This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Convergence Analysis and Iteration Estimation for a Coupled Design Process With Overlap in Redesign

Zheng Wang and Christopher L. Magee

***Abstract*—In Smith and Eppinger's pioneering research on coupled design processes, the dynamics of design iterations is modeled by a linear model, which implies that there is no overlap between redesign work and allows redesign workload to be more than the original design workload. Actually, this overlap often exists in practice and, therefore, reduces the total redesign workload. In this paper, we propose a nonlinear model for a coupled design process with overlap between redesign work and ensure that redesign workload is less than the original design workload. Based on the model, the sufficiency condition for the convergence of such a design process is proposed. A heuristic rule is also given to reduce the possibility of considering a design process that converges as one that does not converge. Furthermore, we develop a sufficiency condition for estimating the number of design iterations before converging as such an estimate is important for planning product development projects. Another heuristic rule is also introduced to increase the accuracy of estimating design iterations. Finally, numerical experiments are conducted to examine the correctness and performance of the proposed sufficiency conditions and heuristic rules.**

***Index Terms*—Coupled design process, design iteration, design structure matrix (DSM), Lyapunov stability.**

I. Introduction

A. Motivation

COUPLING or interdependence usually exists in the design of products. For example, in an aircraft design process [1], the structure design, the aerodynamics design, and the propulsion design are coupled with each other (see Fig. 1). The aerodynamics design receives two design parameters from the structure design, i.e., total weight w_{T} and effective wing arc change due to twist Θ, and receives one design parameter from the propulsion design, i.e., engine scale factor (ESF); meanwhile, it outputs the lift L to the structure design and the drag D to the propulsion design. Another example is the coupling between the mechanical system design and the control system design of dc motors [2]. If a group of design tasks are coupled with each other, then a design task is affected by other design tasks from which it receives data, and also affects other design tasks to which it provides data, which results in design iterations. Two important issues that product designers are interested in are 1) will the design iteration converge? and 2) how many times will the design process iterate before it converges?

To analyze the convergence of coupled design processes, a lot of work has been done based on a linear model proposed in the pioneering work of Smith and Eppinger [3]. Smith and Eppinger [3] identified the controlling features of coupled design processes by analyzing the eigenvalues and eigenvectors of the work transformation matrix. In this linear model, the workload of a design task in the current design iteration is the sum of the workloads caused by all the tasks it depends on. That might lead to a phenomenon that the redesign workload of a task in a later design iteration becomes more than the workload of its original design. For instance, suppose that design task 1 depends on design tasks 2 and 3, and that if either one of tasks 2 and 3 affects task 1 independently, task 1 will redo 60% of its original design workload. If tasks 2 and 3 affect task 1 simultaneously, the redesign workload of task 1 is 120% of its original design workload [see Fig. 2(a)]. In some cases such as the development of a new product with an innovative design, it is possible that the redesign will in fact need more total work, because the revisions due to tasks 2 and 3 may create some new problems for task 1. In this situation, the linear model might capture the properties of a coupled design process. However, for the case of designing a mature product without great technological change, the redesign work caused by different design tasks often overlap. Fig. 2(b) illustrates this situation: although the redesign workload for task 1 caused by tasks 2 and 3 are both 60% of the original design workload, they overlap with each other by 30% of the original design workload. So the total redesign workload is 90% of the original design workload. The overlap in this paper is different from that in [4], because Thomke and Bell [4] call attention to the overlap between the problems discovered during the sequential testing process in new product development, but our research focuses on the overlap between redesign work caused by coupled and parallel design tasks.

A real example of this overlap can be found in the aircraft design process in [1]. Fig. 3 depicts the inputs, outputs, and internal computation of the coupled design tasks of the aircraft design process. From the equation for an aerodynamics design, the drag D, one of the output of the task of aerodynamics design, depends on both Θ and ESF. Therefore, once Θ changes due to the redesign of structure, or ESF changes due to the redesign of propulsion, rework is needed for aerodynamics design.

Manuscript received September 25, 2010; revised April 14, 2011, July 23, 2011, and September 7, 2011; accepted September 18, 2011. This work was supported in part by the National Natural Science Foundation of China under Grant 60974096 and by the China Scholarship Council. Review of this manuscript was arranged by Department Editor S. (Sri) Talluri.

Z. Wang is with the School of Automation, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: wangz@seu.edu.cn).

C. L. Magee is with the Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA 02129 USA, and also with Singapore–MIT International Design Centre, Singapore University of Technology and Design, Changi 279623, Singapore (e-mail: cmagee@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2011.2170691

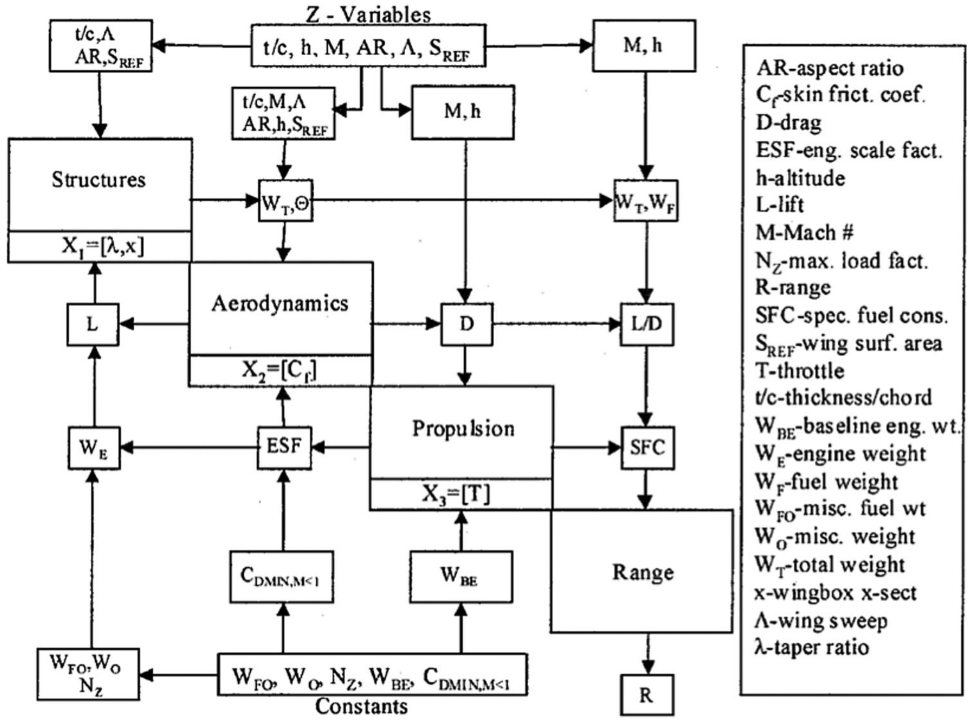

Fig. 1. Coupled design tasks in an aircraft design process (see [1, Fig. 4]).

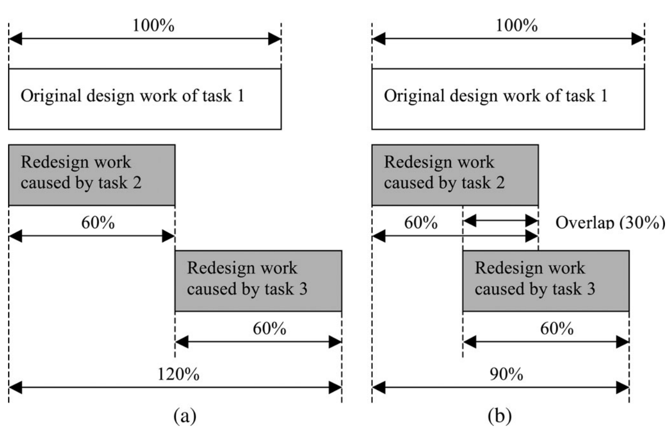

Fig. 2. Redesign of task 1 caused by tasks 2 and 3. (a) If the redesign work caused by tasks 2 and 3 does not overlap with each other, the total redesign workload will be 120% of the original design workload. (b) If the redesign work caused by tasks 2 and 3 overlaps with each other, the total redesign workload will be 90% of the original design workload.

Obviously, when Θ and ESF change at the same time, we only need to redesign the aerodynamics system once, not twice. That is the overlap between the aerodynamics redesign caused by the redesign of structure and propulsion. However, according to the linear model, we would redesign the aerodynamics system twice.

From the aforementioned analysis, we find that the linear model cannot capture the overlap between redesign work. How can one model the coupled design process with overlap between redesign work in order to analyze its convergence? It is the problem investigated in this paper. In the remainder of this section, we will first give a brief review on the existing research on coupled design processes, and then give a more detailed definition of our problem.

B. Literature Review

Research on coupled design processes is usually done from three perspectives: first, model them as a design optimization problem and find the algorithm to solve it; second, model it by using the design structure matrix (DSM) technique and manage it by reorganizing the DSM or analyzing its convergence; third, model it by using the social network technique and analyze the mechanism of design problem solving.

If we focus on a coupled design problem itself, design optimization techniques are often applied. For example, the technique of adjoint coupled postoptimality sensitivity analysis is often applied to construct the algorithm for solving the coupled aerostructural design optimization problem in aircraft design [5]. In the design of complex products or systems, multidisciplinary design optimization is often applied [6], in which coupling between design tasks exists extensively (e.g., the coupling between the fluid dynamic analysis, thermal finite element analysis, and structure finite element analysis in the design of an aircraft engine component [7]). Wang *et al.* [8] proposed a proportional–integral control policy of the coupled design optimization processes of a type of manipulator. Several techniques have also been proposed to coordinate the design optimization process of complex products [9], [10].

However, if we focus on the process of coupled design, the DSM [11] is a very strong tool for modeling and analysis. For example, the DSM method has been applied to an integrated concurrent engineering environment for the conceptual design of a space system with 172 design parameters and 682

BB	Inputs	Internal	Outputs
Structures	$AR, \Lambda, t/c, S_{REF}, W_{FO}, W_O, W_E, L, N_Z, \lambda, x$	$t = \frac{t/c\, S_{REF}}{\sqrt{S_{REF} AR}}$; $b/2 = \sqrt{S_{REF} AR/2}$; $R = \frac{1+2\lambda}{3(1+\lambda)}$; $\Theta = pf(x, b/2, R, L)$; $Fo1 = pf(x)$; $W_w = (0.0051(W_T N_Z)^{0.557} S_{REF}^{0.649} AR^{0.5} (t/c)^{-0.4} (1+\lambda)^{0.1} (0.1875 S_{REF})^{0.1} / \cos(\Lambda)) Fo1$; $W_{FW} = (5S_{REF}/18)(2/3\, t)42.5$; $W_F = W_{FW} + W_{FO}$; $W_T = W_O + W_w + W_F + W_E$; $\sigma 1 \rightarrow \sigma 5 = pf(t/c, L, x, b/2, R)$;	W_T, W_F, Θ
		Constraints: $\sigma 1 \rightarrow \sigma 5 \le 1.09$; $0.96 \le \Theta \le 1.04$	
Aero-dynamics	$M, h, \Lambda, AR, t/c, S_{REF}, W_T, \Theta, ESF, C_{Dmin,M<1}, C_f$	if $h < 36089$ft, $V = M1116.39\sqrt{1-(6.875e-06)h}$, $\rho = (2.377e-03)(1-(6.875e-06h))^{4.2561}$; $V = M968.1$, $\rho = (2.377e-03)e^{-(h-36089)/20806.7}$, if $h > 36089$ ft; $C_L = \frac{W_T}{0.5\rho V^2 S_{REF}}$; $Fo1 = pf(ESF, C_f)$; $C_{Dmin} = C_{Dmin,M<1} Fo1 + 3.05(t/c)^{5/3} \cos(\Lambda)^{3/2}$; $k = 1/(\pi 0.8 AR)$; $Fo2 = pf(\Theta)$; $C_D = (C_{Dmin} + kC_L^2)Fo2$; $L = W_T$; $D = C_D 0.5\rho V^2 S_{REF}$; $dp/dx = pf(t/c)$	$L, D, \frac{L}{D}$
		Constraints: $dp/dx \le 1.04$, evaluated at system level	
Propulsion	M, h, D, W_{BE}, T	$\bar{T} = T*16168.6$; $Temp = pf(M, h, T)$; $ESF = (D/3)/\bar{T}$; $SFC = 1.1324 + 1.5344M - (3.2956e-05)h - (1.6379e-04)\bar{T} -031623M^2 + (8.2138e-06)Mh - (10.496e-05)\bar{T}M - (8.574e-11)h^2 + (3.8042e-09)\bar{T}h + (1.0600e-08)\bar{T}^2$; $W_E = 3W_{BE} ESF^{1.05}$; $\bar{T}_{UA} = 11484 + 10856M - 0.50802h + 3200.2M^2 - 0.29326Mh + (6.8572e-06)h^2$	SFC, W_E, ESF
		Constraints: $0.5 \le ESF \le 1.5$; $\bar{T} \le \bar{T}_{UA}$; $Temp \le 1.02$	
Range	$M, h, \frac{L}{D}, W_T, W_F, SFC$	$\theta = 1 - 6.875e-06 * h$, if $h < 36089$ft; $\theta = 0.7519$ if $h > 36089$ft; $R = \frac{M(L\ D)661\sqrt{\theta}}{SFC}\ln\left(\frac{W_T}{W_T - W_F}\right)$	R
Constants	$W_{FO} = 2000lb$; $W_O = 25000lb$; $N_Z = 6g$; $W_{BE} = 4360lb$; $C_{Dmin,M<1} = 0.01375$		
Side Constraints	$0.1 \le \lambda \le 0.4$; $0.75 \le x \le 1.25$; $0.75 \le C_f \le 1.25$; $0.1 \le T \le 1.0$; $0.01 \le t/c \le 0.09$; $30000 \le h \le 60000$; $1.4 \le M \le 1.8$; $2.5 \le AR \le 8.5$; $40 \le \Lambda \le 70$; $500 \le S_{REF} \le 1500$		

Fig. 3. Inputs, outputs, and internal computations of the coupled design tasks in an aircraft design tasks (see [1, Table A1]).

dependences [12]. Browning [13] broadly surveys the methods of modeling, analyzing, and reorganizing the product development process based on the DSM model.

Using the DSM model, we can analyze the structure of a complex product development process or the organization of a product development team. For example, MacCormack and Baldwin [14] analyzed the modularity of complex software such as the Linux operating system and the Mozilla browser by using the DSM method. Sosa *et al.* [15] also used the DSM model to identify the modular and the integrative systems of a large commercial aircraft engine. By diagonalizing or triangularizing a DSM, we can modularize the structure of products, or resequence design tasks to shorten the development duration [16], [17]. The DSM model is also applied to the organizational aspect of a product development project. For example, Batallas and Yassine [18] combined the DSM model and the technique of social network analysis (SNA) to identify key information leaders in the development team of a large commercial aircraft engine. With the help of the DSM tool, Sosa *et al.* [19] investigated a complex product development process by combining the point of views of both product architecture and organizational structure, and explained the misalignment between them.

The DSM model can also be used to analyze the convergence of coupled design processes, which is of particular interest in this paper. In this field, Smith and Eppinger [3], [20] did pioneering work by analyzing the convergence of purely parallel and pure sequential coupled design processes based on the DSM model. Meier *et al.* [21] proposed a competent genetic algorithm to sequence a design process based on a DSM model of the process so that design iterations can be decreased and the coupled design process can converge faster. As a complement of the pure sequential or parallel design processes, Joglekar *et al.* [22] considered the case that overlap between coupled design activities is allowed, and proposed a performance generation model to develop the insight of managing the overlap degree for coupled design processes. Mihm *et al.* [23] analyzed the oscillation phenomenon in a coupled design process based on the Jacobi matrix between product performances and design parameters, which can also be considered as a DSM. Yassine *et al.* [24] used the DSM model to analyze how information hiding in product development leads to the design churn effect.

Besides the DSM model, the SNA technique is also applied to capture and analyze coupled design processes. Braha and Bar-Yam [25], [26] reported that the network composed of product development tasks has the properties of small-world network and there are some critical tasks with relatively much higher in-degree or out-degree acting as the central information generator or information consumer. Based on the social network model, Braha and Bar-Yam [27] also analyzed the impact of the network topology on the convergence of the product development process.

C. Problem Statement

In this research, we will focus on the convergence analysis of coupled design processes with overlap between redesign work. To deal with this overlap, we will construct a new nonlinear model for a coupled design process of a mature product, in which the redesign workload of every design task in later iterations is not more than the workload of its original design. Based on the nonlinear model, we will find the sufficiency condition for the convergence of such coupled design processes. This is the first issue addressed in this paper.

However, in real world product development, it is not enough to only evaluate whether a coupled design process converges. From the point of view of product developers, estimation of product development cycle time is critical in managing a product development project. For this purpose, simulation-based methods [28]–[30] or Markov chain analysis technique [31] are often used. Different from the existing work on product development time estimation, in this paper, we explore a similar problem from another perspective, i.e., to investigate the method of estimating the number of iterations that is needed for the workload of this coupled design process to reduce to an acceptance level (e.g., 1% of the original design workload). This is the second issue addressed in this paper. We pay attention to the number of iterations before converging instead of the product development cycle time because: 1) the duration of all design tasks and their probability distributions in a complex product design process are not usually known with any accuracy. Thus, estimation of product development cycle time becomes very difficult. However, estimation of design iterations does not require the knowledge of design task duration, which makes it possible to analyze the convergence of a coupled design process quantitatively with less data; 2) given any specific distribution of design tasks duration, the cycle time of a coupled product design process is determined by the number of design iterations. Therefore, estimating the number of design iterations is a more fundamental issue compared to the estimation of product development cycle time. To our knowledge, investigation of this issue is lacking in the literature.

The rest of this paper is organized as follows. In Section II, the sufficiency condition for the convergence of a coupled design process with overlap between redesign work is proposed based on a nonlinear model, as well as a heuristic rule for reducing the possibility of considering a design process that converges as one that does not converge. In Section III, we propose the sufficiency condition of estimating the number of iterations that is needed for the workload of a coupled design process to reduce to an acceptance level. Another heuristic rule is introduced to increase the accuracy of the estimation of the number of design iterations. The results of numerical experiments are reported in Section IV to test the correctness and the effectiveness of the proposed conditions and rules. We summarize this paper in Section V.

II. Convergence Analysis

A. Model of the Coupled Design Process With Overlap Between Redesign Work

Suppose that a coupled design process consists of I design tasks that are carried out in a purely parallel way. The workload of every design task can be measured by the time and personnel that are needed for completing it, for the original design or for redesign in later iterations. For the convenience of measuring the mutual impacts between different design tasks under the same metric, we normalize the measurement of design workload to the interval [0, 1] with 1 being the maximum (original) workload. With this normalized measurement, we denote the workload of design task i at the nth iteration by $x_i(n)$ ($\in$[0, 1]).

To capture the coupling between design tasks, we use a nonnegative parameter $a_{ij}\in[0, 1]$ to represent the influence of the workload of task j on the workload of task i. If $a_{ij} > 0$, then task i depends on task j directly; if $a_{ij} = 0$, then task i does not depend on task j directly. The value of a_{ij} gives the proportion of redesign workload of task i to its original design workload when the redesign workload of task j is 1 and task i only depends on task j. The matrix $\boldsymbol{A} = (a_{ij})_{I\times I}$ is a DSM capturing this coupled design process.

If task i depends on more than one task, the redesign work of task i caused by other tasks may overlap. We use the following nonlinear difference equations to model the dynamics of a coupled design process with overlap between redesign work:

$$x_i(n+1) = \frac{1}{\beta_i}\left\{1 - \exp\left[-\beta_i \sum_{j=1}^{I} a_{ij}x_j(n)\right]\right\} \quad \text{for } i = 1, \ldots, I. \quad (1)$$

Equation (1) is an increasing and concave function that captures how the design workload of task i at the nth iteration is affected by the design workloads of other design tasks at the nth iteration. We use the parameter $\beta_i \in (0, 1)$ to capture how the number (denoted by η_i) of the design tasks that task i depends on influences the redesign workload of task i. If $\eta_i = 1$, i.e., task i only depends on one design task, then β_i is very small, which makes (1) approximate to a linear equation $x_i(n+1) = a_{ij}x_j(n)$. β_i becomes larger when η_i increases, which enhances the nonlinearity of (1) and, therefore, captures the overlap in redesign work. For a given DSM $\boldsymbol{A} = (a_{ij})_{I\times I}$, all the β_i, $i = 1, \ldots, I$, are constants. For simplicity, we can let β_i be equal to a sufficiently small positive number for $\eta_i = 1$, and let $\beta_i = 1$ for $\eta_i \geq 2$. We also assume that the initial values of the workloads are always 1, i.e., $x_i(0) = 1$ for $i = 1, \ldots, I$, which means that the original design always has the largest workload. From (1), we can see that if β_i, $i = 1, \ldots, I$, are chosen carefully, the redesign workload of a task caused by the redesign workloads of other tasks will not be larger than 1 (e.g., let β_i be equal to a sufficiently small positive number for $\eta_i = 1$, and let $\beta_i = 1$ for $\eta_i \geq 2$), which captures the overlap between redesign work. If a_{ij} is very small, then (1) is approximately equal to $x_i(n+1) = \sum_{j=1}^{I} a_{ij} x_j(n)$, which is actually the linear model in [3]. That is to say, if the influences of other design tasks on task *i* are very small, then their influences are approximately independent of each other. However, if a_{ij} is large, that approximate linear equation does not hold anymore. In this case, the influences of other design tasks on task *i* overlap with each other. Therefore, (1) captures both the linear property of the coupled design process when a design task depends on only one other task, and the nonlinear property when a design task depends on multiple design tasks.

Braha and Bar-Yam [27] proposed nonlinear differential equations to capture the dynamics of the complex product development process based on a social network model for the process (see [27, eq. (9), eq. (15), and eq. (EC1)] and its electronic companion). In some ways, our model is similar to theirs, but the analysis methods are quite different. The similarity between our model and Braha and Bar-Yam's model are twofold: first, the state variables of the two models are similar. In our model, the state variable is the normalized redesign workload [$x_i(n)$ in (1)], while in Braha and Bar-Yam's model, the state variable is the density of unsolved tasks. They can be considered as equivalent in the mean field sense. Second, the nonlinear functions in the two models have similar characteristics, i.e., increasing and concave. (Note that we had to rewrite the nonlinear differential functions in Braha and Bar-Yam [27] to difference equations and then compared them with our model.) The difference between our model and Braha and Bar-Yam's model is that our model captures the change of the redesign workload of every design task, while Braha and Bar-Yam's model captures the change of the global density of unsolved tasks from the point of view of the entire product development project, or the changes of the density of unsolved tasks with different in-degrees in the social network model. The analysis of the two models is also quite different. In Braha and Bar-Yam [27], a linearization technique is applied to analyze the stability of the design process when the density of unsolved tasks is very small. However, as will be discussed in the next section, we analyze the convergence of a coupled design process by using the Lyapunov stability theory based on our nonlinear model, which does not need linearization.

B. Convergence of the Coupled Design Process With Overlap Between Redesign Work

The convergence of a coupled design process with overlap between redesign work is equivalent to the asymptotic stability of the nonlinear system captured by (1) when the number of design iterations (i.e., *n*) goes to infinity. We can obtain the sufficiency condition of the convergence of a coupled design process by Lyapunov stability theory as follows.

Theorem 1: The sufficiency condition for the convergence of a coupled design process with overlap between design work modeled by (1) is that the absolute values of the real parts of all the eigenvalues of the matrix $\boldsymbol{A}^T\boldsymbol{A}$ ($\boldsymbol{A} = (a_{ij})_{I\times I}$) is less than one, i.e.,

$$\left|\mathrm{Re}\left(\lambda_i\left(\boldsymbol{A}^T\boldsymbol{A}\right)\right)\right| < 1, \quad \text{for } i = 1, \ldots, I. \tag{2}$$

Proof: See Appendix A.

Theorem 1 is a sufficiency condition but not a necessary condition of the convergence of such a coupled design process. If (2) is satisfied, this coupled design process must converge. This judgment can be called *the first type of correct judgment* (CJ1). If (2) is *not* satisfied and the coupled design process does *not* converge, we call it *the second type of correct judgment* (CJ2). If (2) is satisfied but the coupled design process does not converge, we call it *the first type of incorrect judgment* (ICJ1). Since (2) is the sufficiency condition, this type of incorrect judgment will not happen. However, even if (2) is not satisfied, the coupled design process may still converge. According to the proof of Theorem 1, this phenomenon happens because 1) $|\mathrm{Re}\lambda_i| < 1$ is the sufficiency condition but not the necessary condition of the negative definiteness of $G(\boldsymbol{x})$ (or $\Delta W(\boldsymbol{x}(n))$) (see Appendix A for the definition of $G(\boldsymbol{x})$ and $\Delta W(\boldsymbol{x}(n))$; and 2) even if $G(\boldsymbol{x})$ (or $\Delta W(\boldsymbol{x}(n))$) is not negative definite, the coupled design process might also converge. This might be explained as follows. Equation (2) ensures that $G(\boldsymbol{x})$ is negative definite for all the $\boldsymbol{x}$ in the state space. However, if $G(\boldsymbol{x})$ is not negative definite but the $\boldsymbol{x}$ that make $G(\boldsymbol{x}) > 0$ are confined in a very small domain of the state space, then there is still a good chance for $G(\boldsymbol{x}) < 0$ to hold for the $\boldsymbol{x}$ on the state trajectories of the coupled design process captured by (1). In this case, according to the Lyapunov stability theory, the coupled design process converges with a large probability. Therefore, we only require $G(\boldsymbol{x}) < 0$ for most of the $\boldsymbol{x}$ and allows $G(\boldsymbol{x}) > 0$ for a small part of $\boldsymbol{x}$.

Let us take a coupled design process composed of only three design tasks for example. The DSM of this coupled design process is

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0.1 & 0.1 \\ 0.1 & 0 & 0.1 \\ 0.7 & 0.8 & 0 \end{bmatrix}.$$

The maximum eigenvalue of the matrix $\boldsymbol{A}^T\boldsymbol{A}$ is 1.1402. According to Theorem 1, this coupled design process may not converge. However, simulation based on (1) shows that it converges, because the workloads of the two design tasks become less than 0.001 after ten iterations. By checking the function $G(\boldsymbol{x})$ of this example, we find that the $\boldsymbol{x}$ that make $G(\boldsymbol{x}) > 0$ are confined in a very small part of the whole state space $[0, 1] \times [0, 1] \times [0, 1]$, which are shown by the dark domain in Fig. 4. For the convenience of depicting the 3-D vector $\boldsymbol{x}$ on a 2-D plane, in Fig. 4, the horizontal axis is x_1 while the vertical axis is $x_2 + x_3$. Although it does not depict the true distribution of the $\boldsymbol{x}$ satisfying $G(\boldsymbol{x}) > 0$ exactly, this approximation captures the fundamental feature of this distribution. The black circles in Fig. 4 represent the workloads of the two design tasks at all iterations, i.e., the trajectory of the workloads in the design process. From this

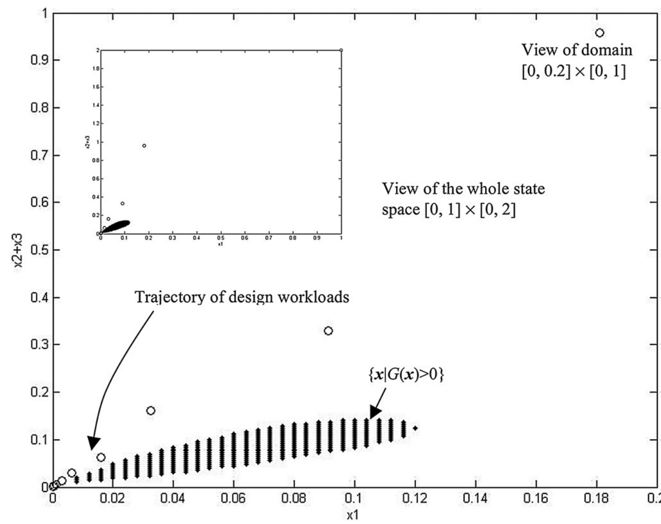

Fig. 4. Distribution of the $\boldsymbol{x}$ that makes $G(\boldsymbol{x}) > 0$ and the trajectory of design workloads.

trajectory, we can see that the circles representing the workloads do not enter the dark domain that represents $G(\boldsymbol{x}) > 0$.

Therefore, the sufficiency condition in Theorem 1 is too strict. If we use Theorem 1 to evaluate the convergence of a coupled design process, we might make an incorrect judgment: considering a coupled design process that converges as one that does not converge, which can be called *the second type of incorrect judgment* (ICJ2). We need to find a relaxed condition to reduce the possibility of making the second type of incorrect judgment.

Let us start with (A4) (see Appendix A), which can be rewritten as

$$G(\boldsymbol{x}) = \sum_{i=1}^{I} \lambda_i \left(\sum_{k=1}^{I} u_{ik} x_k \right)^2 - \sum_{i=1}^{I} \frac{1}{\beta_i^2} \left[\ln(1 - \beta_i x_i) \right]^2 \quad (3)$$

where u_{ik} is the kth entry of the eigenvector $\boldsymbol{u}_i = (u_{i1}, \ldots, u_{iI})^T$ corresponding to the eigenvalue λ_i. This is because there exists a orthogonal matrix such that $\boldsymbol{U}^T \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{U} = \boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_I\}$, where $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_I)^T$. Actually, the distribution of $\boldsymbol{x}$ that satisfies $G(\boldsymbol{x}) > 0$ in the domain $[0, 1] \times \cdots \times [0, 1]$ determines the stability of the design process. It is not easy to compute this distribution, but from Fig. 4 and some other simulations we can conjecture that the set $\{\boldsymbol{x} | G(\boldsymbol{x}) > 0\}$ usually does not cover the straight line $x_1 = \cdots = x_I$, and the trajectory of design workloads is usually close to the straight line $x_1 = \cdots = x_I$. Therefore, to make the coupled design process converge, it might only be necessary to require that $G(\boldsymbol{x})$ is negative definite for the $\boldsymbol{x}$ on the straight line $x_1 = \cdots = x_I$, instead of requiring $G(\boldsymbol{x}) < 0$ for all the $\boldsymbol{x}$ in the state space. This is a relaxation of the sufficiency condition (2). In this case, (3) becomes

$$G(x_1) = \sum_{i=1}^{I} G_i(x_1) = \sum_{i=1}^{I} \left\{ \lambda_i \left(\sum_{k=1}^{I} u_{ik} \right)^2 \times x_1^2 - \frac{1}{\beta_i^2} \left[\ln(1 - \beta_i x_1) \right]^2 \right\}. \quad (4)$$

Similar to the analysis in the proof of Theorem 1 [see (A4)–(A9)], we find that if $\lambda_i (\sum_{k=1}^{I} u_{ik})^2 < 1$ holds for all the $i = 1, \ldots, I$, then $G(x_1)$ is negative definite. So we have a heuristic rule for evaluating the convergence of the coupled design process.

Rule 1: An approximate condition for a coupled design process captured by (1) to converge is

$$\lambda_i \left(\sum_{k=1}^{I} u_{ik} \right)^2 < 1, \quad \text{for } i = 1, \ldots, I. \quad (5)$$

If $\beta_1 = \cdots = \beta_I$, (5) becomes $\sum_{i=1}^{I} \lambda_i (\sum_{k=1}^{I} u_{ik})^2 < I$. Here, $\lambda_i (\sum_{k=1}^{I} u_{ik})^2$ or $\sum_{i=1}^{I} \lambda_i (\sum_{k=1}^{I} u_{ik})^2$ is called the *judgment parameter*.

Rule 1 is neither a sufficiency nor a necessary condition for the convergence of a coupled design process. We can also define the two types of correct judgments and two types of incorrect judgments for Rule 1 like those for Theorem 1. Numerical experiments in Section IV will show that the possibility of making the second type of incorrect judgment will be reduced greatly by using Rule 1 compared to using Theorem 1. And the possibility of making the first type of incorrect judgment is very close to zero if Rule 1 is applied.

III. Estimation of the Number of Iterations

In a real-world product development process, making a correct judgment of the convergence of a coupled design process is not enough. Product development managers are quite concerned about the number of design iterations before a coupled design process converges. If we can estimate the number of design iterations that is needed for a coupled design process to converge, then it will be quite helpful for planning a product development project. From the point of view of practice, if the design workloads fall to an acceptance level, say 1% of the original design workloads, in finite iterations, then we can consider that the design process converges practically. Therefore, if we can find the sufficiency condition that the workload of every design task is less than an acceptance level δ after N iterations, we can estimate the number of iterations before converging. That is what Theorem 2 determines.

Theorem 2: Given that the initial value of the design workloads of all the design tasks are $x_1(0) = \cdots = x_I(0) = 1$, the sufficiency condition that the design workload of every design task is less than an acceptance level δ after N design iterations is that there exist δ_{in}, $n = 0, 1, \ldots, N$, such that

$$\sum_{j=1}^{I} a_{ij} < \min \left\{ -\frac{\ln(1 - \beta_i \delta_{in})}{\beta_i \delta_{i,n-1}}, \quad \text{for } n = 1, \ldots, N \right\}, \quad \text{for } i = 1, \ldots, I \quad (6)$$

holds, where $\delta_{i0} = 1$, $\delta_{iN} = \delta$ and $\delta < \delta_{in} < 1$, for $n = 1, \ldots, N$, and $i = 1, \ldots, I$.

Proof: See Appendix B.

Here arises a problem: how to obtain the δ_{in}, $n = 1, \ldots, N$, $i = 1, \ldots, I$, in Theorem 2? Of course, for any δ_{in}, $n = 1, \ldots, N$, $i = 1, \ldots, I$, as long as (6) is satisfied, the conclusion of Theorem 2 is correct. However, we hope to find a group of δ_{in}, $n = 1, \ldots, N$, $i = 1, \ldots, I$, to maximize the right-hand side of (6), because that will relax the constraint on the left-hand side of (6) to the largest extent, so that stronger interdependence (i.e., larger a_{ij}) between design tasks is allowed for given N and δ. Theorem 3 gives the condition of maximizing the right-hand side of (6).

Theorem 3: 1) For a given i, the δ_{in}, $n = 1, \ldots, N$, which satisfies

$$-\frac{\ln(1-\beta_i\delta_{i1})}{\beta_i\delta_{i0}} = -\frac{\ln(1-\beta_i\delta_{i2})}{\beta_i\delta_{i1}} = \cdots = -\frac{\ln(1-\beta_i\delta_{iN})}{\beta_i\delta_{i,N-1}} \tag{7}$$

can maximize $\min\{-\ln(1-\beta_i\delta_{in})/(\beta_i\delta_{i,n-1}), \text{for } n = 1, \ldots, N\}$. 2) Therefore, the sufficiency condition for the workload of every design task to be less than δ after N design iterations is

$$\sum_{j=1}^{I} a_{ij} < -\frac{\ln(1-\beta_i\delta_{i1})}{\beta_i\delta_{i0}} = -\frac{\ln(1-\beta_i\delta_{i2})}{\beta_i\delta_{i1}} = \cdots = -\frac{\ln(1-\beta_i\delta_{iN})}{\beta_i\delta_{i,N-1}} \quad \text{for } i = 1, \ldots, I \tag{8}$$

where $\delta_{i0} = 1$, $\delta_{iN} = \delta$, and $\delta < \delta_{in} < 1$ for $n = 1, \ldots, N$ and $i = 1, \ldots, I$.

Proof: See Appendix C.

Actually, Theorem 3 is a special case of Theorem 2. It gives the upper bound of the right-hand side of (6). It is also a sufficiency condition for estimating the number of design iterations but not a necessary condition. We may overestimate the number of iterations of a coupled design process before converging by using Theorem 3. Namely, it is possible that the design workload can be less than δ only after N' iterations, where $N' < N$. The numerical experiments in Section IV will show this case. Furthermore, the difference between the estimated number of iterations and that obtained by simulation becomes larger when $\sum_{j=1}^{I} a_{ij}$ becomes larger. The reason for this phenomenon can be explained as follows.

From the proof of Theorem 3, (8) ensures that the workloads of all the design tasks are less than or equal to δ_{in} after n iterations. Actually, even if the workloads of all the design tasks are equal to δ_{in} after n iterations, it will take only $N-n$ iterations for them to become less than δ. Therefore, (8) makes the convergence faster than expected. In this sense, (8) is a relatively strict constraint. If we relax this constraint by requiring that the sum of all the entries of the DSM $\boldsymbol{A}$ be less than the sum of $-\ln(1-\beta_i\delta_{i1})/(\beta_i\delta_{i0}) = -\ln(1-\beta_i\delta_{i2})/(\beta_i\delta_{i1}) = \cdots = -\ln(1-\beta_i\delta_{iN})/(\beta_i\delta_{i,N-1})$ over i, then the estimation of iterations might be more accurate. That is the following heuristic rule for estimating the number of iterations.

Rule 2: An approximate condition for the workload of every design task of a coupled design process to be less than δ after N design iterations is

$$\sum_{i=1}^{I}\sum_{j=1}^{I} a_{ij} < -\sum_{i=1}^{I}\frac{\ln(1-\beta_i\delta_{i1})}{\beta_i\delta_{i0}} = \cdots = -\sum_{i=1}^{I}\frac{\ln(1-\beta_i\delta_{iN})}{\beta_i\delta_{i,N-1}} \tag{9}$$

where $\delta_{i0} = 1$, $\delta_{iN} = \delta$, and $\delta < \delta_{in} < 1$, for $n = 1, \ldots, N$. If $\beta_1 = \cdots = \beta_I$, (9) becomes

$$\frac{1}{I}\sum_{i=1}^{I}\sum_{j=1}^{I} a_{ij} < -\frac{\ln(1-\delta_1)}{\delta_0} = \cdots = -\frac{\ln(1-\delta_N)}{\delta_{N-1}} \tag{10}$$

where $\delta_0 = 1$, $\delta_N = \delta$, and $\delta < \delta_n < 1$, for $n = 1, \ldots, N$.

Rule 2 is not a sufficiency condition for design workload to become less than δ after N iterations. Perhaps more iterations than estimated by Rule 2 are needed to make the workloads become less than δ, but the number of iterations should be closer to N than that estimated by Theorem 3. Numerical experiments in the next section will support this conclusion.

IV. Numerical Experiments

In this section, numerical experiments are conducted to examine the correctness of the sufficiency conditions for evaluating the convergence of a coupled design process and the number of iterations before converging, as well as the effectiveness of the two heuristic rules for improving the accuracy of convergence judgment and iteration estimation.

A. Experiment 1: Convergence Analysis of an Aircraft Design Process

In the first experiment, we analyze the convergence of the aircraft design process in [5], which is composed of four design tasks: structure design, aerodynamics design, propulsion design, and range design. The interdependence between design tasks is depicted in Fig. 2. The input, output, and equations for internal computation of every design task are described in Fig. 3. According to Figs. 2 and 3, we can construct the DSM of the aircraft design process as follows:

$$\begin{array}{r} \text{structure} \\ \text{aerodynamics} \\ \text{propulsion} \end{array} \begin{bmatrix} 0 & 1/3 & 1/3 \\ 1 & 0 & 1/2 \\ 0 & 1/2 & 0 \end{bmatrix}.$$

The first, the second, and the third row (or column) of the DSM is corresponding to the structure design, the aerodynamics design, and the propulsion design, respectively. We do not include the range design in the matrix because only the first three design tasks are coupled with each other. The nonzero entries in the DSM is determined in this way: if task i has p_i outputs, and the output of task j affects q_{ji} outputs of task i, then the entry in the ith row and the jth column of the DSM is q_{ji}/p_i. For example, the task of aerodynamics design has two outputs: the lift L and the drag D. This task depends on the output ESF from the task of propulsion design. Since ESF only affects D (see the equations in

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

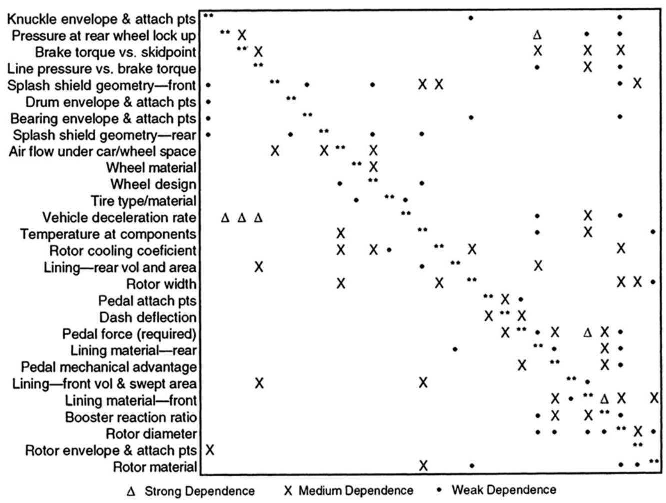

Fig. 5. DSM of a brake system design process (i.e., [3, Fig. 7]).

Fig. 3), the entry at the cross point of the row of "aerodynamic" and the column of "propulsion" is 1/2. If the linear model in Smith and Eppinger [3] is applied, we can obtain the maximum eigenvalue of this DSM, which is 0.87921 (<1). Therefore, this coupled design process converges. By simulation, the number of design iterations before converging (i.e., the redesign workload of every design task is less than 0.01) is 58. If the nonlinear model proposed in this paper is applied, we can also determine that this coupled design process converges according to Rule 1, because the judgment parameter is 2.94, which is less than 3, i.e., the size of the DSM. By simulation, we find that the number of design iteration before converging is 45, according to the nonlinear model. Obviously, the number of design iterations based on the nonlinear model is much less than that based on the linear model. This is because the nonlinear model captures the overlap between the aerodynamics redesign work caused by the redesign of structure and propulsion.

B. Experiment 2: Evaluate the Convergence of the Brake System Design Process

Smith and Eppinger [3] gave an example of designing a brake system. They construct its DSM model and identify a coupled block (a 28 $\times$ 28 DSM, see Fig. 5). Every off-diagonal entry in this coupled block is an estimation of the workload (in % of original workload) that the design task in the column of the entry creates for the task in the row of the entry. Every entry has three possible values, i.e., 0.5, 0.25, and 0.05, corresponding to strong, medium, and weak dependence between design tasks, respectively. For this coupled block, we apply Theorem 1 and Rule 1 to evaluate its stability. The maximum eigenvalue is 1.2115, which is greater than 1; and the value of the judgment parameter is 15.4675, which is less than the dimension of the DSM, i.e., $I = 28$. According to Theorem 1, we cannot say that the design process converges; however, according to Rule 1, we can say that it converges. Simulation of this coupled design process based on the nonlinear model [i.e., (1)] show that it does converge. Therefore, this result supports the judgment based on Rule 1.

TABLE I
COMPARISON OF THE PERFORMANCE OF THEOREM 1 AND RULE 1.
(a) PERFORMANCE OF THEOREM 1 (FOR 30-BY-30 DSM).
(b) PERFORMANCE OF RULE 1 (FOR 30-BY-30 DSM)

(a)

UB	0.31	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39	0.40
CJ1	100	100	100	100	97	100	96	88	83	80
CJ2	0	0	0	0	0	0	0	0	0	0
ICJ1	0	0	0	0	0	0	0	0	0	0
ICJ2	0	0	0	0	3	0	4	12	17	20
UB	0.41	0.42	0.43	0.44	0.45	0.46	0.47	0.48	0.49	0.50
CJ1	63	65	46	35	27	19	22	9	3	3
CJ2	0	0	1	0	0	0	1	0	1	4
ICJ1	0	0	0	0	0	0	0	0	0	0
ICJ2	27	35	53	65	73	81	77	91	96	93

(b)

UB	**0.31**	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39	0.40
CJ1	100	100	100	100	100	100	100	100	100	100
CJ2	0	0	0	0	0	0	0	0	0	0
ICJ1	0	0	0	0	0	0	0	0	0	0
ICJ2	0	0	0	0	0	0	0	0	0	0
UB	0.41	0.42	0.43	0.44	0.45	0.46	0.47	0.48	0.49	0.50
CJ1	100	100	99	99	99	95	93	92	81	82
CJ2	0	0	0	0	0	0	1	0	1	4
ICJ1	0	0	0	0	0	1	0	0	0	0
ICJ2	0	0	0	1	1	5	6	8	18	14

UB is the upper bound of the nonzero entries in the DSM.
CJ1 is the number of the first type of correct judgment. CJ2 is the number of the second type of correct judgment.
ICJ1 is the number of the first type of incorrect judgment. ICJ2 is the number of the first type of incorrect judgment.

C. Experiment 3: Comparison of the Performances of Theorem 1 and Rule 1

This experiment is to explore the performances of Theorem 1 and Rule 1. We generate 100 30 $\times$ 30 DSMs randomly, which has a size similar to the DSM of the brake system design process (28 $\times$ 28) and whose entries are a_{ij} s in (1). We also assume that $\beta_1 = \cdots = \beta_I = 1$ holds in (1) for simplicity. The density of the nonzero entries in the DSM (i.e., the number of nonzero entries divided by the number of all the entries) is about 10%, which is close to that of the DSM of the brake system design process. The values of the nonzero entries are uniformly distributed in the interval (0, a_{ub}], where a_{ub} is the upper bound of the nonzero entries. Each DSM $\boldsymbol{A} = (a_{ij})_{30\times 30}$ is corresponding to a coupled design process. We assume that the initial workload of every design task is 1. Each design process is simulated to see whether the workload of every design task is less than 0.001 in 1000 iterations. If they are, we say that this design process converges; otherwise, it does not converge.

First, we compare the performances of Theorem 1 and Rule 1 for different upper bounds of DSM entries. The results are shown in Table I(a) and (b), respectively. From Table I(a), we see that the number of the first type of incorrect judgment is zero, which supports the conclusion of Theorem 1 that (2) is the sufficient condition for the convergence of the coupled design process. However, when a_{ub} increases, the number of the second type of incorrect judgment increases greatly if Theorem 1 is applied. For example, when $a_{\mathrm{ub}} = 0.48$, 0.49, and 0.50, more than 90% of the judgments are incorrect. Namely, although the maximum eigenvalue is greater than 1, the design process still converges. However, if Rule 1 is applied, the number of the second type of incorrect judgment becomes much less than the case that Theorem 1 is applied [see Table I(b)]. From Table I(b), we see that when $a_{\mathrm{ub}} = 0.48$, 0.49, and 0.50, the numbers of incorrect judgments are only 8, 18, and 14 out of 100, respectively. When

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
IMPACT OF THE DENSITY AND THE UPPER BOUND OF NONZERO ENTRIES IN THE DSM ON THE JUDGMENT OF THE CONVERGENCE OF COUPLED DESIGN PROCESSES. (a) PERFORMANCE OF THEOREM 1. (b) PERFORMANCE OF RULE 1

(a)

(ρ, a_{ub})	(0.1, 0.38)	(0.1, 0.40)	(0.1, 0.42)	(0.1, 0.44)	(0.1, 0.46)	(0.1, 0.48)	(0.1, 0.50)	(0.1, 0.52)
ICJ	12	22	54	67	85	85	93	87
(ρ, a_{ub})	(0.1, 0.54)	(0.1, 0.56)	(0.1, 0.58)	(0.1, 0.60)	(0.1, 0.62)	(0.1, 0.64)	(0.1, 0.66)	(0.1, 0.68)
ICJ	87	90	76	61	47	37	21	23
(ρ, a_{ub})	(0.1, 0.70)	(0.1, 0.72)	(0.1, 0.74)	(0.1, 0.76)	(0.1, 0.78)	(0.1, 0.80)	(0.1, 0.82)	(0.1, 0.84)
ICJ	15	13	13	8	2	3	4	1
(ρ, a_{ub})	(0.1, 0.92)	(0.2, 0.26)	(0.2, 0.28)	(0.2, 0.30)	(0.2, 0.32)	0.2, 0.34)	(0.2, 0.36)	(0.2, 0.38)
ICJ	1	15	36	68	73	50	34	18
(ρ, a_{ub})	(0.2, 0.40)	(0.2, 0.42)	(0.3, 0.20)	(0.3, 0.22)	(0.3, 0.24)	(0.3, 0.26)	(0.3, 0.28)	(0.4, 0.14)
ICJ	6	2	36	59	27	2	1	
(ρ, a_{ub})	(0.4, 0.16)	(0.4, 0.18)	(0.5, 0.12)	(0.5, 0.14)	(0.6, 0.12)	(0.7, 0.10)	(0.8, 0.08)	(0.9, 0.08)
ICJ	43	14	1	28	11	23	1	4

(b)

(ρ, a_{ub})	(0.1, 0.44)	(0.1, 0.46)	(0.1, 0.48)	(0.1, 0.50)	(0.1, 0.52)	(0.1, 0.54)	(0.1, 0.56)	(0.1, 0.58)
ICJ	4	6	10	18	21	48	55	54
(ρ, a_{ub})	(0.1, 0.60)	(0.1, 0.62)	(0.1, 0.64)	(0.1, 0.66)	(0.1, 0.68)	(0.1, 0.70)	(0.1, 0.72)	(0.1, 0.74)
ICJ	51	43	35	20	21	15	13	13
(ρ, a_{ub})	(0.1, 0.76)	(0.1, 0.78)	(0.1, 0.80)	(0.1, 0.82)	(0.1, 0.84)	(0.1, 0.92)	(0.2, 0.28)	(0.2, 0.30)
ICJ	8	2	3	4	1	1	5	22
(ρ, a_{ub})	(0.2, 0.32)	(0.2, 0.34)	(0.2, 0.36)	(0.2, 0.38)	(0.2, 0.40)	(0.2, 0.42)	(0.3, 0.20)	(0.3, 0.22)
ICJ	39	31	27	18	6	2	5	32
(ρ, a_{ub})	(0.3, 0.24)	(0.3, 0.26)	(0.3, 0.28)	(0.4, 0.16)	(0.4, 0.18)	(0.5, 0.14)	(0.6, 0.12)	(0.7, 0.10)
ICJ	22	2	1	15	11	20	8	14
(ρ, a_{ub})	(0.8, 0.08)	(0.9, 0.08)						
ICJ	1	3						

ρ is the number of nonzero entries in the DSM. a_{ub} is the upper bound of the nonzero entries in the DSM.
ICJ is the number of incorrect judgment (ICJ = ICJ1 + ICJ2). For other ρ and a_{ub}, the number of incorrect judgment is 0.

$a_{\rm ub} = 0.31, \ldots, 0.43$, there is no incorrect judgment for Rule 1. However, for Theorem 1, incorrect judgment happens when $a_{\rm ub} \geq 0.35$. By comparing Table I(a) and (b), we determine that Rule 1 can reduce the number of the second type of incorrect judgment greatly. Although it cannot avoid the first type of incorrect judgment, its number is very small. Table I(b) shows that if $a_{\rm ub} = 0.46$, the number of the first type of incorrect judgment is one; and for other cases, there are no first type of incorrect judgments.

Second, we examine the influence of the density and upper bound of nonzero DSM entries. We vary the density of nonzero entries from 0.1 to 0.9 with the increment of 0.1 and vary the upper bound of the values of nonzero entries from 0.02 to 1.00 with the increment of 0.02. For each pair of density and upper bound, we generate 100 30 $\times$ 30 DSMs randomly and examine the number of incorrect judgments of convergence. The results are shown in Table II(a) and (b) for Theorem 1 and Rule 1, respectively. By comparing Table II(a) and (b), we find that for different density and upper bound of nonzero entries, the number of incorrect judgment when Rule 1 is used is much less than the number when Theorem 1 is used. If we define the ratio of incorrect judgments as the total number of incorrect judgments divided by the total number of experiments, then the ratio of incorrect judgments of Theorem 1 is 1560/(100 $\times$ 9 $\times$ 50) = 0.034667, and the ratio of incorrect judgments of Rule 1 is 730/(100 $\times$ 9 $\times$ 50) = 0.016222. The same numerical experiments were conducted for 100 $\times$ 100 DSMs. The density of nonzero entries also varies from 0.1 to 0.9 with the increment of 0.1, and the upper bound of nonzero entries varies from 0.01 to 0.25 with the increment of 0.01. In this case, the ratio of incorrect judgments of Theorem 1 is 403/(100 $\times$ 9 $\times$ 25) = 0.017911; and the ratio of incorrect judgments of Rule 1 is 192/(100 $\times$ 9 $\times$ 25) = 0.008533. The ratio of incorrect judgments is reduced by > 50% for DSMs of both sizes.

D. Experiment 4: Estimate the Number of Iterations by Theorem 3 and Rule 2

This experiment is to explore the correctness of Theorem 3 and Rule 2. First, we need to solve (7) to find $\delta_{i1}, \ldots, \delta_{i,N-1}$ satisfying $\delta_{iN} = \delta = 0.01$ for $i = 1, \ldots, I$, and $N = 2, 3, \ldots, 20$. Without loss of generality, we assume that $\beta_1 = \cdots = \beta_I = 1$ holds in (7) for simplicity. The results, i.e., the relation between the maximum value of the sum of the entries in DSM rows and the estimated number of iterations that is needed for the workloads of design tasks to become less than the acceptance level 0.01, are depicted by the stepwise curve in Fig. 6(a). "$*$" in Fig. 6(a) represents the actual number of iterations before the workloads of design tasks become less than 0.01. The

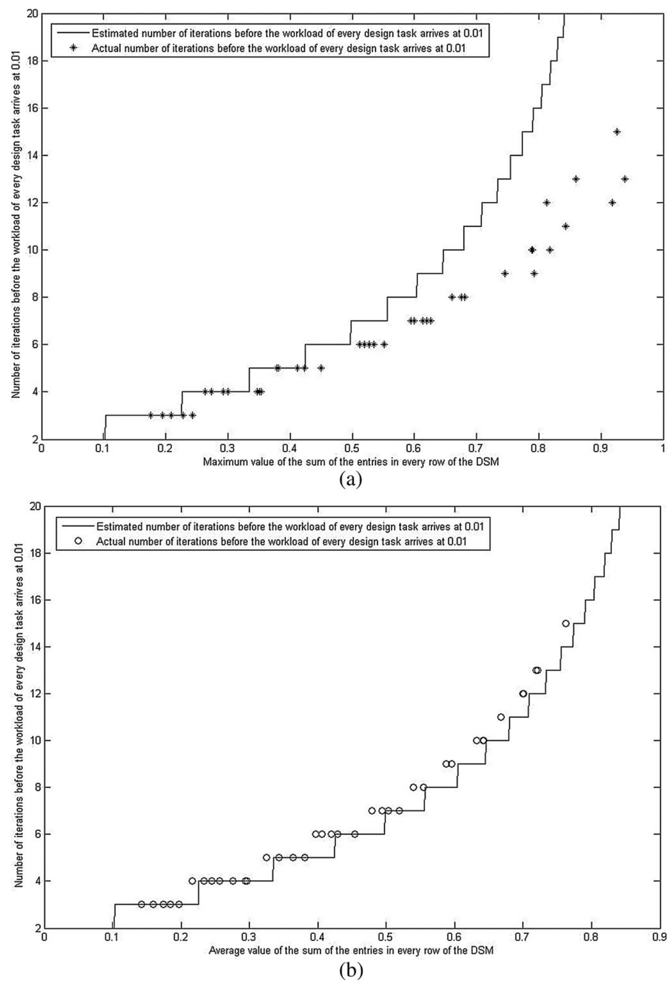

Fig. 6. Iteration estimation of a coupled design process for 30 × 30 DSMs. (a) Iteration estimation by using Theorem 3. (b) Iteration estimation by using Rule 2.

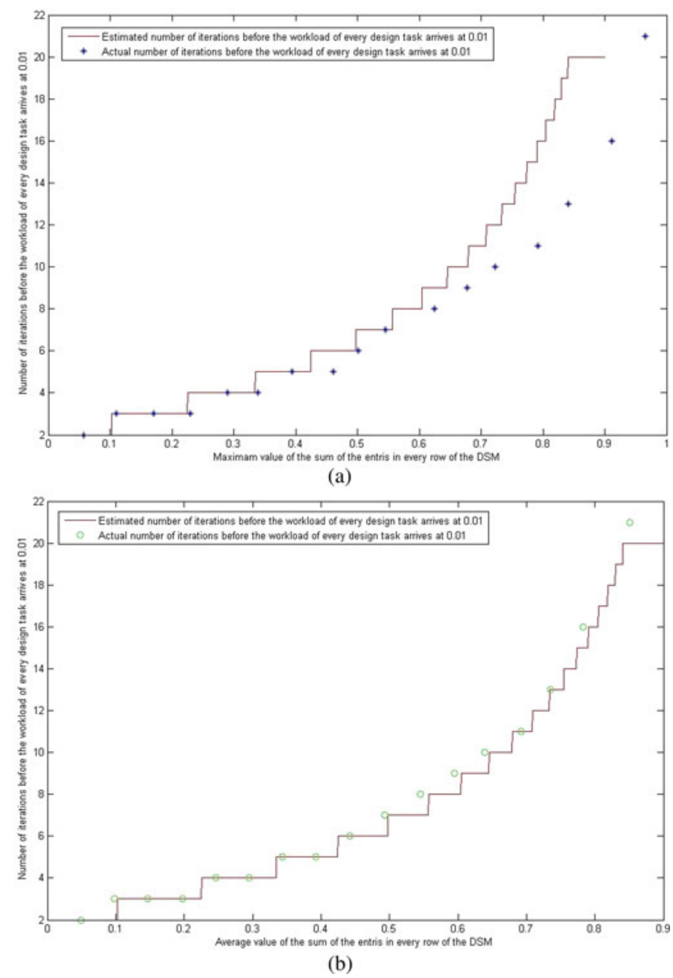

Fig. 7. Iteration estimation of a coupled design process for 100 × 100 DSMs. (a) Iteration estimation by using Theorem 3. (b) Iteration estimation by using Rule 2.

actual numbers of iterations are obtained by simulation, in which the sizes of the DSMs are all 30 × 30. From Fig. 6(a), we see that the actual numbers of iterations are always less than or equal to the estimated numbers of iterations, which supports the correctness of Theorem 3. We can also see that the difference between the actual and estimated number of iterations is small if the maximum value of the sum of the entries in DSM rows is small. However, the difference becomes large if this maximum value becomes large. Therefore, the iterations are usually overestimated by using Theorem 3.

However, if Rule 2 is used to estimate the number of iterations, the difference between the estimated number of iterations and the actual number of iterations becomes smaller. We also generate some 30 × 30 DSMs randomly. The results of these experiments are shown in Fig. 6(b). In Fig. 6(b), the stepwise curve has the same meaning as in Fig. 6(a). "o" represents the actual number of iterations before converging. Fig. 6(b) shows that the difference between the estimated design iterations and the actual design iterations is no more than one for any value of the row sum, although the design iterations are usually underestimated by Rule 2. Fortunately, the estimation error is acceptable and is much smaller than the estimation error when Theorem 3 is applied. We also did numerical experiments for some 100 × 100 DSMs. The results are depicted in Fig. 7, from which we can draw the same conclusion as for the case of the 30 × 30 DSMs.

E. Experiment 5: The Influence of Acceptance Level on the Number of Design Iterations

This experiment is to examine how the acceptance level (i.e., δ) of the convergence of a coupled design process influences the number of design iterations. We solve (7) for different iteration numbers N and acceptance level $\delta_N = \delta$, where $N = 2, 3, \ldots,$ 20, and $\delta_N = \delta = 0.1, 0.2, \ldots, 1.0$. Without loss of generality, we assume that $\beta_1 = \cdots = \beta_I = 1$ holds. Each pair of N and δ_N is corresponding to the maximum or average value of the sum of the entries in every row of the DSM, i.e., the right-hand side of (8) and (9). We plot the relationship between the acceptance level (i.e., δ) and the maximum or average value of the sum of the entries in every row of the DSM for $N = 2, 3, \ldots, 20$ (see Fig. 8). Fig. 8 shows that 1) for a given acceptance level (i.e., δ), the larger the maximum or average value of the sum of the

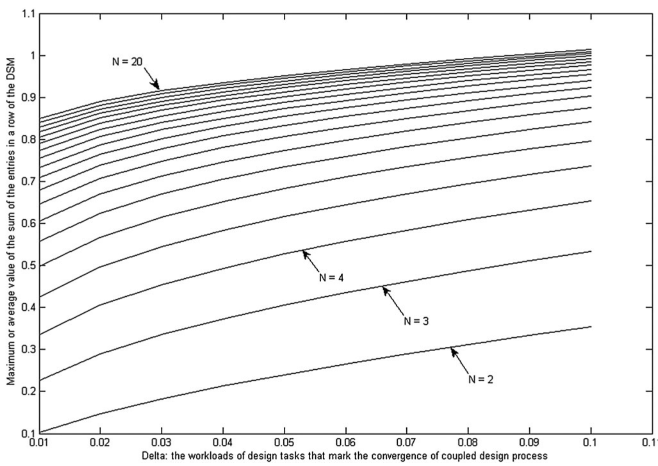

Fig. 8. Acceptance level (δ) of the convergence of a coupled design process versus maximum or average value of the sum of the entries in every row of the DSM.

entries in every row of the DSM, the larger the number of design iterations; and 2) for a given maximum or average value of the sum of the entries in every row of the DSM, larger acceptance levels allow smaller iteration numbers.

V. Conclusion

In this paper, the convergence of a coupled design process with overlap between redesign work has been investigated. To capture this overlap, we model such a coupled design process by a group of nonlinear difference equations instead of the linear model used previously in the literature. Based on the nonlinear model, we derive a sufficiency condition for evaluating convergence of this type of coupled design process. To estimate the number of iterations before converging, we also derive a sufficiency condition for the workload of every design task to reduce to an acceptance level after an expected number of iterations. The results of numerical experiments support the correctness of the two sufficiency conditions.

However, because they are only sufficiency conditions but not necessary ones, incorrect judgment of convergence and overestimation of design iterations usually occur. To improve the accuracy of convergence evaluation and iteration estimation, we also propose two heuristic rules. The first rule is to reduce the possibility of considering a coupled design process that converges as one that does not converge. The second rule is to reduce the difference between the estimated iterations and the actual ones. The results of numerical experiments show that by using the first rule, the number of incorrect judgments for convergence can be reduced greatly; and that by using the second rule, the differences between the estimated iterations and the actual ones are not more than one, although the number of iterations is usually underestimated.

Although the two heuristic rules can improve the accuracy of judging the convergence and estimating the number of iterations, an error still exists. We have not found the necessary conditions of evaluating the convergence of coupled design processes and estimating the number of iterations that are needed for the workload of every design task to reduce to an acceptance level. Those are issues for future research.

This research provides the fundamental theory of managing a coupled design process with overlap between design work. By using the proposed sufficiency conditions and heuristic rules, we can evaluate the convergence of such a coupled design process. Furthermore, we can identify the DSM entries that most strongly influence the convergence of the coupled design process, and then regulate the strength of the interdependence between design tasks to ensure the convergence of the design process. By using the proposed sufficiency conditions and the heuristic rules, we can also estimate the number of design iterations so as to evaluate the convergence rate of such a coupled design process. Furthermore, by regulating the strength of the interdependence between design tasks, we can achieve the expected convergence rate and then arrive at a reliable schedule for the product design project. These issues, i.e., identification of the most influential DSM entries and regulation of the strength of the interdependence between design tasks, are of great interest and left for future research.

Appendix

A. *Proof of Theorem 1*

Equation (1) can be rewritten as

$$\boldsymbol{A}\boldsymbol{x}(n) = -\left\{ \frac{1}{\beta_1} \ln\left[1 - \beta_1 x_1(n+1)\right], \ldots, \frac{1}{\beta_I} \ln \times \left[1 - \beta_I x_I(n+1)\right] \right\}^T \tag{A1}$$

where $\boldsymbol{x}(n) = [x_1(n), \ldots, x_I(n)]^T$. Construct a Lyapunov function of this dynamic system

$$W(\boldsymbol{x}(n)) = \boldsymbol{x}^T(n)\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}(n). \tag{A2}$$

Obviously, $W(\boldsymbol{x}(n))$ is positive definite. Its rate of change is

$$\Delta W(\boldsymbol{x}(n)) = W(\boldsymbol{x}(n)) - W(\boldsymbol{x}(n-1)) = \boldsymbol{x}^T(n)\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}(n) - \sum_{i=1}^{I} \frac{1}{\beta_i^2} \left\{\ln\left[1 - \beta_i x_i(n)\right]\right\}^2. \tag{A3}$$

According to the Lyapunov stability theorem, if the function $\Delta W(\boldsymbol{x}(n))$ is negative definite, then the nonlinear system (1) is asymptotically stable at the equilibrium state $(x_1, \ldots, x_I)^T = (0, \ldots, 0)^T$. Let

$$G(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - \sum_{i=1}^{I} \frac{1}{\beta_i^2} \left\{\ln(1 - \beta_i x_i)\right\}^2. \tag{A4}$$

Then, we have

$$\nabla G(\boldsymbol{x}) = 2\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} + 2\left\{ \frac{\ln(1 - \beta_1 x_1)}{\beta_1(1 - \beta_1 x_1)}, \ldots, \frac{\ln(1 - \beta_I x_I)}{\beta_I(1 - \beta_I x_I)} \right\} \tag{A5}$$

and

$$\nabla^2 G(\boldsymbol{x}) = 2\boldsymbol{A}^T\boldsymbol{A} + 2\mathrm{diag}\left\{ \frac{\ln(1 - \beta_1 x_1) - 1}{(1 - \beta_1 x_1)^2}, \ldots, \times \frac{\ln(1 - \beta_I x_I) - 1}{(1 - \beta_I x_I)^2} \right\}. \tag{A6}$$

Because $\boldsymbol{A}^T\boldsymbol{A}$ is symmetric, there exists a orthogonal matrix such that $\boldsymbol{U}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{U} = \boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_I\}$, where $\lambda_1, \ldots, \lambda_I$ are the eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_I)$ is the matrix composed of the eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_I$ of $\boldsymbol{A}^T\boldsymbol{A}$. So (A6) can be rewritten as

$$\nabla^2 G(\boldsymbol{x}) = 2\boldsymbol{U}\text{diag}\left\{\lambda_1 + \frac{\ln(1-\beta_1 x_1) - 1}{(1-\beta_1 x_1)^2}, \ldots, \lambda_I + \frac{\ln(1-\beta_I x_I) - 1}{(1-\beta_I x_I)^2}\right\}\mathbf{U}^T. \quad \text{(A7)}$$

Because $[\ln(1-\beta x) - 1]/(1-\beta x)^2 \leq -1$ for $x \in [0, 1]$ and $\beta \in [0, 1]$, it follows that if $|\text{Re}\lambda_i| < 1$ holds for $i = 1, \ldots, I$, then the matrix $\nabla^2 G(\boldsymbol{x})$ is negative definite. Therefore, $G(\boldsymbol{x})$ is concave.

Now, we prove that $G(\boldsymbol{x})$ must be negative definite given that $\nabla^2 G(\boldsymbol{x})$ is negative definite by contraposition. If $G(\boldsymbol{x})$ is not negative definite, then there must exist an $\boldsymbol{x}^*$ such that $G(\boldsymbol{x}^*) > 0$. Since $G(\boldsymbol{x})$ is concave, we have

$$\begin{aligned} G(\alpha\mathbf{0} + (1-\alpha)\boldsymbol{x}^*) &> \alpha G(\mathbf{0}) + (\mathbf{1}-\alpha)\mathbf{G}(\boldsymbol{x}^*) \\ &= (1-\alpha)G(\boldsymbol{x}^*) > 0 \quad \text{for } 0 < \alpha < 1. \end{aligned} \quad \text{(A8)}$$

On the other hand, because $\nabla^2 G(\boldsymbol{x})$ is negative definite, we have

$$\begin{aligned} G(\alpha\mathbf{0} + (1-\alpha)\boldsymbol{x}^*) &= G(\alpha\mathbf{0}) + (1-\alpha)\nabla G(\mathbf{0})\boldsymbol{x}^* \\ &\quad + \frac{1}{2}(1-\alpha)^2 \boldsymbol{x}^{*T}\nabla G(\theta(1-\alpha)\boldsymbol{x}^*)\,\boldsymbol{x}^* \\ &= \frac{1}{2}(1-\alpha)\boldsymbol{x}^{*T}\nabla^2 G(\theta(1-\alpha)\boldsymbol{x}^*)\,\boldsymbol{x}^* < 0 \end{aligned} \quad \text{(A.9)}$$

where $\theta \in [0, 1]$. Here arises a contradiction. Therefore, $G(\boldsymbol{x})$ must be negative definite. So (1) is asymptotically stable, i.e., the coupled design process captured by (1) converges.

B. Proof of Theorem 2

This theorem can be proved by mathematical induction. First, we try to find the condition that the design workload of every design task is less than δ_{i1} after one design iteration. According to (1), we let $\sum_{j=1}^{I} a_{ij} < -(1/\beta_i)\ln(1-\beta_i\delta_{i1})$, for $i = 1, \ldots, I$. Then, we have

$$\begin{aligned} x_i(1) &= \frac{1}{\beta_i}\left\{1 - \exp\left[-\beta_i \sum_{j=1}^{I} a_{ij}x_j(0)\right]\right\} \\ &= \frac{1}{\beta_i}\left\{1 - \exp\left[-\beta_i \sum_{j=1}^{I} a_{ij}\right]\right\} < \delta_{i1}, \text{for } i = 1, \ldots, I \end{aligned}$$

where $x_1(0) = \cdots = x_I(0) = 1$ is applied.

So

$$\sum_{j=1}^{I} a_{ij} < -\frac{\ln(1-\beta_i\delta_{i1})}{\beta_i\delta_{i0}} = -\frac{1}{\beta_i}\ln(1-\beta_i\delta_{i1}) \quad \text{for } i = 1, \ldots, I$$

is the sufficiency condition for $x_i(1) < \delta_{i1}$, $i = 1, \ldots, I$. Assume that $\sum_{j=1}^{I} a_{ij} < -\frac{1}{\beta_i}\ln(1-\beta_i\delta_{i1})$ and $\sum_{j=1}^{I} a_{ij} < -\frac{\ln(1-\beta_i\delta_{in})}{\beta_i\delta_{i,n-1}}$, for $n = 2, \ldots, N-1$, and $i = 1, \ldots, I$, are the sufficiency conditions for $x_i(1) < \delta_{i1}, \ldots, x_i(N-1) < \delta_{i,N-1}$, for $i = 1, \ldots, I$. Let $\sum_{j=1}^{I} a_{ij} < -\frac{\ln(1-\beta_i\delta_{iN})}{\beta_i\delta_{i,N-1}} = -\frac{\ln(1-\beta_i\delta)}{\beta_i\delta_{i,N-1}}$, then we have

$$\begin{aligned} x_i(N) &= \frac{1}{\beta_i}\left\{1 - \exp\left[-\beta_i \sum_{j=1}^{I} a_{ij}x_j(N-1)\right]\right\} \\ &< \frac{1}{\beta_i}\left\{1 - \exp\left[-\beta_i\delta_{i,N-1} \sum_{j=1}^{I} a_{ij}\right]\right\} \\ &< \delta_{i,N} = \delta \quad \text{for } i = 1, \ldots, I. \end{aligned}$$

Therefore, $\sum_{j=1}^{I} a_{ij} < \min\{-\frac{\ln(1-\beta_i\delta_{in})}{\beta_i\delta_{i,n-1}}, n = 1, \ldots, N\}$, for $i = 1, \ldots, I$, is the sufficiency condition for $x_i(1) < \delta_{i1}, \ldots, x_i(N) < \delta_{iN} = \delta$, for $i = 1, \ldots, I$.

C. Proof of Theorem 3

We prove the first conclusion briefly by contraposition. For a given i, assume that there exist a group of δ_{in}^*, $n = 1, \ldots, N$, that can maximize $\min\{-(\ln(1-\beta_i\delta_{in})/\beta_i\delta_{i,n-1})$, for $n = 1, \ldots, N\}$, but (7) does not hold for δ_{in}^*, $n = 1, \ldots, N$. Then, we can always find an $\tilde{n}$ such that $-(\ln(1-\beta_i\delta_{i\tilde{n}}^*)/\beta_i\delta_{i,\tilde{n}-1}^*) = \min\{-(\ln(1-\beta_i\delta_{in}^*)/\beta_i\delta_{i,n-1}^*)$, for $n = 1, \ldots, N\}$ and $-(\ln(1-\beta_i\delta_{i\tilde{n}}^*)/\beta_i\delta_{i,\tilde{n}-1}^*) < -(\ln(1-\beta_i\delta_{i,\tilde{n}+1}^*)/\beta_i\delta_{i\tilde{n}}^*)$ holds or $-(\ln(1-\beta_i\delta_{i\tilde{n}}^*)/\beta_i\delta_{i,\tilde{n}-1}^*) = \min\{-(\ln(1-\beta_i\delta_{in}^*)/\beta_i\delta_{i,n-1}^*)$, for $n = 1, \ldots, N\}$ and $-(\ln(1-\beta_i\delta_{i\tilde{n}}^*)/\beta_i\delta_{i,\tilde{n}-1}^*) < -(\ln(1-\beta_i\delta_{i,\tilde{n}-1}^*)/\beta_i\delta_{i,\tilde{n}-2}^*)$ holds.

According to the monotonicity of the functions $-\ln(1-x)/a$ and $-\ln(1-b)/x$ for fixed a and b, the x maximizing $\min\{-(\ln(1-x)/a), -(\ln(1-b)/x)\}$ satisfies $\ln(1-x)/a = \ln(1-b)/x$. Therefore, we can always find another $\delta_{i\tilde{n}}^{**}$ that satisfies $-\ln(1-\beta_i\delta_{i\tilde{n}}^{**})/\beta_i\delta_{i,\tilde{n}-1}^* = -\ln(1-\beta_i\delta_{i,\tilde{n}+1}^*)/\beta_i\delta_{i\tilde{n}}^{**}$ and $-\ln(1-\beta_i\delta_{i\tilde{n}}^{**})/\beta_i\delta_{i,\tilde{n}-1}^* > -\ln(1-\beta_i\delta_{i\tilde{n}}^*)/\beta_i\delta_{i,\tilde{n}-1}^*$, or another $\delta_{i,\tilde{n}-1}^{**}$ that satisfies $-\ln(1-\beta_i\delta_{i\tilde{n}}^*)/\beta_i\delta_{i,\tilde{n}-1}^{**} = -\ln(1-\beta_i\delta_{i,\tilde{n}-1}^{**})/\beta_i\delta_{i,\tilde{n}-2}^*$ and $-(\ln(1-\beta_i\delta_{i,\tilde{n}-1}^{**})/\beta_i\delta_{i,\tilde{n}-2}^*) > -(\ln(1-\beta_i\delta_{i\tilde{n}}^*)/\beta_i\delta_{i,\tilde{n}-1}^*)$. For the aforementioned two situations, we can substitute $\delta_{i\tilde{n}}^*$ by $\delta_{i\tilde{n}}^{**}$ or $\delta_{i,\tilde{n}-1}^*$ by $\delta_{i,\tilde{n}-1}^{**}$. Then, $-\ln(1-\beta_i\delta_{i\tilde{n}}^*)/\beta_i\delta_{i,\tilde{n}-1}^*$ is not the minimum value. This contradiction happens because we assume that (7) does not hold. So the first conclusion is correct. The second conclusion can be drawn directly according to Theorem 2 and the first conclusion.

Acknowledgment

The authors would like to thank the anonymous reviewers and the editors for their comments and suggestions on this paper.

References

[1] J. Sobieszczanski-Sobieski, J. S. Agte, and R. R. Sandusky, Jr., "Bi-level integrated system synthesis (BLISS)," NASA Langley Res. Center, Hampton, VA, Tech. Rep. NASA/TM-1998-208715, 1998.

[2] J. A. Reyer and P. Y. Papalambros, "Combined optimal design and control with application to an electric DC motor," *J. Mech. Design*, vol. 124, pp. 183–191, 2002.
[3] R. P. Smith and S. D. Eppinger, "Identifying controlling features of engineering design iteration," *Manage. Sci.*, vol. 43, no. 3, pp. 276–293, 1997.
[4] S. Thomke and D. E. Bell, "Sequential testing in product development," *Manage. Sci.*, vol. 47, no. 2, pp. 308–323, 2001.
[5] I. R. Chittick and J. R. R. A. Martins, "Aero-structural optimization using adjoint coupled post-optimality sensitivities," *Struct. Multidisciplinary Optimization*, vol. 36, no. 1, pp. 59–70, 2008.
[6] J. Sobieszczanski-Sobieski and R. T. Haftka, "Multidisciplinary aerospace design optimization: Survey of recent developments," *Struct. Optimization*, vol. 14, pp. 1–23, 1997.
[7] R. V. Tappeta and S. Nagendra, "A multidisciplinary design optimization approach for high temperature aircraft engine components," *Struct. Optimization*, vol. 18, pp. 134–145, 1999.
[8] Z. Wang, Q. Chen, J. Hu, and H. Huang, "A proportional–integral control policy of two coupled concurrent design optimization processes," *Concurrent Eng.: Res. Appl.*, vol. 18, no. 1, pp. 55–63, 2010.
[9] I. Y. Kim and O. de Weck, "Adaptive weighted sum method for multiobjective optimization, structural and multidisciplinary optimization," *Struct. Multidisciplinary Optimization*, vol. 31, no. 2, pp. 105–116, 2006.
[10] C. D. McAllister, T. W. Simpson, K. Hacker, K. Lewis, and A. Messac, "Integrating linear physical programming within collaborative optimization for multi-objective multidisciplinary design optimization," *Struct. Multidisciplinary Optimization*, vol. 29, pp. 178–189, 2005.
[11] D. V. Steward, "The design structure system: A method for managing the design of complex systems," *IEEE Trans. Eng. Manage.*, vol. 28, no. 3, pp. 71–74, Aug. 1981.
[12] M. S. Avnet and A. L. Weigel, "An application of the design structure matrix to integrated concurrent engineering," *Acta Astronaut.*, vol. 66, pp. 937–949, 2010.
[13] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: A review and new directions," *IEEE Trans. Eng. Manage.*, vol. 48, no. 3, pp. 292–306, Aug. 2001.
[14] J. R. MacCormack and C. Y. Baldwin, "Exploring the structure of complex software design: An empirical study of open source and proprietary code," *Manage. Sci.*, vol. 52, no. 7, pp. 1015–1030, 2006.
[15] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "Identifying modular and integrative systems and their impact on design team interactions," *ASME J. Mech. Design*, vol. 125, pp. 240–252, 2003.
[16] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala, "A model-based method for organizing tasks in product development," *Res. Eng. Design*, vol. 6, no. 1, pp. 1–13, 1994.
[17] A. Kusiak, N. Larson, and J. Wang, "Reengineering of design and manufacturing processes," *J. Comput. Ind. Eng.*, vol. 26, no. 3, pp. 521–536, 1994.
[18] D. A. Batallas and A. A. Yassine, "Information leaders in product development organizational networks: Social network analysis of the design structure matrix," *IEEE Trans. Eng. Manage.*, vol. 53, no. 4, pp. 570–582, Nov. 2006.
[19] M. Sosa, S. D. Eppinger, and C. Rowles, "The misalignment of product architecture and organizational structure in complex product development," *Manage. Sci.*, vol. 50, pp. 1674–1689, 2004.
[20] R. P. Smith and S. D. Eppinger, "A predictive model of sequential iteration in engineering design," *Manage. Sci.*, vol. 43, no. 8, pp. 1104–1120, 1997.
[21] C. Meier, A. A. Yassine, and T. R. Browning, "Design process sequencing with competent genetic algorithms," *ASME J. Mech. Design*, vol. 129, pp. 566–585, 2007.
[22] N. R. Joglekar, A. A. Yassine, S. D. Eppinger, and D. Whitney, "Performance of coupled product development activities with a deadline," *Manage. Sci.*, vol. 47, no. 12, pp. 1605–1620, 2001.
[23] J. Mihm, C. Loch, and A. Huchzermeier, "Problem-solving oscillations in complex engineering projects," *Manage. Sci.*, vol. 49, no. 6, pp. 733–750, 2003.
[24] A. Yassine, N. Joglekar, D. Braha, S. Eppinger, and D. Whitney, "Information hiding in product development: The design churn effect," *Res. Eng. Design*, vol. 14, pp. 145–161, 2003.
[25] D. Braha and Y. Bar-Yam, "The topology of large-scale engineering problem-solving networks," *Phys. Rev. E*, vol. 69, no. 1, pp. 016113-1–016113-7, 2004.
[26] D. Braha and Y. Bar-Yam, "Information flow structure in large-scale product development organizational networks," *J. Inf. Technol.*, vol. 19, no. 4, pp. 244–251, 2004.
[27] D. Braha and Y. Bar-Yam, "The statistical mechanics of complex product development: Empirical and analytical results," *Manage. Sci.*, vol. 53, no. 7, pp. 1127–1145, 2007.
[28] S. H. Cho and S. D. Eppinger, "A simulation-based process model for managing complex design projects," *IEEE Trans. Eng. Manage.*, vol. 52, no. 3, pp. 316–328, Aug. 2005.
[29] H. M. E. Abdelsalam and H. P. Bao, "A simulation-based optimization framework for product development cycle time reduction," *IEEE Trans. Eng. Manage.*, vol. 53, no. 1, pp. 69–85, Feb. 2006.
[30] A. Karniel and Y. Reich, "From DSM-based planning to design process simulation: A review of process scheme logic verification issues," *IEEE Trans. Eng. Manage.*, vol. 56, no. 4, pp. 636–649, Nov. 2009.
[31] M. Carrascosa, S. D. Eppinger, and D. E. Whitney, "Using the design structure matrix to estimate a product development time," in *Proc. ASME Design Autom. Conf.*, Atlanta, GA, 1998, pp. 1–10.

Zheng Wang received the B.S. degree in mechanical engineering from the Changchun Institute of Optics and Fine Mechanics, Changchun, China, in 1993, and the M.S. and Ph.D. degrees in control theory and control engineering from Southeast University, Jiangsu, China, in 1998 and 2001, respectively.

He is currently an Associate Professor in the School of Automation, Southeast University. From February 2009 to January 2010, he was a Visiting Associate Professor at Engineering Systems Division, Massachusetts Institute of Technology, Cambridge. He has published papers on international refereed journals such as IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, *European Journal of Operational Research*, *International Journal of Production Research*, *Journal of Intelligent Manufacturing*, *Concurrent Engineering: Research and Applications, and International Journal of Advanced Manufacturing Technology*. His research interests include modeling and analysis of product design process, and manufacturing systems control.

Dr. Wang is a member of the Institute for Operations Research and the Management Sciences.

Christopher L. Magee received the B.S., M.S., and Ph.D. degrees in metallurgy and materials science from the Carnegie Institute of Technology (now Carnegie Mellon University), Pittsburgh, PA, and the M.B.A. degree from Michigan State University, East Lansing.

He is currently Professor of the Practice, Engineering Systems Division and the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, and the Co-Director of International Design Centre, Singapore University of Technology and Design, Changi, Singapore, being codeveloped by MIT and Singapore. His recent research has emphasized innovation and technology development in complex systems.

Dr. Magee was elected to the National Academy of Engineering while with Ford Motor Company for contributions in advanced vehicle development in 1997. He was a Ford Technical Fellow in 1996 and a Fellow of the American Society for Materials.