RESEARCH ON PRODUCT DEVELOPMENT ITERATIONS BASED ON FEEDBACK CONTROL THEORY IN A DYNAMIC ENVIRONMENT

RENBIN XIAO¹, TINGGUI CHEN² AND CHUNHUA JU²

¹Institute of Systems Engineering
Huazhong University of Science and Technology
Wuhan, 430074, P. R. China
rbxiao@mail.hust.edu.cn

²College of Computer Science and Information Engineering
Zhejiang Gongshang University
Hangzhou, 310018, P. R. China
ctgsimon@gmail.com; juchunhua@hotmail.com

Received January 2010; revised June 2010

ABSTRACT. The complicated interactions among tasks lead to iterations in product development and unexpected disturbances may destabilize the entire product development process. This paper makes an investigation on disposing product development iterations in a dynamic environment. Firstly, the uncertain factors, such as task durations, output branches of tasks, and resource allocations, existing in product development are discussed. Secondly, a satisfaction degree-based feedback control approach is put forward. Such an approach includes two scenarios: identifying a satisfaction degree and monitoring and controlling of iteration process. The former is used to represent the accepted iterative development process and the latter is realized by a resource regulating matrix based on feedback control theory. Finally, an example of a crane development project is provided to illustrate and verify the proposed approach.

Keywords: Design structure matrix (DSM), Feedback control theory, Fuzzy satisfaction degree, Product development iteration

1. Introduction. Product development and innovation has become a key link to obtain competitive advantage for modern enterprises in recent years. To maintain the market share, enterprises should effectively manage their product development project and bring their products to market as early as possible. However, as we all know, the interdependencies among development tasks give rise to complex information flows as the execution of a development task may create new information or conditions that affect other interdependent tasks. As information flow is also random, and all the coupled tasks are executed concurrently, development decision made using incomplete or imperfect information are revisited in what is termed development iteration [1]. Osborne [2] found that iteration accounted for between one third and two thirds of total development time for projects at a major semiconductor producer. He also found that unpredictable iteration is the main cause of variability in the lead times of projects at this firm. Therefore, it is necessary to explore iterative development process. Moreover, in practice, each product development project is usually unique in nature [3]. It often encounters situations where the duration of particular tasks cannot be given precisely at the project initialization stage. Unless a product development project being scheduled is similar to previous projects, previous experience is of limited relevance [4]. It means that the durations of tasks are of uncertainty at the initial stage of product development process. In addition, when a
certain task finishes, one or more successive tasks will be triggered or some preceding ones will be performed again. It also means output branches of tasks are of uncertainty [5]. As a result, uncertain and dynamic development environment mentioned above often exists which will seriously threaten to finish the entire development process successfully. In particularly, uncertainties associated with development iterations will further lead to instability of the entire development process. Thus, the motivation of the paper is to use feedback control method to tackle development iterations and achieve a satisfied stability of the entire development process in a dynamic environment.

Currently, traditional representations of product development process, such as digraph, Critical Path Method (CPM), Program Evaluation and Review (PERT), cannot model development iterations. Due to compact matrix form, design structure matrix (DSM) has widely applied to product design and development [6] such as development project management [7], design optimization [8], tasks sequencing [9], product configuration [10] and so on. It is also a useful tool for analyzing highly complicated dependencies, inclusive of feedback and coupled tasks [11]. Owing to these reasons mentioned above, in this paper, we used DSM and its extension form named work transformation matrix (WTM) [12] to put forward an approach to analysis of development iterations based on feedback control theory. The paper is organized as follows: in Section 2, we survey the existed literatures on disposal of coupling. Section 3 analyzes uncertainties existing in product development process. Section 4 proposes an approach to analysis of development iteration based on feedback control theory. Section 5 applies the approach to a product development example from a crane manufacturer. Section 6 offers the conclusions in this paper and some perspectives of the future research.

2. Related Works. Today’s dynamic development environment is characterized by sudden disturbances which threaten to undermine the stability of the entire development process. External disturbances such as improvement in technologies and changing customer preferences may necessitate a re-examination of the entire development process so as to reduce cost and shorten the lead time. Due to these reasons, resources have to be re-allocated among various development tasks in order to achieve the goals. In literature [13], the authors presented a model for solving coupled task sets based on resource leveling strategy. However, it is hypothesized that once resources allocated to coupled task sets are ascertained, then in all iterations’ process, they no longer changes. It dose not exactly accord with the real product development process. So how to establish a model reflecting both features of dynamic allocation of resources and uncertain factors existed in development process requires further research. Furthermore, some researchers introduced feedback control theory to analyze product development process. For instance, Huang and Gu [14] viewed a product development process as a dynamic system with feedback based on the feedback control theory and used DSM to capture the interaction and feedback of development information. Nevertheless, the authors also pointed out that the insufficiency of the process dynamic analysis as well as the impropriety of both the process evaluation and the information feedback increased the instability of a system to a great extent. Ong et al. [15] adopted the state-space concept of modern control theory to model and understand the iteration process. The model was developed using the state-space concept to analyze the stability and convergence rate of convergent development tasks, and to identify the most slowly converging task. However, unfortunately, few extant studies fully utilize the well-developed analysis techniques of control engineering with DSM representation. Some of the results are even misleading and confusing due to inadequate adoption of the analysis techniques. As a consequence, Kim [16] developed a systematic representation of the work transformation matrix method, with a discrete state-space description of the
development process. With this representation, the dynamics of the development process can be easily investigated and predicted, using well-established discrete system analysis and control synthesis techniques. Moreover, Lee et al. [17] developed non-homogenous and homogeneous state-space concepts, where non-homogenous one monitored and controlled the stability and the convergence rate of development tasks and at the same time predicted the number of development iterations; homogenous one did not consider external disturbances and its response was only due to initial conditions. However, all these researches mentioned above didn’t consider uncertain factors existing in the development iteration process, such as task durations, output branches of tasks and so on. These uncertainties often make product development process uncontrollable and bring much more additional cost and prolong the lead time. Thus, further researches are necessary in order to consider the disturbances from external environments.

In this paper, we take coupled development tasks as an object to investigate the effects of uncertainty factors on development iterations based on feedback control theory. In addition, an example deriving from a jib-bridge luffing system of a crane is used to illustrate the detailed applications of the method in a dynamic environment.

3. Analysis of Uncertainties in Product Development Process. In this section, three uncertain factors, including uncertainty of task durations, uncertainty of output branches of tasks and uncertainty of resource allocations which are existed in a dynamic environment, are investigated respectively. At the same time, their effects on development process are also studied.

3.1. Uncertainty of task durations. Because task durations are affected by customer requirements, personal experiences, development resources, and so on, it is difficult to limit them to precise values. In general, three-point estimate method is adopted, viz., an optimistic or best case value “a”, a most likely value “m”, and a pessimistic or worst case value “b”. In doing so, we can obtain a normal distribution function whose mean $\mu$ and variance $\sigma^2$ can be expressed as $\mu = (a + 4m + b)$ and $\sigma^2 = (b - a)^2/6$ respectively [18]. Its probability density $y_d$ and cumulative probability density $y_F$ can be expressed as $y_d = f(x | \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ and $y_F = F(x | \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$ respectively.

If we select a random number produced in an interval between 0 and 1 to represent $y_F$, variable $x$ satisfied to formula (1) may be taken as a sample value of task duration. The meaning of $x$ can be shown in Figure 1(a).

$$x = F^{-1}(y_F | \mu, \sigma) = \{ x : F(x | \mu, \sigma) = y_F \} \tag{1}$$

In practice, engineers are able to estimate a possible confidence interval of task durations by their experiences. Therefore, we can select satisfaction value $\beta$ less than 1 to make $y_F$ between $\alpha$ and $1 - \alpha$ as shown in Figure 1(b). It is helpful to accelerate the convergence of iteration process. At this moment, $y_F$ can be determined by formula (2).

$$y_F = \alpha + (1 - 2\alpha) \times \text{rand}() \tag{2}$$

3.2. Uncertainty of output branches of tasks. Uncertainty of output branches of tasks is caused by repetition probability. Smith and Eppinger proposed two models: a sequential iteration model [19] and a parallel iteration model [12]. The first one assumes coupled development tasks are executed one after the other and the rework is governed by a probabilistic rule. The second one assumes coupled development tasks are all executed in parallel and iteration is governed by a linear rework rule. However, these two models cannot reflect all iteration phenomena in product development process at all because of
complexity of iteration process. For example, in Figure 2, it includes three matrices: rework amount matrix \( RA \), repetition probability matrix \( RP \) and duration matrix \( DM \) where rework amount matrix gives rise to the transfer of work involved in the iterations, and repetition probability matrix represents the probability that a certain task will be repeated in the next iteration. In each iteration stage, there are one or more tasks executed by a probabilistic rule. Under this situation, the iteration process belongs to neither sequential iteration nor parallel iteration. It means output branches of tasks are of uncertainty.

\[
\begin{align*}
\text{RA}^{(t+1)}(i, j) &= RA(i, j) & \text{if } V^{(t+1)}(j) = 1 \text{ and } RP(i, j) \geq \text{rand}() \\
\text{RA}^{(t+1)}(i, j) &= 0 & \text{others}
\end{align*}
\]

where \( RA^{(t+1)} \) denotes rework amount matrix composed of executive tasks. It means that if current state of task \( j \) is executive state (i.e., \( V^{(t+1)}(j) = 1 \)) and its successive tasks will be repeated (i.e., \( RP(i, j) \geq \text{rand}() \)), we can get \( RA^{(t+1)}(i, j) = RA(i, j) \), otherwise \( RA^{(t+1)}(i, j) = 0. \)
3.3. **Uncertainty of resource allocations.** Uncertainty of resource allocations lies in uncertainty of both task durations and output branches of tasks. On the one hand, owing to uncertainty of task durations, resources allocated among various tasks are only determined according to past experiences. On the other hand, due to uncertainty of output branches of tasks, there are different executive states in different iteration stage for all tasks, which will cause hybrid iteration mode. All these reasons lead to instability of the entire product development process.

However, managers usually expect that the makespan of the entire project can be fluctuated in a small zone. Excessive disturbances make it difficult to predict the development cost, and monitor and control the entire development process. Therefore, an automatic regulating device should be developed to regulate all kinds of uncertainties existing in the process. From this viewpoint, we introduce an appropriate regulating matrix $K$ to monitor and control disturbances in a system based on feedback control theory. It is defined as the degree of influence of other tasks on the resource usage of a task. After introducing the matrix $K$, formula (3) can be revised as

$$u(t + 1) = RA^{(t+1)}u(t) - BKu(t)$$

where the matrix $B$ represents the proportion of common resources shared by two or more tasks. If we assume that resources are not shared among the various tasks, $B$ is a unit matrix. We can see from formula (5) that extra resources are required to make the makespan of the entire development project approach to an expected value.

4. Feedback Control Approach to Analysis of Product Development Iteration. In this section, we will firstly introduce resource regulating matrix $K$ and set up analytical model of product development iteration based on feedback control. Then, the calculation principle and method of the matrix $K$ is illustrated under different dynamic environments. Finally, implementation steps of the approach to iteration analysis are given.

4.1. **The analytical model of development iteration based on feedback control theory.** In this section, the following four assumptions are made before feedback control model used to analyze development iteration is presented.

1. All tasks are repeated by a constant probability. In addition, rework performed is a function of the work done in the previous iteration stage.
2. Resources are sufficient and none of tasks will be delayed because of resource constraints.
3. All tasks are executed concurrently in the first iteration stage. After that, tasks are done by a probabilistic selection in the following iteration stages.
4. Overlapping [20], time-window [21] and other dependencies among tasks are not considered.

The first assumption is convenient from a mathematical standpoint but more difficult to argue from an empirical perspective. However, if the number of iterations is not very large, the changes of parameters over time haven’t a great impact on the tasks. The second assumption avoids the lag phenomenon caused by resource conflicts in order to make successive tasks be executed randomly. The third assumption assures that all tasks may be executed at least one time. The fourth assumption ignoring the effects of both overlapping and time-window dependencies among tasks on the makespan is an idealization of an observation we have made concerning many development projects. Based on these assumptions, feedback control model can be built as follows.

According to discussion in section 3, uncertainties of both task durations and output branches of tasks can be taken as disturbance factors in a system. The matrix $K$ can
be used as feedback gain matrix to regulate dynamic property of the system in order to
achieve the desired stability. As we all know, the magnitude of the maximum pole can
ascertain systemic stability. So, we may carry out pole assignment of a system through
selecting the appropriate regulating matrix $K$ (also called feedback gain matrix). In doing
so, on the one side, make the system reach stabilization; on the other side, make iteration
process converge fast. The feedback closed-loop system model is shown in Figure 3.

![Figure 3. Feedback closed-loop system model](image)

In Figure 3, $\eta(t)$ denotes an input disturbing signal caused by the uncertainty of task
durations. A matrix $B$ is a control or input matrix mentioned in Section 3.3. A matrix
$A(t)$, reflecting both rework amount and repetition probability, is a state matrix and
determined by formula (4). It is easy to find that it changes over time. In addition, a
matrix $C$ is an output matrix. It reflects execution times of all tasks in every iteration
stage. A matrix $K$ is feedback gain matrix, which is used to regulate the disturbing
signals caused by $\eta(t)$ and $A(t)$. A matrix $D$ is a feedforward matrix. Because disturbing
signal of durations, $\eta(t)$, does not directly affect output signal, $y(t)$, the matrix $D$ is a null matrix. According to principles of automatic control, the stability of the closed-loop
system depends on the eigenvalues of the matrix $(A(t)-K)$. If we select an appropriate
matrix $K$, the desirable performance of the system can be achieved. As can be seen from
further research, in Figure 3, the state of the system at the $(k+1)$th iteration stage,
$u(k+1)$, is a function of the preceding state $u(k)$ and the matrix $K$. When the state
of the preceding iteration stage $k$, $u(k)$, deviates from the expected value owing to some
external disturbances including task durations and execution modes, the matrix $K$ can
be used to cope with this deviation.

In practice, it is not necessary or possible to limit the makespan of development projects
to a precise value. Generally, it is satisfying when the makespan locates in a feasible in-
terval. Therefore, we introduce a satisfaction degree to describe this situation. However,
because evaluation process is usually random, fuzzy and uncertain, membership function
based on a satisfaction degree is used to express this condition. It is important to select
suitable membership function to express fuzzy concept. For various problems, member-
ship function curves have a lot of different kinds of shapes, such as triangle, trapezium,
campaniform, bar, and so on, as shown in Figure 4. Owing to simple shapes, convenient
understanding and computing and excellent description features, triangle and trapezium
membership function curves are widely used. So, in this paper, we will adopt these two
shapes to discuss the stability of the entire development project.

Take a camera development process as an example. This development process com-
prises four coupled development tasks, such as development shutter mechanism (task $C$),
development viewfinder (task $D$), development camera body (task $E$) and development
film mechanism (task $F$). If none of task is executed after the first iteration stage, the
minimum makespan $M_{\text{min}}$ of the iteration process appears. On the contrary, if all tasks
are executed concurrently in every iteration stage, the maximum makespan $M_{\text{max}}$ appears. We can achieve the range of fluctuation of the makespan $M (M_{\text{min}} \leq M \leq M_{\text{max}})$. Generally, without losing design quality, less iteration time is desirable due to market competition. Therefore, we can give a feasible range of makespan which denotes the accepted iterative development process in the specific situation. Here, $\gamma$ is introduced to represent a tolerance of satisfaction degree $M$. Figure 5 illustrates the satisfaction degree of a camera development. In Figure 5, we hypothesize that $M_{\text{min}}$ is equal to 19 units and $M_{\text{max}}$ 64.5 units. Moreover, the acceptable makespan of the iteration process is no more than 40 units, so the value of $M_{\text{accept}}$ in Figure 5 corresponds to 40. That is to say, the makespan greater than 40 is unacceptable.

4.2. Calculation of the Regulating Matrix $K$. In Section 4.1, we mention that the resource regulating matrix $K$ can be used to assign the poles of a system in order to achieve the stability under desirable states. However, because of many different environments the system exists in, how to correctly select appropriate matrix $K$ is very critical. Therefore, the calculation principle of the regulating matrix $K$ will be specially introduced in this section.

Firstly, the analytical model of development iterations based on feedback control theory is built according to the method mentioned in Section 4.1. Besides, related repetition matrix $RP$, rework amount matrix $RA$ and duration matrix $DM$ of coupled task set, corresponding to the model, are determined at the same time. Subsequently, through analysis and investigation, both the accepted satisfaction degree of development iteration process and its convergence conditions should be ascertained, where the convergence of iteration process has to subject the following two conditions: 1. development iteration process should converge to the range of the accepted satisfaction degree $M_{\text{accept}}$. 2. a task is considered to be no longer iterated when the remaining work, so-called rework amount, is no more than $\varepsilon$ (in general, $\varepsilon$ is no more than 0.1) of the original. The regulating effects of the model lies in selecting both appropriate matrix $K$ and the number of iterations $n$ to subject the condition of satisfaction degree as well as convergence condition of task rework as can be shown in formulae (6) and (7). At last, according to the matrix $K$, detailed allocative decision of resources can be drawn up.

\[
\sum_{0}^{n} \max_{i}[DM(RA^{(t+1)} - K)^{n}u_{0}]^{(i)} \leq \sum_{0}^{n} \max_{i}[DM(RA - K)^{n}u_{0}]^{(i)} \leq M_{\text{accept}} \tag{6}
\]

\[
\max_{i}[(RA - K)^{n}u_{0}]^{(i)} \leq \varepsilon \tag{7}
\]

Nevertheless, if the scale of tasks in coupled set is very large, there possibly exist many solutions of $n$ and matrix $K$ satisfying formulae (6) and (7) because the matrix $K$ includes
many parameters. At the moment, considering the magnitude of the maximum pole can ascertain systemic stability, we can only regulate the magnitude of the maximum eigenvalue of the system state matrix to make iteration process converge faster. As a result, we can assume that the matrices $RA$ and $K$ have the same eigenvectors and eigenvalues except for the maximum eigenvalue. Furthermore, according to related knowledge of matrix theory, if the matrix has linearly independent eigenvectors (the eigenvector matrix is invertible), then the matrix $RA$ can also be expressed as

$$RA = SAS^{-1}$$  \(\text{(8)}\)

where $S$ is the eigenvector matrix of the matrix $RA$.

Because both the matrix $K$ and the matrix $RA$ have the same eigenvectors and eigenvalues except for the maximum eigenvalue, if we let the maximum eigenvalues of matrix $RA$ and matrix $K$ be $\lambda_{max}$ and $\lambda_{max}^*$ respectively, then $(RA - K)$ can be illustrated as formula $(9)$ according to $K = SA^*S^{-1}$.

$$(RA - K) = (SAS^{-1} - SA^*S^{-1}) = [S(\Lambda - \Lambda^*)S^{-1}] = S\begin{bmatrix} \lambda_{max} - \lambda_{max}^* & 0 \\ 0 & 0 \end{bmatrix}S^{-1}$$ \(\text{(9)}\)

Adopting formula $(9)$ to simplify the matrix $K$, we can achieve the minimum solutions of both the matrix $K$ and the number of iterations $n$ satisfying formulae $(6)$ and $(7)$.

In order to explain the calculation process of the matrix $K$ in detail, take the camera development process as an example mentioned in Section 4.1. Its rework amount matrix $RA$ can be shown as follows. The rows represent the development tasks of development shutter mechanism (task $C$), development viewfinder (task $D$), development camera body (task $E$) and development film mechanism (task $F$), respectively. Each column also corresponds to the tasks in the same order.

$$RA = \begin{bmatrix} 0 & 0.1 & 0.2 & 0.3 \\ 0.3 & 0 & 0.4 & 0.2 \\ 0.1 & 0.3 & 0 & 0.5 \\ 0.1 & 0.1 & 0.2 & 0 \end{bmatrix}$$

At the initial iteration stage, the initial work vector, $u_0$, is a column vector of ones and can be shown as $u_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$. The satisfaction degree curve of the iteration process is shown in Figure 5. If the makespan, $M$, is greater than $M_{accept}$ ($M_{accept}=40$), feedback control is employed. The regulating matrix $K$ is computed as:

$$\sum_0^n \max_i [DM(RA^{(t+1)} - K)^nu_0]^{(i)} \leq \sum_0^n \max_i [DM(RA - K)^nu_0]^{(i)} \leq 40 \quad \text{(10)}$$

where $RA$ is a rework amount matrix and $RA^{(t+1)}$ is calculated by formula $(4)$. $DM$ is a duration matrix whose diagonal elements are taken by formulae $(1)$ and $(2)$. $[\bullet]^{(i)}$ is the $i$th element of the vector within the brackets. The symbol $n$ is the terminated stage of iterations. In this example, if we assume that a task is considered to be terminated when the remaining work is 10% of the original, $n$ is computed as follows.

$$\max_i [(RA - K)^nu_0]^{(i)} \leq 0.1 \quad \text{(11)}$$

where $u_0$ is an initial column vector of ones. $[\bullet]^{(i)}$ is the same as formula $(10)$. 
The eigenvector matrix $S$ and the eigenvalue diagonal matrix $\Lambda$ of the matrix $RA$ can be expressed as:

$$
\Lambda = \begin{bmatrix}
0.674 & -0.392 & -0.141 + 0.060i \\
-0.141 - 0.060i & -0.141 + 0.060i & -0.141 - 0.060i \\
0.410 & -0.067 & 0.657 & 0.657 \\
0.624 & -0.613 & 0.060 - 0.570i & 0.060 + 0.570i \\
0.580 & 0.758 & 0.0395 + 0.073i & -0.0395 - 0.073i \\
0.326 & 0.213 & -0.065 + 0.274i & -0.065 - 0.274i \\
\end{bmatrix}
$$

$$
S = \begin{bmatrix}
0.674 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
$$

However, because the matrix $K$ includes 16 parameters, according to formulae (10) and (11), there exist many solutions. According to the principle mentioned above, the maximum eigenvalue of the matrix $RA$ is equal to 0.674 and $(RA - K)$ can therefore simplify to:

$$(RA - K) = (S \Lambda S^{-1} - S \Lambda^* S^{-1}) = [S(\Lambda - \Lambda^*)S^{-1}] = S \begin{bmatrix}
0.674 - \lambda_{\text{max}} & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix} S^{-1}
$$

Substituting formula (12) into formulae (10) and (11), we can achieve the minimum solutions of both the matrix $K$ and the number $n$. For instance, we randomly obtain 3-group datum produced by formulae (1) and (2) and the calculating results, including task durations, matrix $K$ and number $n$, are listed in Table 1.

**Table 1. Calculating results of 3-group datum**

<table>
<thead>
<tr>
<th>group</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>Matrix K</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>6</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>15</td>
<td>12</td>
<td>8</td>
<td>-0.091 -0.001 0.062 0.127 0.162 -0.153 0.190 -0.063 -0.029 0.158 -0.195 0.256 0.028 0.020 0.091 0.137</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>18</td>
<td>15</td>
<td>10</td>
<td>-0.079 0.013 0.081 0.151 0.180 -0.133 0.218 -0.028 -0.011 0.177 -0.169 0.289 0.038 0.031 0.105 -0.119</td>
<td>4</td>
</tr>
</tbody>
</table>

In Table 1, the elements of the matrix $K$ denote the degree of other tasks on the resource usage of a task. Negative elements imply that fewer additional resources and less additional work are needed. The diagonal elements denote working on its own task which will help to reduce its own rework. Moreover, zero elements indicate no extra resources are required. Note that due to the effects of repetition probability, the terminated stage of iterations, $n$, calculated by formulae (10) and (11), is the maximum possible value. In general, the iteration process has been accomplished before the $n$th iteration stage because of the improvement of experiences and technologies.
4.3. **Implementation steps of the approach.** According to the analysis mentioned above and related discussions in Section 3, implementation steps of the approach to iteration analysis of product development based on feedback control theory are given as follows.

(1) Set up DSM to describe information dependency relationships among development tasks in product development process. Then, identify development iterations caused by coupled task sets. Moreover, determine repetition probability matrix \((RP)\), rework amount matrix \((RA)\) and duration matrix \((DM)\) of coupled task sets;

(2) Build feedback closed-loop system model in terms of assumptions mentioned in Section 4.1 and establish corresponding relations of the matrices in the model;

(3) Identify satisfaction degree with issues and adopt resource regulating matrix \(K\) to real-time monitor and control so as to make the system stability within an accepted range of makespan;

(4) Draw up detailed control strategy according to the results.

In the next section, we take a crane jib-bridge luffing system as an example to illustrate the analysis and disposing process.

5. **Example of a Crane Luffing System Development Project.** To illustrate the approach presented in the paper, an example of jib-bridge luffing system of a crane is investigated. Firstly, the background of an issue is introduced. Secondly, we adopt the approach presented in the paper to cope with coupled problems existing in a crane jib-bridge luffing system. Finally, the calculating results are also analyzed and discussed.

5.1. **Background.** In crane development process, a manufacturer often faces with the conditions that market demands change over time. For instance, if customer requirements change and they expect to improve work efficiency of a crane and shorten loading time, then lifting capacity of a crane should be improved from original 45t to current 80t. In addition, considering the factor of market competition, the manufacturer has to finish the development process within originally planned time. As a result, in order to satisfy the changes of customer requirements, the development program for a crane based on small lifting capability needs modifying and decoupled design is possible to turn into coupled one in development process. Due to these reasons, it is necessary to cope with the condition when coupling appears.

To satisfy the change of the lifting capability, great changes in the whole program have taken place. However, arm support system is the main mechanism to realize load-transmitting and also decides the bearing capacity of a crane. Additionally, luffing system is connected directly with arm support system and drives arm support to move horizontally. Owing to these reasons, the emphasis of this section is to study the development program of a jib-bridge luffing system and other mechanisms are not discussed due to the length limitation of the paper.

In development process of a crane with small lifting capability, in general, both a single jib luffing system based on rope compensation scheme and a driven system based on rack luffing scheme are used to realize the amplitude of variation. However, a single jib luffing system has some drawbacks: 1. there are little useful space below arm support which will affect its use; 2. suspending wirerope is too long to benefit to loading and uploading; 3. hoist rope wears very quickly. Also hunting zone is very large when single jib is located on small amplitude; 4. when a crane bears a large load, rack is easy to be worn out. From these reasons, development process of a crane jib-bridge luffing system with medium lifting capability should adopt a jib system of elastic four-bar linkage and a lead screw driven device to substitute a single jib system and a rack driven device,
respectively. The jib system of elastic four-bar linkage can not only make up the lacks of a single jib system but also realize horizontal displacement compensation through selecting appropriate size of arm support. Owing to compact structure, smooth transmission and low noise, a lead screw driven device has been suit for all kinds of working conditions. In the following section, we will adopt the method presented in the paper to cope with the coupling existing in a jib-bridge luffing system.

5.2. Problem solving process. According to related analysis mentioned in Section 5.1, a jib-bridge luffing system can be divided into small-scale subsystem modules, i.e., a jib system of elastic four-bar linkage and a luffing system, where a luffing system includes a horizontal displacement compensation system, a balancing self-weight device for a jib system, a luffing driven system and a safe-assistant device. Elastic four-bar linkage includes fly-jib, rigid pull rod, jib and stander. A balancing self-weight device for a jib system is mainly comprised of counterpoise and small pull rod. A luffing driven system is consisted of transmission layout scheme, electric motor, gear reducer, brake device, coupling device and lead screw driven device. A safe-assistant device includes buffering and damping device, indicating device and safety device. In addition, because hoist wirerope, leading sheave and sheerleg are directly connected with the luffing system, these three components should be considered in development process of a jib-bridge luffing system.

The design structure matrix model of a jib-bridge luffing system is set up according to the following steps: 1. identify 18 subtasks that development process of the luffing system includes and build corresponding rows and columns of the matrix; 2. determine input/output relationships among tasks to obtain Boolean DSM as shown in Figure 6(a); 3. examine dependent relationships in DSM, remove unnecessary ones and increase omissive ones; 4. adopt partitioning algorithm to find out coupled sets existing in DSM so as to optimize development process as shown in Figure 6(b). We can see from Figure 6 that there exists a coupled task set comprised of 8 tasks owing to selecting different jib system and luffing driven device. Therefore, how to cope with the coupled set is very important.

Adopt the method presented in the paper to cope with the coupled task set shown in Figure 6(b). Firstly, build rework amount matrix (RA) and repetition probability matrix (RP) reflecting iteration characteristics of coupled tasks. The matrix RA can be transformed into dependence matrix suggested by literature [12]. In lieu of precise numerical value in the matrix RA, the individual elements are estimated to be either of weak, medium, or strong dependence. We may use the values 0.3, 0.2, 0.1 for strong, medium, and weak dependence, respectively, in terms of fuzzy evaluation method as well as engineers’ experiences (see Figure 7(a)). The matrix RP is mainly influenced by both information evolution of the upstream tasks and information sensitivity of the downstream ones. Analytic hierarchy process (AHP) based on pairwise comparisons [22] are used to determine the element values in the matrix RP (see Figure 7(b)). Furthermore, task durations are affected by customer requirements, personal experiences, development resources, and so on, so three-point estimate method is adopted to express task durations. The engineers in the development organization are asked to describe how long tasks will be accomplished. We then assign precise optimistic, likely and pessimistic values to the task durations described by engineers (see Figure 7(c)).

Secondly, the accepted iteration makespan of the coupled set, $M_{accept}$, can be determined according to customer requirements. If $M$ (actual iteration makespan) is greater than $M_{accept}$, then feedback control strategy is employed and the regulating matrix $K$ is used to monitor and control the makespan. We assume that a task is considered to be terminated when the remaining work is 5% of the original. The matrix $K$ has to meet the following formulae (13) and (14), where formula (13) is applied to assure that the iteration process
**Figure 6.** Design structure matrix model of a jib-bridge luffing system

(a) original DSM

(b) DSM after partitioning
Figure 7. Iteration process description of a jib-bridge luffing system
can be finished within the planned time and formula (14) is employed to determine the terminated stage of iterations. In this example, $M_{\text{accept}}$ is equal to 10 weeks and $n$ denotes the terminated stage of iterations.

$$
\sum_0^n \max_i \left[ DM(\mathbf{R}\mathbf{A}^{(t+1)} - K)^n u_0 \right]^{(i)} \leq \sum_0^n \max_i \left[ DM(\mathbf{R}\mathbf{A} - K)^n u_0 \right]^{(i)} \leq 10 \quad (13)
$$

$$
\max_i \left[ (\mathbf{R}\mathbf{A} - K)^n u_0 \right]^{(i)} \leq 0.05 \quad (14)
$$

The convergence rate of iteration process is associated with the maximum eigenvalue, $\lambda_{\text{max}}$, of the matrix $(\mathbf{R}\mathbf{A} - K)$. When $\lambda_{\text{max}}$ approaches to -1, an iteration process will converge quickly. On the contrary, when $\lambda_{\text{max}}$ approaches to 1, it will converge slowly [12]. Therefore, the iteration process can be accelerated through regulating the magnitude of the maximum eigenvalue $\lambda_{\text{max}}$, which will be realized by selecting the appropriate matrix $K$. The calculation principle of the matrix $K$ has been introduced in detail in Section 4.2, so we directly give the final results of the matrix $K$ and the number $n$ corresponding to different task durations (see Table 2).

5.3. **Analysis and discussion of results.** In Table 2, if task durations take optimistic values, feedback control may not be launched and the matrix $K$ is a null matrix as shown in the second row and the tenth column in Table 2. That is to say, the makespan of iteration process is acceptable and Figure 8 shows the result of iterations for all the development tasks. For the other two circumstances, $K$ is not a null matrix as shown in Table 2. It means feedback control needs employing and extra resources are required. Figures 9 and 10 are corresponded to the convergence results of iterations when the task durations select likely and pessimistic values, respectively. Furthermore, in order to clearly reflect the changes of tasks with feedback control theory when taking different durations, we choose task 9 as an object and its convergence trend is shown in Figure 11. It can be seen that task 9 converges to 5% of its original after only six and five iterations when taking likely and pessimistic duration, comparing to seven iterations without employing feedback control method when task 9 takes optimistic duration. Besides, using the proposed feedback control method may not only accelerate iteration process but also maintain the stability of iterative development process. At the same time, by computing the regulating matrix $K$, the corresponding amount of extra resources required by each task can be determined. Table 3 lists rework of all tasks and the additional resources required by them with feedback control. For example, when the task durations take likely values, in order to achieve the target of 5% of the remaining work within the planned time, it is necessary to allocate additional resources to tasks to accelerate the convergence of iterations, so at the outsets, additional resources required by tasks are $(0.0330, 0.0377, 0.0486, 0.0487, 0.0319, 0.0157, 0.0181, 0.0442)$.

Consider the different stage of iteration, that is to say, $k=1$. At $k=1$, the amount of remaining work for tasks would have gone down from $(0.5, 0.6, 0.8, 0.7, 0.4, 0.3, 0.5, 0.9)$ to $(0.4670, 0.5623, 0.7514, 0.6513, 0.3681, 0.2843, 0.4819, 0.8558)$ due to allocating additional resources to tasks. In addition, when considered other stages of iteration, the iteration process can also be accelerated owing to allocating additional resources to tasks as shown in Table 3. The whole makespans of iterations are 7.37, 10.05, and 10.49, respectively. Although the last two are slightly greater than $M_{\text{accept}}$, the results are still accepted owing to the effects of repetition probability. In addition, we can see from Table 3 that for the second and the third groups of datum, if we accelerate only one task, any other tasks are all converged faster after employing feedback control. It is mainly because all the tasks are coupled each other. It shows that only regulating a certain task will
<table>
<thead>
<tr>
<th>group</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>15</th>
<th>matrix $K$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.5</td>
<td>2</td>
<td>0.5</td>
<td>0.5</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2.5</td>
<td>0 0 0 0 0 0 0 0</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1.5</td>
<td>3</td>
<td>4</td>
<td>-0.043 -0.214 -0.072 -0.048 -0.103 -0.035 -0.043 0.113</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.151 -0.098 -0.082 -0.055 -0.018 -0.041 -0.049 0.200</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.063 0.173 -0.106 -0.071 0.148 -0.052 -0.063 0.071</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.063 0.073 0.194 -0.071 -0.052 -0.053 -0.063 -0.029</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.041 -0.083 0.131 0.154 -0.099 -0.034 -0.042 -0.085</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.020 -0.041 -0.034 -0.023 0.251 -0.017 -0.020 -0.042</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.023 -0.047 -0.039 -0.026 0.144 0.281 -0.024 -0.048</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.043 -0.015 0.004 0.036 -0.038 0.053 0.243 -0.117</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.5</td>
<td>3.5</td>
<td>2</td>
<td>1.5</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>-0.038 0.223 -0.064 -0.043 -0.092 -0.032 -0.038 0.122</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.157 -0.088 -0.073 -0.049 -0.005 -0.036 -0.044 0.211</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.056 0.187 -0.094 -0.063 0.165 -0.047 -0.056 0.085</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.056 0.087 0.206 -0.063 -0.035 -0.047 -0.057 -0.015</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.037 -0.074 0.139 0.159 -0.089 -0.031 -0.037 -0.075</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.018 -0.036 -0.030 -0.020 0.257 -0.015 -0.018 -0.037</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.021 -0.042 -0.035 -0.024 0.150 0.283 -0.021 -0.043</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.049 -0.003 0.015 0.043 -0.023 0.058 0.249 -0.104</td>
<td></td>
</tr>
</tbody>
</table>
Figure 8. Iteration process taken optimistic durations

Figure 9. Iteration process taken likely durations
Figure 10. Iteration process taken pessimistic durations

Figure 11. Natural and control convergence process of task 9 taken different durations
Table 3. Rework and additional resource required by tasks with feedback control

<table>
<thead>
<tr>
<th>Task durations (1, 3, 5, 7, 10, 15, 20, 25)</th>
<th>Additional resources required by tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$ $w_2$ $w_3$ $w_4$ $w_5$ $w_6$ $w_7$ $w_8$</td>
<td>$r_1$ $r_2$ $r_3$ $r_4$ $r_5$ $r_6$ $r_7$ $r_8$</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
</tr>
<tr>
<td>0 1 1 1 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 0.5 0.8 0.1 0.7 0.4 0.3 0.5 0.9</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>2 0.19 0.41 0.48 0.49 0.39 0.12 0.17 0.48</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>3 0.216 0.246 0.399 0.304 0.194 0.060 0.096 0.297</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>4 0.1272 0.1453 0.1554 0.1020 0.1226 0.0582 0.0658 0.1550</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>5 0.6759 0.0721 0.1120 0.1130 0.0947 0.0385 0.0425 0.1022</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>6 0.6866 0.0531 0.0890 0.0890 0.0452 0.0224 0.0269 0.0626</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>7 0.0385 0.0265 0.0421 0.0421 0.0076 0.0135 0.0155 0.0393</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

Before using an analytical approach to development iterations based on feedback control theory, the crane manufacturer copes with coupled task sets in terms of rework amount of tasks, which not only increases the number of iterations but also delays the makespan of iterations. However, after adopting the approach, the makespan of iterations shortens to ten weeks and the number of iterations reduces to no more than seven. From the analysis above, we can see that with the feedback control method, the amount of remaining work for each task is monitored and controlled by the matrix $K$. Once undesirable situations occur, the matrix $K$ will be launched to bring them back within the desired work.

6. Concluding Remarks. In this paper, we propose an approach to analyze product development iteration based on feedback control theory in a dynamic environment. Firstly, the uncertain factors, including uncertainty of task durations, uncertainty of output branches of tasks and uncertainty of resource allocations, existing in product development process are discussed. Secondly, feedback control model used to monitor and control these uncertainties are presented by introducing the resource regulating matrix $K$. At the same time, the concept of satisfaction degree is introduced and a satisfaction degree-based
feedback control method is proposed. Additionally, the detailed calculating principle and method of the matrix $K$ is described. Lastly, an example of a crane development is provided to illustrate the approach. The results show that the issue discussed in this paper is an extension of DSM, and also gives a further insight into complex product development. Moreover, with feedback control method, the remaining work for each development task can be monitored and controlled in a dynamic environment. If the convergence rate of the tasks is within the desired state at every stage of iteration $k$, the method can be used to predict the states of the following stage of tasks. However, once undesirable circumstances appear, feedback control method is employed to bring tasks back within the desired states. Therefore, it can be used to improve the stability and the convergence rate of development tasks when development process is in a dynamic environment.

However, due to the complexity of the product development, several issues should be taken into account in the future researches: 1. the effects of other eigenvalues except for the maximum one on system stability are the next challenge; 2. the mathematical analysis of non-linear systems is far more complex than for linear systems, hence, the modeling and analysis of time-varying, non-linear development tasks need further research; 3. to accelerate the convergence of the iteration process, the re-organization tasks of a process should be implemented in parallel based on the feedback information. It is also another area to be studied; 4. other approaches, such as disruption management strategy [23], should be considered to deal with the dynamic changes in product development, which is also an area to be explored; 5. how to incorporate artificial intelligence technologies including artificial immune system [24], fuzzy-neural system [25] and so on, to tackle other more complex aspects of dynamic product development such as job shop scheduling problem [26] and production scheduling problem [27] is also required to be studied.

Acknowledgment. This research is supported by the National Natural Science Foundation of China (Grant No. 50575083, 70671094, 70701014) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 200804870070).

REFERENCES


