Técnicas de diseño óptimo multidisciplinario

Multidisciplinary design optimisation techniques

Andrés Tovar,[1] Nelson Arzola de la Peña[2] y Alexander Gómez[3]

RESUMEN

El proceso de optimización de un proyecto multidisciplinario en ingeniería involucra la descomposición de un sistema en distintas disciplinas y la posterior asociación de sus contribuciones. El objetivo de este trabajo es presentar los esquemas de desagregación y asociación más utilizados actualmente en procesos de diseño óptimo multidisciplinario – *Multidisciplinary Design Optimization* (MDO). Entre los esquemas de desagregación se presentan la descomposición jerárquica y no jerárquica, así como los métodos computacionales más comunes. Los esquemas de asociación incluyen: formulación de un solo nivel (e.g., optimización integrada y análisis y diseño simultáneos), formulación de múltiples niveles (e.g., optimización en espacios concurrentes y optimización colaborativa) y diseño robusto. El trabajo presenta también un ejemplo de aplicación resuelto numéricamente.

Palabras clave: diseño óptimo multidisciplinario, descomposición de un sistema, optimización colaborativa.

ABSTRACT

Design optimisation of a multidisciplinary project in engineering involves the decomposition of a system into disciplines and the subsequent association of their contributions. This work was aimed at presenting the most common decomposition and association techniques currently used in multidisciplinary design optimisation (MDO). Amongst the decomposition techniques this work includes hierarchical and non-hierarchical approaches as well as the most popular numerical procedures. The association techniques include: one-level methods (e.g. all-at-once optimisation and simultaneous analysis and design), multilevel methods (e.g. concurrent subspace optimisation and collaborative optimisation) and robust design. This work also incorporates an illustrative numerical example.

Keywords: multidisciplinary design optimisation, system decomposition, collaborative optimisation.

Recibido: septiembre 28 de 2006
Aceptado: marzo 1 de 2007

Introducción

El diseño en ingeniería involucra la participación de grupos de trabajo en distintas disciplinas. Normalmente cada grupo es responsable del diseño de un subsistema del proyecto completo. Aunque durante el proceso de diseño los objetivos de cada subsistema deben satisfacer los objetivos y restricciones del sistema, es muy factible que estos entren en conflicto con los objetivos de otros subsistemas. Para acomodar todas las funciones objetivo en proyectos multidisciplinarios se ha instituido un nuevo campo de estudios denominado diseño óptimo multidisciplinario – *multidisciplinary design optimization* (MDO). MDO es un concepto que ha existido en la mente de los diseñadores por muchas décadas (Kron, 1953); sin embargo, solo se formalizó como un área de estudio en los años ochenta (AIAA, 1991; Lewis, 1996).

El diseño óptimo multidisciplinario es una estrategia de ingeniería aplicable al diseño de sistemas complejos de gran escala, cuya solución se ve afectada por los diseños óptimos de los subsistemas que los componen. Proyectos de ingeniería tales como el diseño de automóviles, barcos o aviones, son por naturaleza de carácter multidisciplinario, ya que involucran la participación de grupos de trabajo de distintas especialidades que interactúan a través de canales de comunicación interdisciplinarios. Una característica

[1] Ingeniero mecánico. M.Sc., en Automatización industrial, Universidad Nacional de Colombia, Bogotá. M.Sc., en Ingeniería Mecánica, University of Notre Dame, Indiana, USA. Ph.D., en Ingeniería mecánica y aeroespacial, University of Notre Dame, Indiana, USA., Director, grupo de investigación en Diseño Óptimo Multidisciplinario – OptimUN. Profesor asociado, Departamento de Ingeniería Mecánica y Mecatrónica, Universidad Nacional de Colombia, Bogotá. atovarp@unal.edu.co.

[2] Ingeniero mecánico, Universidad de Cienfuegos, Cuba. Doctor, Ciencias Técnicas, Universidad Central de las Villas, Cuba. Codirector, grupo de investigación en Diseño Óptimo Multidisciplinario – OptimUN. Profesor asociado, Departamento de Ingeniería Mecánica y Mecatrónica, Universidad Nacional de Colombia, Bogotá. narzola@unal.edu.co.

[3] Ingeniero mecánico. Ph.D., en Ingeniería Mecánica, Universidad de Kassel, Alemania. Codirector, grupo de investigación Biomasa y Optimización de Procesos Térmicos – BIOT. Profesor asociado, Departamento de Ingeniería Mecánica y Mecatrónica, Universidad Nacional de Colombia, Bogotá. agomezm@unal.edu.co.

importante que complica el proceso de diseño y optimización en este tipo de proyectos, es el hecho de que cada disciplina o grupo de trabajo busca optimizar sus propios objetivos de diseño satisfaciendo restricciones individuales. Como resultado, se obtiene un problema de optimización multiobjetivo que involucra funciones potencialmente conflictivas entre sí. Por ejemplo, en el diseño de un automóvil se requiere una estructura rígida para soportar los pesos de los componentes mecánicos; sin embargo, es necesario que la estructura sea al mismo tiempo deformable, para mejorar las condiciones de seguridad durante un impacto, y además liviana, para mejorar la relación peso/potencia y el rendimiento.

La solución de un problema de MDO se inicia a través de la descomposición del sistema en sus subsistemas componentes. Estos subsistemas están conectados entre sí a través de canales que comunican su función, diseño y desempeño. Los métodos MDO normalmente utilizan técnicas en las cuales es necesario analizar cada subsistema individualmente. Luego, los resultados de los análisis son recolectados y procesados usando técnicas de enlace que garantizan la compatibilidad entre subsistemas. En la actualidad existe una gran variedad de métodos MDO. Una revisión sobre algunos de los más tradicionales puede consultarse en AIAA (1991), Balling y Sobieszczanski-Sobieski (1996), Sobieszczanski-Sobieski y Haftka (1997) y Lewis y Mistree (1998). El propósito de este trabajo es presentar las técnicas actuales más comunes para la solución de problemas MDO.

Esquemas de descomposición

En la actualidad es virtualmente imposible encontrar un profesional que domine todo el conocimiento necesario para el desarrollo de un proyecto complejo, multidisciplinario y de gran escala en ingeniería, pues de hecho, requiere de especialistas en distintas áreas. De acuerdo con la forma en la que se interrelacionan las áreas de especialización o subsistemas, un proyecto se puede descomponer de múltiples formas, siendo las más comunes los métodos de descomposición jerárquica (Figura 1) y los de descomposición no jerárquica (Figura 2). Una revisión de varios métodos de descomposición puede encontrarse en Barthelemy (1988). Renaud (1992) hace una descripción completa sobre estos dos tipos de descomposición.

Descomposición jerárquica

El esquema de descomposición jerárquica, como el mostrado en la Figura 1, fue introducido por Sobieszcznanski-Sobieski (1982) e implementado por Barthelemy y Sobieszczanski-Sobieski (1983) como un método para determinar derivadas en análisis de sensibilidad. Con este tipo de descomposición, la información para el análisis fluye de arriba hacia abajo, transmitiéndose de "padre" a "hijo". Por ejemplo, un análisis de esfuerzo sobre toda la estructura de un automóvil puede ser un "padre" que le transmite fuerzas en las fronteras a los ejes que representan un "hijo", y a los alerones, que representarían otro "hijo".

Con el esquema jerárquico, el análisis en cada subsistema puede hacerse secuencialmente. De esta forma, se llega a una descripción estable del sistema (e.g., variables de estado y variables de sistema) y de sus funciones objetivo (e.g., desempeño y costo). El modelamiento y simulación de un sistema jerárquico es relativamente simple, ya que toda la información requerida por un subsistema está disponible una vez se ha ejecutado el análisis en las disciplinas anteriores. No se requiere que la información sea devuelta o que haya la de doble flujo entre los subsistemas. De esta misma forma, el proceso de optimización se transmitiría desde abajo hacia arriba. Algunos ejemplos de este tipo de descomposición se pueden encontrar en Sobieszczanski-Sobieski *et al*. (1985) y Wrenn y Dovi (1988).

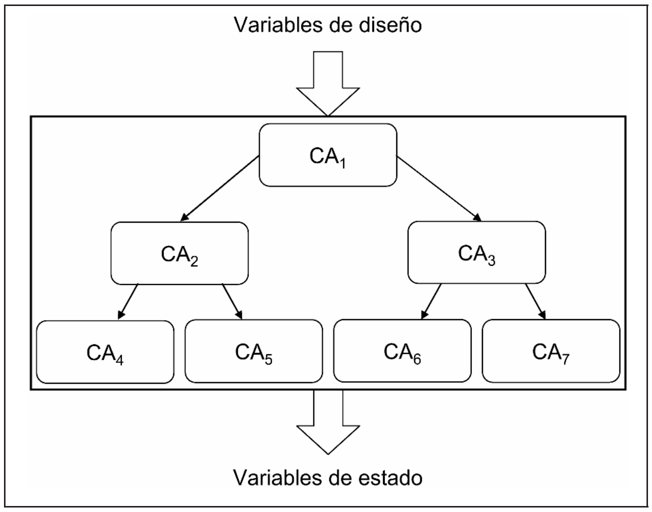


Figura 1. Descomposición de un sistema complejo utilizando una estructura jerárquica. Con esta descomposición, el análisis de contribución (CA) en cada subsistema se hace secuencialmente

Descomposición no jerárquica

El concepto matemático del esquema no jerárquico fue introducido por Sobieszczanski-Sobieski (1990a, 1990b). Sin embargo, la descomposición no jerárquica ha sido aplicada con anterioridad al diseño de aeronaves (Sobieszczanski-Sobieski, 1988b; Abi *et al*., 1988), diseño estructural (Adelman y Haftka, 1986) y diseño aerodinámico (Yates, 1987). Un sistema no jerárquico, como el mostrado en la Figura 2, es aquel en el cual no hay una secuencia predeterminada para el análisis en cada subsistema, permitiendo la transmisión de información en forma multidireccional. El sistema no jerárquico no puede organizarse en una forma piramidal "padre – hijo". La interacción compleja entre las distintas disciplinas o subsistemas obliga a usar una gran cantidad de ciclos para obtener información de diseño confiable. Por su complejidad, el proceso de optimización puede ser ejecutado como una operación integral de todo el sistema.

Métodos computacionales

La descomposición de un sistema es la base del diseño óptimo multidisciplinario. Cuando no hay suficiente información disponible sobre un sistema, su descomposición se puede ver beneficiada por una gran variedad de métodos computacionales (AIAA, 1991). El objetivo de estos métodos es convertir un conjunto de subespacios, aleatoriamente ordenados, en un conjunto ordenado jerárquicamente no jerárquicamente o de una forma híbrida combinando estos dos esquemas. Un algoritmo computacional normalmente iniciaría con una distribución aleatoria de los subsistemas y de sus conexiones. La Figura 3 ilustra una distribución diagonal de subsistemas interconectados matricialmente. Las conexiones sobre la diagonal representan la alimentación de datos mediante una transmisión *downstream*. Las conexiones bajo la diagonal indican retroalimentación de datos mediante transmisión *upstream*.

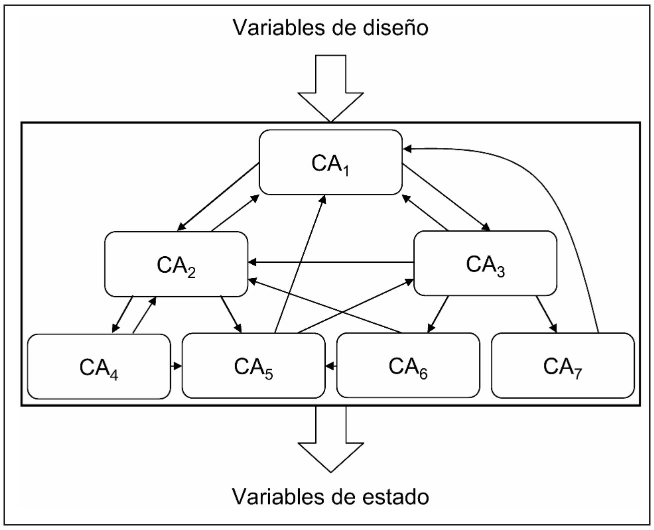


Figura 2. Descomposición de un sistema complejo utilizando una estructura no jerárquica. Con esta descomposición, la secuencia de los análisis de contribución (CA) en cada subsistema no está predeterminada

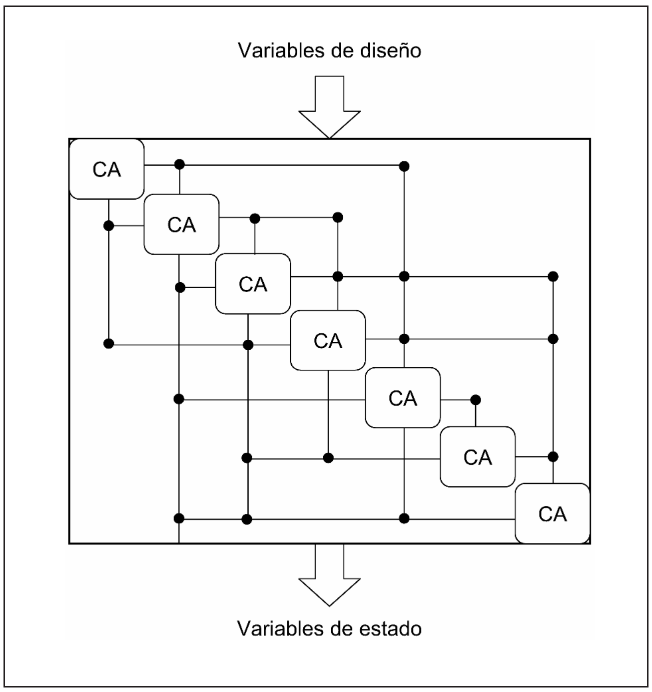


Figura 3. Distribución diagonal de subsistemas con una interconexión matricial aleatoria

Mediante permutaciones sucesivas entre filas y columnas y reorganización de subsistemas, la distribución y conexión inicial se transforma en una secuencia ordenada, como la que se presenta en la Figura 4. El objetivo de esta transformación es eliminar tantas líneas de retroalimentación como sea posible. En este caso, las líneas de retroalimentación fueron reducidas y encajonadas en *clusters* de subsistemas. En dicha configuración híbrida hay un esquema jerárquico de *clusters* cuyos subsistemas tienen una organización no jerárquica.

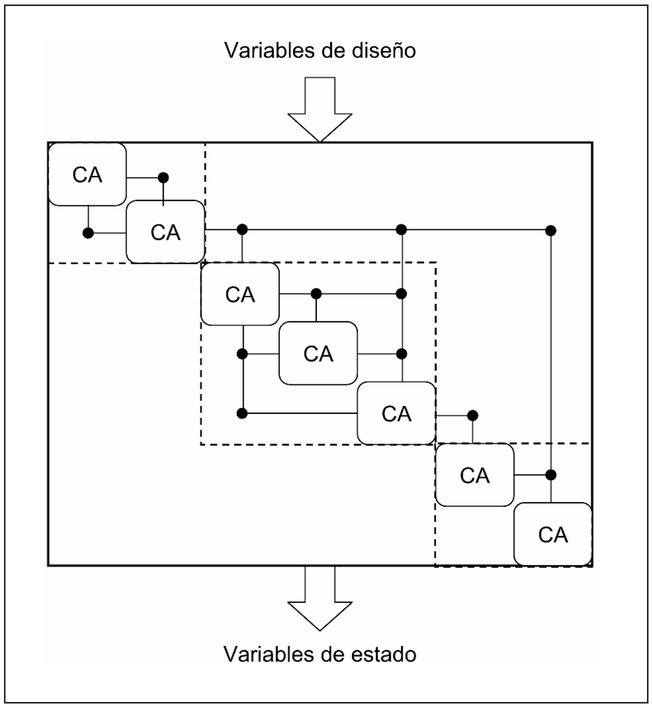


Figura 4. Distribución diagonal de subsistemas con una interconexión matricial ordenada. En este esquema híbrido los clusters (representados por líneas punteadas) están organizados en un esquema jerárquico; los subsistemas en cada cluster tienen una organización no jerárquica

Esquemas de asociación

La formulación matemática básica utilizada en la solución de un problema MDO es consistente con la estructura de un problema de optimización no lineal, es decir,

$$
\begin{aligned}
&\min && f(x) \\
&\text{s.t.} && g(x) \leq 0 \qquad\qquad (1) \\
& && h(x) = 0,
\end{aligned}
$$

donde los valores de las variables de diseño, x, minimizan la función objetivo, $f(x)$, satisfaciendo las restricciones de desigualdad, $g(x)$, y de igualdad, $h(x)$. La solución de este problema de optimización requiere, algunas veces, de un análisis de alta fidelidad (e.g., por elementos finitos). Tales rutinas reciben el nombre de análisis de contribución – *contributing analysis* (CA), las cuales se ejecutan a nivel de subsistema. Las técnicas más utilizadas en la formulación de problemas MDO pueden ser de dos tipos: de un solo nivel y de múltiples niveles.

Formulación de un solo nivel

La formulación de un solo nivel es la más simple. Esta se caracteriza porque los análisis, o llamados funciones, se encuentran en un único nivel (i.e., nivel de sistema). Su expresión más común y tradicional es la optimización integrada – *All-At-Once Optimization* (AAO), que se muestra en la Figura 5. Bajo este esquema se pretende encontrar los valores de las variables de diseño x que minimizan la función $f(x)$ y satisfacen las restricciones $g(x)$ y $h(x)$. Las relaciones entre los subsistemas componentes se encuentran implícitas dentro del análisis integrado. Para ciertos valores de variables de diseño el análisis integrado le transmite a la rutina de optimización los valores de la función objetivo y las restricciones. En sistemas complejos y de gran escala el análisis integrado de todos los subsistemas puede resultar demasiado complejo tanto conceptual como computacionalmente.

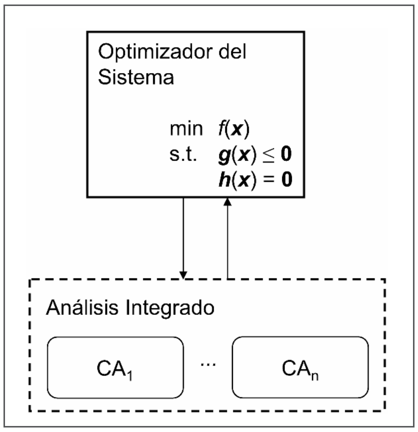


Figura 5. Formulación de un solo nivel con una estructura tradicional de optimización integrada – All-At-Once Optimization (AAO)

Una formulación alternativa es el análisis y diseño simultáneos – *Simultaneous Analysis and Design* (SAND), propuesta por Balling y Sobieszczanski-Sobieski (1994), la cual se ilustra en la Figura 6. La principal diferencia con la formulación AAO es que, utilizando SAND, el análisis en cada subsistema se ejecuta independientemente. Eso facilita la incorporación de esta metodología a sistemas complejos y de gran escala. Como el análisis en los subsistemas se realiza independientemente, el optimizador debe incorporar una serie de restricciones de compatibilidad, (x_c), en términos de las variables de acoplamiento, x_c, las cuales son comunes a varios subsistemas. Los valores de estas variables son calculados por la rutina de optimización y transmitidos a los subsistemas, en los cuales se ejecutan los respectivos análisis de contribución (CA) y sus resultados son devueltos al optimizador. Mediante las restricciones de compatibilidad, el optimizador verifica que los valores iniciales de las variables de acoplamiento no hayan cambiado y de esta forma asegura la compatibilidad entre los subsistemas (Herskovits, 2005).

En la formulación de un solo nivel la rutina de optimización se ejecuta únicamente al de sistema. Esta es la base de los métodos de ingeniería concurrente – *Concurrent Engineering* (CE) (Hall, 1991; Ziemke & Spann, 1991; Prasad, 1996). Aunque los esquemas CE son considerados ideales en muchas situaciones (Marston y Mistree, 1998), en problemas demasiado complejos es conveniente ejecutar rutinas de optimización internamente en los subsistemas. Por ejemplo, en el diseño de un auto de carreras resulta ventajoso realizar el análisis y optimización independientemente de cada uno de los alerones, delantero, trasero y laterales, para que el aire cumpla su función de direccionamiento, adherencia y refrigeración en el lugar que le corresponde (McAllister *et al*., 2002). Del mismo modo, en algunos casos la organización de la estructura del proyecto hace difícil la interacción de disciplinas en un solo nivel debido a barreras de comunicación o geográficas (Womack *et al*., 1990; Lewis y Mistree, 1997). El análisis y optimización a nivel de subsistema es la base de la formulación de múltiples niveles en problemas MDO, e.g., optimización de subespacios concurrentes y optimización colaborativa.

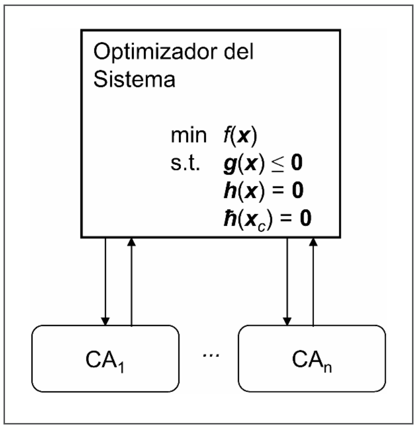


Figura 6. Formulación de un solo nivel con una estructural para análisis y diseño simultáneos – Simultaneous Analysis And Design (SAND)

Formulación de múltiples niveles

Un primer esquema de formulación de múltiples niveles es la denominada optimización de subespacios concurrentes – *Concurrent Subspace Optimizacion* (CSSO), propuesta por Sobieszczanski-Sobieski (1988a), que se muestra en la Figura 7. En CSSO se propone un coordinador a nivel de sistema, el cual únicamente evalúa las restricciones de compatibilidad y obtiene los valores de las variables de diseño que satisfacen esta condición. Así el coordinador asegura la factibilidad de los resultados de los optimizadores de los susbsistemas o subespacios. Por tal razón este esquema solamente es apropiado en problemas en los cuales no se requiere ni variables de diseño ni función objetivo a nivel de sistema.

La implementación del CSSO se basa en el uso de ecuaciones de sensibilidad global – *Global Sensitivity Equations* (GSE), las cuales se derivan del teorema de la función implícita (Sobieszczanski-Sobieski, 1988a). Para sistemas acoplados, no jerárquicos, las GSE permiten el cálculo exacto de las primeras derivadas de los estados de los subsistemas con respecto a las variables de diseño, $\mathrm{d}f_i/\mathrm{d}\boldsymbol{x}$. De esta forma el análisis de sensibilidad requiere de la concatenación de un vector de diseño. Renaud (1992) plantea un esquema CSSO modificado que elimina la necesidad de concatenación mediante una aproximación de segundo orden.

Se ha demostrado que el CSSO reduce significativamente el número de análisis de contribución (Balling y Sobieszczanski-Sobieski, 1996), pero puede presentar problemas de convergencia (Balling y Wilkinson, 1997). Sin un optimizador a nivel de sistema es difícil realizar un verdadero control sobre los estados de los subsistemas (McAllister *et al*., 2005). Dicha carencia motivó el desarrollo de una de las técnicas actuales más populares en MDO la optimización colaborativa.

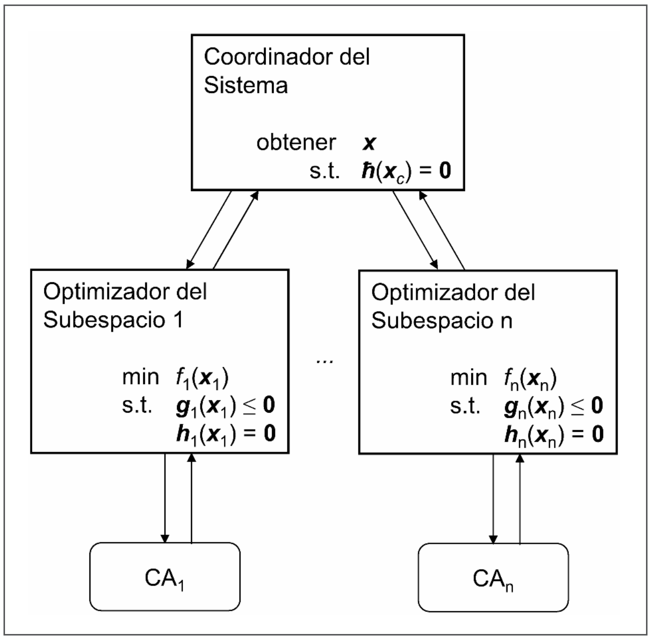


Figura 7. Formulación de múltiples niveles con una estructura de optimización de subespacios concurrentes – Concurrent Subspace Optimization (CSSO)

La optimización colaborativa – *Collaborative Optimization* (CO) fue propuesta por Kroo *et al*. (1994) y Braun *et al*. (1996) y posteriormente desarrollada por Braun y Kroo (1997). Tal como se muestra en la Figura 8, en el esquema CO cada subespacio tiene un optimizador cuya función es minimizar la violación de las restricciones de compatibilidad satisfaciendo las restricciones propias del subsistema. Así mismo, a diferencia del CSSO, el esquema CO implementa un optimizador que actúa sobre una función objetivo a nivel de sistema, Lo que representa una ventaja significativa del CO sobre el CSSO. Sin embargo, la implementación del CO es computacionalmente costosa debido al número de iteraciones requeridas para satisfacer las condiciones de compatibilidad a nivel de sistema, las cuales aseguran la igualdad de las variables comunes en los subsistemas.

Aplicaciones de esquemas CO incluyen: diseño de vehículos espaciales (Braun *et al*., 1997), diseño de alas de aviones (Sobieski y Kroo, 1996), vehículos subacuáticos (McAllister *et al*., 2000) y diseño de carros de fórmula uno (McAllister *et al*., 2005). Algunas modificaciones sobre la formulación original del CO incluyen, entre otras, las siguientes: problemas multiobjetivo usando sumas ponderadas (Tappeta y Renaud, 1997), programación por objetivos (McAllister *et al*., 2000), diseño basado en confiabilidad (Gu y Renaud, 2001; McAllister y Simpson, 2003), diseño basado en decisiones (Gu *et al*., 2002) y diseño multiobjetivo con programación física (Tappeta *et al*., 2000; McAllister *et al*., 2005).

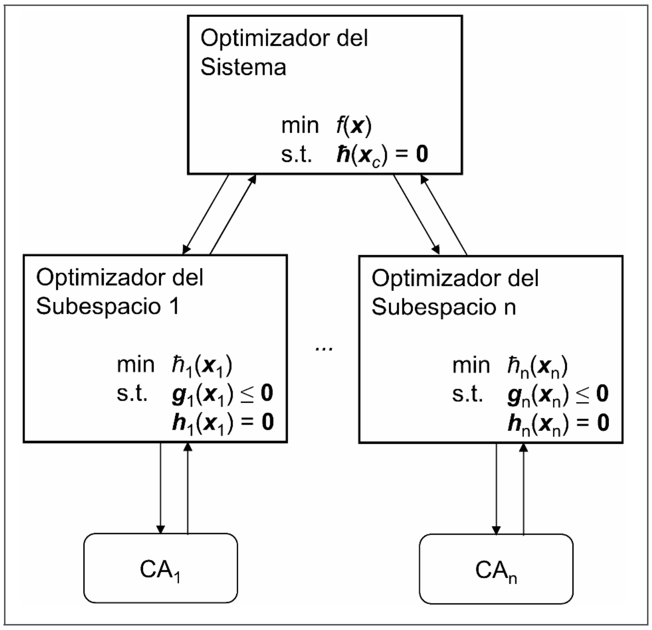


Figura 8. Formulación de múltiples niveles con una estructura de optimización colaborativa – Collaborative Optimization (CO)

Diseño robusto

El proceso de optimización en proyectos multidisciplinarios se fundamenta en decisiones tomadas por cada grupo de trabajo. Estos grupos, responsables del diseño de los subsistemas de todo el proyecto, están normalmente acoplados, es decir, manejan variables de diseño controladas por otros grupos (Chen y Lewis, 1999; Kalsi *et al*., 2001). En teoría, los procesos de diseño colaborativos basados en la interacción de los distintos grupos de trabajo, deben arrojar resultados inmejorables; sin embargo, en la práctica la colaboración completa es casi imposible (Danesh, 2001). Los esquemas secuenciales representan con más fidelidad los procesos diseño más comunes en la industria actual (Kalsi *et al*., 2001). De esta forma se han propuesto técnicas y modelos matemáticos del proceso de diseño que le permiten a cada disciplina resolver su problema de optimización independientemente del resto del sistema (Bloebaum *et al*., 1992; Balling y Sobieszczanski-Sobieski, 1996; Gu *et al*., 2000). Este diseño con información incompleta, incierta o variable, da origen al diseño robusto.

El diseño robusto se basa en la minimización del efecto de la variación en los parámetros de diseño sin eliminar la fuente de incertidumbre (Phadke, 1989). La aplicación de técnicas de diseño robusto ha demostrado ser efectiva en la solución de una gran variedad de problemas multidisciplinarios en ingeniería (Gu *et al*., 2000; Batill *et al*., 2000, Agarwal *et al*., 2004, Agarwal y Renaud, 2004). En diseño robusto se distinguen dos tipos de problemas (Chen *et al*., 1996). En diseño robusto Tipo I, el objetivo es minimizar la variación causada por factores de ruido incontrolables (e.g., temperatura ambiente, ambiente de operación). En diseño robusto Tipo II, el objetivo es minimizar la variación causada por desviaciones en los factores de control (i.e., variables de diseño).

Ejemplo de aplicación

Para ilustrar la aplicación del MDO en un problema de ingeniería se ha planteado una gran cantidad de problemas algebraicos (Shankar *et al*., 1993; Alexandrov y Lewis, 2002). Uno particularmente simple es el propuesto por Shankar *et al*., (1993), el cual se ha usado para evaluar una gran cantidad de métodos MDO. El problema consta de una función objetivo cuadrática, escrita en términos de dos variables, y es acoplado mediante dos restricciones lineales. Aunque no es particularmente diseñado para resolverse con un esquema CO, su solución será de ayuda para entender la implementación de esta metodología. El problema algebraico puede escribirse de la siguiente forma:

$$\begin{aligned} \text{min} \quad & f(\boldsymbol{x}) = x_1^{\,2} + x_2^{\,2} \\ \text{s.t.} \quad & g_1(\boldsymbol{x}) = x_1 + \beta x_2 \text{ £ } 4 \\ & g_2(\boldsymbol{x}) = \beta x_1 + x_2 \text{ ³ } 2 \\ & l_1 < x_1 < u_1 \\ & l_2 < x_2 < u_2. \end{aligned} \qquad (2)$$

Su descomposición en dos subsistemas sugiere la copia local de las variables del sistema. De esta forma, x_{11} y x_{21} serán la copia de x_1 y x_2 en el subsistema 1, y x_{12} y x_{22} serán la copia de x_1 y x_2 en el subsistema 2. Los problemas de optimización en cada subsistema se pueden definir así:

$$\begin{aligned} \text{min} \quad & {}_1(\boldsymbol{x}) = (x_{11} - x_1)^2 + (x_{21} - x_2)^2 \\ \text{s.t.} \quad & g_1(\boldsymbol{x}) = x_{11} + \beta x_{21} \text{ £ } 4, \end{aligned} \qquad (3)$$

para el subsistema 1, y para el subsistema 2,

$$\begin{aligned} \text{min} \quad & {}_2(\boldsymbol{x}) = (x_{12} - x_1)^2 + (x_{22} - x_2)^2 \\ \text{s.t.} \quad & g_2(\boldsymbol{x}) = \beta x_{21} + x_{22} \text{ ³ } 2. \end{aligned} \qquad (4)$$

El problema de optimización a nivel de sistema se desarrolla así:

$$\begin{aligned} \text{min} \quad & f(\boldsymbol{x}) = x_1^{\,2} + x_2^{\,2} \\ \text{s.t.} \quad & {}_1(\boldsymbol{x}) = 0 \\ & {}_2(\boldsymbol{x}) = 0 \end{aligned} \qquad (5)$$

Este problema puede resolverse a nivel de sistema (sin descomposición) utilizando programación secuencial cuadrática – *Sequential Quadratic Programming* (SQP). El algoritmo se encuentra implementado en la función fmincon del *toolbox* de optimización de Matlab®, desarrollado por The MathWorks (ver Anexo). SQP también podría utilizarse para solucionar el problema CO (descompuesto) a nivel de cada subsistema; sin embargo, las soluciones a nivel de subsistema pueden derivarse analíticamente de manera simple (Anexo).

Comentarios finales

El diseño óptimo multidisciplinario (MDO) se fundamenta en dos procesos: descomposición y asociación. Los dos esquemas de descomposición clásicos en MDO son descomposición jerárquica y descomposición no jerárquica. Así mismo, se utilizan comúnmente dos métodos computacionales para interconectar las distintas disciplinas de un sistema no jerárquico: interconexión matricial aleatoria e interconexión matricial ordenada. En cuanto a los esquemas de asociación, existen dos formulaciones básicas: la de un solo nivel (e.g., AAO, SAND, CE) y la de múltiples niveles (e.g., CSSO, CO). Estos esquemas de asociación arrojarían resultados inmejorables, pero en la práctica existen limitantes que impiden la perfecta comunicación entre las distintas disciplinas. En la actualidad hay un gran desarrollo en técnicas de diseño robusto que permiten suplir estas falencias. Un sencillo ejemplo, resuelto numérica y analíticamente, ha ilustrado los procesos de descomposición y asociación en un problema multidisciplinario.

MDO es un campo de estudio formalizado hace algo más de una década. Sus desarrollos más importantes se han dado, en su mayoría, en el presente milenio. Sus aplicaciones en procesos de diseño en ingeniería crecen continuamente y demuestran su importancia. Esto avala una revisión de sus fundamentos, desarrollos y aplicaciones tal como la que se presenta en este artículo.

Agradecimientos

Este trabajo ha contado con la financiación de la Dirección de Investigación Sede Bogotá de la Universidad Nacional de Colombia a través del proyecto "optimización estructural con autómatas celulares híbridos" (DIB 20601003551).

Nomenclatura

f( )	Función objetivo
g( )	Restricción de desigualdad
( )	Restricción de compatibilidad
h( )	Restricción de igualdad
l	Límite inferior
L( )	Lagrangiano
Lx( )	Derivada parcial del lagrangiano respecto a *x*
u	Límite superior
x	Variable de diseño
x_c	Variable de compatibilidad
β	Factor independiente
λ	Multiplicador de Lagrange

Bibliografía

Abi, F. F., Ide, H., Shankar, V. J., and Sobieszczanski-Sobieski, J., Optimization for Nonlinear Aeroelastic Tailoring Criteria., International Council for Aeronautical Sc., Proceedings of 16th Congress, Jerusalem, Vol. 2, 1988, pp. 1083-1091.

Adelman, H. A. and Haftka, R. T., Sensitivity Analysis of Discrete Structural Systems., AIAA J., Vol. 24, No. 5, 1986, pp. 823–832.

Agarwal, H. and Renaud, J. E., Reliability based design optimization using response surfaces in application to multidisciplinary systems., Engineering Optimization, Vol. 36, No. 3, 2004, pp. 291–311.

Agarwal, H., Renaud, J. E., Preston, E. L. and Padmanabhan D., (2004), Uncertainty quantification using evidence theory in multidisciplinary design optimization., Reliability Engineering & System Safety, Vol. 85. No. 1–3, 2004, pp. 281–294.

AIAA Technical Committee on Multidisciplinary Design Optimization (MDO)., White Paper on Current State of the Art., Disponible en: http://endo.sandia.gov/AIAA_MDOTC/sponsored/aiaa_paper.html, 1991.

Alexandrov, N. M. and Lewis, R. M., Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design., AIAA J., Vol. 40, No. 2, 2002, pp. 301-309.

Balling, R. J. and Sobieszczanski-Sobieski, J., Optimization of Coupled Systems: A Critical Overview of Approaches., AIAA J., Vol. 34, No. 1, 1996, pp. 6–17.

Balling, R. J. and Wilkinson, C. A., Execution of multidisciplinary design optimization approaches on common test problems., AIAA J., Vol. 35, No. 1, 1997, pp. 178–186.

Balling, R.J. and Sobieszczanski-Sobieski, J., Optimization of Coupled Systems: A Critical Overview of Approaches., AIAA-94-4330-CP, Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, USA, 1994, pp. 697–707.

Barthelemy, J. F., Engineering Design Applications of Heuristic Multilevel Optimization Methods., Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Hampton, VA, Sep. 28-30, NASA CP - No 3031, 1988.

Barthelemy, J. F. and Sobieszczanski-Sobieski, J., Optimum Sensitivity Derivatives of Objective Functions in Nonlinear Programing., AIAA J., Vol. 22, No. 6, 1983, pp. 913–915.

Batill, S. M., Renaud, J. E. and Gu, X., Modeling and Simulation Uncertainty in Multidisciplinary Design Optimization., AIAA 200-4803, En: 8th AIAA/NASA/USAF/ ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA, Long Beach, California, Sep. 5-8, 2000.

Bloebaum, C. L., Hajela, P., and Sobieski, J., Non-Hierarchic System Decomposition in Structural Optimization., Engineering Optimization, Vol. 19, No. 3, 1992, pp. 171–186.

Braun, R. D. and Kroo, I. M., Development and application of the collaborative optimization architecture in a multidisciplinary design environment., En: N. M. Alexandrov y M. Y. Hussaini (Eds.) Multidisciplinary Design Optimization: State of the Art, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, Hampton, VA, SIAM, 1997.

Braun, R. D., Gage, P., Kroo, I. and Sobieski, I., Implementation and performance issues in collaborative optimization., En: 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellavue, WA, AIAA-96-4017, 1996, pp. 295–305.

Braun, R. D., Moore, A. A. and Kroo, I. M., Collaborative architecture to launch vehicle design., J. Spacecr. Rockets, Vol. 34, No. 4, 1997, pp. 478–486.

Chen, W., Allen, J. K., Tsui, K-L., and Mistree, F., A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors., ASME J. Mech. Des., Vol. 118, No. 4, 1996, pp. 478–485.

Chen, W. and Lewis, K., A Robust Design Approach for Achieving Flexibility in Multidisciplinary Design., AIAA J., Vol. 37, No. 8, 1999, pp. 982–990.

Danesh, M.R. and Jin, Y., An Agent-Based Decision Network for Concurrent Engineering Design., Concurrent Engineering, Vol. 9, No. 1, 2001, pp. 37–47.

Gu, X. Y. and Renaud, J. E., Implicit uncertainty propogation for robust collaborative optimization., En: Diaz A (Ed.) ASME Design Engineering Technical Conferences – Design Automation Conference, Pittsburgh, PA, DETC2001/DAC-21118, 2001.

Gu, X. Y., Renaud, J. E., Ashe, L. M., Batill, S. M., Budhiraja, A. S., and Krajewski, L. J., Decision-based collaborative optimization., ASME J. Mech. Des., Vol. 124, No. 1, 2002, pp. 1–13.

Gu, X., Renaud, J. E., Batill, S. M., Brach, R. M. and Budhiraja, A. S., Worst case propagated uncertainty of multidisciplinary systems in robust design optimization., Structural and Multidisciplinary Optimization, Vol. 20, No. 3, 2000, pp. 190–213.

Gu, X., Renaud, J.E., Ashe, L.M. and Batill, S.M., Decision-Based Collaborative Optimization under Uncertainty., En: ASME Design Engineering Technical Conferences, Baltimore, Maryland, DETC2000/DAC-14297, 2000.

Hall, D., Concurrent engineering: Defining terms and techniques., IEEE Spectrum, Vol. 28, No. 7, 1991, pp. 24-25.

Herskovits J., Mappa, P., Goulart, E. and Soares C. M. M., Mathematical programming models and algorithms for engineering design optimization., Computer Methods in Applied Mechanics and Engineering, Vol. 194, No. 30–33, 2005, pp. 3244–3268.

Kalsi, M., Hacker, K. and Lewis, K., A Comprehensive Robust Design Approach for Decision Trade-Offs in Complex Systems Design., ASME J. Mech. Des., Vol. 123, No. 1, 2001, pp. 1–10.

Kron, G., A Set of Principals to Interconnect the Solutions of Physical Systems., J. Appl. Physics, Vol. 24, No. 8, 1953, pp. 965.

Kroo, I., Altus, S., Braun, R., Gage, P. and Sobieski, I., Multidisciplinary Optimization Methods for Aircraft Preliminary Design., En: 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, AIAA 94-4325, 1994.

Lewis, K., and Mistree, F., Modeling Interactions in Multidisciplinary Design: A Game Theoretic Approach., AIAA J., Vol. 35, No. 8, 1997, pp. 1387–1392

Lewis, K., and Mistree, F., Collaborative, Sequential, and Isolated Decisions in Design., ASME J. Mech. Des., Vol. 120, No. 4, 1998, pp. 643–652.

Lewis, R.M., A Trust Region Framework for Managing Approximation Models in Engineering Optimization., AIAA-

96-4101-CP, En: 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September 4-6, 1996.

Marston, M. and Mistree, F., An Implementation of Expected Utility Theory in Decision Based Design., En: ASME Design Engineering and Technical Conferences, Atlanta, GA, DETC98/DTM-5670, 1998.

McAllister, C. D. and Simpson, T. W., Multidisciplinary robust design optimization of an internal combustion engine., ASME J. Mech. Des., Vol. 125, No. 1, 2003, pp. 124–130.

McAllister, C. D., Simpson, T. W., Hacker, K. and Lewis, K., Application of Multidisciplinary Design Optimization to Racecar Design and Analysis., En: 9th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Atlanta, GA, 2002.

McAllister, C. D., Simpson, T. W., Hacker, K., Lewis, K. and Messac, A., Integrating linear physical programming within collaborative optimization for multiobjective multidisciplinary design optimization., Struct. Multidisc. Optim., Vol. 29, No. 3, 2005, pp. 178-189.

McAllister, C. D., Simpson, T. W., Lewis, K. and Messac, A., Robust Multiobjective Optimization through Collaborative Optimization and Linear Physical Programming., En: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, 2004.

McAllister, C. D., Simpson, T. W. and Yukish, M., Goal programming applications in multidisciplinary design optimization., En: 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, AIAA-2000-4717, 2000.

Phadke, M. S., Quality Engineering Using Robust Design., Prentice Hall, Englewood Cliffs, New Jersey, 1989.

Prasad, B., Concurrent engineering fundamentals., Upper Saddle River, NJ: Prentice-Hall Inc., 1996.

Renaud, J. E., Sequential Approximation in Non-Hierarchic System Decomposition and Optimization: A Multidisciplinary Design Tool., Doctoral Dissertation, Rensselaer Polytechnic Institute, 1992.

Shankar, J., Ribbens, C., Haftka, R. and Watson, L., Computational study of nonhierarchical decomposition algorithm., Computational Optimization and Applications, Vol. 2, 1993, pp. 273–293.

Sobieski, I. P. and Kroo, I. M., Aircraft design using collaborative optimization., En: 34th Aerospace Sciences Meeting and Exhibition, Reno, NV, AIAA-96-0715, 1996.

Sobieszczanski-Sobieski, J., Optimization by decomposition: A step from hierarchic to non-hierarchic systems., En: 2nd NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Hampton, VA, 1988, NASA CP 3031, 1988a, pp. 1–27.

Sobieszczanski-Sobieski, J., Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design: Recent Advances and Results., Int'l Council for Aeronautical Sc., En: 16th Congress, Jerusalem. Aug.- Sept., Vol 2, 1988b, pp. 953-964.

Sobieszczanski-Sobieski, J., On the Sensitivity of Complex, Internally Coupled Systems., AIAA J., Vol. 28, No. 1, 1990a.

Sobieszczanski-Sobieski, J., Sensitivity Analysis of Complex Coupled Systems Extended to Second and Higher-Order Derivatives., AIAA J., Vol. 28, No. 4, 1990b.

Sobieszczanski-Sobieski, J., and Haftka, R. T., Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments., Structural Optimization, Vol. 14, No. 1, 1997, pp. 1–23.

Sobieszczanski-Sobieski. J., James, B. B. and Dovi, A. R., Structural Optimization by Multilevel Decomposition, AIAA J., Vol. 23, No. 11, 1985, pp. 1775–1782.

Sobiesznanski-Sobieski, J., A Linear Decomposition Method for Large Optimization Problems - Blueprint for Development., NASA TM 83248, 1982.

Tappeta, R. V. and Renaud, J. E., Multiobjective collaborative optimization., ASME J. Mech. Des., Vol. 119, No. 3, 1997, pp. 403–411.

Tappeta, R. V., Renaud, J. E., Messac, A. and Sundararaj, G. J., Interactive Physical Programming: Tradeoff Analysis and Decision Making in Multidisciplinary Optimization., AIAA J., Vol. 38, No. 5, 2000, pp. 917-926.

Womack, J., Jones, D. and Roos, D., The Machine that Changed the World., Rawsan Associates, New York, New York, 1990.

Wrenn, G. A. and Dovi, A. R., Multilevel Decomposition Approach to the Preliminary Sizing of a Transport Aircraft Wing., AIAA J. of Aircraft, Vol. 25, No. 7, 1988, pp. 632–638.

Yates, E. C., Aerodynamic Sensitivities from Subsonic, Sonic, and Supersonic Unsteady, Nonplanar Lifting-Surface Theory., NASA TM 100502, 1987.

Ziemke, M. C., and Spann, M. S., Warning: Don't be half-hearted in your efforts to employ concurrent engineering., Industrial Engineering, Vol. 23, No. 2, 1991, pp. 45-49.

Anexo

Para la implementación del problema a nivel de sistema en Matlab® utilizando la función fmincon del *toolbox* de optimización, se comienza escribiendo un archivo M que evalúe la función objetivo f en cualquier punto x.

```
function f = myfun(x)
    f = x(1)^2 + x(2)^2;
```

Las restricciones se reescriben en forma de "menor o igual que una constante", es decir,

$$g_1(\boldsymbol{x}) = x_1 + \beta x_2 \pounds 4$$

$$g_2(\boldsymbol{x}) = -\beta x_1 - x_2 \pounds -2. \qquad (6)$$

Como las dos restricciones son lineales, la región factible puede escribirse de la forma $\boldsymbol{Ax} = \boldsymbol{b}$, donde

```
beta = 0.5;
A = [1, beta; -beta, -1];
b = [4; -2];
```

A continuación, se determina el diseño inicial y se llama a la rutina de optimización,

```
x0 = [1; 1];
[x, fval] = fmincon(@myfun,x0,A,b)
```

Después de tres iteraciones y quince llamados a la función, el algoritmo converge en

```
x =
      0.8000
      1.6000
fval =
      3.2000
```

es decir, los valores óptimos son $x_1 = 0.8$ y $x_2 = 1.6$. Cuando el problema se descompone utilizando un esquema CO, los de optimización a nivel de subsistema se pueden resolver analíticamente de una manera sencilla. Considere, por ejemplo, el (4), en donde se debe obtener una solución para las variables locales x_{12} y x_{22}. El lagrangiano de este problema de optimización puede escribirse como

$$L = (x_{12} - x_1)^2 + (x_{22} - x_2)^2 + \lambda_2(-\beta x_{21} - x_{22} + 2), \qquad (7)$$

donde λ_2 es el multiplicador de Lagrange de la restricción del susbsistema 2. De la condición de optimalidad se obtiene que,

$$Lx_{21} = 2(x_{12} - x_1) - \beta\lambda_2 = 0 \qquad (8)$$

$$Lx_{22} = 2(x_{22} - x_2) - \lambda_2 = 0, \qquad (9)$$

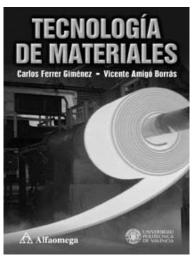
donde Lx_{21} y Lx_{22} representan las derivadas parciales del Lagrangiano respecto a las variables locales de diseño. La condición de complementaridad se satisface si

$$\lambda_2(-\beta x_{21} - x_{22} + 2) = 0 \qquad (10)$$

Combinando (8) y (9) en (10), se obtiene una ecuación de segundo nivel para resolver λ_2. La solución trivial de esta ecuación, $\lambda_2 = 0$, arroja, como resultado $x_{12} = x_1$ y $x_{22} = x_2$, con lo cual se satisface la restricción a nivel de sistema y arroja como resultado $x_1 = x_2 = 0$.

La otra solución, $\lambda_2 = 4/(\beta^2 + 1) > 0$, que también cumple con la condición de no negatividad, implica que $x_{12} = 2\beta/(\beta^2 + 1) + x_1$ y $x_{22} = 2/(\beta^2 + 1) + x_2$. Reemplazando por un valor $\beta = 0.5$, se obtiene $x_{12} = 0.8$ y $x_{22} = 1.6$. Con esto se comprueba que $x_1 = 0.8$ y $x_2 = 1.6$.

Alfaomega Colombiana S.A.



TECNOLOGÍADE MATERIALES

FERRER, Carlos y
AMIGÓ, Vicente
Rústica, 17 x 23 cm
ISBN 970-15-0879-3
EAN: 9789701508794
Coedición: Alfaomega-Universidad Politécnica de Valencia

NOVEDAD

Presenta una visión general de las propiedades de los materiales básicos que se usan en la industria de la fabricación, los ensayos que se les practicas para determinar propiedades y características. Permite conocer de manera sencilla y fácil algunos criterios acerca de los requerimientos básicos de los procesos industriales enfocados a la obtención de piezas máquinas o dispositivos, a partir de materiales específicos.

Resumen del contenido:
El concepto de tecnología de materiales Los procesos de fractura Procesos de deterioro superficial no corrosivo. Lubricación y desgaste Técnicas para la detección de defectos. Ensayos no destructivos Procesos de colada Conformado por deformación plástica Proceso de sinterización Procesos de tratamiento térmico Procesos de unión Procesos de modificación de superficies.

Adquiera nuestros textos en el punto de venta **Alfaomega Carrera 15 No 64a - 29** o en las principales librerías del país.

Afíliese a nuestro **CLUB DEL CONOCIMIENTO** a través de nuestra página web, y reciba descuentos en nuestro punto de venta, contenidos actualizados vía Internet, información de novedades, prioridad en productos promocionales y entregas a domicilio sin costo adicional.

Visite nuestra página Web:
www.alfaomega.com.co