. Martins and A. B. Lambe, “Multidisciplinary design optimization: a survey of architectures,” *AIAA J.*, vol. 51, no. 9, pp. 2049–2075, Sep. 2013, doi: 10.2514/1.j051895

[12] D. R. Herber and J. T. Allison, “Nested and simultaneous solution strategies for general combined plant and control design problems,” *J. Mech. Des.*, vol. 141, no. 1, Jan. 2019, doi: 10.1115/1.4040705

[13] H. K. Fathy, P. Y. Papalambros, A. G. Ulsoy, and D. Hrovat, “Nested plant/controller optimization with application to combined passive/active automotive suspensions,” in *American Control Conference*, 2003, doi: 10.1109/acc.2003.1244053

[14] J. A. Reyer and P. Y. Papalambros, “Combined optimal design and control with application to an electric DC motor,” *J. Mech. Des.*, vol. 124, no. 2, pp. 183–191, May 2002, doi: 10.1115/1.1460904

[15] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, Jan. 2010, doi: 10.1137/1.9780898718577

[16] W. K. Belvin and K. C. Park, “Structural tailoring and feedback control synthesis - an interdisciplinary approach,” *J. Guid. Control. Dynam.*, vol. 13, no. 3, pp. 424–429, May 1990, doi: 10.2514/3.25354

[17] J. Onoda and R. T. Haftka, “An approach to structure/control simultaneous optimization for large flexible spacecraft,” *AIAA J.*, vol. 25, no. 8, pp. 1133–1138, Aug. 1987, doi: 10.2514/3.9754

[18] L. T. Biegler, “An overview of simultaneous strategies for dynamic optimization,” *Chem. Eng. Process. Process Intensif.*, vol. 46, no. 11, pp. 1043–1053, Nov. 2007, doi: 10.1016/j.cep.2006.06.021

[19] ——, *Nonlinear Programming*. Society for Industrial and Applied Mathematics, Jan. 2010, doi: 10.1137/1.9780898719383

[20] D. R. Herber, “Advances in combined architecture, plant, and control design,” Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Dec. 2017.

[21] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso, “The complex-step derivative approximation,” *ACM Trans. Math. Software*, vol. 29, no. 3, pp. 245–262, Sep. 2003, doi: 10.1145/838250.838251

[22] D. R. Herber and A. K. Sundarrajan, “On the uses of linear-quadratic methods in solving nonlinear dynamic optimization problems with direct transcription,” in *International Mechanical Engineering Congress & Exposition*, no. IMECE2020-23885, Nov. 2020, doi: 10.1115/IMECE2020-23885

[23] D. R. Herber and J. T. Allison, “A problem class with combined architecture, plant, and control design applied to vehicle suspensions,” *J. Mech. Des.*, vol. 141, no. 10, May 2019, doi: 10.1115/1.4043312

[24] X. Hu, S. J. Moura, N. Murgovski, B. Egardt, and D. Cao, “Integrated optimization of battery sizing, charging, and power management in plug-in hybrid electric vehicles,” *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 1036–1043, May 2016, doi: 10.1109/TCST.2015.2476799

[25] J. A. Reyer, H. K. Fathy, P. Y. Papalambros, and A. G. Ulsoy, “Comparison of combined embodiment design and control optimization strategies using optimality conditions,” in *Design Engineering Technical Conference*, no. DETC2001/DAC-21119, 2001.

[26] G. Clarizia, “Co-design optimization of a tethered multi drone system,” M.S. Thesis, Politecnico di Milano, Milano, Italy, 2019.

[27] J. Dongarra, J. L. Martin, and J. Worlton, “Computer benchmarking: paths and pitfalls,” *IEEE Spectr.*, vol. 24, no. 7, pp. 38–43, 1987, doi: 10.1109/MSPEC.1987.6448963

[28] “The DTQP project,” [Online], url: https://github.com/danielrherber/dt-qp-project