AIAA JOURNAL
Vol. 36, No. 5, May 1998

Control-Structure Integrated Design with Closed-Form Design Metrics Using Physical Programming

Achille Messac*
Northeastern University, Boston, Massachusetts 02115

The technological area of control-structure integrated design (CSID) has witnessed significant evolution since the early 1980s. This paper builds on recent developments in the area of computational optimization and on new analytical methods developed herein to make three distinct but synergistic contributions. First, it explores the application of the physical programming optimization method to the CSID problem. Second, a new approach is developed for the closed-form evaluation of time-domain performance metrics for high-order systems, e.g., settling time. Third, a method for design-variable-linking that uses splines is proposed and is shown to be computationally beneficial while also leading to more physically meaningful structural designs. The numerical results obtained for a controlled idealized rotating structure suggest that the analytical developments of this paper might have practical applicability.

I. Introduction

RESEARCH in the area of control-structure integrated design (CSID) is motivated by the risk that the traditional sequential structure/control design might not adequately exploit interdisciplinary dependence.[1-20] Under the traditional sequential design approach, the structure is designed first, and the controller is designed next. This results in a lack of design integration that typically requires several ad hoc redesigns to converge to a satisfactorily performing system.

The basic classical steps of the design process for controlled structures can be outlined as follows. 1) The structural engineer designs a structure by primarily addressing the structural objectives, such as the overall mass and dimensions of the structure, stress requirements, and such quantities as natural frequencies. 2) Subsequently, the control engineer designs a control system for the structure, which has already been designed to meet the structural objectives. At this stage, the control engineer deals with a different set of objectives, such as good settling time and command-following properties, robustness to disturbance and unmodeled plant dynamics, and good active vibration control. 3) Finally, because the resulting controller does not usually fulfill the overall design requirements, the previous steps are repeated until—one hopes—all of the design goals are achieved.

To understand why the sequential approach often fails to capture the features of an optimal system design, we observe that the CSID problem is both multidisciplinary and multiobjective in nature. The governing design metrics are typically highly nonlinear in terms of the design parameters and usually are mutually conflicting. We also note that the discipline that is addressed first (in this case structure) generally does not properly account for the second disciplinary design (in this case control). This barrier of communication is at the heart of the deficiency that plagues any sequential multidisciplinary system design effort. The preceding statement is therefore not limited to the CSID problem. Although the CSID problem was motivated by the difficulties posed by the design of large space structures, later research has broadened to include a wider base of applications. High-speed aircraft also suffer from control/structure interaction problems.[1,2] Similar problems are observed in flexible, high-speed positioning robot arms, which are widely used in personal computer board assembly lines.[3-5]

Over the past 15 years, CSID methodological approaches have evolved significantly. The initial CSID methods were based on linear quadratic regulator (LQR) optimal control techniques.[6,7] In this approach, the control variables are eliminated from the analysis by representing the optimal control metric as a function of structural variables. Belvin and Park[8] used the same approach together with modal-space control for structural tailoring. An extended version of this approach is to create a scalar objective function that aggregates the weighted sum of the optimal control quadratic index and the total structural mass.[9-11] As is discussed in the following, the major shortcoming of this approach is the determination of the proper weights that yield a desirable solution.

Lust and Schmit[12] simultaneously optimized the structural and controller gain parameters. Instead of using a single cost function, they iteratively generated approximate problems in an effort to minimize numerical difficulties and to promote faster convergence. Their original work addressed harmonically loaded structures and collocated control problems. Thomas et al.[13] extended the approach to noncollocated problems and improved the quality of the approximations. Gilbert and Schmidt[14] proposed a different approach to combine structure and control design via multilevel optimization techniques. They considered the design of the structure and the control system separately and then combined these solutions using multilevel optimization techniques. Canfield and Meirovich[15] used independent modal space control to design structural and control systems together. Their approach invoked the LQR optimal control problem in conjunction with independent modal space control to design structures with good active vibration control properties. Messac and Malek[16] developed a scalar measure that integrates the disturbance rejection, control effort, and tracking properties of the structural system.

Some of the more recent research focused on robustness issues and on the application of more advanced control theory to the CSID problem. Slater and McLaren[1] developed a disturbance model for the CSID framework. Khot and Heise[17] considered the structured and unstructured plant uncertainties using H_∞ control design methods. Niewoehner and Kaminer[2] introduced a new CSID methodology for aircraft design problems by representing the design objectives as linear matrix inequalities.

CSID concepts are also applied to the design of high-speed flexible robot arms. Asada et al.[4] considered the design of a flexible single-link manipulator. They modeled the robot arm as a flexible beam via the finite element method and searched for an optimal shape that yields the desired closed-loop, pole-zero locations. Park and Asada[3] considered a two-link manipulator and performed CSID to increase the high-speed positioning ability.

In a recent work, Cheng and Li[18] addressed the importance of formulating CSID as multiobjective optimization with a properly generated objective function. They formed a payoff matrix by performing optimization with each design objective. Then they constructed an aggregate objective function for overall system performance.

Received Dec. 18, 1996; revision received Dec. 7, 1997; accepted for publication Jan. 2, 1998.

*Associate Professor, Mechanical Engineering Department. E-mail: messac@coe.neu.edu. Associate Fellow AIAA.

Next, we provide some brief comments on the use of weights in forming objective functions. To facilitate the discussion, we use the LQR method as an example.

Under a common CSID approach, the LQR performance index is used as the objective function to be minimized while other objectives are introduced to the optimization problem as hard constraints.[6,7] A different approach is to minimize the total mass while the control requirements are treated as hard constraints. Unfortunately, using one of the criteria as the metric to be minimized (or maximized) and treating the others as hard constraints (inequalities) is often a poor means of addressing multiobjective optimization problems. Real-world designer preferences are more adequately expressed in degrees of desirability than they are in hard statements. In line with that thinking, some researchers propose to form an aggregate weighted sum of the total mass and the LQR performance index.[9–11] This aggregation takes the form

$$J = \alpha W + \beta \frac{1}{2} \int_0^\infty (x^T Q x + u^T R u)\, dt \tag{1}$$

where W is the total structural mass, Q and R are the weighting matrices for the states and the control effort, respectively, and α and β are, respectively, the weights for the mass and LQR cost index.[1] As emphasized by several researchers, the values of the weights α and β, as well as the values of the LQR index weighting matrices Q and R, are quite difficult to determine.[1,19] Frequently, the weight matrices Q and R are formed as $Q = \text{diag}\,(M, K)$ to minimize the structure's total energy, and $R = B^T K^{-1} B$ to minimize the energy transferred to the structure by the controller.[8,15,20] The matrices M, K, and B are, respectively, the mass, stiffness, and actuator influence coefficient matrices. Unfortunately, there is no guarantee that these formulations will yield the solution that the designer seeks, in a local sense, e.g., minimization of a given nodal deformation. Instead, these approaches to forming the weights matrices address the performance objectives in an aggregate fashion, e.g., total potential energy and total kinetic energy minimization. Similarly, proper assignment for the weights α and β is a tedious task that usually requires several iterations. We also note that the situation is further complicated by the possible need to add more design metrics to the aggregate objective function of Eq. (1). Unfortunately, to date there is no general reliable and robust method for determining the correct values for weights.

At this point, it is appropriate to reflect on the practical consequences of the preceding discussion regarding weights. First, borrowing from Messac,[21] we note that computational design optimization typically involves three distinct phases: 1) Model the physical system in terms of design parameters and design metrics, 2) form an aggregate objective function in terms of the design metrics, and 3) minimize the aggregate objective function using an optimization code. Analytical and computational tools available to perform the first and third phases can be considered robust. On the other hand, analytical tools available for constructing the objective function in phase 2 are remarkably simplistic and generally involve difficult-to-obtain weights. Because the optimum solution is only as effective as the aggregate objective function, any deficiency in the formation of the latter significantly impacts the outcome. Second, we note that nearly all computational optimization approaches require choosing weights explicitly or implicitly. Finally, in light of the preceding comments, we believe that methodological advances in the area of weight determination, or in the elimination of the need for choosing weights, would be singularly beneficial to the field of computational design.

A new approach to computational design optimization entitled physical programming has recently been developed.[22] The approach entirely removes the necessity to chose weights and allows designers to reflect their design preferences in terms of each design metric in an explicit and flexible manner. Under the physical programming paradigm, the designers' preference for a generic design metric is expressed in terms of six ranges of differing degrees of desirability. Specifically, the designers provide numerical values that define the boundaries of each of these ranges for each design metric. For example, if mass is one of the design metrics that the designers wish to minimize—among many others—in a design effort, they might express their mass-related preference as follows:

Highly desirable	<20 kg
Desirable	20–30 kg
Tolerable	30–40 kg
Undesirable	40–50 kg
Highly undesirable	50–80 kg
Unacceptable	>80 kg

The physical programming method uses the preceding information—and similar statements regarding other metrics—to automatically form the aggregate objective function.

The effectiveness of physical programming to perform multiobjective design has been examined in several publications.[21–24] Messac and Wilson[23] used physical programming to design a robustly stable controller for a benchmark problem that consists of a mass-spring-mass noncollocated control system. Their approach was to simulate the closed-loop system and to obtain the desired design metrics from the time-history simulation results. Although this approach works well for low-order systems, the large number of time simulations that is required during optimization becomes computationally prohibitive for high-order systems. Motivated by the realization of this shortcoming, this paper develops an efficient closed-form method for evaluating time-domain metrics. This method eliminates the need for time simulations during the optimization process.

The main focus of this paper is to explore the effectiveness of physical programming in the CSID context, assuming computationally efficient and physically meaningful design metrics are available. In this spirit, we also develop efficient means of evaluating design metrics, and we examine the use of splines for design-variable-linking.

The paper is organized as follows. Section II presents a synopsis of the physical programming method. In Sec. III, the closed-loop system equations are developed. The computationally efficient time-domain closed-form design metrics are developed in Sec. IV. In Sec. V, the formulation of the multiobjective optimization problem is discussed. The use of a convex spline to link design variables is presented in Sec. VI. Section VII presents a numerical example with results, where a rotating structure is designed. Concluding remarks are provided in Sec. VIII.

II. Physical Programming Synopsis

This section presents a synopsis of the physical programming method. For a comprehensive presentation and related additional information, please see Refs. 21–24. To apply physical programming, one may either code the procedure outlined in Ref. 22 or use the software package PhysPro[25] (implemented in the MATLAB environment), which embodies the physical programming method.

Within the physical programming procedure, designers express their wishes with respect to each design metric using four different classes, defined shortly. Each class comprises two possibilities, hard and soft, referring to the sharpness of the preference. The resulting eight possibilities are referred to as subclasses. Designers assign a subclass to each design metric. Figure 1 depicts the qualitative meaning of each of the four soft subclasses. The generic design metric μ_i is a function of the design variable vector $\boldsymbol{d}$. The value of the criterion under consideration, μ_i, is represented on the horizontal axis, and the pertaining function that will be minimized, which is called a preference function, P_i, is on the vertical axis. For a preference function of a given design metric, the lower the value of the preference function, the more desirable the related design metric value. All preference functions become additive constituent components of the aggregate objective function, which is minimized.

As stated earlier, the desired behavior of a generic design metric is described by one of the eight subclasses, four soft (S) and four hard (H). These subclasses are as follows.

Soft:

Class 1S	smaller-is-better, i.e., minimization
Class 2S	larger-is-better, i.e., maximization
Class 3S	value-is-better
Class 4S	range-is-better

Table 1 Preference table for spacecraft example

Metrics:	Highly desirable	Desirable	Tolerable	Undesirable	Highly undesirable
Parameter	ν_{i1}	ν_{i2}	ν_{i3}	ν_{i4}	ν_{i5}
t_{tip}	150	180	210	230	250
u_{tip}	4e–5	7e–5	10e–5	12e–5	14e–5
t_{hub}	110	120	135	145	155
m	42	44	46	48	50

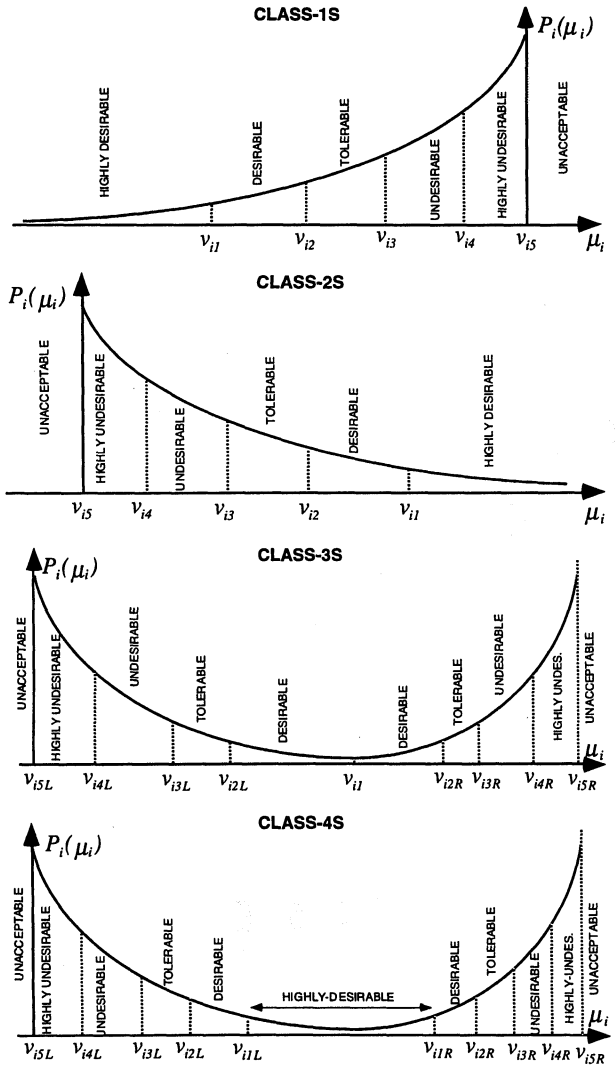


Fig. 1 Preference functions for design metrics.

Hard:

Class 1H	must be smaller, i.e., $\mu_i \leq \nu_{i,\max}$
Class 2H	must be larger, i.e., $\mu_i \geq \nu_{i,\min}$
Class 3H	must be equal, i.e., $\mu_i = \nu_{i,\text{val}}$
Class 4H	must be in range, i.e., $\nu_{i,\min} \leq \mu_i \leq \nu_{i,\max}$

The structure of the preference functions provides the means for designers to express the ranges of differing levels of preference for each soft design metric.

To construct the preference functions, the physical programming lexicon comprises terms that characterize the degrees of desirability of each metric, using six types of ranges. The ranges are defined as follows, in order of decreasing preference: 1) *highly desirable* range: ($\mu_i \leq \nu_{i1}$), an acceptable highly desirable range; a range over which the improvement that results from further change in the design metric is desired but is of minimal additional value; 2) *desirable* range: ($\nu_{i1} \leq \mu_i \leq \nu_{i2}$), an acceptable range that is desirable; 3) *tolerable* range: ($\nu_{i2} \leq \mu_i \leq \nu_{i3}$), an acceptable, tolerable range; 4) *undesirable* range: ($\nu_{i3} \leq \mu_i \leq \nu_{i4}$), a range that, while acceptable, is undesirable; 5) *highly undesirable*: ($\nu_{i4} \leq \mu_i \leq \nu_{i5}$), a range that, while still acceptable, is highly undesirable; and 6) unacceptable range: ($\nu_{i5} \leq \mu_i$), the range of values that the design metric must not take.

The range-boundary values ν_{ij} are prescribed by the designers as part of the problem statement. Using the preceding values and terminology, the physical programming problem model takes the following form:

$$\min_{\boldsymbol{d}} P(\boldsymbol{\mu}) = \frac{1}{n_{\text{SC}}} \sum_{i=1}^{n_{\text{SC}}} P_i[\mu_i(\boldsymbol{d})] \qquad \text{(for soft classes)} \tag{2}$$

subject to

$\mu_i(\boldsymbol{d}) \leq \nu_{i5}$	(for class 1S metrics)
$\mu_i(\boldsymbol{d}) \geq \nu_{i5}$	(for class 2S metrics)
$\nu_{i5L} \leq \mu_i(\boldsymbol{d}) \leq \nu_{i5R}$	(for class 3S metrics)
$\nu_{i5L} \leq \mu_i(\boldsymbol{d}) \leq \nu_{i5R}$	(for class 4S metrics)
$\mu_i(\boldsymbol{d}) \leq \nu_{i,\max}$	(for class 1H metrics)
$\mu_i(\boldsymbol{d}) \geq \nu_{i,\max}$	(for class 2H metrics)
$\mu_i(\boldsymbol{d}) = \nu_{i,\text{val}}$	(for class 3H metrics)
$\nu_{i,\min} \leq \mu_i(\boldsymbol{d}) \leq \nu_{i,\max}$	(for class 4H metrics)
$d_{j,\min} \leq d_j \leq d_{j,\max}$	(for design parameter constraints)

where $\nu_{i,\min}$, $\nu_{i,\max}$, $\nu_{i,\text{val}}$, $d_{j,\min}$, and $d_{j,\max}$ represent prescribed values. The range limits are provided by the designers (see Table 1 and Fig. 1), n_{SC} is the number of soft metrics that the problem comprises, and $\boldsymbol{\mu}$ is the vector of design metrics. In Fig. 1, only the soft subclasses are shown, as the hard subclasses are inequalities that are self-explanatory. Note that the aggregate preference function P only comprises preference functions, which are associated with soft metrics. The hard metrics are treated as constraints.

The next section develops the system closed-loop equations, which are used in the design metrics evaluation.

III. Closed-Loop System Equations

In this section, the closed-loop equations of a structural plant with a controller are developed.

Open-Loop Plant Equations

The equations of motion for a linear structural system can be written as

$$M\ddot{\boldsymbol{q}} + D\dot{\boldsymbol{q}} + K\boldsymbol{q} = B\boldsymbol{u} \tag{3}$$

where M, D, K, and B, respectively, denote the mass, damping, stiffness, and the force influence coefficient matrices; $\boldsymbol{q}$ is the vector of nodal coordinates; and u is the generalized force vector. For later developments, B and u are partitioned as

$$B = [B_{qc} \quad B_{qd}], \qquad \boldsymbol{u} = \begin{Bmatrix} \boldsymbol{u}_c \\ \boldsymbol{u}_d \end{Bmatrix} \tag{4}$$

where u_c and u_d, respectively, represent the control forces and external disturbances. By defining

$$\boldsymbol{x} = \begin{Bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{Bmatrix} \tag{5}$$

as the state vector, the related state equations take the form

$$\dot{\boldsymbol{x}} = A_q\boldsymbol{x} + B_q\boldsymbol{u} \tag{6}$$

$$y = C_q\boldsymbol{x} + D_q\boldsymbol{u} \tag{7}$$

where the system and influence matrices are given by

$$A_q = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}D \end{bmatrix}, \qquad B_q = \begin{bmatrix} 0 & 0 \\ M^{-1}B_{qc} & M^{-1}B_{qd} \end{bmatrix} \tag{8}$$

The output vector is partitioned as

$$y = \begin{Bmatrix} z_p \\ y_0 \\ u_c \end{Bmatrix} \tag{9}$$

where, from top to bottom, the constituent components are the plant error, the observation, and the control vectors. The corresponding influence coefficient matrices take the form

$$C_q = \begin{bmatrix} C_{qz} \\ C_{qy} \\ 0 \end{bmatrix}, \qquad D_q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I & 0 \end{bmatrix} \tag{10}$$

As usual, for high-order systems modal decomposition is performed. We assume the coordinate transformation

$$q = \phi\eta \tag{11}$$

where ϕ is the modal matrix, which satisfies the relations

$$\phi^T M\phi = I, \qquad \phi^T K\phi = \Lambda = \text{diag}(\lambda_i) \tag{12}$$

$$\phi^T D\phi = D_\eta = \text{diag}(2\zeta_i\omega_i), \qquad \omega_i = \sqrt{\lambda_i} \tag{13}$$

which lead to the governing equation

$$\ddot{\eta} + D_\eta\dot{\eta} + \Lambda\eta = \phi^T Bu \tag{14}$$

The modal state equations can be expressed as

$$\dot{x}_\eta = A_\eta x_\eta + B_\eta u \tag{15}$$

$$y = C_\eta x_\eta + D_\eta u \tag{16}$$

where

$$x_\eta = \begin{Bmatrix} \eta \\ \dot{\eta} \end{Bmatrix} \tag{17}$$

$$A_\eta = \begin{bmatrix} 0 & I \\ -\Lambda & -D_\eta \end{bmatrix}, \qquad B_\eta = [B_{\eta c} \quad B_{\eta d}] \tag{18}$$

with

$$B_{\eta c} = \begin{bmatrix} 0 \\ \phi^T B_{qc} \end{bmatrix}, \qquad B_{\eta d} = \begin{bmatrix} 0 \\ \phi^T B_{qd} \end{bmatrix} \tag{19}$$

The modal output coefficient matrices are written as

$$C_{\eta z} = C_{qz}\begin{bmatrix} \phi & 0 \\ 0 & \phi \end{bmatrix}, \qquad C_{\eta y} = C_{qy}\begin{bmatrix} \phi & 0 \\ 0 & \phi \end{bmatrix}, \qquad D_\eta = D_q \tag{20}$$

Compensator Realization

The multi-input/multi-output compensator is assumed to have the following realization:

$$\dot{x}_c = A_c x_c + B_c e_c \tag{21}$$

$$u_c = C_c x_c + D_c e_c \tag{22}$$

where x_c is the controller state vector and e_c is the error vector, which is fed back to the controller and defined as

$$e_c = r - y_o \tag{23}$$

where r is the reference signal that should be tracked.

Closed-Loop Equations

The closed-loop state equations can be expressed as

$$\dot{x}_t = A_t x_t + B_t u_t \tag{24}$$

$$y_t = C_t x_t + D_t u_t \tag{25}$$

The inputs of the closed-loop system are the reference signals and the disturbance forces. If we let

$$x_t = \begin{Bmatrix} x_\eta \\ x_c \end{Bmatrix}, \qquad u_t = \begin{Bmatrix} u_d \\ r \end{Bmatrix}, \qquad B_t = [B_{td} \quad B_{tr}] \tag{26}$$

then the closed-loop system matrix and the influence matrices can be written as

$$A_t = \begin{bmatrix} A_\eta - B_{\eta c}D_c C_{\eta y} & B_{\eta c}C_c \\ -B_c C_{\eta y} & A_c \end{bmatrix} \tag{27}$$

$$B_{td} = \begin{bmatrix} B_{\eta d} \\ 0 \end{bmatrix}, \qquad B_{tr} = \begin{bmatrix} B_{\eta c}D_c \\ B_c \end{bmatrix} \tag{28}$$

The explicit definition of the output vector in Eq. (25), which is used in constructing the design metrics, is provided in the following section.

IV. Development of Closed-Form Design Metrics

When the physical system under consideration has simple dynamical properties, the corresponding mathematical model is usually of low order. For such systems, the simulation time for the plant is computationally inexpensive, and the evaluation of time-domain design metrics required for the optimization is feasible. Such design metrics might include settling time, maximum control effort, and average nodal displacement. These quantities can be gathered through simulations.[23] However, this approach is not practical for realistic structural plants. In these cases, the closed-loop model of the system is usually both of high order and of high bandwidth (requiring small integration time steps). The difficulty of dealing with such systems in an optimization setting usually forces researchers to use indirect (aggregate) measures of system performances. This section addresses this problem by developing a class of computationally efficient system performance metrics or design metrics. These metrics are used later for the formulation of the optimization problem in accordance with the physical programming framework.

Closed-Form Evaluation of Settling Time

We start with the development of a new approach to define and to evaluate the settling time of a stable system. The settling time of a signal is often defined as the time required for this signal to settle within a certain percentage of its final value. Common practice is to choose a percentage in the range of 2–5. The particular choice typically changes for different applications. We note that, when the final value of the signal is zero, the preceding definition degenerates because every percentage of zero is zero. Some minor redefinitions can address that problem. For example, one could use the percentage of the difference between the initial and final values of the signal.

In practice, analytical expressions for the settling time of a system of order 1 or 2 are used to determine system performance. However, when the system is of order higher than 2, the situation is significantly more complicated. In this case, the time-domain system response can be explicitly examined but at significant computational expense. And when this time-domain simulation is required repeatedly during optimization, the computational nature of the procedure is particularly taxing.

To address the difficulty defined earlier, we propose the solution of the following equation as a new way to define and to evaluate the settling time t_s for high-order systems. We let

$$\frac{I^\infty - I^{t_{sp}}}{I^\infty} = \tilde{p}, \qquad \tilde{p} = \frac{p}{100}, \qquad (0 \le \tilde{p} \le 1), \qquad (0 \le p \le 100) \tag{29}$$

for a given signal $z(t)$, where

$$I^\infty = \int_0^\infty z^2(t)\,dt, \qquad I^{t_{sp}} = \int_0^{t_{sp}} z^2(t)\,dt \tag{30}$$

and p is a specified area percentage value. We will call the value of t_{sp} obtained from Eq. (29) the area settling time to distinguish it from the classical (percent-magnitude) settling time. We observe that the basic idea of the area settling time is the same as that of the classical settling time. Equation (29) simply states that we use the percent-area instead of the percent-magnitude of the signal. As will be seen in the following, the advantage of this definition is that it allows the determination of an effective settling time by using integrals whose values can be easily evaluated.[16]

We now investigate the relationship between the area settling time (AST) and the classical settling time (CST). To do so, we evaluate and compare the AST and the CST for a generic exponential function $[z(t) = \exp -\alpha t, \alpha > 0]$. It is easy to show that the classical $\gamma\%$ settling time $t_{s\gamma}$ is given by

$$\text{CST} = t_{s\gamma} = -(\ell_n \tilde{\gamma}/\alpha), \qquad \tilde{\gamma} = (\gamma/100)$$
$$(0 \le \tilde{\gamma} \le 1), \qquad (0 \le \gamma \le 100) \quad (31)$$

where γ is the percentage of the peak below which the signal is considered settled. We emphasize that p is a percentage of area, whereas γ is a percentage of magnitude. Using the generic exponential function, Eqs. (29) and (30) lead to

$$I^{\infty} = 1/2\alpha, \qquad I^{t_{sp}} = 1/2\alpha(1 - e^{-2\alpha t_{sp}}), \qquad \tilde{p} = e^{-2\alpha t_{sp}} \quad (32)$$

which provide the area settling time as

$$t_{sp} = -(\ell_n \tilde{p}/2\alpha) \quad (33)$$

Equating $t_{s\gamma}$ [Eq. (31)] to t_{sp} [Eq. (33)] leads to the relationships

$$\tilde{p} = \tilde{\gamma}^2, \qquad p = 0.01\gamma^2 \quad (34)$$

which are, interestingly, independent of α and state that the area-fraction is equivalent to the square of the magnitude-fraction. Equation (34) provides an empirical relationship for assigning a value to p in Eq. (29) to evaluate the AST, when the desired percent-magnitude γ for CST is prescribed. For instance, if a percent-magnitude of $\gamma = 10\%$ is desired, then we can set the corresponding percent-area to $p = 1\%$ in Eq. (29) for the AST. Although Eq. (34) is exact only for an exponential function, it has been used with reasonable success for nonexponential decaying functions to get an approximate value for p (as is shown in the following).

The time-domain behavior of the system will be quantified by using the AST defined in Eq. (33). Consider the output of the closed-loop system given by Eq. (25). The overall output of the closed-loop system will consist of three basic quantities: 1) the deformation of some selected structural nodes (as a measure of vibration level), 2) the plant output $\boldsymbol{y}_o$ (as a measure of command-following properties), and 3) the control effort. To find the settling time of these signals, we use Eqs. (29) and (30). Let $\boldsymbol{y}_t$ be a general output vector for Eq. (25), which is repeated here as

$$\boldsymbol{y}_t = C_t \boldsymbol{x}_t + D_t \boldsymbol{u}_t \quad (35)$$

and let y_t^i be the ith element of this vector. Then

$$I_i^{\infty} = \int_0^{\infty} y_t^{i^2}(t)\,dt = \int_0^{\infty} y_t^T Q^i y_t\,dt \quad (36)$$

$$I_i^{t_s} = \int_0^{t_s} y_t^{i^2}(t)\,dt = \int_0^{t_s} y_t^T Q^i y_t\,dt \quad (37)$$

where Q^i is a properly dimensioned square matrix of zeros except at some selected diagonal elements $(Q^i)_{ii} = 1$. We will assume that disturbances are composed of impulses and that the reference signals are step functions, i.e.,

$$\boldsymbol{u}_d(t) = \bar{\boldsymbol{u}}_d \delta(t), \qquad \boldsymbol{r}(t) = \begin{cases} 0 & t < 0 \\ \bar{\boldsymbol{r}} & t \ge 0 \end{cases} \quad (38)$$

To evaluate the preceding integrals, it is convenient to perform modal decomposition first and let

$$A_t = X \Lambda Y, \qquad XY = I \quad (39)$$

where

$$\lambda_i = \text{eig}(A_t), \qquad \Lambda = \text{diag}(\lambda_i) \quad (40)$$

It can be shown that[16]

$$I_i^{\infty} = I_{rr}^{\infty} + 2I_{rd}^{\infty} + I_{dd}^{\infty}, \qquad I_i^{t_s} = I_{rr}^{t_s} + 2I_{rd}^{t_s} + I_{dd}^{t_s} \quad (41)$$

where

$$I_{rr}^{\infty} = \bar{\boldsymbol{r}}^T \bar{B}_r^T S_{rr}^{\infty} \bar{B}_r \bar{\boldsymbol{r}}, \qquad I_{rd}^{\infty} = \bar{\boldsymbol{r}}^T \bar{B}_r^T S_{rd}^{\infty} \bar{B}_d \bar{\boldsymbol{u}}_d$$
$$I_{dd}^{\infty} = \bar{\boldsymbol{u}}_d^T \bar{B}_d^T S_{dd}^{\infty} \bar{B}_d \bar{\boldsymbol{u}}_d \quad (42)$$

$$I_{rr}^{t_s} = \bar{\boldsymbol{r}}^T \bar{B}_r^T S_{rr}^{t_s} \bar{B}_r \bar{\boldsymbol{r}}, \qquad I_{rd}^{t_s} = \bar{\boldsymbol{r}}^T \bar{B}_r^T S_{rd}^{t_s} \bar{B}_d \bar{\boldsymbol{u}}_d$$
$$I_{dd}^{t_s} = \bar{\boldsymbol{u}}_d^T \bar{B}_d^T S_{dd}^{t_s} \bar{B}_d \bar{\boldsymbol{u}}_d \quad (43)$$

and if we let λ^* be the complex conjugate of λ, we obtain

$$(S_{rr}^{\infty})_{ij} = -\frac{(\bar{Q})_{ij}}{\lambda_i^*(\lambda_i^* + \lambda_j)\lambda_j}, \qquad (S_{rd}^{\infty})_{ij} = -\frac{(\bar{Q})_{ij}}{\lambda_i^*(\lambda_i^* + \lambda_j)} \quad (44)$$

$$(S_{dd}^{\infty})_{ij} = -\frac{(\bar{Q})_{ij}}{(\lambda_i^* + \lambda_j)}$$

$$(S_{rr}^{t_s})_{ij} = (S_{rr}^{\infty})_{ij}\left[1 - e^{(\lambda_i^* + \lambda_j)t_s}\right]$$

$$(S_{rd}^{t_s})_{ij} = (S_{rd}^{\infty})_{ij}\left[1 - e^{(\lambda_i^* + \lambda_j)t_s}\right] \quad (45)$$

$$(S_{dd}^{t_s})_{ij} = (S_{dd}^{\infty})_{ij}\left[1 - e^{(\lambda_i^* + \lambda_j)t_s}\right]$$

$$\bar{Q} = \bar{C}^T Q \bar{C}, \qquad \bar{C} = C_t X, \qquad \bar{B}_r = Y B_{tr}, \qquad \bar{B}_d = Y B_{td} \quad (46)$$

The settling time due to an impulse signal that is represented in the form of Eq. (35) can now be calculated via Eqs. (41–46) and Eq. (29). If the settling time for the elements of the plant error vector is required, we let

$$C_t = [C_{\eta z} \quad 0], \qquad D_t = [0 \quad 0] \quad (47)$$

in Eq. (35). Similarly, to find the settling time for the plant output, we let

$$C_t = [C_{\eta y} \quad 0], \qquad D_t = [0 \quad 0] \quad (48)$$

in Eq. (35).

The preceding settling-time derivation is used to develop physically meaningful design metrics for the system vibration level and control effort. As a measure of the vibration level, we use the (pseudo) root mean square (rms) of the signal

$$\text{rms}(y_t^i) = \sqrt{\frac{1}{t_s}\int_0^{t_s} y_t^{i^2}(t)\,dt} = \sqrt{\frac{I^{t_s}}{t_s}} \quad (49)$$

The same measure is used for the control effort. We define

$$\text{con}(u_c^i) = \sqrt{\frac{1}{t_s}\int_0^{t_s} u_c^{i^2}(t)\,dt} = \sqrt{\frac{I_{u,i}^{t_s}}{t_s}} \quad (50)$$

The integrals

$$I_{u,i}^{t_s} = \int_0^{t_s} u_c^{i^2}(t)\,dt, \qquad I_{u,i}^{\infty} = \int_0^{\infty} u_c^{i^2}(t)\,dt \quad (51)$$

can be evaluated as before by letting

$$C_t = [-D_c C_{\eta y} \quad C_c], \qquad D_t = [0 \quad D_c] \quad (52)$$

in Eq. (35). The closed-form evaluation of the settling time developed in this section is used in the optimization problem formulation of the next section.

V. Formulation of the Optimization Problem

Design Metrics

The design metrics development was presented in the preceding section. Depending on the system and on the design requirements, physically meaningful design metrics for the vibration level and the control effort, as well as settling times for selected outputs, can be evaluated.

Design Variables

The system design variables can be written as

$$\boldsymbol{d} = [\boldsymbol{d}_s \quad \boldsymbol{d}_c] \tag{53}$$

where $\boldsymbol{d}_s$ and $\boldsymbol{d}_c$, respectively, represent the structural and the controller design variable vectors. The structural variables consist of parameters that determine the geometrical properties of the structure and, by implication, its mass, viscous damping, and stiffness distributions.

The controller design variables d_c are the parameters that affect the properties of the controller. If the controller realization given by Eqs. (21) and (22) has the transfer function matrix $G_c(s)$, the constituent elements of this matrix can be expressed as follows:

$$[G_c(s)]_{ij} = \left[\frac{g \prod_{k=1}^{n1} (s + a_k) \prod_{k=1}^{n2} (s^2 + b_k s + c_k)}{\prod_{k=1}^{n3} (s + d_k) \prod_{k=1}^{n4} (s^2 + e_k s + f_k)} \right]_{ij} \tag{54}$$

The parameters $g, a_k, b_k, c_k, d_k, e_k$, and f_k collectively constitute the set of control design variables, namely $\boldsymbol{d}_c$. Let us consider the variables that characterize the ijth element of the transfer function matrix $G_c(s)$. We define

$$\begin{aligned} a_{ij} &= \{a_1, \ldots, a_{n1}\}_{ij}, & b_{ij} &= \{b_1, \ldots, b_{n2}\}_{ij} \\ c_{ij} &= \{c_1, \ldots, c_{n2}\}_{ij}, & d_{ij} &= \{d_1, \ldots, d_{n3}\}_{ij} \\ e_{ij} &= \{e_1, \ldots, e_{n4}\}_{ij}, & f_{ij} &= \{f_1, \ldots, f_{n4}\}_{ij} \end{aligned} \tag{55}$$

Then these vectors can be combined as

$$d_c^{ij} = \{a, b, c, d, e, f, g\}_{ij} \tag{56}$$

which provides the definition of the control design variable vector

$$\boldsymbol{d}_c = \{d_c^{11}, \ldots, d_c^{1p}, \ldots, d_c^{k1}, \ldots, d_c^{kp}\} \tag{57}$$

where p and k are, respectively, the number of controller inputs and outputs.

Objective Function: Aggregate Preference Function

As a departure from many conventional methods, designers do not need to engage in a tedious trial-and-error effort to construct the objective function using numerical weights. As was discussed in Sec. II, physical programming evaluates the objective function according to the designers' preferences. In light of the preceding comment, in this paper we call the objective function the aggregate preference function. Next, we discuss the structure of the structural design variables used in forming the aggregate preference function.

VI. Design-Variable-Linking Using a New Convex Spline

A key goal of this paper is to explore the use of a different form of structural tailoring, which avoids some difficulties associated with the nonlinked design variables approach while decreasing the number of structural design variables significantly. We propose to link the structural design variables by using splines to characterize the shape of the appendages. Messac and Sivanandan[26] recently developed a new family of convexity-preserving splines of order n, entitled the CPn spline, that are well suited to address the present needs.

We present in the following a degenerate special case of the CPn spline that is adequate for our purpose. A generic spline segment $s(x)$ in a specified interval, $x_i \le x \le x_j$, can be expressed in terms of the local coordinate ξ

$$\xi = \frac{x - x_i}{x_j - x_i} \tag{58}$$

with the following prescribed boundary conditions:

$$s(x_i) = s_i, \qquad s(x_j) = s_j \tag{59}$$

$$\frac{\partial s}{\partial x}(x_i) = s_i', \qquad \frac{\partial s}{\partial x}(x_j) = s_j' \tag{60}$$

Explicitly, the CPn spline is expressed as

$$s(\xi) = T_i(\xi) s_i + T_j(\xi) s_j + \bar{T}_i(\xi) s_i' + \bar{T}_j(\xi) s_j' \tag{61}$$

where the functions $T_i(\xi)$, $T_j(\xi)$, $\bar{T}_i(\xi)$, and $\bar{T}_j(\xi)$ are given by

$$T_i^n = [1/(n-2)][\xi^n - (\xi - 1)^n - n(\xi - 1) - 1] \tag{62}$$

$$T_j^n = [1/(n-2)][-\xi^n + (\xi - 1)^n + n\xi - 1] \tag{63}$$

$$\bar{T}_i^n = \frac{\lambda}{n(n-2)}[\xi^n - (n-1)(\xi - 1)^n - n(\xi - 1) - 1] \tag{64}$$

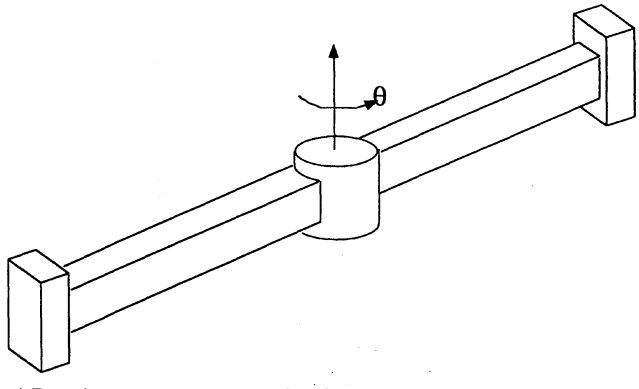
$$\bar{T}_j^n = \frac{\lambda}{n(n-2)}[(n-1)\xi^n - (\xi - 1)^n - n\xi + 1] \tag{65}$$

The parameter n is any even integer greater than or equal to 4. Different values of n result in different curvatures of the spline. In this paper, we will use the value of 4.

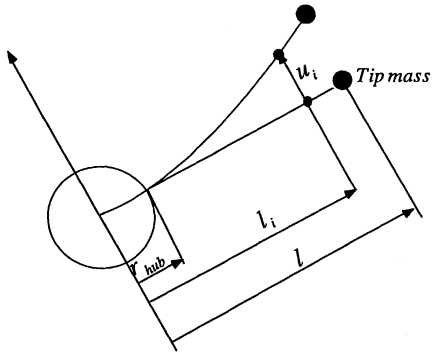
VII. Numerical Example and Results

Numerical Example

The effectiveness of the approach proposed in this paper is explored using a spacecraft example that was considered by Messac and Malek.[16] The spacecraft consists of a central rigid hub and two flexible appendages, each carrying a lumped mass at its tip (Fig. 2). The equations of motion are developed in Ref. 16 by using the finite element method in conjunction with the Newton–Euler approach. In the following, only the final equations are presented. In the optimization process, designers will have the ability to express their soft preference in the form of a preference table (Table 1) regarding tip-displacement settling time, tip displacement, hub-displacement



a) Rotating structure



b) Structure idealization

Fig. 2 Rotating structure example. Numerical properties: tip mass, 2 kg; hub inertia, 4000 kg m^2; appendage length, 20 m; hub radius, 1 m; beam depth (constant), 0.05 m; beam volumetric density, 1000 kg/m^3; beam modulus, 10^{11} N/m^2; and damping ratio, 0.5%.

settling time, and total mass. As previously stated, these quantities will form constituent components of the aggregate objective function. Hard preference (inequality constraint) regarding minimum appendage thickness is expressed as class 2H. These metrics are developed next.

The equations that govern the system dynamics in physical coordinates can be written as

$$M\ddot{q} + Kq = PF \tag{66}$$

where

$$q = \{\theta, u_1, \ldots, u_n\}^T \tag{67}$$

$$M = \begin{bmatrix} \frac{1}{2}M & S^T \\ S & M_{tt} \end{bmatrix}, \qquad K = \begin{bmatrix} 0 & 0 \\ 0 & K_{tt} \end{bmatrix} \tag{68}$$

$$M_{tt} = \text{diag}(m_1, \ldots, m_n), \qquad S = \{m_1 l_1, \ldots, m_n l_n\}^T \tag{69}$$

$$P = \begin{bmatrix} 0.5 & l_1 & \ldots & l_n \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix}, \qquad F = \{T_h, f_1, \ldots, f_n\}^T \tag{70}$$

where T_h is the torque applied to the hub (zero in the present case) and f_i is the generic force applied at the ith node of the appendage (nonzero only for $l = l_{i,\text{act}}$). The properties of the spacecraft are given in the Fig. 2 caption.

The actuators, which apply the control forces, are located antisymmetrically along the appendages 13.8 m away from the center of the hub ($l_{i,\text{act}}$). The hub displacement is fed back to a proper third-order controller, yielding a noncollocated control problem. The design metrics used for the problem are the following.

1) Stability: To ensure stability, we require that

$$\varphi = \max\{\text{Re}[\lambda(A_t)]\} < 0 \tag{71}$$

which is the maximum real part of the closed-loop eigenvalues ($Re[\,]$ denotes the vector containing the real parts of a complex vector). In the parlance of physical programming, the left-hand side of Eq. (71) is a class 1H design metric. Theoretically, the relationship (71) is not necessary because it will automatically be satisfied if other design metrics, e.g., the controller effort, are finite. However, if the closed-loop system is unstable, the design metrics that are defined only for a stable system are meaningless. During the search for the optimal solution, if the controller design variables yield an unstable system, the optimization algorithm may perform poorly. To accommodate the realities/peculiarities of numerical optimization, the zero in inequality (71) is replaced by a small negative number, e.g., -0.0001, in the computational implementation. The subject of how to avoid regions in optimization space that are singular is beyond the scope of this paper. To allow the code to work, even in the case of unstable systems, the user may proceed as follows. If $\varphi < 0$, do nothing. If $\varphi > 0$, do not evaluate the metrics that become undefined and replace their evaluations by the expression $c + d\varphi$, where c and d are appropriate large numbers. More sophisticated methods exist to address this problem. But fortunately, once the optimization search is in a stable region, it tends to remain in that stable region.

2) Settling time for the tip displacement (t_{tip}): The output matrices in Eq. (48) are arranged to yield the tip displacement, and Eq. (29) is used to calculate the settling time. Because this settling time should be minimized, it is a class 1S metric. A $\gamma = 10\%$ (magnitude) settling time is used, leading to a $p = 1\%$ area settling time [Eq. (29)].

3) Tip displacement (u_{tip}): The rms of the tip displacement is also penalized. Once the settling time is calculated, Eq. (49) yields the rms of the tip displacement. Because the amplitude of the vibrations should be minimized, this metric is class 1S.

4) Settling time for the hub displacement (t_{hub}): This is a class 1S design metric that is treated in a fashion similar to the tip displacement settling time. Again, $p = 1\%$ is used in Eq. (29) for the hub settling time.

5) Total mass (m): As mentioned in the Introduction, total mass is often included in the optimization problem as a hard constraint due to the difficulties associated with assigning numerical weights in an aggregate preference function. To expose the ability of physical programming to deal with this issue, the total structural mass is treated as a soft objective (class 1S design metric) and becomes a component of the physical programming aggregate preference function.

6) Minimum appendage thickness (w_{min}): The minimum thickness that the appendage can have is limited to 0.01 m, by making it a class 2H design metric.

The preferences for the soft design metrics are given in Table 1. These preferences apply to an initially quiescent structure that is commanded a step rotational maneuver of 1 rad.

The controller design variables are the coefficients of the numerator and denominator polynomials that describe the third-order, single-input (hub response), single-output (control effort) controller transfer function

$$G(s) = \frac{g(s+a)(s^2+bs+c)}{(s+d)(s^2+es+f)} \tag{72}$$

Hence, for this problem we have seven controller variables that form the controller design variable vector in Eq. (57) as follows:

$$d_c = \{a, b, c, d, e, f, g\} \tag{73}$$

The common approach to determine the optimum structural design variables for structural tailoring is to model the beam using the finite element method and to set the dimensions of the elements as the structural design variables.[4,8,15,16] This approach, hereby referred to as the nonlinked design parameter approach (NLPD), has two major disadvantages. 1) The profile of the beam can be adequately tailored only if a sufficiently large number of elements (and design parameters) is used. This large number of elements unduly increases the size of the optimization problem, which—in addition to the resulting computational difficulties—tends to yield local optima. This difficulty can be alleviated by starting with a smaller number of structural design parameters and performing several successive optimizations while increasing the number of parameters, which is a time-consuming process. 2) Making the dimension (width) of each finite element a design variable may exacerbate modeling errors. In addition, using as many design parameters as there are modeling finite elements negatively impacts the optimization process in a significant way. The results for the spacecraft example that are obtained via this approach are discussed later (case I). The appendages are modeled using 15 elements, the widths of which are the structural design variables. Together with the 7 controller variables, there is a total of 22 design variables. We note that, when design-variable-linking is invoked, the number of structural variables is reduced. Physical programming was used in all optimization cases.

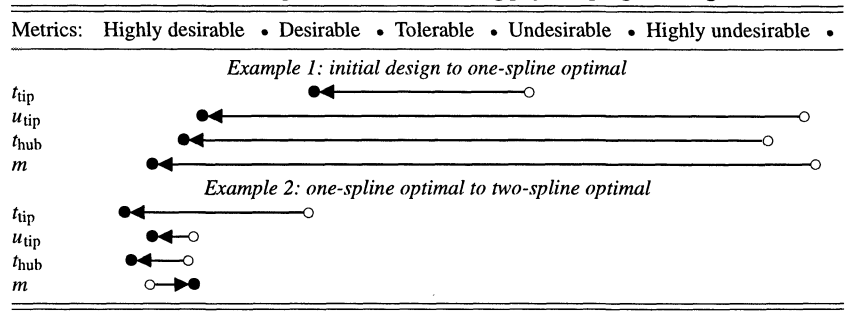
Results

Results are provided for three cases. In case I, the nonlinked design variable approach is discussed. In case II, one spline per appendage is used. Two splines per appendage are used for case III. All cases represent CSID. The preferences pertaining to each design metric are represented in Table 1. In Table 2, the overall improvements that resulted from optimization are depicted in the form of arrows that denote the improvement or worsening of each metric. As can be seen in Table 3, the goodness of the design metrics between cases I and II is not significantly different. The differences between case I and case II are discussed next. Because of these similarities, we chose to depict in Table 2 the behavior of the metrics for two situations: 1) from initial to case I and 2) from case I to case II. These arrows primarily show the impact of using two splines vs one spline. We discuss each case next.

Case I: Nonlinked Design Variables

The results for this NLDP approach are presented in Fig. 3 (left-hand column) and in Table 3. This case involves a total of 22 design variables. We observe from Fig. 3 (left-hand column) that the resulting optimal profile is not at all smooth. Clearly, such an appendage profile is not practical. As expected, however, the response of the

Table 2 CSID optimization results using physical programming[a]



[a]○: initial value; ●: final value.

Table 3 Initial and final values of the design and system parameters

Description	Parameters	Initial values	Nonlinked desirable parameter case	One-spline case	Two-spline case
Controller desirable variable	a	0.4592	0.5161	0.4553	0.8022
	b	0.0001	3.9*e*–6	3.9*e*–6	3.9*e*–6
	c	0.0761	0.1475	0.0647	0.5442
	d	1.6844	2.7848	2.3438	3.8669
	e	0.0290	0.0312	0.0350	0.0651
	f	5.8373	6.2743	4.1062	7.4969
	g	3.2780	4.6846	3.2681	1.8613
Structural desirable variable	s_0	0.05	——	3.0637*e*–2	3.5918*e*–2
	s_1	——	——	——	0.6716*e*–2
	s_2	0.05	——	3.1274*e*–2	3.1305*e*–2
	s_0'	0	——	0.1122*e*–2	0.1478*e*–2
	s_1'	——	——	——	– 0.0142*e*–2
	s_2'	0	——	1.0071*e*–2	0.8775*e*–2
Design metrics	t_{tip}	214.23	167.93	160.83	52.80
	u_{tip}	13.26*e*–5	1.92*e*–5	2.29*e*–5	2.21*e*–5
	t_{hub}	150.09	105.66	93.63	84.50
	m	50	43.06	35.11	39.53
	w_{min}	0.050	0.011	0.013	0.010
Other	Total inertia	2.12*e*4	1.48*e*4	1.41*e*4	1.52*e*4
	Aggregate cost	5.7*e*4	1.21	0.33	0.00005

system is improved as a result of the optimization. Such a design is symptomatic of the possible existence of local optima. The discontinuous lines represent the initial design, and the solid lines represent the optimal design. As is discussed next, linking the design variables results in a better behaved design space.

Case II: One Spline Used

To explore the use of splines in CSID, we consider the case for the spacecraft example where one spline per appendage is used to represent the appendage profile. This case involves a total of 11 design variables. The results for this case are presented in the middle column of Fig. 3. As expected, the use of the spline leads to a low total of four structural design variables, which are the values and the slopes of the spline segment at the right and left ends of the appendage. Explicitly, we have

$$\boldsymbol{d}_s = \{s_0,\ s_2,\ s_0',\ s_2'\} \tag{74}$$

where $x_0 = 0$ m (left end of the appendage) and $x_2 = 20$ m (right end of the appendage). Details are provided in Eq. (61).

The optimization starts with a uniform appendage profile whose initial width is 0.05 m. The profiles as well as the responses of the system for the initial case (uniform profile) and the one-spline case (case II) are shown in the middle column of Fig. 3. The values of the controller parameters and of the corresponding design metrics are given in Table 3. From Fig. 3, we observe that the structural responses expectedly improve from optimization. Interestingly, the improvement is almost identical to that obtained using the NLDP approach (compare Fig. 3, columns 1 and 2). However, the resulting profile is more acceptable from a practical point of view, using the spline method.

Comparing the preferences in Table 1 with the results in Table 3, we observe that the performance of the initial system is quite poor (see also Table 2). The settling time for the tip vibration is in the undesirable region, and the tip displacement and the hub settling time are in the highly undesirable region. The total mass of the system is 50 kg, the limit of the unacceptable region. The profile at the beginning is uniform, hence initially $s_i = 0.05$ and 0 for $i = 0$ and 2 for case II. The shape of the profile obtained as the solution of the first case is used as an initial guess for the two-spline case (case III). As a result of optimization, the tip settling time is improved and moved from the undesirable region to the desirable region. All of the other design metrics, namely the hub settling time, the tip displacement, and the total mass, are improved significantly and are moved to the highly desirable region. The new shape of the appendage, the tip, and the hub responses are depicted in the middle column of Fig. 3.

Case III: Two Splines Used

In case III, we use two splines to further improve the system. The results for this case are presented in the third column of Fig. 3. This case involves a total of 13 design variables. The height and the slope of the beam profile at the point where the actuator is located are added to the structural design variable vector

$$\boldsymbol{d}_s = \{s_0,\ s_1,\ s_2,\ s_0',\ s_1',\ s_2'\} \tag{75}$$

where $x_0 = 0$ m, $x_1 = 12.8$ m (actuator location), and $x_2 = 20$ m.

Although the total mass is increased with respect to the one-spline case (case II), it remained in the highly desirable region. It is interesting to note what occurred. From the expression of preference (Table 1), the designers say that any value of the mass that is less than 42 is highly desirable, which is equivalent to a small value of the effective scalar-optimization-weight in this neighborhood of the design space. This situation is exploited to further improve the overall design by increasing the mass, even though we wish it to be minimized.

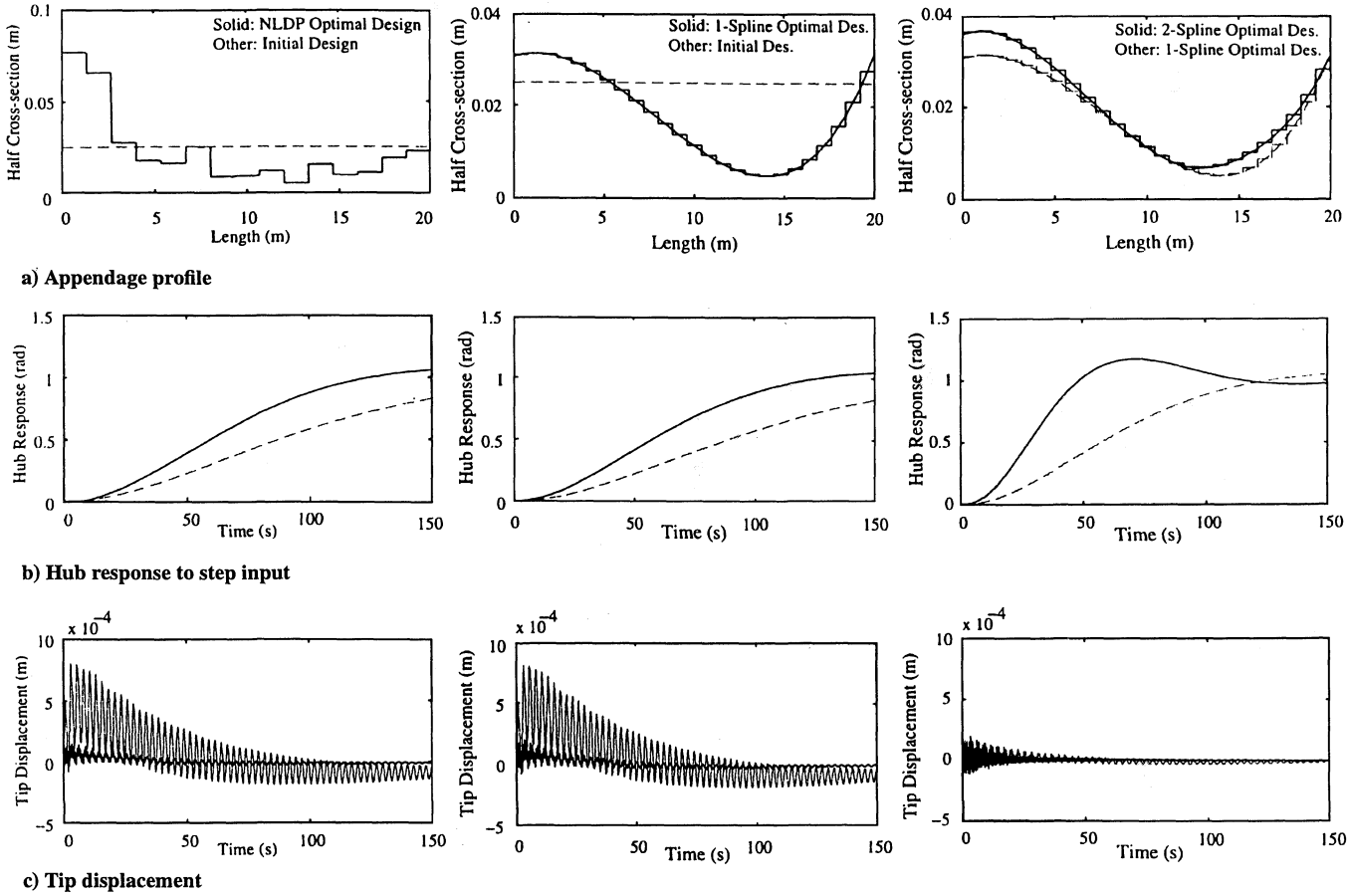


Fig. 3 Left, nonlinked design parameter approach vs initial design; middle, one-spline case vs initial design; right, two-spline case vs one-spline case.

Once the mass would move into another region (for example, the desirable region), the effective scalar-optimization-weight would automatically increase. This ability of physical programming to assess the design locally in design space is impossible to duplicate in the typical constant-weight formulation, which forms the basis of most optimization frameworks. In addition, the hub and tip settling times are further improved without increasing the tip displacement. Improved outer profile and the corresponding tip and hub responses are shown in the right-hand column of Fig. 3.

General Comments

After optimization, if the designers wish to explore new possibilities, they simply need to change the region limits defined in the preference table. We recall that the region limits are physically meaningful to the designers and that this flexibility is not available in the classical weighted-sum approach. As is discussed in detail in previous physical programming papers, changing the numbers in the preference table to explore the design space is fundamentally more deterministic than iterating on numerical weights in the aggregate objective function. A change in any one of the weights may have unpredictable consequences, particularly when the aggregate preference function is nonlinearly dependent on the design parameters, which is common in engineering problems.

VIII. Conclusion

In this paper, the new optimization method, physical programming, is used for control-structure integrated design. The proposed method largely circumvents the difficulty of the weighted-sum approach and allows designers to reflect their preferences in terms of physically meaningful quantities. In addition, new design metrics for structural design are developed. These metrics are physically meaningful and are appropriate for physical programming application and optimization-aided design in general. Finally, a new approach for structural tailoring, which uses a new family of splines, is successfully applied to a structural design example.

Acknowledgments

The author hereby gratefully acknowledges the support of this work by the National Science Foundation, Dynamic Systems and Control Program Grant CMS-9622652, and expresses his special thanks to the anonymous reviewers for their good insights.

References

[1]Slater, G. L., and McLaren, M. D., "Disturbance Model for Control/Structure Optimization with Full State Feedback," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 13, 1996, pp. 523–533.

[2]Niewoehner, R. J., and Kaminer, I. I., "Integrated Aircraft-Controller Design Using Linear Matrix Inequalities," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 445–452.

[3]Park, J.-H., and Asada, H., "Integrated Structure/Control Design of a Two-Link Nonrigid Robot Arm for High Speed Positioning," *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, 1992.

[4]Asada, H., Park, J.-H., and Rai, S., "A Control-Configured Flexible Arm: Integrated Structure/Control Design," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 2356–2362.

[5]Rai, S., and Asada, H., "Integrated Structure/Control Design of High Speed Flexible Robots Based on Time Optimal Control," *Advances in Robotics, Mechatronics, and Haptic Interface*, DS-Vol. 49, American Society of Mechanical Engineers, 1993, pp. 199–209.

[6]Hale, A. L., Lisowski, R. J., and Dahl, W. E., "Optimal Simultaneous Structural and Control Design of Maneuvering Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 1, 1985, pp. 86–93.

[7]Messac, A., and Turner, J. D., "Dual Structural-Control Optimization of Large Space Structures," AIAA Paper 84-1042, May 1984.

[8]Belvin, W. K., and Park, K. C., "Structural Tailoring and Feedback Control Synthesis: An Interdisciplinary Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 424–429.

[9]Messac, A., Turner, J. D., and Soosaar, K., "An Integrated Control and Minimum-Mass Structural Optimization Algorithm for Large Space Structures," Jet Propulsion Lab. Workshop on Identification and Control of Flexible Space Structures, JPL 85-29, San Diego, CA, June 1984.

[10]Salama, M., Hamidi, M., and Demsetz, L., "Optimization of Controlled

Structures," *Proceedings of the JPL Workshop on Identification and Control of Flexible Space Structures*, San Diego, CA, 1984.

[11]Onoda, J., and Haftka, R. T., "An Approach to Structure/Control Simultaneous Optimization for Large Flexible Spacecraft," *AIAA Journal*, Vol. 25, No. 8, 1987, pp. 1133–1138.

[12]Lust, R. V., and Schmit, L. A., "Control Augmented Structural Synthesis," *AIAA Journal*, Vol. 26, No. 1, 1988, pp. 86–95.

[13]Thomas, H. L., Sepulveda, A. E., and Schmit, L. A., "Improved Approximations for Control Augmented Structural Synthesis," *AIAA Journal*, Vol. 30, No. 1, 1992, pp. 171–179.

[14]Gilbert, G. G., and Schmidt, D. K., "Integrated Structure/Control Law Design by Multilevel Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 1001–1007.

[15]Canfield, R. A., and Meirovich, L., "Integrated Structural Design and Vibration Suppression Using Independent Modal Space Control," *AIAA Journal*, Vol. 32, No. 10, 1994, pp. 2053–2060.

[16]Messac, A., and Malek, K., "Control-Structure Integrated Design," *AIAA Journal*, Vol. 30, No. 8, 1992, pp. 2124–2131.

[17]Khot, N. S., and Heise, S. A., "Consideration of Plant Uncertainties in the Optimum Structural-Control Design," *AIAA Journal*, Vol. 32, No. 3, 1994, pp. 610–615.

[18]Cheng, F. Y., and Li, D., "Multiobjective Optimization of Structures with and Without Control," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 392–397.

[19]Hale, A. L., and Lisowski, R. J., "Characteristic Elastic Systems of Time-Limited Optimal Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 5, 1985, pp. 628–636.

[20]Venkayya, V. B., and Tischler, V. V., "Frequency Control and Its Effect on the Dynamic Response of Flexible Structures," *AIAA Journal*, Vol. 23, No. 11, 1985, pp. 291–298.

[21]Messac, A., "From the Dubious Art of Constructing Objective Functions to the Application of Physical Programming," *Proceedings of the AIAA/USAF/NASA/ISSMO 6th Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, 1996, pp. 1857–1867.

[22]Messac, A., "Physical Programming: Effective Optimization for Computational Design," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 149–158.

[23]Messac, A., and Wilson, H. W., "Physical Programming for Computational Control," *AIAA Journal*, Vol. 36, No. 2, 1998, pp. 219–226.

[24]Messac, A., and Hatttis, P., "Physical Programming Design Optimization for High Speed Civil Transport," *Journal of Aircraft*, Vol. 33, No. 2, 1996, pp. 446–449.

[25]"PhysPro: A Physical Programming Software Package for Optimal Design," Optimal Systems, Lexington, MA, Jan. 1997.

[26]Messac, A., and Sivanandan, A., "A New Family of Convex Splines for Data Interpolation," *Computer Aided Geometric Design Journal*, Vol. 15, 1997, pp. 39–59.

A. D. Belegundu
Associate Editor