The Cost on System Performance of Requirements on Differentiable Variables

George A. Hazelrigg[1]
Fellow ASME
Research Professor
Department of Mechanical Engineering,
George Mason University,
Fairfax, VA 22030
e-mail: ghazelri@gmu.edu

Philip Stolfi
Department of Mechanical Engineering,
George Mason University,
Fairfax, VA 22030
e-mail: pstolfi@masonlive.gmu.edu

System design is commonly thought of as a process of maximizing a design objective subject to constraints, among which are the system requirements. Given system-level requirements, a convenient management approach is to disaggregate the system into subsystems and to "flowdown" the system-level requirements to the subsystem or lower levels. We note, however, that requirements truly are constraints, and they typically impose a penalty on system performance. Furthermore, disaggregation of the system-level requirements into the flowdown requirements creates added sets of constraints, all of which have the potential to impose further penalties on overall system performance. This is a highly undesirable effect of an otherwise beneficial system design management process. This article derives conditions that may be imposed on the flowdown requirements to assure that they do not penalize overall system performance beyond the system-level requirement.

[DOI: 10.1115/1.4049336]

Keywords: design methodology, design optimization, design process, design theory and methodology, systems engineering

1 Introduction

Modern systems engineering often comprises a system design process based on requirements. The common perception is that the requirements are a set of directives that define what the customer wants and what the system has to do to meet the needs and wants of the customer. In fact, however, requirements do not define what the customer wants. They are a set of constraints that define what the customer will not accept, and they do not enable ranking of system alternatives. For example, *the system shall not weigh more than 100 pounds* simply says that the customer will not accept the system if it weighs more than 100 pounds. It does not say how the customer values weights that meet this requirement, for example, whether 70 pounds is better than 80 pounds. Thus, requirements would not serve to define a preference or an objective function for system optimization. In fact, requirements are constraints, and as shown by Hazelrigg and Saari [1], constraints have the potential to significantly reduce system performance as measured by the system design objective. Thus, one would normally prefer to minimize the imposition of constraints.

Current systems practice, however, involves a process of requirements flowdown, wherein a system-level requirement, such as a weight restriction, is flowed down to the subsystem level by assigning weight requirements at that level. The idea in setting the flowdown requirement is that, if the sum of the subsystem weight requirements is not greater than the system weight requirement, the system as a whole will meet its weight requirement. The flowdown requirements then enable the overall system design project to be broken down into a set of well-defined design tasks that empower design teams to preform the necessary subsystem and component level designs.

The problem with this practice is that the flowdown process introduces a large number of new requirements, each of which constitutes another constraint on the system, and each additional constraint has the potential to further degrade the system. Constraints may be classified as either inactive or active. An inactive constraint is one that is satisfied by an unconstrained optimal solution. In other words, the constraint would be satisfied by the solution if it were not stated at all. An active constraint is one that requires modification of the optimal solution to be satisfied. Thus, active constraints always impose penalties on the unconstrained (or less constrained) optimal solution. Clearly, as the requirements flowdown process introduces many new constraints (perhaps thousands in a complex system), it can be expected that many of these will impose penalties on the final system performance. But the flowdown constraints are self-imposed by the systems engineering process as a convenience to enable disaggregation of the system design process. This article addresses this problem for the case of requirements on differentiable variables such as weights, costs, volumes, power demands, and component reliabilities.

2 Background

While it is clear that large engineered systems have been designed and constructed for millennia—pyramids, the Roman aqueducts, and the Taj Mahal—the "science" of systems engineering appears to have had its beginnings in the early 1900s in the Bell Laboratories [2]. Hall was tasked with the establishment of a systems engineering course for the lab and eventually compiled extant systems knowledge into an early text on the subject [3]. In this book, he coins the term *objectives*, consisting of quantifiable statements describing what the system is intended to do. These statements would appear to be the first formal use of *requirements* as we know them today. They show further that Hall had grasped the concept of hierarchically structured objectives. Hall describes five phases of systems engineering: system studies, exploratory planning including selecting objectives and system design optimization, development planning, system development and test, and late-stage or operational engineering. Fagen [4] reviews applications of systems engineering during World War II, and in 1946, the RAND Corporation was created to assist what would later become the Air Force in the conduct of systems analysis. Also, in the 1940s systems engineering became an important aspect of missile and missile-defense systems [5].

Since its founding in 1990, the International Council On Systems Engineering (INCOSE) has been a major contributor to the theory and practice of systems engineering and has given considerable attention to the definition, statement, and flowdown of requirements. The INCOSE Systems Engineering Handbook [6] recognizes requirements as a key to the processes of system management, integration, verification, validation, operation, maintenance, and disposal. The Handbook notes, "successful projects depend on meeting the needs and requirements of the stakeholder/customer," and goes on to say, "a great deal of literature exists on how to write and manage requirements." The Handbook then enumerates how to "elicit and capture requirements, generate a concept of operations, define system capabilities and performance

[1] Corresponding author.

Contributed by the Design Theory and Methodology Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received May 18, 2020; final manuscript received November 18, 2020; published online January 29, 2021. Assoc. Editor: Christopher Hoyle.

objectives, and define non-functional requirements." However, the Handbook does not address the management of requirements flowdown. A later INCOSE document [7] recognizes a "complex relationship between requirements, the design choices made to address each requirement, and the system-level consequences of the sum of design choices across the full set of performance requirements···" Neither does this document address the penalties that requirements can impose on a system.

The National Aeronautics and Space Administration systems engineering handbook [8] recognizes the requirements flowdown process. This document recognizes four "system design processes," stakeholder expectation definition, technical requirements definition, logical decomposition, and design solution definition. The relevant process here is the logical decomposition process, which NASA describes as "used to improve understanding of the defined technical requirements and the relationships among the requirements ··· and to transform the defined set of technical requirements into a set of logical decomposition models and their associated set of derived technical requirements for lower levels of the system and for input to the design solution definition process." This document also does not address the potential penalties that requirements can impose on a system.

Collopy, in a number of unpublished presentations, has clearly recognized the penalties that requirements can impose on system performance. He has specifically addressed the problem of flowdown requirements, noting that they are constraints that impose potentially significant penalties. In addition, Collopy and coworkers [9] studied Department of Defense acquisition programs noting that current system procurement processes lead to an estimated loss on the order of $200 million per day. It is partly for this reason that we address the losses that flowdown requirements of continuous variables such as weight, cost, power consumption, thermal load, and reliability can impose on system performance and provide conditions that assure that these penalties are minimized.

3 Deterministic Formulation of the Problem

Let a system be described by a set of statements, $\mathbf{x}$. The elements of $\mathbf{x}$ may include continuous and integer values, verbal statements such as colors or textures, descriptions of a system configuration, manufacturing process descriptions, and even probabilistic statements or beliefs. We will consider the elements of $\mathbf{x}$ to be of two types, $\mathbf{x}^T = [\mathbf{x}_d, \mathbf{x}_n]$, where $\mathbf{x}_d$ are those components of $\mathbf{x}$ that are differentiable and $\mathbf{x}_n$ are the nondifferentiable components of $\mathbf{x}$, which we shall refer to as the *system configuration*. Let $P(\mathbf{x})$ be a real scalar function that denotes the system performance or objective function such that candidate system designs, $\mathbf{x}$, are evaluated and ranked by values of $P(\mathbf{x})$. Next, let system-level requirements on differentiable variables be denoted by $\mathbf{r}$, composed of elements r_j, $j = 1, 2, \ldots, n$, where n is the number of system-level requirements on differentiable variables. The r_j may derive from statements such as, "the system shall not weigh more than 100 pounds." We then decompose the system-level requirements into subsystem flowdown requirements $\mathbf{q}_j$, with elements q_{jk}, $k = 1, 2, \ldots, m$, where m is the number of relevant subsystems.

With this notation, the system design is subject to constraints that accommodate the requirements,

$$\begin{aligned} \boldsymbol{f}_j(\mathbf{q}_j) &\le \mathbf{r}_j \\ \boldsymbol{g}_j(\mathbf{x}) &\le \mathbf{q}_j \\ \mathbf{q}_j &\ge \mathbf{0} \end{aligned} \tag{1}$$

where the notation $\boldsymbol{f}_j(\mathbf{q}_j)$ refers to the vector of functions $f_j(\mathbf{q}_j)$ with each element of this vector associated with its corresponding element of $\mathbf{r}_j$, and $\boldsymbol{g}_j(\mathbf{x})$ refers to the vector of functions $g_j(\mathbf{x})$ corresponding to the vector of flowdown requirements for each subsystem, j.

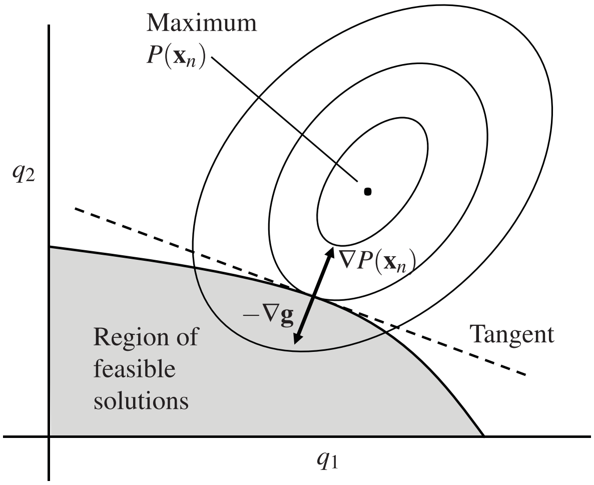

Fig. 1 Graphical representation of an optimization with an inequality constraint

We will assume that the constraint functions are convex so that the set of feasible solutions is a convex set. We shall also assume that the first partial derivatives of $P(\mathbf{x})$ with respect to $\mathbf{x}_n$ and $\boldsymbol{g}_j$ are defined. The solution to the maximization of $P(\mathbf{x})$ can be obtained via a Lagrangian formulation invoking the Karush–Kuhn–Tucker (KKT) conditions [10]. This optimization is shown in Fig. 1.

A requirement that is satisfied by an optimal design without imposing the requirement does not impose a penalty on the optimal solution, and it is unnecessary. This would be the case if the region of feasible solutions, that is, the solutions that satisfy the inequality constraint on $\mathbf{r}$, encompasses the maximum point. In this case, we say that the constraint is *inactive*. The curves circling the maximum point are lines of constant $P(\mathbf{x})$. For all requirements that are not satisfied by an optimal design as shown in the figure, the requirement imposes a penalty on the optimal solution, and to minimize that penalty, the solution is the point of tangency between boundary of the constraint region and a surface of constant $P(\mathbf{x})$. We refer to these constraints as *active*.

With these conditions noted, we can frame the system design problem as follows. First, choose a candidate system configuration, $\mathbf{x}_n$, and flowdown requirements, $\mathbf{q}_j$. Second, for this system configuration and flowdown requirements, values of $\mathbf{x}_d$ are determined by the following optimization as written in the Lagrangian form [10,11]:

$$\begin{aligned} \mathrm{Max}_{wrt\,\mathbf{x}_d} J(\mathbf{x}) = P(\mathbf{x}) &+ \boldsymbol{\lambda}_\mathbf{r}\{\mathbf{r} - \boldsymbol{f}_j(\mathbf{q}_j)\} \\ &+ \sum_j \boldsymbol{\lambda}_{\mathbf{q}j}\{\mathbf{q}_j - \boldsymbol{g}_j(\mathbf{x})\} \end{aligned} \tag{2}$$

The KKT conditions for the maximization of $J(\mathbf{x})$ with respect to $\mathbf{x}_d$ are as follows:

$$\begin{aligned} &\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}_d} = 0 \\ &\boldsymbol{f}_j(\mathbf{q}_j) \le \mathbf{r} \\ &\boldsymbol{g}_j(\mathbf{x}) \le \mathbf{q}_j \\ &\boldsymbol{\lambda}_\mathbf{r}\{\mathbf{r} - \boldsymbol{f}_j(\mathbf{q}_j)\} = \mathbf{0} \\ &\boldsymbol{\lambda}_{\mathbf{q}j}\{\mathbf{q}_j - \boldsymbol{g}_j(\mathbf{x})\} = \mathbf{0}, \quad j = 1, \ldots, m \\ &\boldsymbol{\lambda}_\mathbf{r} \ge \mathbf{0} \\ &\boldsymbol{\lambda}_{\mathbf{q}j} \ge \mathbf{0}, \quad j = 1, \ldots, m \end{aligned} \tag{3}$$

Solution to conditions (3) yields values for the multipliers $\boldsymbol{\lambda}_\mathbf{r}$ and $\boldsymbol{\lambda}_{\mathbf{q}j}$ and the optimum values for the design parameters $\mathbf{x}_d$ subject to the given values of $\mathbf{x}_n$ and $\mathbf{q}_j$. Note that, if for any requirement $\lambda_r = 0$, that requirement is inactive.

In this formulation, the $\boldsymbol{\lambda}_{\mathbf{q}j}$ represent the marginal penalties, namely, $-\partial P(\mathbf{x})/\partial \mathbf{q}_j$, of the flowdown requirements on the system performance. Thus, a condition that assures that the flowdown requirements do not impose additional constraints on the system that further penalize performance beyond the penalty imposed by the system-level requirements $\mathbf{r}$ is that the $\boldsymbol{\lambda}_{\mathbf{q}}$'s satisfy a *transversality* condition. That is, the $\boldsymbol{\lambda}_{\mathbf{q}}$'s must be transverse (perpendicular or normal) to the plane of the requirements defined by $\boldsymbol{f}_j(\mathbf{q}_j)$ at the point of tangency for the optimal solution as shown in Fig. 1. Note that this point defines the optimal values of $\mathbf{q}$. Furthermore, the requirement to satisfy the transversality condition makes clear that if the system-level requirement is inactive, it must be the case that all flowdown requirements are also inactive, that is, all $\lambda_{\mathbf{q}} = 0$. Contrariwise, if a particular $\boldsymbol{\lambda}_{\mathbf{r}}$ is nonzero positive, then that constraint is active and the corresponding constraints on all components of the associated $\mathbf{q}_j$ must also be active. This is because it is not possible to satisfy the transversality condition otherwise, and in this case, we know immediately that the equality conditions on $\mathbf{r}$ and $\mathbf{q}$ must apply.

In practice, this can be a very simple requirement to implement. For example, suppose the flowdown requirement allocates weights to various subsystems. Then, the flowdown requirement plane is defined by the following equation:

$$q_1 + q_2 + \cdots + q_m = r \tag{4}$$

For a hyperplane defined by an equation of the form $a_1\, y_1 + a_2\, y_2 + \cdots + a_m\, y_m = c$, the basis for vectors that are perpendicular to this plane is simply $[a_1, a_2, \ldots, a_m]$. Thus, for a flowdown requirement plane of the form (4), a vector normal to the plane is simply $[1,1, \ldots, 1]$, and the magnitude of this vector is $\sqrt{m}$. The transversality condition states that the projection of the $\boldsymbol{\lambda}_q$'s for each flowdown requirement onto the requirement plane must have magnitude 0. This means that the vector defined by the components λ_q must align with the vector $[1,1, \ldots, 1]$. In other words, for each requirement, q, $\lambda_{q1} = \lambda_{q2} = \cdots = \lambda_{qm}$ or, in the more general case where all $a_i \neq 1$, $\lambda_{q1}/a_1 = \lambda_{q2}/a_2 = \cdots = \lambda_{qm}/a_m$. Furthermore, if this condition is not met, this vector sum will have a finite projection onto the requirement plane, and that projection will show the direction in which one must adjust the q's to seek a more optimal allocation of the flowdown requirements. Stated verbally, for a given flowdown requirement, each design team should first obtain a "best" design and then estimate the improvement in performance achievable if the requirement is relaxed by a given small amount. These "sensitivities" are the respective λ's. If the λ's satisfy the aforementioned transversality condition, then the flowdown requirements are allocated optimally, and they will impose no performance penalty in addition to that imposed by the system-level requirement r. If the λ's do not satisfy the aforementioned conditions, then their vector sum will have a finite projection onto the requirement plane, and this projection will denote the relative changes to the q's one should make to seek a more optimal allocation of the flowdown requirements. Thus, in practice it is not necessary to find the projection of the vector sum onto the requirements plane as it is only necessary to assure that the magnitude of the projection is 0.

A key advantage of this approach to the optimization of flowdown requirements is that it enables the disaggregation of design tasks in the same manner that the current requirements flowdown process does. At each step in this process, the subsystem design teams will have access to flowdown requirements that enable them to provide candidate designs. Yet, as the design iterates to a final, optimal design, the flowdown requirements will converge to a set that imposes no penalties to the system performance beyond that imposed by the system-level requirements.

It is also worth to note that the $\boldsymbol{\lambda}_{\mathbf{r}}$ represent the cost per unit of the system-level requirements, namely, $-\partial P(\mathbf{x})/\partial \mathbf{r}$. These data could be useful in determining whether the system-level requirements are reasonably determined.

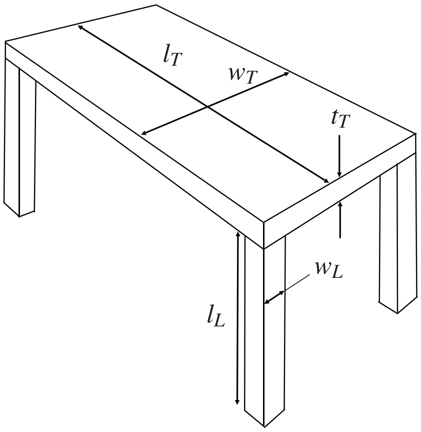

Fig. 2 The example table

4 Example Problem

We will now consider a simple example problem involving a single system-level requirement and two subsystem flowdown requirements. It is an illustrative problem only with parameters not intended to represent a real design. This example problem can be envisioned as the design of a table such as that shown in Fig. 2, where there is a weight requirement, r_W, on the assembled table that is flowed down to weight requirements on the table top and the legs taken as a group, namely, $f_W(q_W) = q_T + q_L$, and with an objective of minimum cost. The weight of the table is given as the sum of the volumes of the parts of the table times the densities, ρ, of these parts. For simplicity, we will take the outer dimensions of the parts to comprise rectangular cuboids, which will be lightened by material removal resulting in a final volume of η times the original volume of the cuboid, namely, $V = \eta lwt$, where l, w, and t are the length, width, and thickness of the parts, respectively, and $(1 - \eta)$ is the fraction of the material that is removed to lighten each part, $0 < \eta < 1$.[2] To prevent the trivial result that the table has zero surface area, we will take the values of l_T and w_T to be fixed and given. In addition, we will assume that the table is to be of a specified height, h, such that $h = (l_L + t_T)$ is also fixed and given. Thus, the weights of the table parts are given by

$$\begin{aligned} g_T &= \eta_T l_T w_T t_T \rho_T = W_T \\ g_L &= 4\eta_L l_L w_L^2 \rho_L = 4\eta_L (h - t_T) w_L^2 \rho_L = W_L \end{aligned} \tag{5}$$

where W_T and W_L are the weights of the top and legs, respectively, noting that the table has four legs and taking their unmachined cuboids to be of equal width and thickness.

Next, we develop a cost model. For this example, it is convenient to assume that the cost of the table is composed of a materials cost C, a cost of machining M, and a cost of assembly A. The material cost will be taken to be proportional to the weights of the unmachined parts, namely, $W_T^* = \rho_T l_T w_T t_T$ and $W_L^* = 4\rho_L (h - t_T) w_L^2$. Accordingly, the material cost is

$$\begin{aligned} C_T &= W_T^* P_T = P_T \rho_T l_T w_T t_T \\ C_L &= W_L^* P_L = 4 P_L \rho_L (h - t_T) w_L^2 \end{aligned} \tag{6}$$

where P_T and P_L are the prices per unit weight of the table top and table leg materials, respectively. Next, we shall use the following relationships for the cost of machining for the purpose of weight

[2] We use the concept of material removal here to emphasize that we have assured continuous differentiability of $\mathbf{g}(\mathbf{x}_n)$.

reduction,

$$M_T = 90(W_T - 20)^{-0.4}$$
$$M_L = 10\left(1 - \frac{W_L}{W_L^*}\right)^{1.5}\left(\frac{W_L}{W_L^*} - 0.05\right)^{-0.5} \tag{7}$$

These functional forms result in greater cost to achieve lighter designs. The final term relates to the assembly cost. Here, we assume that the cost of assembly is a weakly increasing function of weight.

$$A = A_0 + \delta(W_T + W_L) \tag{8}$$

It follows that the total cost of production is expressed as follows:[3]

$$C_T + C_L + M_T + M_L + A = -P(\mathbf{x}) \tag{9}$$

We now see that the differentiable design variables, $\mathbf{x}_d$, include the dimensions of the table top and legs, namely, l_T, w_T, t_T and l_L, w_L, and η_T and η_L. Of these, however, only t_T, w_L, $W_T = W_T^*\eta_T$, and $W_L = W_L^*\eta_L$ are free to be optimized. Furthermore, the relationships between t_T and η_T and between w_L and η_L would normally be constrained by relationships that determine the required strength and stiffness. Thus, to keep this example relatively simple, we shall also take t_T and w_L to be given, leaving only W_T and W_L to be optimized. These design variables are determined by maximizing the function:

$$\begin{aligned}\text{Max}_{wrt\ \mathbf{x}_d} J(\mathbf{x}) = P(\mathbf{x}) + \lambda_r\{r_W - q_T - q_L\}\\ + \lambda_{qT}\{q_T - W_T\} + \lambda_{qL}\{q_L - W_L\}\end{aligned} \tag{10}$$

To maximize this function, we must satisfy conditions (3). Taking the partial derivatives of $J(\mathbf{x})$ with respect to the remaining free variables of $\mathbf{x}_d$, namely, W_T and W_L.

$$\begin{aligned}\frac{\partial J}{\partial W_T} &= 36(W_T - 20)^{-1.4} - \delta - \lambda_{qT} = 0\\ \frac{\partial J}{\partial W_L} &= \frac{5(1 - W_L/W_L^*)^{1.5}}{W_L^*(W_L/W_L^* - 0.05)^{1.5}}\\ &\quad + \frac{15(1 - W_L/W_L^*)^{0.5}}{W_L^*(W_L/W_L^* - 0.05)^{0.5}} - \delta - \lambda_{qL} = 0\end{aligned} \tag{11}$$

Solving for the λ's,

$$\begin{aligned}\lambda_{qT} &= \frac{\partial P}{\partial W_T}\\ \lambda_{qL} &= \frac{\partial P}{\partial W_L}\end{aligned} \tag{12}$$

In this example, there are only two flowdown requirements defined by the equation $r_W = q_T + q_L$. Hence, the requirement hyperplane is a line with normal vectors defined by the direction $\{1,1\}$ corresponding to the q_T and q_L axes, and the optimality condition is expressed as follows:

$$\lambda_{qT} = \frac{\partial P}{\partial W_T} = \frac{\partial P}{\partial W_L} = \lambda_{qL} \tag{13}$$

As noted earlier, condition (4) would be different if the coefficients of the equation for r_W, 1 and 1, were different. It is also possible for the flowdown requirements to combine nonlinearly in which case the flowdown requirement surface is not a plane. This typically would be the case if the flowdown requirements are on component reliabilities. The more general case of condition (4) is presented in Appendix A.

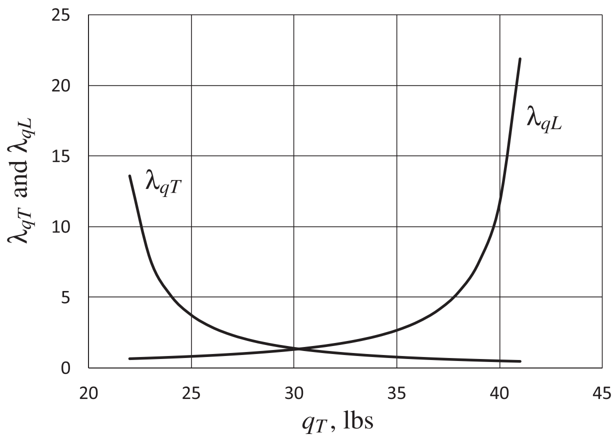

Fig. 3 Plot of λ_{qT} and λ_{qL} as a function of q_T

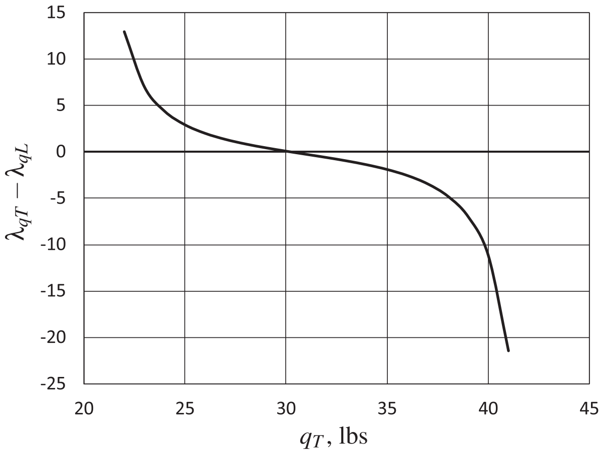

Fig. 4 Plot of $\lambda_{qT} - \lambda_{qL}$ as a function of q_T

To illuminate the example case further, we choose the following data:

Oak table top
Density = 39.33 lbs/cu-ft
Price = \$1.0679/lb
$W_T^* = 78.66$ lbs
$C_T = \$84.001$
316 stainless steel legs
Density = 496.32 lbs/cu-ft
Price = \$0.75/lb
$W_L^* = 142.46$ lbs
$C_L = \$106.845$
Assembly cost
$A_0 = \$15$
$\delta = \$0.05/\text{lb}$
Weight requirement
$r \leq 50$ lbs

With these data, convergence to a solution is obtained easily within ten iterations using a simple gradient search. Figure 3 is a plot of λ_{qT} and λ_{qL} as a function of q_T, with an optimal solution of $q_T = 30.2295$ lbs. As both λ_{qT} and λ_{qL} are positive, we are assured that the constraint on $\mathbf{r}$ is active, and the equality conditions of the constraints on $\mathbf{r}$ and $\mathbf{q}$ apply. Figure 4 shows the projection of the vector sum $\lambda_{qT} + \lambda_{qL}$ onto the requirement plane. The zero crossing depicts the optimal solution. Figure 5 plots the total cost of manufacture as a function of q_T.

[3] Note that we wish to minimize total cost, which is equivalent to maximizing performance, P($\mathbf{x}$), expressed as the negative of total cost.

5 Nondeterministic Formulation of the Problem

The formulation for the determination of optimal flowdown requirements given earlier can be adapted to more complex cases where there is uncertainty, in which case it may be desirable to specify the flowdown requirements with "margins" to provide added confidence that the system-level requirements can be met. In one case, the margins may be used at the design stage to assure that some exceedance of the flowdown requirements will not result in failure of the final design to meet the system-level requirement. Alternatively, the margins may be used to account for manufacturing variance, again to assure that the manufactured product will meet the system-level requirement. It is also possible to accept different interpretations of the flowdown requirements. On the one hand, project management could elect to hold all design teams responsible for meeting their individually assigned flowdown requirements. On the other hand, management could elect only to manage requirements at the system level, accepting exceedances of some flowdown requirements provided they are accommodated by underages in others. Each of these cases fit within the overall framework provided here, however with some modification of the logic by which the payoff function is determined. While it might be reasonable to expect that the deterministic formulation will converge to an optimal solution as uncertainties are reduced through the iterative design process, it may prove more expeditious to begin with the nondeterministic formulation.

Extension to the nondeterministic case relies on reformulation of the problem to maximize expected utility of the selection of the flowdown requirements [12]. For this case, we must consider more than the marginal penalties of the flowdown requirements themselves. We must acknowledge the potential that inadequate provision of margins can lead to cases where the final design fails to meet the system-level requirements and is, therefore, a failure that bears a cost to the project. Thus, the formulation given earlier must be augmented to provide an estimate of the benefit of a successful design (one that meets all system-level requirements) as a function of the flowdown requirements and an estimate of the cost of a design failure. Note that, while increasing the margin on the flowdown requirements may increase the probability of achieving a successful design, doing so penalizes the expected performance of the successful design. Thus, maximization of the expected utility must account for both the utility of successful designs and that of failed designs.

For the successful designs, the system expected utility is given by

$$\mathrm{E}\{u_s[J(\mathbf{x})]\} = \mathrm{E}\left\{u\left[-P(\mathbf{x}) + \lambda_{\mathbf{r}}\{\mathbf{r} - \boldsymbol{f}_j(\mathbf{q}_j)\} + \sum_j \lambda_{\mathbf{q}j}\{\mathbf{q}_j - \boldsymbol{g}_j(\mathbf{x})\}\right]\right\} \tag{14}$$

In this formulation, it is no longer the case that $\mathbf{q}_j = \boldsymbol{g}_j(\mathbf{x})$. Rather the inequality, $\mathbf{q}_j \le \boldsymbol{g}_j(\mathbf{x})$, will now apply. We must next consider the possibility of design failure, that is, where uncertainty in the outcome of a design choice leads to an exceedance of a requirement, either at the system level or at the flowdown level depending on the project management approach. We shall consider the case of design choices failing to meet the system-level requirement. For the case of the table example presented earlier, this means that the total table weight exceeds a system-level weight requirement. We shall assume that, when this happens, there is a cost imposed such that the design outcome has negative value. We denote the utility of this loss as u_f. Then, the expected utility of a choice of flowdown requirements, in the example case, q_T and q_L, is given by

$$\mathrm{E}\{u[J(\mathbf{x})]\} = p_s \mathrm{E}\{u_s[J(\mathbf{x})]\} + (1 - p_s)u_f \tag{15}$$

where p_s is the probability that the design meets the system-level requirement.

Interestingly, the introduction of uncertainty leads to a requirement for additional data, including both data on the nature of the uncertainty itself and on the context within which the uncertainty lies. We will consider a very simple case of uncertainty here, where the choice of the requirements q_T and q_L leads to uncertain component weights W_T and W_L, and where it is required that $W_T + W_L \le r$. This requirement leads to significant added complexity in the solution of (5). For this reason, we resort to Monte Carlo simulation for the evaluation of choices of q_T and q_L. We also assume that the decision maker is risk neutral, that is, the utility of money equals money. The additional data we need for this case are the following:

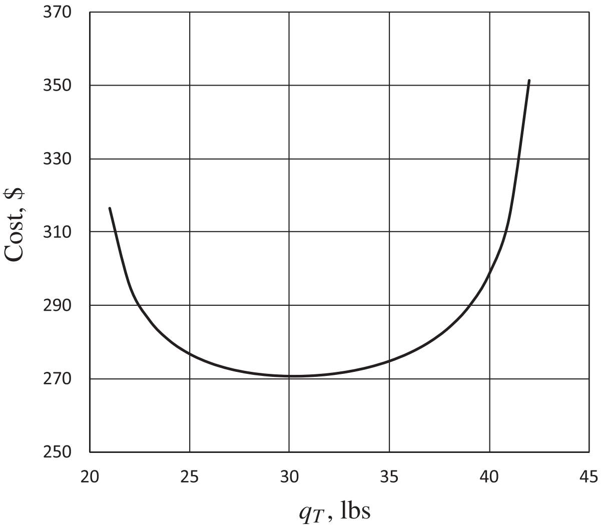

Fig. 5 Total cost of manufacture as a function of q_T

Uncertainty in W_T
    Normal distribution
    Mean = q_T lbs
    Standard deviation = 0.5 lbs
Uncertainty in W_L
    Normal distribution
    Mean = q_L lbs
    Standard deviation = 0.5 lbs
Economic data
    Failure cost = $2000
    Sale price = $325
    Demand at the sale price = 100 units

Note that, if the decision maker were not risk neutral, for example, if the decision maker's risk preferences were expressed as the utility of money equals the log of money, then in addition to the aforementioned data we would also require data on the financial status of the decision maker.

A solution to the aforementioned case is shown in Fig. 6. Clearly, there would be a considerable penalty to specifying values of q_T and q_L equal to the deterministic solution. The labels on the contours shown in this figure denote the expected utility of the choice of q_T and q_L, which in this case equates to the expected profit. The "+" sign shows the approximate location of the maximum point with an expected utility of 5132.

Although we have shown through this example that the problem formulation given is capable of dealing with cases that involve uncertainties, much work remains to fully exploit this capability.

6 Optimization of System-Level Requirements

Recognize that the $\boldsymbol{\lambda}_{\mathbf{r}}$ represents the penalty, $\partial P(\mathbf{x})/\partial \mathbf{r}$, that the system-level requirements impose on the system performance. Thus, these values can prove useful in evaluating the desirability of these requirements. For example, if the penalty of the requirement, *the system shall not weigh more than 100 pounds*, seems excessive, it could encourage the customer to relax the requirement

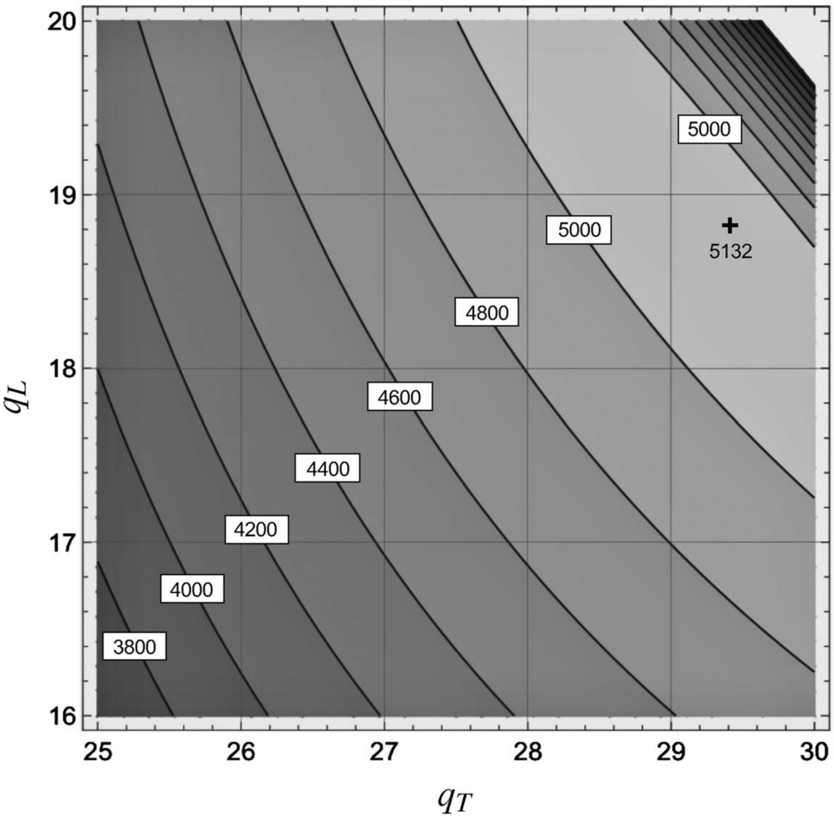

Fig. 6 Expected utility of choices of q_T and q_L

somewhat. These $\boldsymbol{\lambda}_{\mathbf{r}}$'s might also be used to understand tradeoffs among system-level requirements. For example, consider the case of an airplane designed for long-distance routes. System-level requirements might include cruise speed and range, with more of each being desirable. However, point-to-point trip times may be higher because range limitations require refueling stops on long flights, such that the overall trip time could actually be reduced by allowing a lower, more fuel-efficient, cruise speed.

Furthermore, it could be useful to determine the system-level requirements for which the $\boldsymbol{\lambda}_{\mathbf{r}} = \mathbf{0}$ as this condition allows the system performance to be maximized in the absence of the requirements. The process for finding the values of the requirements that satisfy this condition is the same as that for finding the values of the flowdown requirements that do not penalize the system performance.

7 Conclusions

System design by requirements and requirements flowdown is a well-established and presumably well-understood process. Unfortunately, it imposes added constraints on system design that have the potential to translate into serious performance penalties. To alleviate these penalties, we have derived a condition that, when imposed on the flowdown requirements, assures that they impose no additional penalty on system performance. The mathematics of the Lagrangian formulation together with the KKT conditions used in this approach leads to a convenient and powerful method that enables consideration of nonlinear cases and has the potential to extend to the case of uncertainty where we seek to optimize the system with respect to the expected utility of a system performance measure. The method also extends easily to multiple levels of flowdown requirements. A significant advantage of this method of setting flowdown requirements is that, at all steps during the design process, system-level and flowdown requirements are available to the design teams, allowing the design process to remain essentially unchanged while reducing the performance penalties adherent to the current requirements flowdown systems engineering approach.

By implementing this approach to the selection of requirements at all levels of system design, it may be possible to significantly reduce penalties associated with requirements on continuous variables, while requirements remain available to the design teams so that their work can proceed as usual. Furthermore, it seems reasonable that the concepts employed here can be extended to cases where the requirements are on variables that are not continuously differentiable. Noting that constraints imposed by requirements might be made inactive through the proper choice of a system performance measure, the approach provided here could pave the way to enable the conversion of system design by requirements into a case of design by preference while leaving the actual design process and its management largely unchanged.

Acknowledgment

This work has been supported by the National Science Foundation under award CMMI-1923164.

Conflict of Interest

There are no conflicts of interest.

Appendix A: The Transversality Condition

The transversality condition may be stated, in order that the flowdown requirements impose no performance penalty on the system, the vector $\boldsymbol{\lambda}_q = [\lambda_{q1}, \lambda_{q2}, \ldots, \lambda_{qm}]$ must be normal to the requirement hyperplane defined by $r = f(\mathbf{q})$ at the point $\mathbf{q}$ for the optimum design point $\mathbf{x}$. In the case that $f(\mathbf{q})$ takes the form

$$f(\mathbf{q}) = a_1 q_1 + a_2 q_2 + \cdots + a_m q_m \tag{A1}$$

a vector normal to this hyperplane is simply $\mathbf{a} = [a_1, a_2, \ldots, a_m]$. Thus, if $\boldsymbol{\lambda}_q = \alpha \mathbf{a}$, where α is a positive scalar, the transversality condition is met.

The transversality condition is a bit more complex in the case that $r = f(\mathbf{q})$ defines a nonplanar surface. Requiring that $\boldsymbol{\lambda}$ be normal to the requirement surface at point $f(\mathbf{q})$ is equivalent to requiring that $\boldsymbol{\lambda}_q$ be normal to the hyperplane that is tangent to $f(\mathbf{q})$ at point $\mathbf{q}$. But, from (A1), we can see that the tangent hyperplane is given by

$$f(\mathbf{q}) = a_1 q_1 + a_2 q_2 + \cdots + a_m q_m = r \tag{A2}$$

where the coefficients are given by

$$a_i = \left.\frac{\partial r}{\partial q_1}\right|_{\mathbf{q}} \tag{A3}$$

Another way of stating the transversality condition is that, in order that the flowdown requirements impose no performance penalty on the system, the projection of the vector $\boldsymbol{\lambda}_q$ onto the tangent hyperplane at point $\mathbf{q}$ must have magnitude 0. The projection of $\boldsymbol{\lambda}_q$ onto the requirement hyperplane is simply $\boldsymbol{\lambda}_q$ minus the vector normal to the tangent hyperplane, $\mathbf{n}$, from $\boldsymbol{\lambda}_q$ to the hyperplane. Note that the direction of $\mathbf{n}$ is the same as that of $\mathbf{a}$ and its magnitude is obtained from the dot product of $\boldsymbol{\lambda}_q$ and $\mathbf{a}$:

$$|n| = \frac{\boldsymbol{\lambda}_q \cdot \mathbf{a}}{|a|} \tag{A4}$$

and

$$\mathbf{n} = \frac{|n|}{|a|}\mathbf{a} \tag{A5}$$

References

[1] Hazelrigg, G. A., and Saari, D. G., 2020, “Toward a Theory of Systems Engineering,” Proceedings of the ASME 2020 IDETC/CIE Conference, Paper No. IDETC2020-22004.

[2] Buede, D. M., 2009, *The Engineering Design of Systems*, John Wiley & Sons Inc., Hoboken, NJ.

[3] Hall, A. D., 1962, *A Methodology for Systems Engineering*, Van Nostrand, Princeton, NJ.

[4] Fagen, M. D., 1978, *A History of Engineering and Science in the Bell System: National Service in War and Peace (1925–1975)*, Bell Telephone Laboratories Inc., Murray Hill, NJ.

[5] Goode, H. H., and Machol, R. E., 2001, *System Engineering—An Introduction to the Design of Large-Scale Systems*, McGraw-Hill, New York.

[6] INCOSE, 2006. *Systems Engineering Handbook, Version 3*, incose-tp-2003-002-03 ed., International Council on Systems Engineering, Hoboken, NJ.

[7] Cilli, M. V., and Parnell, G. S., 2014, “Systems Engineering Tradeoff Study Process Framework,” INCOSE, San Diego, CA.

[8] NASA, 2020, NASA Systems Engineering Processes and Requirements. NPR 7123.1C.

[9] Maddox, I. D., Collopy, P. D., and Farrington, P. A., 2013, “Value-Based Assessment of DoD Acquisition Programs,” Procedia Comput. Sci., **v16**, 1161–1169.

[10] Cooper, L., and Steinberg, D., 1970, *Introduction to Methods of Optimization*, W. B. Saunders Company, Philadelphia, PA.

[11] Aoki, M., 1971, *Introduction to Optimization Techniques, Fundamentals and Applications of Nonlinear Programming*, The Macmillan Company, New York.

[12] Hazelrigg, G. A., 2012, *Fundamentals of Decision Making for Engineering Design and Systems Engineering*, Self published by G. A. Hazelrigg, Vienna, VA.