



UNIVERSIDAD NACIONAL DEL SUR

TESIS DE DOCTOR EN INGENIERIA QUIMICA

**“RESOLUCION EFICIENTE DE PROBLEMAS
EN INGENIERIA DE PROCESOS: APROVECHAMIENTO
DE LA ESTRUCTURA DE LA MATRIZ DE INCIDENCIA”**

Alejandro Omar Domancich

BAHIA BLANCA

ARGENTINA

2009

PREFACIO

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctor en Ingeniería Química, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito del Departamento de Ingeniería Química durante el período comprendido entre el 13 de Septiembre de 2005 y el 3 de Noviembre de 2009, bajo la dirección de la Dra. Patricia Mónica Hoch, Profesora Adjunta del Dpto. de Ingeniería Química e Investigadora Adjunta del CONICET y la Dra. Nélida Beatriz Brignole, Profesora Adjunta del Dpto. de Ciencias e Ingeniería de la Computación e Investigadora Independiente del CONICET.

Alejandro Omar Domancich



UNIVERSIDAD NACIONAL DEL SUR
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el/..../..... , mereciendo la calificación de(.....)

AGRADECIMIENTOS

Deseo expresar en primer término mi más profundo agradecimiento a las Dras. Nélide Beatriz Brignole y Patricia Mónica Hoch, por su invaluable colaboración y espíritu de trabajo, por la excelencia de la formación que me brindaron y constante estímulo. Su dirección conjunta resultó un pilar fundamental para la concreción de esta Tesis, transmitiéndome sus conocimientos en los diversos campos de investigación en los que se enfocó este trabajo.

Al Consejo Nacional de Investigaciones Científicas y Técnicas, a la Planta Piloto de Ingeniería Química y al Departamento de Ingeniería Química, mi agradecimiento por haberme facilitado los medios para realizar esta tarea.

Mi agradecimiento al Colegio Don Bosco por brindarme la formación ética y moral para realizarme como ser humano, por su apoyo y su constante aliento durante mis doce años en dicha Institución.

Mi especial reconocimiento al Dr. Ignacio Ponzoni, quien me brindó en estos años su valiosa y desinteresada colaboración, sin la cual hubiese resultado imposible llevar adelante el desarrollo de esta Tesis.

A todas las personas que, durante estos años, me acompañaron y ayudaron, me aportaron ideas y con quienes compartí horas de trabajo, brindándome su conocimiento en áreas ajenas a mi especialidad.

Mi especial agradecimiento a Andrea, quien siempre estuvo a mi lado, tanto para apoyarme y ayudarme a superar los momentos difíciles transcurridos en este tiempo como para disfrutar conmigo de los momentos felices que se presentaron en este período.

Y por último, quiero expresar mi mayor gratitud a mis padres, mis hermanos, mi abuela y mis cuñadas, quienes con su afecto y constante aliento hicieron posible que concretara esta tesis. A ellos mi agradecimiento por el cariño y apoyo brindados.

A mis padres
A Andrea

RESUMEN

El objetivo de esta Tesis ha sido desarrollar métodos para análisis estructural de modelos matemáticos y resolución de problemas en ingeniería de procesos. Estos métodos pueden ser aplicados a los campos de diseño de instrumentación, simulación y optimización de procesos químicos. Se utilizó como herramienta el Método Directo (MD), que es una técnica novedosa de análisis estructural. El MD es un nuevo algoritmo de particionamiento que efectúa un reordenamiento estructural de la matriz de incidencia correspondiente al modelo de estado estacionario de un proceso, para llevarla a una forma triangular inferior en bloques y establecer una descomposición en subsistemas y un orden de precedencia.

La Tesis puede dividirse en dos partes: un análisis cualitativo de procesos químicos (estudio de modelos desde un punto de vista estructural) y un análisis cuantitativo (resolución de modelos matemáticos). El primer tipo de análisis resulta aplicable al campo de diseño de instrumentación, mientras que el segundo es aplicable a los campos de simulación y optimización de procesos.

La contribución original realizada en el campo de diseño de instrumentación está dada por el desarrollo de una versión modificada del MD, denominada Método Directo Extendido (MDE). Este nuevo algoritmo consiste en la implementación de un módulo que cuantifica la no linealidad inherente a un sistema de ecuaciones. Este módulo guía al MD tradicional hacia la descomposición del modelo matemático en subsistemas con el menor grado de no linealidad posible, es decir potencialmente más fáciles de resolver. Mediante la aplicación del MDE, se obtiene una reducción en la no linealidad de los subsistemas obtenidos a través del análisis de observabilidad. Al mismo tiempo, el algoritmo es capaz de incrementar la cantidad de bloques generados de mínimo tamaño. La aplicación del MDE favorece el proceso global de diseño de instrumentación en tres aspectos principales: incremento en la velocidad de procesamiento de datos, provisión de conocimiento adicional acerca del nivel de redundancia del modelo y simplificación en la tarea de reconciliación de datos de una planta.

Con respecto a los campos de simulación y optimización de procesos, la contribución realizada consistió en desarrollar un paquete completo que permite generar, particionar y resolver modelos matemáticos pertenecientes a plantas químicas; no sólo tratando los casos cuadrados, sino también los modelos que dan lugar a estructuras rectangulares. La resolución se realiza luego de efectuadas dos etapas. En la primera, el usuario obtiene de manera

automática el sistema de ecuaciones que representa el funcionamiento de una planta de procesos a partir del ingreso al programa de la topología de la misma. En una segunda etapa, el programa particiona el modelo obtenido (utilizando de manera interna el MDE) y brinda una estrategia de resolución en bloques. Luego, se lleva a cabo la simulación u optimización del modelo dependiendo de las características del mismo. El paquete desarrollado es sencillo de utilizar y flexible para cualquier proceso químico con el que se desee trabajar, combinando además las ventajas que naturalmente posee el enfoque de simulación secuencial modular con las del enfoque orientado a ecuaciones.

ABSTRACT

The general purpose of this thesis was to develop methods for both the structural analysis of mathematical models and the resolution of problems in process engineering. These methods can be applied to instrumentation design, simulation and optimization fields. The Direct Method (DM), which is a novel technique for structural analysis, was employed. Based on graph theory, the DM carries out a structural reordering of the incidence matrix that corresponds to the steady-state model of a process. This restructuring leads to a block-lower triangular form. The partitioning implicitly defines both a block-decomposition and the precedence order among blocks.

The thesis can be divided in two main parts: a qualitative analysis of chemical processes (from a structural point of view, it is a model study) and a quantitative analysis (resolution of mathematical models). The first kind of examination is practical for instrumentation design, while the second one can be applied to process simulation and optimization fields.

The original contribution made in the field of instrumentation design is the development of a modified DM version, which was called the Extended Direct Method (EDM). This algorithm consists in the implementation of a module that quantifies the nonlinearity inherent to an equation system. This module guides the traditional DM towards the decomposition of the mathematical model in subsystems with the smallest possible degree of nonlinearity, i.e. potentially easier to solve. The results show that the EDM is able to reduce the nonlinearity of the subsystems obtained through the observability analysis procedure. At the same time, it can successfully increase the number of minimum size subsystems. The EDM application contributes to the improvement of the DSS' global performance in three main aspects: it increases the data processing speed; it provides additional knowledge about the redundancy level of the model and it simplifies the data-reconciliation task for a plant.

Regarding the fields of process simulation and optimization, the main contribution consists in the development of a complete package for the generation, the partition and the resolution of mathematical models that belong to chemical plants, not only treating the square models, but also dealing with rectangular structures. The resolution is carried out after having completed two main stages. Firstly, the user obtains automatically the equation system that

represents the process plant behaviour, by feeding the plant topology to the software programme. In the second stage, the programme partitions the model (by internally using the EDM) and proposes a block-resolution strategy. Then, the simulation and optimization are carried out, depending on the model characteristics. The developed package is user-friendly and flexible for any chemical process, thus combining the advantages of the sequential modular approach with those of the equation-oriented one.

ÍNDICE

Prefacio	i
Resumen	v
Abstract	vii
Índice	ix
Índice de figuras	xvii
Índice de tablas	xxi
Lista de Publicaciones	xxiii

CAPÍTULO 1: INTRODUCCIÓN

1.1 INTRODUCCIÓN	1
1.2 OBJETIVOS	3
1.3 CONTENIDOS Y ESTRUCTURA	6

PARTE 1: ANÁLISIS CUALITATIVO DE PROCESOS

CAPÍTULO 2: SISTEMA DE SOPORTE A LA DECISIÓN PARA DISEÑO DE INSTRUMENTACIÓN

2.1 INTRODUCCIÓN	7
2.2 DEFINICIÓN DE LA MATRIZ DE INCIDENCIA	9
2.3 CLASIFICACIÓN DE LAS VARIABLES Y ECUACIONES DE UN MODELO	10
2.4 EJEMPLO DE ANÁLISIS DE OBSERVABILIDAD	11
2.5 EJEMPLO DE ANÁLISIS DE REDUNDANCIA EN EL MODELO MATEMÁTICO DE UN EQUIPO	15
2.6 ESTRUCTURA GENERAL DEL DSS	17
2.7 ETAPAS PARA LLEVAR A CABO EL DISEÑO DE INSTRUMENTACIÓN DE UNA PLANTA DE PROCESOS	19
2.8 CONCLUSIONES	21
2.9 NOMENCLATURA DEL CAPÍTULO	21

CAPÍTULO 3: MÉTODO DIRECTO PARA ANÁLISIS DE OBSERVABILIDAD

3.1	INTRODUCCIÓN	23
3.2	CONCEPTOS BÁSICOS DE TEORÍA DE GRAFOS	24
3.2.1	GRAFOS, DIGRAFOS Y NOCIONES DE CONECTIVIDAD.....	24
3.2.1.1	CONCEPTOS BÁSICOS	25
3.2.1.2	CONCEPTO DE CAMINATA EN UN GRAFO.....	26
3.2.2	GRAFOS BIPARTITOS	26
3.2.3	PAREAMIENTOS MAXIMALES	27
3.3	ARQUITECTURA DEL MÉTODO DIRECTO	28
3.3.1	ETAPA 0: INICIALIZACIÓN.....	29
3.3.2	ETAPA 1: DESCOMPOSICIÓN GRUESA.....	29
3.3.3	ETAPA 2: DESCOMPOSICIÓN FINA.....	31
3.3.4	ETAPAS 3, 4 Y 5: TEST DE ADMISIÓN, REASIGNACIÓN Y REDUCCIÓN.....	31
3.3.5	ETAPA 6: REORDENAMIENTO FINAL.....	33
3.4	ALGORITMO DEL MÉTODO DIRECTO	33
3.5	EJEMPLO DE APLICACIÓN DEL MÉTODO DIRECTO	36
3.6	CONCLUSIONES	40
3.7	NOMENCLATURA DEL CAPÍTULO	41

CAPÍTULO 4: MÉTODO DIRECTO EXTENDIDO

4.1	INTRODUCCIÓN	43
4.2	DEFINICIÓN DEL GRADO DE NO LINEALIDAD	44
4.2.1	GRADO DE NO-LINEALIDAD EN LAS ECUACIONES.....	44
4.2.2	GRADO DE NO-LINEALIDAD EN LAS VARIABLES:.....	45
4.3	CUANTIFICACIÓN DEL GRADO DE NO LINEALIDAD	45
4.3.1	CUANTIFICACIÓN EN ECUACIONES.....	46
4.3.2	CUANTIFICACIÓN EN VARIABLES:	46
4.3.3	EJEMPLOS DE CUANTIFICACIÓN DEL GRADO DE NO LINEALIDAD	46

4.4	INTEGRACIÓN DEL GRADO DE NO LINEALIDAD.....	49
4.5	MÓDULOS Y ENCABEZADOS QUE CONFORMAN EL MDE.....	51
4.6	ARCHIVOS DE ENTRADA Y SALIDA DEL MDE	53
4.6.1	ARCHIVOS DE ENTRADA	53
4.6.2	ARCHIVOS DE SALIDA.....	53
4.7	COMPARACIÓN DE DESEMPEÑO ENTRE EL MDE Y EL MD	
	ORIGINAL.PRESENTACIÓN DE CASOS DE ESTUDIO.....	54
4.7.1	COLUMNA DE DESTILACIÓN.....	55
4.7.1.1	DESCRIPCIÓN DEL PROCESO INDUSTRIAL.....	55
4.7.1.2	EVALUACIÓN EXPERIMENTAL DE LOS PESOS A UTILIZAR	55
4.7.1.3	ANÁLISIS DE OBSERVABILIDAD UTILIZANDO EL MD Y EL MDE	58
4.7.1.4	ANÁLISIS DE REDUNDANCIA DEL PROCESO A PARTIR DE LAS DESCOMPOSICIONES OBTENIDAS POR EL MD Y EL MDE	58
4.7.1.5	RESUMEN DEL IMPACTO DEL MDE EN EL PROCESO GLOBAL DE DISEÑO Y ANÁLISIS DE INSTRUMENTACIÓN	62
4.7.2	PLANTA DE SÍNTESIS DE AMONÍACO.....	63
4.7.2.1	DESCRIPCIÓN DEL PROCESO INDUSTRIAL.....	63
4.7.2.2	ANÁLISIS DE OBSERVABILIDAD UTILIZANDO EL MD Y EL MDE	64
4.7.3	PLANTA DE PRODUCCIÓN DE ETANO	65
4.7.3.1	DESCRIPCIÓN DEL PROCESO INDUSTRIAL.....	65
4.7.3.2	ANÁLISIS DE OBSERVABILIDAD UTILIZANDO EL MD Y EL MDE	67
4.8	CONCLUSIONES	67
4.9	NOMENCLATURA DEL CAPÍTULO	69

PARTE 2: ANÁLISIS CUANTITATIVO DE PROCESOS

CAPÍTULO 5: GENERADOR AUTOMÁTICO DE MODELOS PARA SIMULACIÓN Y OPTIMIZACIÓN DE PROCESOS

5.1	INTRODUCCIÓN	71
5.2	SIMULACIÓN DE PROCESOS	73

5.2.1	ETAPAS EN EL DESARROLLO DE UN MODELO PARA SIMULACIÓN	74
5.2.2	CLASIFICACIÓN DE LOS SIMULADORES DE PROCESOS.....	75
5.2.2.1	ENFOQUE SECUENCIAL MODULAR.....	75
5.2.2.2	ENFOQUE ORIENTADO A ECUACIONES	76
5.3	OPTIMIZACIÓN DE PROCESOS	79
5.3.1	MÉTODOS DE OPTIMIZACIÓN.....	80
5.4	GENERADOR DE MODELOS PARA SIMULACIÓN Y OPTIMIZACIÓN DE PROCESOS INDUSTRIALES.....	80
5.4.1	OBJETIVOS DE DISEÑO DEL SOFTWARE Y CARACTERÍSTICAS PRINCIPALES DEL MISMO.....	80
5.4.2	MODELADO RIGUROSO PARA SIMULACIÓN Y OPTIMIZACIÓN	82
5.4.3	DESCRIPCIÓN DEL SOFTWARE.....	84
5.4.4	CASO DE ESTUDIO: COLUMNA DE DESTILACIÓN REACTIVA.....	92
5.4.4.1	DESCRIPCIÓN DEL PROCESO DE DESTILACIÓN REACTIVA.....	92
5.4.4.2	SIMULACIÓN Y OPTIMIZACIÓN DE LA COLUMNA REACTIVA	95
5.5	CONCLUSIONES	95
5.6	NOMENCLATURA DEL CAPÍTULO	96

CAPÍTULO 6: MP4SO: UN SOFTWARE DE PARTICIÓN DE MODELOS PARA LA SIMULACIÓN Y OPTIMIZACIÓN DE PROCESOS INDUSTRIALES

6.1	INTRODUCCIÓN.....	97
6.2	UTILIZACIÓN DEL MDE Y GAMS PARA SIMULACIÓN Y OPTMIZACIÓN DE PROCESOS INDUSTRIALES.....	99
6.2.1	DESCOMPOSICIÓN DE MODELOS MATEMÁTICOS MEDIANTE LA UTILIZACIÓN DEL MÉTODO DIRECTO EXTENDIDO.....	99
6.2.2	SISTEMA GENERAL DE MODELADO ALGEBRAICO (GAMS)	100
6.2.3	RESOLUCIÓN EN BLOQUES LLEVADA A CABO MEDIANTE LA UTILIZACIÓN DEL MÉTODO DIRECTO EXTENDIDO Y GAMS	101

6.3	MP4SO: UN SOFTWARE PARA LA INTEGRACIÓN DEL MÉTODO DIRECTO EXTENDIDO Y GAMS	103
6.3.1	ETAPAS QUE CONFORMAN MP4SO	103
6.3.2	MP4SO: PRESENTACIÓN DEL SOFTWARE.....	108
6.4	ANÁLISIS DE PERFORMANCE DEL SOFTWARE	113
6.4.1	SIMULACIÓN DE PROCESOS	114
6.4.1.1	CRAQUEO CATALÍTICO DE GASOIL	114
6.4.1.2	COLUMNA DE DESTILACIÓN REACTIVA	115
6.4.2	OPTIMIZACIÓN DE PROCESOS.....	116
6.4.2.1	RED DE INTERCAMBIADORES DE CALOR	116
6.4.2.2	UBICACIÓN ÓPTIMA DE CATALIZADOR EN UN REACTOR TUBULAR.....	117
6.4.2.3	PROCESO DE HDA PARA PRODUCCIÓN DE BENCENO.....	119
6.4.3	ANÁLISIS DE LOS RESULTADOS OBTENIDOS	121
6.5	CONCLUSIONES	121

CAPÍTULO 7: CONCLUSIONES Y TRABAJOS FUTUROS

7.1	CONCLUSIONES	126
7.1.1	SECCIÓN I: ANÁLISIS CUALITATIVO DE PROCESOS.....	126
7.1.2	SECCIÓN II: ANÁLISIS CUANTITATIVO DE PROCESOS	128
7.2	TRABAJOS FUTUROS	130

APÉNDICE A: METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA INGENIERÍA

A.1	INTRODUCCIÓN	133
A.2	DEFINICIÓN DE LA METODOLOGÍA APLICADA EN LA TESIS PARA DESARROLLO DE SOFTWARE	134

APÉNDICE B: EXTENSIÓN DE LAS FACILIDADES DE SIMULACIÓN DE HYSYS: INCORPORACIÓN DE MÓDULOS DEFINIDOS BAJO EL ESTÁNDAR CAPE-OPEN

B.1	INTRODUCCIÓN	139
B.2	PROYECTO CAPE OPEN	141
B.2.1	ALCANCES DEL PROYECTO CAPE-OPEN	141
B.2.2	INTERFACES DE LAS UNIDADES DE OPERACIÓN	143
B.3	DESARROLLO DE EXTENSIONES DE FORTRAN A HYSYS.....	143
B.3.1	ESTRATEGIA DE MIGRACIÓN DE MODELOS DE FORTRAN A HYSYS	144
B.3.2	CASO DE ESTUDIO: PLANTA DE AMONÍACO.....	146
B.3.3	RESULTADOS	147
B.3.3.1	CÓDIGO EN VISUAL BASIC PARA EL REACTOR DE AMONÍACO.....	147
B.3.3.2	REACTOR DE AMONÍACO PROBADO EN HYSYS.....	149
B.4	DESARROLLO DE EXTENSIONES DE GAMS A HYSYS.....	150
B.4.1	ESTRATEGIA DE MIGRACIÓN DE MODELOS DE GAMS A HYSYS.....	150
B.4.1.1	DESARROLLO DE MÓDULOS COMPATIBLES CON CAPE-OPEN.....	150
B.4.1.2	DEFINICIÓN DE LOS PASOS A SEGUIR PARA DESARROLLAR MÓDULOS COMPATIBLES CON CAPE OPEN A PARTIR DE MODELOS DESARROLLADOS EN GAMS	152
B.4.2	CASO DE ESTUDIO: COLUMNA DE DESTILACIÓN REACTIVA.....	153
B.4.3	RESULTADOS	153
B.4.3.1	CÓDIGO DE VISUAL BASIC PARA LA COLUMNA REACTIVA.....	154
B.5	CONCLUSIONES	156

APÉNDICE C: INCORPORACIÓN DE EQUIPOS AL GENERADOR DE MODELOS POR MEDIO DE SU DEFINICIÓN EN LENGUAJE XML

C.1	INTRODUCCIÓN	159
C.2	LENGUAJE XML PARA INCORPORACIÓN DE EQUIPOS	160

C.2.1	INTRODUCCIÓN AL LENGUAJE XML	160
C.2.2	CARACTERÍSTICAS DEL LENGUAJE.....	160
C.2.3	DOCUMENTOS XML BIEN FORMADOS	161
C.2.4	APLICACIÓN DEL LENGUAJE XML PARA LA INCORPORACIÓN DE EQUIPOS AL GENERADOR AUTOMÁTICO DE MODELOS	161

APÉNDICE D: MODELO MATEMÁTICO DE LA COLUMNA DE DESTILACION REACTIVA

D.1	MODELOS MATEMÁTICOS UTILIZADOS	167
D.1.1	MODELO MATEMÁTICO DEL EQUIPO.....	167
D.1.1.1	ETAPAS NO REACTIVAS	167
D.1.1.2	ETAPAS REACTIVAS	168
D.1.1.3	CONDENSADOR	169
D.1.1.4	REBULLIDOR	170
D.1.2	MODELO DE EQUILIBRIO	170
D.1.3	MODELO GEOMÉTRICO (Douglas, 1988).....	171
D.1.3.1	ALTURA DE LA COLUMNA.....	171
D.1.3.2	DIÁMETRO DE LA COLUMNA.....	171
D.1.3.3	ÁREA DEL CONDENSADOR.....	171
D.1.3.4	ÁREA DEL REBULLIDOR.....	172
D.1.4	COSTOS DE INSTALACIÓN (Douglas, 1988).....	172
D.1.4.1	CASCO DE LA COLUMNA	172
D.1.4.2	PLATOS DE LA COLUMNA.....	172
D.1.4.3	CATALIZADOR	172
D.1.4.4	CONDENSADOR	172
D.1.4.5	REBULLIDOR	172
D.1.5	COSTO ANUAL DE OPERACIÓN	172
D.1.5.1	AGUA DE ENFRIAMIENTO.....	172
D.1.5.2	VAPOR SATURADO	173
D.1.6	COSTOS TOTALES	173
D.1.6.1	COSTO TOTAL DE INSTALACIÓN	173

D.1.6.2	COSTO TOTAL ANUAL DE OPERACIÓN	173
D.1.6.3	COSTO TOTAL ANUALIZADO	173
D.2	SIMULACIÓN DE LA COLUMNA: PRODUCCIÓN DE MTBE	173
D.2.1	INFLUENCIA DEL NÚMERO DE PLATOS REACTIVOS	175
D.2.2	INFLUENCIA DEL CAUDAL DE FONDO	178
D.3	OPTIMIZACIÓN DEL PROCESO DE PRODUCCIÓN DE MTBE	179
D.3.1	FORMULACIÓN DEL PROBLEMA DE OPTIMIZACIÓN	180
D.3.2	RESULTADOS DE LA OPTIMIZACIÓN	182
D.3.3	COMPARACIÓN CON PROCESO CONVENCIONAL	183
D.4	NOMENCLATURA	184
	REFERENCIAS	185

LISTA DE PUBLICACIONES

EN REVISTAS CIENTÍFICAS Y LIBROS CON REFERATO INTERNACIONAL

Domancich A., Durante M., Ferraro S., Hoch P., Brignole N., Ponzoni. “How to improve the model partitioning in a DSS for instrumentation design”, *I. Industrial & Engineering Chemistry Research*, Vol. 48 (7) pp 3513-3525, ISSN 0888-5885. **2009.**

Domancich A.O., Perez V.L., Hoch P.M, Brignole N.B. “Systematic generation of a Cape-Open Compliant Simulation Module From GAMS and FORTRAN Models”, *Chemical Engineering Research and Design*, ISSN 0263-8762. En prensa. doi:10.1016/j.cherd.2009.07.022. **2009.**

Domancich A.O., Maidana M, Hoch P.M, Brignole N.B, Ponzoni, I. “MP4SO: A Model-Partitioning Software for Simulation and Optimization”, *Computer-Aided Chemical Engineering*, Vol. 27, Ed. Elsevier. ISSN 1570-7946, pp 471-476. **2009.**

Domancich A.O., Brignole N.B., Hoch P.M, “Structural analysis of reactive distillation columns”, *VirtualPro* (<http://www.revistavirtualpro.com>), N°84, pp. 18. Editorial VirtualPro. ISSN 1900-6241, **2009.**

Domancich A.O., Perez V.L., Hoch P.M, Brignole N.B. “Development of a CAPE-OPEN Simulation Component for a GAMS-modeled Process Unit”, *Computer-Aided Chemical Engineering*, Vol. 25, Ed. Braunschweig, Joulia, Elsevier. ISBN 978-0-444-53228-2 (versión en CD), **2008.**

Domancich A.O., Brignole N.B., Hoch P.M. "Optimal Structure of Reactive and Non-reactive Stages in Reactive Distillation Processes", *Chemical Engineering Transactions*, Ed. Jiri Klemes, AIDIC Servizi S:R.L. ISBN 88-901915-4-6., Vol. 12, pp .85-90, **2007.**

EN REVISTAS NACIONALES CON SELECCIÓN DE TRABAJOS A CONGRESOS

Domancich A. O., Touceda T., Brignole, N. B., Hoch P.M. “Generador Automático de Modelos para Simulación y Optimización de Procesos Industriales”, *Mecánica Computacional*, en prensa, **2009**.

Domancich A, Ardenghi J.I., Vazquez G.E., Brignole N.B. “Desempeño de un Algoritmo Espectral para Optimización Rigurosa de Plantas de Procesos Industriales”, *Mecánica Computacional*, Vol. XXIII, 3047-3057. ISSN: 1666-6070, **2004**.

EN CONFERENCIAS Y REUNIONES CIENTÍFICAS CON COMITÉ DE REVISIÓN

INTERNACIONALES

Domancich A. O., Touceda T., Brignole, N. B., Hoch P.M. “A Model Generator for Process Simulation and Optimization”, *AIChE Annual Meeting 2009*, Nashville, TN, EEUU, 8-13/11/**2009**.

Domancich A.O., Maidana M, Hoch P.M, Brignole N.B, Ponzoni, I. “MP4SO: A Model-Partitioning Software for Simulation and Optimization”, *PSE'09: 10th International Symposium on Process Systems Engineering*. Salvador Bahia, Brasil. 16-20/08/**2009**.

Domancich A.O., Perez V.L., Hoch P.M, Brignole N.B. “Development of a CAPE-OPEN Simulation Component for a GAMS-modeled Process Unit”, *ESCAPE 18: 18th European Symposium on Computer Aided Process Engineering*, Lyon, France, 1-4/06/**2008**.

Domancich A.O., Brignole N.B., Hoch P.M. "Optimal Structure of Reactive and Non-reactive Stages in Reactive Distillation Processes", *PRES '07. 10th Conference on Process Integration, Modeling and Optimization for Energy Saving and Pollution Reduction*, Ischia, Gulf of Naples, Italy, 24-27/06/**2007**.

Domancich A.O., Brignole N.B., Hoch P.M. “Design of Reactive Distillation Systems Using a Process Systems Engineering Approach”, *AIChE Annual Meeting 2007*, Salt Lake City, EEUU, 4-9/11/ **2007**.

Domancich A.O., Olivera A.C., Brignole N.B., Hoch P.M. “Performance Analysis of Reactive Distillation Columns: Adjustment of Reactive and Non-reactive Stages”, *XXII Interamerican Congress of Chemical Engineering*, ISSN 1850 3535, Buenos Aires, Argentina 1-4/10/2006.

Perez V. L., Domancich A., Vazquez G.E., Brignole N.B. “A Cape-Open Compliant Simulation Module For An Ammonia Reactor Unit”, *ENPROMER 2005: 2nd Mercosur Congress on Chemical Engineering*, ISBN: 85-7650-042-6, Rio das Pedras, Brasil, 14-18/8/2005.

NACIONALES

Domancich A. O., Touceda T., Brignole, N. B., Hoch P.M. “Generador Automático de Modelos para Simulación y Optimización de Procesos Industriales”, *ENIEF 2009: 18° Congreso sobre Métodos Numéricos y sus Aplicaciones*, Tandil, Argentina, 3-6/11/2009.

Domancich A.O., Durante M., Ferraro S., Hoch P.M., Brignole N.B., Ponzoni I. “DSS para Diseño de Instrumentación: Técnicas para Mejorar la Descomposición de Modelos”, *36 JAIIO: 36th Conference on Computer Science and Operational Research-* ISSN 1850-2776, Mar del Plata, Argentina 30/8/2007.

Olivera A.C., Domancich A.O., Hoch P.M. , Brignole N.B. “Building a Reactive Column Model for Both a DSS Frame and a Simulation Tool”, *AST2006: 7th Argentine Symposium On Computing Technology- 35 JAIIO: 35th Conference on Computer Science and Operational Research- 35° Jornadas Argentinas de Informática e Investigación Operativa.,* ISSN 1850-2806, Mendoza, Argentina 4-8/9/2006.

Domancich A, Ardenghi J.I., Vazquez G.E., Brignole N.B. “Desempeño de un Algoritmo Espectral para Optimización Rigurosa de Plantas de Procesos Industriales”, *ENIEF 2004: 14° Congreso sobre Métodos Numéricos y sus Aplicaciones, Bariloche, Argentina, 8-11/11/2004.* Publicado en CD (ISSN 1666-6070)

Carballido J.A., Ferraro S. J., Domancich A.O., Ponzoni I., Brignole N.B., “Un Nuevo AGMO para Diseño de Instrumentación: Aplicación a una Planta de Amoníaco” AADECA 2005: XIX Congreso Nacional de Control Automático, ISBN: 950-99994-3-1, Buenos Aires, Argentina, 30/8-3/9/2004.

ÍNDICE DE FIGURAS

<i>Figura 2.1. Sistema de ecuaciones que representa un proceso industrial.....</i>	<i>9</i>
<i>Figura 2.2. Matriz de incidencia correspondiente al Ejemplo 2.1.....</i>	<i>10</i>
<i>Figura 2.3. Sistema de ecuaciones que representa un proceso industrial.....</i>	<i>11</i>
<i>Figura 2.4. Matriz de incidencia correspondiente al sistema de la figura 2.3.....</i>	<i>12</i>
<i>Figura 2.5. Conjunto arbitrario de mediciones impuesto al sistema de la figura 2.4.....</i>	<i>12</i>
<i>Figura 2.6. Matriz de incidencia resultante de la medición de las variables x_6, x_7 y x_8.....</i>	<i>13</i>
<i>Figura 2.7. Matriz de incidencia resultante de la aplicación del algoritmo de observabilidad... </i>	<i>14</i>
<i>Figura 2.8. Subsistemas obtenidos a partir de la aplicación del algoritmo de observabilidad... </i>	<i>15</i>
<i>Figura 2.9. Mezclador con dos corrientes de entrada y una de salida.....</i>	<i>16</i>
<i>Figura 2.10. Sistema de ecuaciones correspondiente al modelo del mezclador.....</i>	<i>16</i>
<i>Figura 2.11. Conjunto de mediciones necesario para que las variables de la corriente de salida del mezclador resulten observables.....</i>	<i>17</i>
<i>Figura 2.12. Esquema de interacción entre el usuario y el DSS para efectuar el diseño de instrumentación de una planta química.....</i>	<i>18</i>
<i>Figura 3.1. Estructura de la matriz FTiB.....</i>	<i>24</i>
<i>Figura 3.2. Representación gráfica de un grafo no dirigido G.....</i>	<i>25</i>
<i>Figura 3.3. Representación gráfica de un bigrafo G.....</i>	<i>27</i>
<i>Figura 3.4. Algoritmo de búsqueda del PMB.....</i>	<i>28</i>
<i>Figura 3.5. Arquitectura del Método Directo.....</i>	<i>29</i>
<i>Figura 3.6. Clasificación de nodos y descomposición gruesa para ambas técnicas.....</i>	<i>30</i>
<i>Figura 3.7. Algoritmo del Método Directo.....</i>	<i>35</i>
<i>Figura 3.8. Matriz de incidencia del modelo matemático.....</i>	<i>36</i>
<i>Figura 3.9. Pareamiento Maximal del bigrafo $G(N)$.....</i>	<i>36</i>
<i>Figura 3.10. Matriz reordenada a partir de la descomposición gruesa.....</i>	<i>37</i>
<i>Figura 3.11. Descomposición Fina de $N_2 = (SC2, SR2)$.....</i>	<i>38</i>
<i>Figura 3.12. Matriz resultante de la primera reasignación.....</i>	<i>39</i>
<i>Figura 3.13. Matriz resultante de la segunda reasignación.....</i>	<i>39</i>
<i>Figura 4.1. Sistema de ecuaciones que representa un proceso industrial.....</i>	<i>48</i>
<i>Figura 4.2. Versión modificada del algoritmo PMB.....</i>	<i>50</i>
<i>Figura 4.3. Esquema de la columna de destilación.....</i>	<i>55</i>
<i>Figura 4.4. Esquema de la planta de síntesis de amoníaco.....</i>	<i>64</i>

Figura 4.5. Esquema simplificado de la planta de Etano.....	66
Figura 4.6. Sector criogénico de la planta de Etano.....	66
Figura. 5.1. Ejemplo de sistema de ecuaciones generado por el MGM para llevar a cabo el análisis de observabilidad de una planta de procesos.....	83
Figura. 5.2. Ejemplo de sistema de ecuaciones generado por el nuevo generador de modelos para simulación y optimización de procesos.....	83
Figura. 5.3. Captura de pantalla del área principal del software para un caso de estudio....	84
Figura 5.4. Ventana “General” del equipo para definición de nombre del mismo y notas....	86
Figura 5.5. Definición de las corrientes de entrada y salida del equipo.....	87
Figura 5.6. Pantalla de información sobre las fases que permite el equipo.....	87
Figura 5.7. Definición de los parámetros del equipo.....	88
Figura 5.8. Definición de las ecuaciones que forman parte del modelo matemático del equipo..	89
Figura 5.9. Definición de las propiedades de la corriente. Pantalla “General”.....	90
Figura 5.10. Definición de los compuestos que integran la corriente.....	90
Figura 5.11. Definición de las variables medidas de la corriente.....	91
Figura 5.12. Selección de las ecuaciones que modelan las propiedades físicas y químicas de la corriente.....	92
Figura. 5.13. Comparación entre proceso convencional y destilación reactiva.....	93
Figura 6.1. Figuras que conforman MP4SO	104
Figura 6.2. Captura de pantalla del área de texto de MP4SO.....	109
Figura 6.3. Captura de pantalla del menú “Model”.....	110
Figura 6.4. Captura de pantalla del menú “Partition”.....	111
Figura 6.5. Captura de pantalla del menú “Simulation”.....	112
Figura 6.6. Esquema del proceso de craqueo catalítico de gas oil.....	114
Figura 6.7. Esquema del proceso de optimización de la red de intercambiadores de calor....	116
Figura 6.8. Esquema de un reactor tubular flujo pistón.....	118
Figura 6.9. Esquema del proceso de HDA para producción de benceno.....	120
Figura A.1. Esquema de la metodología para desarrollo de software aplicada en esta tesis....	136
Figura B.1. Integración de las diferentes herramientas de CAPE-OPEN.....	142
Figura B.2. Proceso de encapsulado del código de FORTRAN.....	146
Figura B.3. Código desarrollado en Visual Basic para implementar la Unidad de Operación correspondiente al reactor de amoníaco.....	148
Figura B.4. Unidad de Operación CO disponible en HYSYS.....	149

<i>Figura B.5. Unidad de Operación CO incorporada a la simulación de la planta.....</i>	<i>150</i>
<i>Figura B.6. Unidad de Operación CO incorporada a la simulación de la planta.....</i>	<i>152</i>
<i>Figura B.7. Ciclo de transferencia de datos entre GAMS y HYSYS.....</i>	<i>154</i>
<i>Figura B.8. Código de Visual Basic para la implementación de la Unidad de Operaciones perteneciente a la columna de destilación reactiva.....</i>	<i>156</i>
<i>Figura C.1. Diagrama de relaciones entre las etiquetas que conforman un equipo.....</i>	<i>162</i>
<i>Figura D.1. Esquema de la columna de destilación reactiva.....</i>	<i>174</i>
<i>Figura D.2. Conversión obtenida en función del número de etapas reactivas.....</i>	<i>176</i>
<i>Figura D.3. Perfiles de composición molar de MTBE a lo largo de la columna en función del número de etapas reactivas.....</i>	<i>177</i>
<i>Figura D.4. Consumo energético del rebullidor y condensador de la columna en función del número de etapas reactivas.....</i>	<i>177</i>
<i>Figura D.5. Composición de MTBE en función del caudal de fondo (FB).....</i>	<i>178</i>
<i>Figura D.6. Conversión obtenida en función del caudal de fondo (FB).....</i>	<i>179</i>
<i>Figura D.7. Perfiles de temperatura obtenidos para diferentes caudales de fondo.....</i>	<i>179</i>

INDICE DE TABLAS

<i>Tabla 3.1. Definición de los términos en que se clasifican los nodos.....</i>	<i>31</i>
<i>Tabla 4.1. Ponderación del tipo de términos según grado de no linealidad.....</i>	<i>45</i>
<i>Tabla 4.2. Resultados de observabilidad para el modelo de la columna de destilación con el MDE, para diferentes conjuntos de pesos.....</i>	<i>57</i>
<i>Tabla 4.3. Resultados de observabilidad obtenidos para el modelo de la columna de destilación aplicando el MD y el MDE.....</i>	<i>58</i>
<i>Tabla 4.4. Resultado del análisis de redundancias de las variables medidas para las descomposiciones obtenidas por el MD y el MDE.....</i>	<i>60</i>
<i>Tabla 4.5. Resultados de observabilidad obtenidos para el modelo de la planta de síntesis de amoníaco aplicando el MD y el MDE.....</i>	<i>65</i>
<i>Tabla 4.6. Resultados de observabilidad obtenidos para el modelo de la planta de producción de etano aplicando el MD y el MDE.....</i>	<i>67</i>
<i>Tabla 6.1. Descomposición obtenida para la simulación del craqueo catalítico de gasoil...115</i>	<i>115</i>
<i>Tabla 6.2. Descomposición obtenida para la simulación de la columna reactiva.....115</i>	<i>115</i>
<i>Tabla 6.3. Descomposición obtenida para la simulación de la red de intercambiadores.....117</i>	<i>117</i>
<i>Tabla 6.4. Descomposición obtenida para la optimización de la ubicación de un catalizador en un reactor tubular.....118</i>	<i>118</i>
<i>Tabla 6.5. Descomposición obtenida para la optimización de la producción de HDA.....120</i>	<i>120</i>
<i>Tabla A.1. Pasos que constituyen las iteraciones de la metodología aplicada.....135</i>	<i>135</i>
<i>Tabla D.1. Especificaciones de la columna utilizadas para el caso base.....175</i>	<i>175</i>
<i>Tabla D.2. Resultados de la optimización para ambas alternativas.....182</i>	<i>182</i>
<i>Tabla D.3. Costos del proceso convencional de producción de MTBE.....184</i>	<i>184</i>

INTRODUCCIÓN

1.1 INTRODUCCIÓN

El veloz crecimiento que han tenido durante los últimos años las computadoras de alta performance, tanto en velocidad como en capacidad de manejo de datos, transformó la metodología aplicada en la investigación científica. Estos progresos relativos a la construcción de hardware, combinados al desarrollo constante de nuevos algoritmos, permitieron modificar el tipo de experimentos llevados a cabo, expandiendo notablemente la dimensión y complejidad de los problemas estudiados.

La aplicación de métodos numéricos modernos y mejorados a problemas de ingeniería química es esencial para llevar a cabo el diseño, control, simulación y optimización de los procesos químicos, e incluso, de plantas completas. Debido a la complejidad creciente de los proyectos de investigación y desarrollo que deben ser llevados a cabo diariamente, los especialistas, ingenieros e investigadores confían cada vez más en el modelado, simulación y estudio de procesos asistidos por computadora. Es por eso que surgieron paquetes comerciales que tienen como finalidad asistir al ingeniero en las diferentes etapas que comprenden el diseño completo de una planta de procesos, así como también su control y la mejora de su desempeño. Se llevaron a cabo entonces diversos desarrollos e investigaciones en los campos de modelado matemático de plantas químicas, simulación y optimización de procesos, así como también diseño de instrumentación y control.

En referencia al modelado matemático de plantas químicas, Marquardt (1994) presentó un completo trabajo de revisión acerca del modelado asistido por computadora. En él se enumeran las principales herramientas de modelado existentes, comparándolas entre sí y mencionando las ventajas y desventajas de cada una de ellas.

En el campo de simulación y optimización de procesos, existe una amplia gama de programas comerciales dedicados a esta actividad (HYSYS (Hyprotech, 2002), ASPEN (Aspentech home page, 2009), gPROMS (gPROMS home page, 2009), GAMS (Brooke y *co.*, 2004)), los cuales pueden ser agrupados de acuerdo al enfoque utilizado, es decir, secuencial modular u orientado a ecuaciones (ver capítulo 5). En Biegler y *co.* (1997) se presenta una lista completa de sistemas de modelado, simuladores de procesos y herramientas de síntesis y manejo de información.

En lo que respecta al control de plantas de procesos, se han desarrollado varios programas comerciales para la tarea de reconciliación de datos de una planta (Romagnoli, y *co.*, 1980; Marquardt 1994), como por ejemplo: DATACON, VALI, SIGMAFINE. Para llevar a cabo la tarea de diseño de instrumentación de una planta, se desarrolló un Sistema de Soporte a la Decisión o DSS (por sus siglas en inglés) que asiste al ingeniero en dicha tarea (Vazquez y *co.*, 2003).

Una rama de la ciencia especializada en la investigación y desarrollo de técnicas aplicables a la ingeniería de procesos es la Computación Científica (Keil, 1996). Esta disciplina se encarga de llevar a cabo el diseño e implementación de estrategias confiables para solucionar problemas científicos o tecnológicos utilizando las computadoras como herramienta básica. Se trata de una rama de la ciencia inherentemente multidisciplinaria, que lleva a cabo investigación aplicada al tratamiento riguroso de sistemas complejos. En este contexto, y llevando estos conceptos a la ingeniería de procesos, el rol principal del especialista consiste en investigar la manera más eficiente de llevar a cabo la resolución de problemas reales con la ayuda de computadoras y herramientas computacionales, de manera tal de sacar máximo provecho de las mismas.

Mediante el estudio y la aplicación de las técnicas que la Computación Científica brinda, surgió la motivación de desarrollar diversas metodologías y herramientas aplicables a la ingeniería de procesos. Tales desarrollos se llevaron a cabo teniendo como base el aprovechamiento del modelo matemático que representa un determinado proceso químico y la matriz de incidencia asociada a dicho modelo (Stadtherr y *co.*, 1974). Como se explica en detalle en la presente Tesis (ver capítulo 2), la matriz de incidencia es una matriz binaria que contiene información acerca de la relación existente entre las ecuaciones y variables que conforman el modelo matemático de cualquier planta química. Dicha información, sumada al

correcto modelado de un determinado proceso puede resultar sumamente útil para facilitar el análisis, control, simulación y optimización de cualquier planta química.

1.2 OBJETIVOS

El objetivo central de esta Tesis es contribuir al conocimiento general en el desarrollo de metodologías de análisis estructural de modelos matemáticos y resolución de problemas en ingeniería de procesos, abarcando los campos de diseño de instrumentación, simulación y optimización de procesos químicos. De esta manera, la Tesis se encuentra dividida en dos partes principales. La primera parte abarca el análisis cualitativo (estructural) de los procesos químicos, es decir, el estudio de los modelos matemáticos y la dependencia existente entre las variables y ecuaciones que los conforman (Steward, 1962), aplicado al diseño de instrumentación de las plantas de procesos. La segunda parte de la Tesis está dedicada al análisis cuantitativo (resolución) de plantas químicas, realizándose aportes en los campos de simulación y optimización de procesos

Para llevar a cabo el cumplimiento del objetivo general se utilizó como base una nueva técnica de análisis estructural, denominada Método Directo (MD) (Ponzoni y *co.*, 2004). Este método es un algoritmo de particionamiento que efectúa un conveniente y novedoso reordenamiento estructural de la matriz de incidencia correspondiente al modelo de estado estacionario de un proceso, para llevarla a una forma triangular inferior en bloques que establece una descomposición en subsistemas y un orden de precedencia para la resolución eficiente del sistema de ecuaciones algebraicas asociado a dicha matriz. El MD dio excelentes resultados al ser aplicado a la resolución de problemas de monitoreo de procesos, incluyendo casos académicos e industriales de mediano y gran tamaño. La técnica es extremadamente robusta y eficiente en tiempo de cómputo, y su eficacia aumenta a medida que los problemas crecen en tamaño y complejidad

Uno de los objetivos particulares de esta Tesis consiste en diseñar e implementar mejoras al MD, las cuales favorezcan al proceso global de diseño de instrumentación de una planta. Actualmente el MD establece la partición de la matriz de incidencia de manera tal de encontrar la mayor cantidad posible de variables observables (calculables a partir de la resolución del modelo), sin detenerse en la naturaleza y tamaño de los bloques obtenidos. Para conducir a secuencias de cálculo aún mejores desde el punto de vista numérico, resulta deseable introducir algunos criterios para la selección de los bloques. En particular, los

sistemas distribuidos conducen a subsistemas con estructuras ordenadas y definidas, que pueden ser tenidos en cuenta en la tarea de preprocesamiento. Se incluyeron en el análisis indicadores de calidad, tales como el tipo de subsistemas resultantes (lineales, bilineales o fuertemente no lineales), el tamaño de los bloques obtenidos y el grado de facilidad con que las variables asignadas podrían ser despejadas, manteniendo como meta alcanzar el máximo grado posible de observabilidad. Como se verá en el capítulo 4 de esta Tesis, las modificaciones implementadas al algoritmo de observabilidad impactaron positivamente sobre el proceso global de diseño de instrumentación.

Hasta el momento los estudios de eficacia del Método Directo se habían limitado al área de diseño de instrumentación, es decir, a un análisis estructural de los modelos matemáticos que representan un determinado proceso. Sin embargo, el corazón de este método es una nueva estrategia para analizar y/o resolver sistemas de ecuaciones algebraicas, no necesariamente lineales. En particular otros clásicos problemas complejos de ingeniería de procesos podrían beneficiarse de esta metodología. Por lo tanto, un nuevo objetivo específico de esta Tesis es extender el rango de aplicabilidad del MD, haciendo una evaluación de su desempeño como preprocesamiento para la resolución de cualquier sistema de ecuaciones algebraicas, con especial énfasis en aquellos que surgen en otras aplicaciones del campo de la ingeniería de sistemas de procesos. Se desarrolló entonces un sistema de Particionamiento de Modelos para Simulación y Optimización llamado *MP4SO* (por sus siglas en inglés), un software orientado a ecuaciones que basa su funcionamiento en el MD. Este software está diseñado para trabajar de manera interoperativa con GAMS, sacando provecho de la gran cantidad de algoritmos de resolución que este posee para los diferentes tipos de problema a resolver (lineal, no lineal, mezcla entera, etc). El *MP4SO*, de uso sencillo e intuitivo, permite al usuario, dado un determinado modelo matemático (cuadrado o no), obtener una descomposición en subsistemas del mismo, pudiendo resolver cada uno de ellos de manera individual. De esta manera, se ve facilitada la resolución de grandes modelos, contando además con la asistencia del programa que indica qué variables deberían inicializarse para llevar a cabo la resolución. El programa cuenta además con una etapa de depuración, la cual resulta útil en la búsqueda y corrección de errores en la resolución del modelo, permitiendo al usuario llevar a cabo las correcciones sobre el modelo focalizándose únicamente en los bloques no factibles. El software desarrollado genera para cada caso una estrategia que posibilita el análisis y la resolución eficiente de los problemas.

Para llevar a cabo la prueba del *MP4SO* se empleó el mismo en la simulación y optimización de modelos de equipos especializados. Se resolvieron para ello modelos rigurosos de fenómenos de transferencia de masa con y sin reacción química basados en primeros principios, y se modelaron y resolvieron problemas de optimización con restricciones no lineales. En este contexto, se fijó como un nuevo objetivo particular el desarrollo e implementación de modelos rigurosos que conduzcan a sistemas de ecuaciones de gran tamaño y difícil resolución. Con esta finalidad se llevó a cabo el modelado de una columna de destilación reactiva, con lo cual se obtuvo un sistema de ecuaciones de gran tamaño, complejo por su grado de acoplamiento y fuertemente no lineal. Una vez obtenido el modelo, se realizó la simulación y optimización del mismo por medio de la utilización de GAMS, para luego poder comparar la resolución del sistema completo de ecuaciones con la resolución en bloques llevada a cabo mediante la utilización del *MP4SO*. Para llevar a cabo la optimización del modelo, se plantearon dos casos, cada uno con una función objetivo diferente. En primer lugar se llevó a cabo una optimización económica, donde el objetivo fue minimizar el CTA (Costo Total Anualizado) de la columna. En segundo lugar, se realizó una optimización de la calidad del producto de fondo obtenido en la columna. Para eso se planteó como función objetivo la maximización de la composición del producto deseado que se obtiene por el fondo de la columna.

Una vez testeado de manera rigurosa el *MP4SO*, y sacando provecho de los modelos matemáticos ya definidos, se planificó desarrollar un generador de modelos que pueda ser utilizado como paso previo al programa de resolución, incorporando al mismo los modelos ya desarrollados, junto con otros equipos presentes en cualquier planta de procesos. Dicho generador permite obtener el sistema de ecuaciones que representa el funcionamiento de una planta de procesos a partir del ingreso al programa de la topología de la misma. Una vez obtenido el sistema de ecuaciones, se puede llevar a cabo la resolución del mismo por medio de la aplicación del *MP4SO*. Es importante resaltar que la flexibilidad que posee el generador de modelos permite al usuario modelar cualquier planta química con el grado de realismo que sea necesario, dado que permite de manera sencilla y directa llevar a cabo tanto la modificación de los modelos existentes como también la incorporación de nuevos modelos matemáticos. La utilización conjunta del generador de modelos y el *MP4SO* permite combinar las ventajas que naturalmente posee el enfoque de simulación secuencial modular con las ventajas del enfoque orientado a ecuaciones (ver capítulo 5), dando como resultado una

herramienta para simulación y optimización de procesos sencilla de utilizar y flexible a cualquier proceso químico con el que se desee trabajar.

1.3 CONTENIDOS Y ESTRUCTURA

La Tesis está integrada por 7 capítulos y como se explicó anteriormente, se encuentra dividida en dos secciones principales. La primera de ellas está dedicada al análisis cualitativo de procesos (diseño de instrumentación) y abarca los capítulos 2, 3 y 4. La segunda parte, en la cual se realiza un análisis cuantitativo de procesos (simulación y optimización), abarca los capítulos 5, 6 y 7.

En el capítulo 1 se introdujeron brevemente las bases del problema y se delinearon los principales objetivos de la Tesis. El capítulo 2 describe las nociones básicas del diseño de instrumentación a través del análisis de observabilidad, así como también brinda un panorama general del DSS desarrollado para diseño de instrumentación. En el capítulo 3 se presenta en detalle el funcionamiento del Método Directo para análisis de observabilidad. El capítulo 4 describe la versión modificada del Método Directo (Método Directo Extendido) desarrollada en el marco de esta Tesis, así como también las ventajas que tiene su aplicación sobre el proceso global de diseño de instrumentación. El capítulo 5 detalla el software desarrollado para la generación automática de modelos matemáticos de plantas de procesos químicos. A continuación, en el capítulo 6 se describe el *MP4SO* para partición, simulación y optimización de modelos. Finalmente, en el capítulo 7 se exponen las principales conclusiones del trabajo realizado y se enuncian lineamientos para futuras investigaciones, mejoras y nuevas aplicaciones.

PARTE 1

ANÁLISIS CUALITATIVO DE PROCESOS

***SISTEMA DE SOPORTE
A LA DECISIÓN
PARA DISEÑO
DE INSTRUMENTACIÓN***

2.1 INTRODUCCIÓN

Las plantas de procesos químicos están formadas generalmente por una gran cantidad de equipos industriales, los cuales se encuentran relacionados entre sí por medio de corrientes de materia que ingresan y salen de los mismos. Para controlar dichos procesos, es necesario contar con un conjunto de sensores de medición. Estos sensores son ubicados en diversos sectores de la planta, pero no de una manera aleatoria. La cantidad y ubicación de los sensores utilizados para monitorear un determinado proceso tiene influencia directa sobre el grado de conocimiento que se tiene del mismo, haciendo que sea fundamental obtener una óptima disposición de los mismos. De este modo, se persigue alcanzar un alto grado de conocimiento de la planta utilizando el mínimo número de sensores. Este conocimiento resulta necesario para monitorear su correcto funcionamiento de manera eficiente, lo cual redundará tanto en beneficios económicos, como en la mejora de las condiciones de seguridad de la planta (Worden y Burrows, 2001; Muske y Georgakis, 2003; Lohner y Camelli, 2005).

Para llevar a cabo esta tarea, es necesario contar con modelos matemáticos de cada uno de los equipos que integran la planta química (Singh y Hahn, 2006), los cuales relacionan entre sí a todas las variables que intervienen en el proceso mediante ecuaciones algebraicas provenientes de balances de materia y energía, relaciones de equilibrio y ecuaciones termodinámicas que representan el funcionamiento de la planta en estado estacionario

(Bhushan y Rengaswamy, 2000; Singh y Hahn, 2005). Las variables del proceso se dividen básicamente en dos tipos: las medibles y las no medibles. Las variables medibles son aquellas cuyo valor se puede obtener a partir de la instalación de un sensor de medición en el lugar adecuado. Las variables no medibles son variables para las cuales no existe instrumentación capaz de medir su valor en forma directa, y que deberán ser indefectiblemente calculadas a partir del modelo matemático y datos del proceso si se desea conocer su valor. El principal objetivo del análisis es entonces determinar el conjunto óptimo de sensores (variables a medir) que permita maximizar el grado de conocimiento del funcionamiento de una planta. El grado de conocimiento será cuantificado por la cantidad de variables del modelo cuyo valor resultará conocido a partir de dicha instrumentación, intentando a su vez minimizar el costo de instalación de dichos sensores.

Debido a la gran cantidad de equipos que integran una planta de procesos, los modelos matemáticos resultantes son generalmente de grandes dimensiones, haciendo fundamental la utilización de herramientas computacionales para llevar a cabo esta tarea de diseño de instrumentación (Delattre y *co.*, 2004; Maurya y *co.*, 2004; Narasimhan y Rengaswamy, 2007). En este contexto, se ha propuesto el desarrollo de un Sistema de Soporte a la Decisión (“Decision Support System”, DSS) (Vazquez y *co.*, 2003), el cual aborda esta tarea mediante dos procedimientos centrales: el análisis de observabilidad (AO) y el análisis de redundancia (AR) (Sanchez y *co.*, 1992; Romagnoli y Sánchez, 1999). El objetivo del análisis de observabilidad es individualizar aquellas variables cuyos valores pueden ser calculados a partir de un dado conjunto de sensores y del modelo matemático de la planta. Realiza entonces la clasificación de las variables no medidas en observables (calculables) y no observables (no calculables). Por otro lado, el análisis de redundancia brinda información acerca de los sensores que pueden ser removidos de la planta sin reducir el grado de conocimiento que se tiene de la misma.

En este capítulo se presentan los conceptos básicos acerca del diseño de instrumentación por medio de la aplicación del análisis de observabilidad y redundancia. En primer lugar se define lo que es una matriz de incidencia y cómo se construye la misma a partir del sistema de ecuaciones que modela un proceso en particular (sección 2.2). A continuación, la sección 2.3 describe cómo se lleva a cabo la clasificación de las variables y ecuaciones de un modelo. Ya definidos estos conceptos, se incluye un ejemplo a partir del cual se puede apreciar cómo se realiza, paso a paso, el análisis de observabilidad para un

modelo matemático concreto, así como también la información que se obtiene una vez finalizado el mismo (sección 2.4). Posteriormente, la sección 2.5 detalla cómo se lleva a cabo el análisis de redundancia de un modelo, desarrollando un ejemplo sencillo de aplicación. La sección 2.6 describe la estructura general del DSS para diseño de instrumentación desarrollado por nuestro grupo de investigación. En la sección 2.7 se muestra cuál es la serie de etapas estructuradas que detallan cómo debe ser llevado a cabo el diseño completo de instrumentación de una planta química. Dichos pasos fueron establecidos a partir de la experiencia obtenida en el manejo del DSS a lo largo de la realización de la presente Tesis. Finalmente, la sección 2.8 incluye las conclusiones generales acerca de los temas desarrollados en este capítulo.

2.2 DEFINICIÓN DE LA MATRIZ DE INCIDENCIA

La matriz de incidencia (también conocida como matriz de ocurrencia) es una matriz cuyos elementos sólo pueden ser unos o ceros, que se utiliza como una forma de representar relaciones binarias. Aplicada a la ingeniería de procesos, la matriz de incidencia es una manera sencilla de representar la relación existente entre las ecuaciones y variables que conforman el modelo matemático de cualquier planta química. Observando la matriz resultante, uno es capaz de conocer la información acerca de qué variables participan en cada una de las ecuaciones del modelo. En las filas de la matriz se ubican las ecuaciones, mientras que las columnas corresponden a las variables del sistema.

A continuación se incluye un ejemplo sencillo de representación de un sistema de ecuaciones a través de la matriz de incidencia correspondiente:

EJEMPLO 2.1: Dado el sistema de ecuaciones de la figura 2.1, generar la matriz de incidencia correspondiente.

$$\left\{ \begin{array}{l} E_1 : x_1^2 + x_2 + x_2 \cdot x_3 = 0 \\ E_2 : x_1 + x_3 = 0 \\ E_3 : x_2^3 + x_2 \cdot x_3 + 12 = 0 \end{array} \right.$$

Figura 2.1. Sistema de ecuaciones que representa un proceso industrial.

Siguiendo la definición dada anteriormente, la matriz de incidencia resultante es la que se muestra en la figura 2.2:

$$\begin{array}{c}
 \mathbf{E}_1 \\
 \mathbf{E}_2 \\
 \mathbf{E}_3
 \end{array}
 \begin{bmatrix}
 & \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \\
 \mathbf{x} & & \mathbf{x} & \\
 & \mathbf{x} & & \mathbf{x} \\
 & & \mathbf{x} & \mathbf{x}
 \end{bmatrix}$$

Figura 2.2. Matriz de incidencia correspondiente al Ejemplo 2.1.

2.3 CLASIFICACIÓN DE LAS VARIABLES Y ECUACIONES DE UN MODELO

Los algoritmos de clasificación de variables y ecuaciones buscan diferenciar entre cuatro categorías. Las mismas se incluyen a continuación:

- **Variables Redundantes:** variables medidas que pueden ser calculadas a partir de los balances y del resto de las variables medidas.
- **Variables No Redundantes:** variables medidas que no pueden ser calculadas a partir de los balances y de las restantes variables medidas.
- **Variables Observables:** variables no medidas que se pueden obtener a partir de las variables medidas utilizando las ecuaciones del modelo.
- **Variables No Observables:** variables no medidas que no se pueden calcular a partir de los valores de las mediciones, utilizando las ecuaciones del balance..

Por otra parte, las ecuaciones del modelo pueden ser clasificadas en tres grupos:

- **Ecuaciones Asignadas:** son aquellas que se emplean para calcular directamente variables observables.

- **Ecuaciones Redundantes:** son aquellas cuyas variables son todas observables o medidas, y no se utilizan para hallar variables observables.
- **Ecuaciones No Asignadas:** son aquellas que contienen al menos una variable no observable y no pueden emplearse para hallar variables observables.

El análisis de observabilidad tiene como objetivo determinar qué variables no medidas pueden calcularse a partir de las variables medidas y de las ecuaciones del modelo matemático. Más específicamente, su tarea es determinar cuáles de las variables no medidas pertenecen al grupo de las variables observables y cuáles al de las no observables.

2.4 EJEMPLO DE ANÁLISIS DE OBSERVABILIDAD

A continuación se desarrolla en detalle un ejemplo completo de análisis de observabilidad aplicado al sistema de ecuaciones correspondiente a un proceso químico. El sistema de ecuaciones a analizar es el presentado en la figura 2.3:

$$\left\{ \begin{array}{l} E_1 : x_1 + x_4 - x_6 = 0 \\ E_2 : x_2^2 * x_3 * x_4 - x_5 - x_7 = 0 \\ E_3 : x_1 * x_2^{1.7} * (x_4 - 5) - x_8 = 0 \\ E_4 : x_4 - 3x_1 - x_7 = 0 \\ E_5 : x_1 * x_3 - x_5 + x_7 = 0 \\ E_6 : 3x_2 - x_9 + 4x_6 + 3x_{10} = 0 \end{array} \right.$$

Figura 2.3. Sistema de ecuaciones que representa un proceso industrial.

Como se puede apreciar en la figura 2.3, el sistema está integrado por 6 ecuaciones y 10 variables. La matriz resultante correspondiente al sistema de ecuaciones presentado es la que se muestra en la figura 2.4:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
E_1	x			x		x				
E_2		x	x	x	x		x			
E_3	x	x		x				x		
E_4	x			x			x			
E_5	x		x		x		x			
E_6		x				x			x	x

Figura 2.4. Matriz de incidencia correspondiente al sistema de la figura 2.3.

Si se deciden medir arbitrariamente las variables x_6 , x_7 y x_8 , obteniéndose los valores incluidos en la figura 2.5:

$$\left\{ \begin{array}{l} x_6: 10 \\ x_7: 6 \\ x_8: 8 \end{array} \right.$$

Figura 2.5. Conjunto arbitrario de mediciones impuesto al sistema de la figura 2.4.

se obtendrá entonces una nueva matriz de incidencia, la cual excluye las variables medidas debido a que las mismas dejaron de ser incógnitas para convertirse en datos. La elección del conjunto de variables a medir dependerá principalmente del costo que requiere medir dichas variables y de su naturaleza física. Desde el punto de vista económico, resultaría totalmente inviable para una planta realizar mediciones sobre el total de las variables de la misma. Por este motivo, es importante definir un grupo de sensores que permitan obtener la información necesaria para controlar las variables más importantes del proceso. Por otro lado, desde un punto de vista físico, determinadas variables de un modelo matemático no pueden ser medidas directamente en planta (ej: entalpías, variables de equilibrio entre fases, etc), debido a que no

existe un instrumento capaz de llevar a cabo la medición. En estos casos, si resultara indispensable conocer el valor de dichas variables, será necesario realizar otras mediciones que permitan calcularlas de manera indirecta, mediante la resolución del sistema de ecuaciones en las que estas variables se encuentren involucradas.

Para el ejemplo presentado en esta sección, se asume que el conjunto de variables medidas (x_6, x_7, x_8) fue seleccionado en base a uno de los criterios mencionados anteriormente. La matriz resultante, luego de realizadas estas mediciones, es la mostrada en la figura 2.6.

	X_1	X_2	X_3	X_4	X_5	X_9	X_{10}
E_1	x			x			
E_2		x	x	x	x		
E_3	x	x		x			
E_4	x			x			
E_5	x		x		x		
E_6		x				x	x

Figura 2.6. Matriz de incidencia resultante de la medición de las variables x_6, x_7 y x_8 .

Aplicando un algoritmo de observabilidad de detección de componentes fuertes a la matriz resultante, se obtiene la siguiente matriz permutada (figura 2.7):

	X_1	X_4	X_2	X_3	X_5	X_9	X_{10}
E_1	x	x					
E_4	x	x					
E_3	x	x	x				
E_5	x			x	x		
E_2		x	x	x	x		
E_6			x			x	x

Figura 2.7. Matriz de incidencia resultante de la aplicación del algoritmo de observabilidad.

Como se puede apreciar, el algoritmo particiona la matriz de incidencia en subsistemas, brindando un orden de resolución de esos sistemas que permitirá obtener la mayor información posible del sistema. Ahora se cuenta con cuatro subsistemas, tres de los cuales son cuadrados, de modo que sería posible su resolución numérica, la cual puede ser aplicada siguiendo el orden sugerido por el algoritmo de observabilidad. Los subsistemas obtenidos son detallados en la figura 2.8:

$$S_1 \begin{cases} E_1 : x_1 + x_4 - 10 = 0 \\ E_4 : x_4 - 3x_1 - 6 = 0 \end{cases}$$

$$S_2 \begin{cases} E_3 : x_1 * x_2^{1.7} * (x_4 - 5) - 8 = 0 \end{cases}$$

$$S_3 \begin{cases} E_2 : x_2^2 * x_3 * x_4 - x_5 - 6 = 0 \\ E_5 : x_1 * x_3 - x_5 + 6 = 0 \end{cases}$$

$$S_4 \left\{ E_6 : 3x_2 - x_9 + 40 + 3x_{10} = 0 \right.$$

Figura 2.8. Subsistemas obtenidos a partir de la aplicación del algoritmo de observabilidad.

El orden de resolución otorgado por el algoritmo, indica para este caso que, en primer lugar, se pueden obtener los valores de las variables x_1 y x_4 , por medio de la resolución del bloque S1. A continuación, resolviendo el bloque S₂ es posible obtener el valor de la variable x_2 , siendo a esta altura conocidos los valores de las variables x_1 y x_4 . Finalmente, habiendo ya resuelto los dos bloques anteriores, es posible determinar los valores de las variables x_3 y x_5 por medio de la resolución del bloque S₃. El bloque restante, el S₄ contiene una ecuación (E₆) con dos incógnitas (x_9 y x_{10}), por lo que no es posible resolver ese bloque y obtener los valores de las variables. Si se quisiera obtener el valor de alguna de estas dos variables, sería necesario realizar la medición de la otra, de manera tal de que el sistema resulte consistente.

Para este caso en particular, con un proceso químico representado por el sistema de ecuaciones de la figura 2.3, y habiendo realizado las mediciones de las variables x_6 , x_7 y x_8 , el diseño de instrumentación arrojaría los siguientes resultados:

- **Variables Medidas:** x_6, x_7, x_8
- **Variables Observables:** x_1, x_2, x_3, x_4, x_5
- **Variables No Observables:** x_9, x_{10}

2.5 EJEMPLO DE ANÁLISIS DE REDUNDANCIA EN EL MODELO MATEMÁTICO DE UN EQUIPO

A continuación se brinda un ejemplo desarrollado en base al modelo de un mezclador, donde se puede apreciar el fenómeno de redundancia para un modelo matemático determinado. El mismo consiste en un mezclador que consta de dos corrientes de entrada y una de salida. Cada una de las corrientes contiene tres componentes. A continuación se presenta un esquema del equipo (figura 2.9).

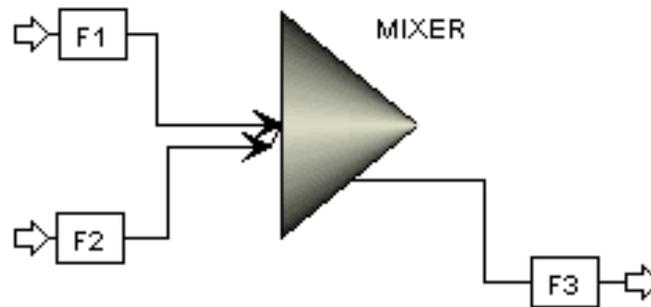


Figura 2.9. Mezclador con dos corrientes de entrada y una de salida.

Si el objetivo del ingeniero fuera lograr calcular la corriente F3 en forma completa, sería necesario que se realizaran las mediciones sobre las corrientes de entrada (F₁ y F₂). De esta manera, la corriente de salida resultaría observable en su totalidad (tanto el caudal como las composiciones). El sistema de ecuaciones generado por el DSS que permitiría que la corriente de salida (F₃) resultara observable tendría la forma (figura 2.10):

$$\left\{ \begin{array}{l} F_1 + F_2 - F_3 = 0 \\ F_1 * x_1^{F1} + F_2 * x_1^{F2} - F_3 * x_1^{F3} = 0 \\ F_1 * x_2^{F1} + F_2 * x_2^{F2} - F_3 * x_2^{F3} = 0 \\ x_1^{F1} + x_2^{F1} + x_3^{F1} = 1 \\ x_1^{F2} + x_2^{F2} + x_3^{F2} = 1 \\ x_1^{F3} + x_2^{F3} + x_3^{F3} = 1 \end{array} \right.$$

Figura 2.10. Sistema de ecuaciones correspondiente al modelo del mezclador.

A este sistema, se debería agregar el siguiente conjunto de mediciones (figura 2.11) para que las variables pertenecientes a la corriente de salida resulten observables:

$$\left\{ \begin{array}{l} F_1; F_2 \\ x_1^{F1}; x_1^{F2} \\ x_2^{F1}; x_2^{F2} \end{array} \right.$$

Figura 2.11. Conjunto de mediciones necesario para que las variables de la corriente de salida del mezclador resulten observables.

Si el ingeniero de procesos decidiera agregar al conjunto de mediciones el caudal de la corriente de salida (F_3), la variable F_3 pasaría ahora a ser una variables “redundante”. Esto se debe a que su valor puede ser obtenido mediante observabilidad (como se explicó anteriormente), pero también ha sido medido por la instalación de un caudalímetro. Mas aún, la primera ecuación de la figura 2.10 (balance de masa global) pasará a ser una ecuación redundante, ya que será utilizada para calcular una variable que ya fue medida y cuyo valor es conocido.

En ocasiones puede resultar de interés que ciertas variables, sobre todo las de mayor importancia dentro del proceso, resulten redundantes (Ferraro y *co.*, 2002). Esto se debe a que da la posibilidad al ingeniero de procesos de corroborar los datos obtenidos por medio de mediciones, comparándolos con los resultados obtenidos de la resolución matemática del modelo.

2.6 ESTRUCTURA GENERAL DEL DSS

El sistema de soporte a la decisión (DSS) propuesto por Vazquez y *co.* (2003) es un software amigable que asiste al ingeniero de procesos en el diseño de instrumentación de plantas químicas. Este DSS permite el modelado riguroso y preciso de problemas reales sin importar qué tan grandes sean los modelos de las plantas a analizar, resultando ser una herramienta eficiente y robusta para el tratamiento de problemas complejos. Otra característica no menos importante es la flexibilidad que el paquete posee para la adición de nuevos algoritmos, la cual permitirá la realización de cualquier mejora o nuevo algoritmo que se quiera implementar, de modo que pueda ser incorporado al software de manera sencilla.

El DSS está constituido por varios módulos, cada uno con una tarea específica. Estos módulos, integrados a través de una interfaz gráfica, permiten al ingeniero realizar el diseño completo de la instrumentación de una planta. Los módulos que forman el DSS son: módulo de

ingreso de datos (MID), generador de modelos (MGM) (Vazquez y *co.*, 2001), módulo robusto para análisis de observabilidad (MAO) (Ponzoni y *co.*, 2004) y módulo para análisis de redundancia (MAR) (Ferraro y *co.*, 2002). En la figura 2.12 se muestra un esquema completo de los módulos y cómo interactúan en el DSS.

En primer lugar, el ingeniero debe definir la topología de la planta, ingresando los equipos de procesos y las cañerías o corrientes que los interconectan. Una vez ingresado el flowsheet completo se debe definir una configuración inicial de sensores, tarea que puede ser llevada a cabo manualmente por el usuario o a través de un algoritmo genético de inicialización especialmente implementado para esta tarea (Carballido y *co.*, 2004; Carballido y *co.*, 2005). El paso siguiente es la generación del modelo matemático completo, haciendo uso del módulo generador de modelos. El modelo generado debe ser luego ingresado en el módulo de análisis de observabilidad, el cual determina cuáles de las variables son observables y cuáles no. Este primer resultado es analizado por el usuario, quien determina si el mismo es satisfactorio. Si alguna de las variables importantes del proceso resulta ser no observable, se deberá recurrir nuevamente al módulo de ingreso de datos, agregando sensores a las variables deseadas y corriendo nuevamente el MAO para obtener una nueva clasificación de variables.

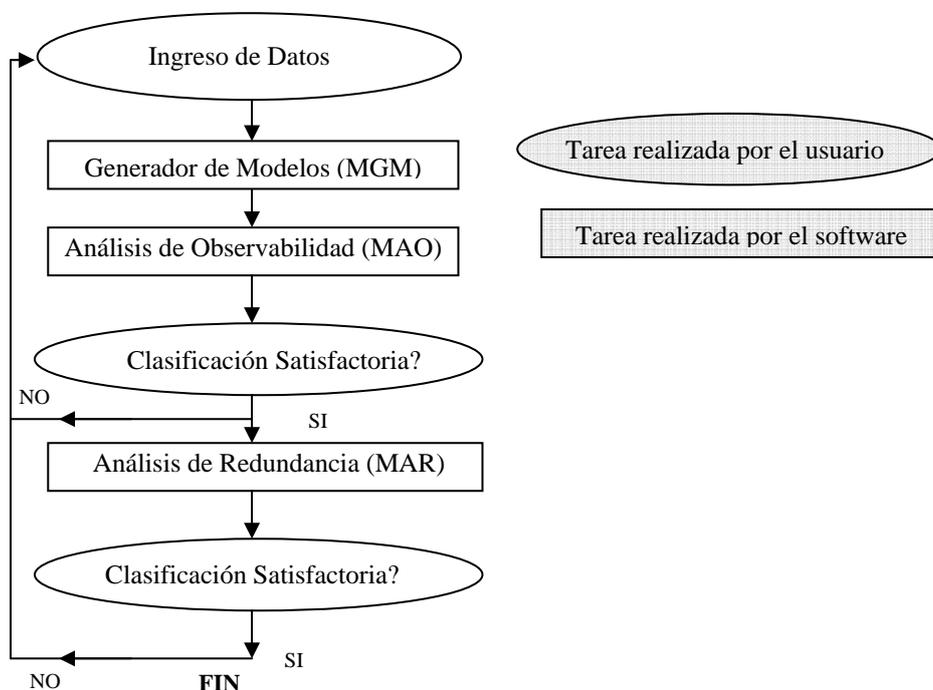


Figura 2.12. Esquema de interacción entre el usuario y el DSS para efectuar el diseño de instrumentación de una planta química.

Una vez que el nivel de observabilidad obtenido es aprobado por el usuario, el paso siguiente es el ingreso de esta clasificación al módulo de análisis de redundancia. Como se explicó anteriormente, el mismo indica cuáles de las variables asociadas a los sensores son además calculables mediante el modelo matemático. Puede haber ocasiones en que resulta necesario tener algunas variables observables medidas, tanto para realizar reconciliación de datos como por cuestiones de seguridad. Si no es el caso, el ingeniero deberá retornar nuevamente a la etapa inicial del diseño (MID) y remover alguno de los sensores redundantes, ejecutar el análisis de observabilidad (MAO) y luego el de redundancia (MAR). Una vez finalizado este proceso iterativo queda establecido un diseño de instrumentación satisfactorio para la planta de procesos.

2.7 ETAPAS PARA LLEVAR A CABO EL DISEÑO DE INSTRUMENTACIÓN DE UNA PLANTA DE PROCESOS

A continuación se sintetizan las etapas que deben ser llevadas a cabo para realizar el diseño de instrumentación de una planta de procesos con la asistencia del DSS.

Etapas 1: Estudio general del proceso - detección de riesgos ambientales: En esta etapa, el ingeniero debe analizar los procesos que serán llevados a cabo en la planta en estudio, sin entrar aún en detalles acerca de la topología de la misma. Se debe tener un conocimiento completo de los procesos químicos y de separación que ocurren, así como también de los desechos tóxicos y contaminantes que puedan emanar la misma. De esta manera se está en condiciones de detectar a priori, posibles peligros ambientales que existan a partir del funcionamiento de la planta. En esta etapa se estudian exclusivamente los riesgos relativos a las sustancias químicas que serán utilizadas o producidas dentro de la planta.

Etapas 2: Estudio detallado del flowsheet – detección de variables críticas para la seguridad y correcta operación de la planta: Una vez estudiado el proceso general, así como también los posibles riesgos existentes, se debe llevar a cabo un análisis minucioso de la planta en cuestión. En esta etapa se debe identificar cada uno de los equipos que forman parte del flowsheet así como las corrientes que los conectan, logrando así un grado de conocimiento tal que permita comprender el funcionamiento de la planta completa. Una vez realizado el análisis, el ingeniero debe estar en condiciones de detectar cuáles de las variables resultan críticas para la seguridad de la planta y el correcto funcionamiento de la misma. El correcto monitoreo de dichas variables dará como resultado una detección temprana de

problemas que permitirá prevenir posibles accidentes, así como también asegurar una correcta especificación en los productos obtenidos.

Etapa 3: Enumeración de variables críticas del proceso: En esta etapa se realiza la enumeración de las variables críticas que deben ser observadas para la planta que se desea monitorear. La identificación de estas variables se debe realizar en base a los estudios llevados a cabo en las etapas 1 y 2. En este paso, se definirán en forma concreta las variables del proceso que deberán ser necesariamente observables. También se debe decidir cuáles de las variables deberán resultar redundantes, es decir, variables que serán medidas, pero cuyo valor podrá ser corroborado por medio del modelo matemático que representa el funcionamiento de la planta.

Etapa 4: Definición de la instrumentación inicial de la planta: El usuario debe en esta etapa definir una instrumentación inicial, la cual será utilizada para realizar el primer análisis de observabilidad. La misma puede ser definida manualmente o a través de la utilización de uno de los algoritmos genéticos de inicialización desarrollados para esta tarea (Carballido y *co.*, 2004; Carballido y *co.*, 2005). Es importante en esta etapa hacer uso de la experiencia que puede llegar a brindar un ingeniero de planta con antecedentes en procesos semejantes al estudiado. La configuración inicial de sensores deberá incluir aquellas variables críticas (definidas en el paso 3) que necesariamente deban resultar redundantes, ya que independientemente de los resultados obtenidos a partir del análisis de observabilidad, siempre deberá ubicarse un sensor para medir su valor.

Etapa 5: Ejecución del Módulo de Análisis de Observabilidad: En esta etapa se determina cuáles de las variables del sistema resultan observables, mediante la aplicación del MAO al modelo matemático generado. El resultado obtenido debe ser analizado por el usuario, quien determina si el mismo es satisfactorio. Si alguna de las variables críticas del proceso resulta ser no observable se vuelve al paso 4, agregando sensores a las variables deseadas y corriendo nuevamente el MAO para obtener una nueva clasificación.

Etapa 6: Ejecución del Módulo de Análisis de Redundancia: Una vez que el resultado del análisis de observabilidad es aprobado por el usuario, el paso siguiente es ingresar esta clasificación al Módulo de Análisis de Redundancia. Como se explicó anteriormente, el MAR indica cuáles de las variables asociadas a los sensores son además calculables mediante el

modelo matemático. El diseño final aprobado por el usuario será aquel que, en base a lo definido previamente en los pasos 1 a 3, posea las siguientes características:

- Ninguna de las variables críticas resulta no observable.
- Existe redundancia en las variables críticas que así lo requieren
- El número y tipo de variables no críticas que resultan observables resulta suficiente
- No existe redundancia en variables que no lo requieren, de manera tal de minimizar el costo de la instrumentación utilizada.

2.8 CONCLUSIONES

En este capítulo se presentaron los conceptos básicos para llevar a cabo el diseño de instrumentación para una planta de procesos químicos, a partir de la aplicación del análisis de observabilidad y el de redundancia al modelo matemático que representa su comportamiento. También se mostró en detalle el DSS para diseño de instrumentación, describiendo cada uno de los módulos que lo conforman. Finalmente, se desarrollaron los pasos a seguir para llevar a cabo el diseño de instrumentación de cualquier planta de procesos a mediante la aplicación del DSS. En particular, la primera parte de esta Tesis (análisis cualitativo de procesos) se focaliza en la mejora de las etapas 5 y 6 del diseño de instrumentación (ver sección 2.7). Para ello se desarrolló un nuevo algoritmo para análisis de observabilidad, cuya aplicación optimiza, tanto la descomposición de la matriz de incidencia, como también el análisis de redundancia realizado posteriormente.

2.9 NOMENCLATURA DEL CAPÍTULO

AO= Análisis de observabilidad

AR= Análisis de redundancia

MAO= Módulo de análisis de observabilidad

MAR= Módulo de análisis de redundancia

MD= Método Directo

MGM= Módulo generador de modelos

MID= Módulo de ingreso de datos

MÉTODO DIRECTO PARA ANÁLISIS DE OBSERVABILIDAD

3.1 INTRODUCCIÓN

En el capítulo 2 se presentó la estructura general del Sistema de Soporte a la Decisión (DSS) para diseño de instrumentación (Vazquez y *co.*, 2003), describiendo también su funcionamiento. A continuación se explica en detalle cómo está conformado uno de los módulos cruciales que integran el DSS: el módulo de análisis de observabilidad (MAO).

Como se explicó anteriormente, el objetivo principal del MAO es identificar en el modelo matemático de una planta de procesos qué variables no medidas pueden ser calculadas a partir de las variables medidas y de las ecuaciones. El MAO determina entonces cuáles de las variables no medidas quedan en el grupo de las observables (calculables) y cuáles en el grupo de la no observables (no calculables).

En particular, la implementación del MAO emplea el algoritmo conocido como Método Directo (MD) (Ponzoni y *co.*, 2004). El MD constituye una técnica de particionamiento de matrices muy robusta y eficiente. La técnica resulta robusta porque se puede aplicar a cualquier tipo de matrices, sin importar su patrón estructural; y es eficiente porque no se conoce otro algoritmo similar que permita incorporar restricciones en la conformación de los bloques para guiar el ordenamiento según pautas especificadas previamente. El método es sumamente flexible dado que es posible generar diversos reordenamientos para un mismo problema hasta encontrar el más satisfactorio.

El resultado de la aplicación del algoritmo es la obtención de una matriz triangular inferior en bloques (FTiB) como la que se muestra en la figura 3.1, donde los bloques S_O de la diagonal son cuadrados y con transversal llena.

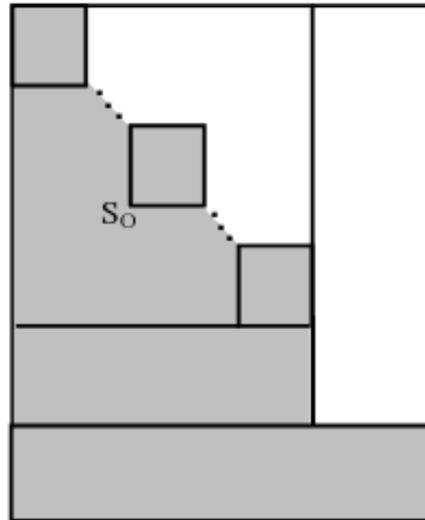


Figura 3.1. Estructura de la matriz FTiB

En este capítulo se presentan, en primer lugar, algunos conceptos básicos de teoría de grafos (sección 3.2), los cuales ayudan a la correcta interpretación del algoritmo que conforma el Método Directo. Luego se incluye un esquema detallado de la arquitectura del algoritmo (sección 3.3), con todas las etapas que lo conforman. En la sección 3.4 se describe el algoritmo completo diseñado para llevar a cabo la implementación del software. Finalmente, en la sección 3.5 se incluye un ejemplo desarrollado a partir de la aplicación del Método Directo a un caso de estudio, de manera tal de facilitar la comprensión del mismo.

3.2 CONCEPTOS BÁSICOS DE TEORÍA DE GRAFOS

Para comprender mejor cómo funciona el módulo de análisis de observabilidad utilizado por el DSS, y en particular el Método Directo, se incluyen las nociones principales de teoría de grafos Gibbons (2004).

3.2.1 GRAFOS, DIGRAFOS Y NOCIONES DE CONECTIVIDAD

Un grafo no dirigido (o simplemente grafo) G consiste de un conjunto N de nodos y una colección A de pares no ordenados de nodos, denominados aristas. Así, un grafo se

representa simbólicamente como $G = (N, A)$. El orden de un grafo queda establecido por su número de nodos, N , mientras que el tamaño de un grafo está dado por su cantidad de aristas, A . Si u y v pertenecen a N y existe un par no ordenado $a = (u, v)$ en A , se dice que a une u y v , ó que existe una arista a entre u y v .

La forma gráfica de representar un grafo es empleando círculos o puntos para especificar los nodos y líneas para trazar las aristas. A continuación se presenta un ejemplo de la notación detallada anteriormente (figura 3.2).

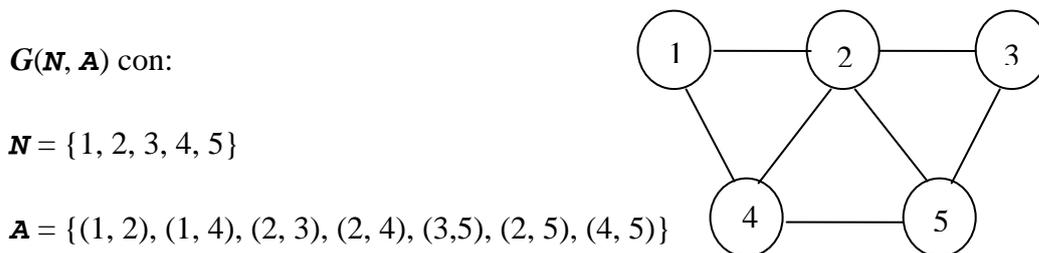


Figura 3.2. Representación gráfica de un grafo no dirigido G

3.2.1.1 CONCEPTOS BÁSICOS

A continuación se incluyen una serie de conceptos básicos utilizados en teoría de grafos, los cuales son útiles para comprender mejor la estructura y funcionamiento del MD:

- **Aristas Paralelas:** Dos o más aristas que unen un mismo par de nodos.
- **Bucle:** Arista cuyos extremos coinciden.
- **Grafo Simple:** Se dice que G es un grafo simple si no contiene aristas paralelas ni bucles.
- **Árbol:** Un grafo simple T se denomina árbol si dados dos nodos u y v de T , existe un único camino entre u y v .
- **Subgrafo:** Un grafo $S = (NS, AS)$ es un subgrafo de $G = (N, A)$ si NS es un subconjunto de N y AS es un subconjunto de A .

- **Subgrafo inducido:** un subgrafo G' de G es un subgrafo inducido por el subconjunto de nodos N' , $N' \subseteq N$, si el grafo G' está conformado por todas las aristas de G con ambos extremos incluidos en N' .
- **Grafo dirigido (dígrafo):** Un grafo $G = (N, A)$ se denomina grafo dirigido o digrafo si N es un conjunto finito de nodos y A un conjunto finito de pares ordenados. La representación gráfica de un digrafo es similar a la empleada para los grafos no dirigidos, sólo que ahora las aristas son flechas cuyas cabezas indican la dirección de las mismas.
- **Arista:** Una arista $e = (u, v)$ de un digrafo G es una arista dirigida desde u hacia v . También se dice que la arista e parte de u y llega a v , ó que v es adyacente a u .
- **Grafo subyacente:** El grafo subyacente a un digrafo es el grafo no dirigido que se obtiene al remover las direcciones de las aristas.
- **Conexión fuerte de dígrafos:** Un digrafo $G = (N, A)$ está fuertemente conectado si existe un camino desde u a v y un camino desde v a u para cada par distinto de vértices u y v , con $u, v \in N$. Si un digrafo no está fuertemente conectado entonces puede ser partido en subgrafos fuertemente conectados; dichos subgrafos se denominan componentes fuertes de G .

3.2.1.2 CONCEPTO DE CAMINATA EN UN GRAFO

Sean u y v dos nodos de un grafo G . Una caminata (*walk*) entre u y v en el grafo es una secuencia finita $u = n_0, a_1, n_1, a_2, n_2, a_3, \dots, a_k, n_k = v$ de nodos y aristas de G tales que la arista a_i en la secuencia une los nodos n_{i-1} y n_i . Los nodos y aristas en una caminata no son necesariamente distintos. La longitud de una caminata está dada por su número de aristas.

3.2.2 GRAFOS BIPARTITOS

Un grafo bipartito (bigrafo) es un grafo simple cuyo conjunto de nodos puede ser particionado en dos subconjuntos disjuntos tales que todas las aristas del grafo posean un extremo en cada subconjunto. La representación matemática de un bigrafo está dada por la terna $G = (N_1, N_2, A)$, donde N_1 y N_2 son los conjuntos de nodos y A el conjunto de aristas. En

la figura 3.3 se muestra la representación gráfica de un bigrafo, en la cual se emplean cuadrados y rombos para esquematizar los nodos de N_1 y N_2 respectivamente.

$G = (N_1, N_2, A)$ con:

$$N_1 = \{n1_1, n1_2, n1_3, n1_4, n1_5, n1_6\}$$

$$N_2 = \{n2_1, n2_2, n2_3, n2_4\}$$

$$A = \{(n1_1, n2_1), (n1_1, n2_2),$$

$$(n1_2, n2_1), (n1_2, n2_2), (n1_2, n2_3),$$

$$(n1_3, n2_1), (n1_3, n2_2), (n1_3, n2_3), (n1_3, n2_4),$$

$$(n1_4, n2_2), (n1_4, n2_3), (n1_4, n2_4),$$

$$(n1_5, n2_1), (n1_5, n2_4),$$

$$(n1_6, n2_1), (n1_6, n2_4)\}$$

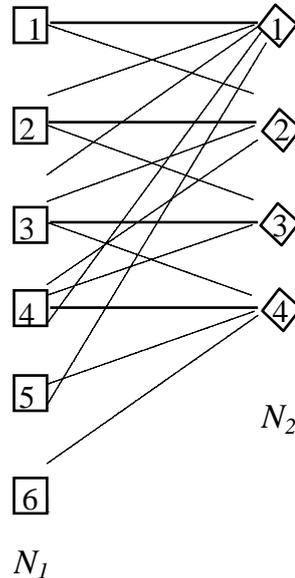


Figura 3.3. Representación gráfica de un bigrafo G

3.2.3 PAREAMIENTOS MAXIMALES

La búsqueda de pareamientos maximales es un problema muy común en aplicaciones modeladas a través de bigrafos (Asratian y *co.*, 1998; Karpinski y Rytter, 1998). Un pareamiento P en un bigrafo $G = (N_1, N_2, A)$ es un subconjunto de A tal que ninguna de las aristas de P tienen extremos en común. La cardinalidad de un pareamiento está dada por el número de aristas que contiene. En tal sentido, se dice que P es un pareamiento maximal para un bigrafo G si no existe otro pareamiento P'' en G tal que la cardinalidad de P'' sea mayor que la cardinalidad de P . Nótese que un mismo bigrafo puede tener más de un pareamiento maximal.

Previo a la presentación del algoritmo para encontrar un pareamiento maximal de un bigrafo (PMB), se presenta una serie de conceptos auxiliares que ayudan a su comprensión:

- **Caminata alternante:** Una caminata alternante sobre un bigrafo G relativa a un pareamiento P es una caminata cuyas aristas alternan en el pareamiento P .
- **Sendero alternante:** es una caminata alternante que no contiene aristas repetidas.

- **Sendero aumentado:** sendero alternante cuyos extremos corresponden a nodos no apareados por P .

A continuación se incluye la figura 3.4, la cual describe el algoritmo utilizado por el Método Directo para encontrar el pareamiento maximal de un bigrafo.

```

Datos de entrada:  $R, C, A$ 
Datos de salida:  $P_M$ 

 $P_M \leftarrow \emptyset.$ 
 $C_U \leftarrow \emptyset.$ 

% Construcción de un pareamiento inicial
Para cada nodo  $c \in C$  hacer:
  Aparear  $c$  con el primer nodo no apareado  $r$ , tal que  $r \in R$ .
  Si no existe tal nodo  $r$  entonces  $C_U \leftarrow C_U \cup \{c\}$ 
fin-para

% Búsqueda de caminos aumentados
 $C_{UN} \leftarrow \emptyset.$ 

Repetir
  Buscar un camino aumentado  $Au$  desde  $c$ , visitando
  Únicamente nodos de  $R$  que no han sido visitados
  anteriormente durante este paso.
  Marcar todos los nodos alcanzados como visitados.
  Si se ha encontrado un camino aumentado  $A$ 
  entonces
    Aumentar  $P_M$  con  $Au$ .
  sino
    Incluir  $c$  en  $C_{UN}$ .
  fin-si.
 $C_U \leftarrow C_U \cup C_{UN}.$ 
 $C_{UN} \leftarrow \emptyset.$ 
hasta: que no se encuentre ningún camino aumentado.

```

Figura 3.4. Algoritmo de búsqueda del PMB.

3.3 ARQUITECTURA DEL MÉTODO DIRECTO

A continuación se incluye un esquema que resume cómo está conformada la arquitectura del Método Directo (figura 3.5), detallando las etapas que forman parte del algoritmo.

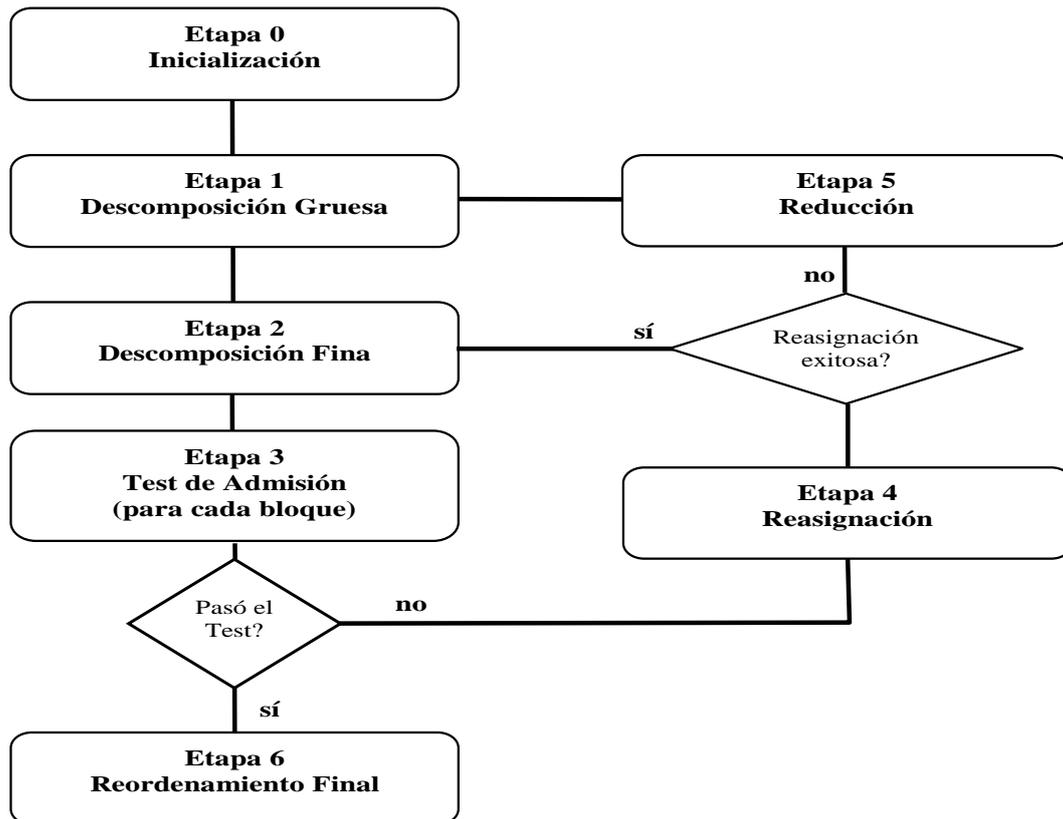


Figura 3.5. Arquitectura del Método Directo

3.3.1 ETAPA 0: INICIALIZACIÓN

En esta etapa, se construye el grafo bipartito $G = (R, C, A)$ asociado a la matriz N , de forma tal que los nodos en R representan a las filas de N , los nodos en C a las columnas, y las aristas en A los elementos no nulos de la matriz, respectivamente.

3.3.2 ETAPA 1: DESCOMPOSICIÓN GRUESA

En esta etapa se realiza el primer particionamiento y se busca un pareamiento maximal (P_m) de G por medio del algoritmo PMB, el cual fue presentado en la figura 3.4. Existen diferentes técnicas para llevar a cabo la permutación de matrices hacia la forma triangular en bloques. Algunas de ellas no resultaban aplicables a matrices rectangulares, estructuralmente singulares, mientras que otras llegaban a distribuciones de bloques que no resultaban útiles. Pothen y Fan (1990) introdujeron una técnica de particionamiento para matrices generales. Esta técnica consta de dos etapas diferentes. En primer lugar, se obtiene una descomposición gruesa por medio de la aplicación de la descomposición de Dulmage-Mendelsohn (Dulmage y Mendelsohn, 1963). La descomposición de Dulmage-Mendelsohn es una de las técnicas más conocidas para permutar una matriz general a una forma triangular en bloques. Una vez que

esta técnica es aplicada, se utiliza el método de Pothen para llevar a cabo la descomposición fina.

La principal diferencia entre la descomposición de Dulmage-Mendelsohn y el Método Directo radica en el hecho de que la primera ubica en el mismo grupo todos los nodos en N_I que pertenecen tanto a VR como a $SR1$, mientras que la clasificación para diseño de instrumentación requiere que se realice una distinción entre ellos. La figura 3.6 muestra las diferencias obtenidas mediante ambas clasificaciones. En la figura 3.6.a se puede apreciar la clasificación de Dulmage-Mendelsohn mientras que la figura 3.6.b muestra cómo deben acomodarse los nodos para obtener una clasificación útil para el diseño de instrumentación. Además de ser cuadrados, una característica importante de los bloques ($SC1, SR1$) y ($SC2, SR2$) es que tienen su transversal llena, debido a que los nodos $SC1, SR1, SC2$, y $SR2$ pertenecen al pareamiento maximal.

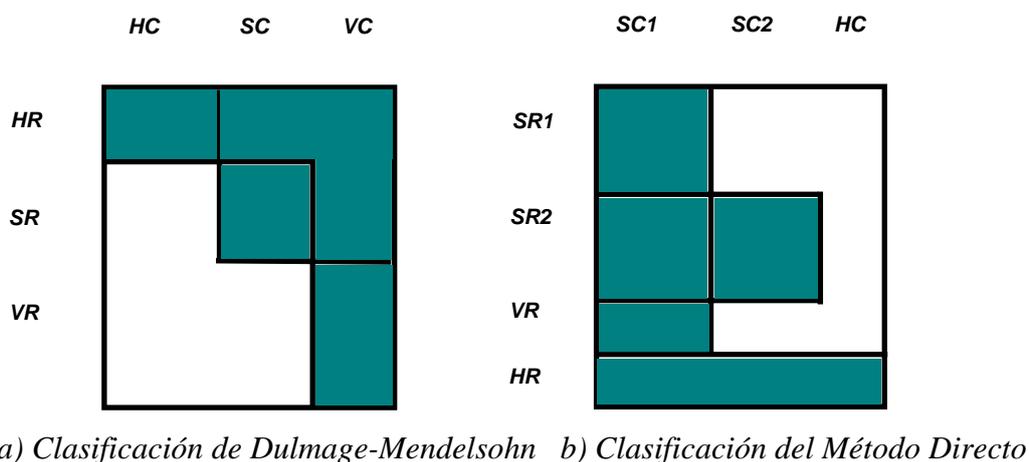


Figura 3.6. Clasificación de nodos y descomposición gruesa para ambas técnicas.

La tabla que se encuentra a continuación incluye la definición de cada uno de los términos en que se clasifica la matriz a partir del reordenamiento obtenido con el Método Directo.

Otra diferencia importante entre ambas propuestas es el hecho de que la descomposición de Pothen y Fan fue presentada para resolver sólo sistemas lineales, mientras que el Método Directo fue diseñado tanto para sistemas lineales como para no lineales. Es sabido que la mayoría de los modelos matemáticos resultantes de las plantas de procesos químicos son sistemas de ecuaciones fuertemente no lineales. Esto constituye una razón fundamental por la cual resultó de suma importancia proponer una nueva técnica de

descomposición, de manera de poder llevar a cabo el diseño de instrumentación asistido por computadora.

VR	=	ecuaciones redundantes: nodos no apareados de R
SRI	=	ecuaciones asignadas: nodos apareados de R alcanzables desde algún nodo no apareado de R por medio de un camino alternante
HR	=	ecuaciones con variables indeterminables: nodos apareados de R alcanzables desde algún nodo no apareado de C mediante un camino alternante
SR2	=	ecuaciones asignadas: $R \setminus (VR \cup SRI \cup HR)$
SCI	=	variables observables: nodos apareados de C alcanzables desde algún nodo no apareado de R mediante un camino alternante
HC	=	variables indeterminables: nodos de C tanto apareados como no apareados alcanzables desde algún nodo no apareado de C mediante un camino alternante
SC2	=	variables observables: $C \setminus (SCI \cup HC)$

Tabla 3.1. Definición de los términos en que se clasifican los nodos.

3.3.3 ETAPA 2: DESCOMPOSICIÓN FINA

En esta etapa se particionan los bloques obtenidos en subsistemas cuadrados no reducibles, por medio del algoritmo SCC (Detección de Componentes Fuertes de un Digrafo), el cual se encuentra detallado en Ponzoni y *co.*(2004). En primer lugar, se realiza la descomposición de $G(N_1)$ donde $N_1 = (SRI, SCI)$, y en segunda instancia $G(N_2)$, con $N_2 = (SR2, SC2)$. Cada una de las componentes fuertes resultantes representa un subconjunto de asignación de FTiB.

3.3.4 ETAPAS 3, 4 Y 5: TEST DE ADMISIÓN, REASIGNACIÓN Y REDUCCIÓN

El **Test de Admisión** verifica la aceptación de cada uno de los subconjuntos de asignación generados en la descomposición fina. Es decir, chequea cada bloque para comprobar que no pertenece al conjunto de restricciones R , donde R indica los subsistemas de N que por alguna razón no resultan admisibles.

Primero examina los bloques N_{I_i} , con $i = 1..p$, de la submatriz $N = (SRI, SCI)$. Si algún bloque N_{I_j} , con $1 \leq j \leq p$ contiene un subconjunto prohibido, el algoritmo pasa a la etapa de **Reasignación**, en donde intercambia una de las ecuaciones del subsistema que fue rechazada por el test de admisión por una de las ecuaciones redundantes. En otras palabras, la reasignación permuta una de las filas de SRI con una de las filas de VR . Luego, el digrafo correspondiente a N_I se construye nuevamente y se reitera la descomposición fina del bloque. Esta permutación evita que se generen los mismos bloques prohibidos una y otra vez (Ponzoni y co., 1998b).

Cuando la reasignación del bloque N_{I_j} es imposible de realizar, el algoritmo procede a la etapa de **Reducción** del bigrafo, en donde los bloques N_{I_i} , con $i < j$, se eliminan del bigrafo $G(N)$, y se agregan a los subconjuntos de asignación válidos. Aquí se escoge un nodo fila de N_{I_j} , llamado fila especial e y se elimina temporalmente de $G(N)$. Luego se vuelva a la etapa de **Descomposición Gruesa**.

El propósito de seleccionar una fila especial y eliminarla del bigrafo $G(N)$, radica en que se debe hallar un pareamiento maximal P_m' distinto al pareamiento generado en la etapa de Descomposición Gruesa. Si no se eliminara temporalmente algún nodo fila de $G(N)$, en la siguiente iteración, el algoritmo construiría un pareamiento maximal P_m' exactamente igual al original, pero con la salvedad que no contendría los nodos de los bloques N_{I_i} eliminados de $G(N)$.

Una vez que el pareamiento maximal P_m' es generado, la fila especial e se reincorpora al bigrafo, y se clasifica según las siguientes reglas:

- $e \in VR$ si todas sus aristas conducen a columnas de SCI ó $SC2$.
- $e \in SRI$ si está conectada a una única columna de HC y a ninguna columna de $SC2$ (en este caso dicha columna y e forman un bloque de orden 1).
- $e \in SR2$ si e está conectada a una única columna de HC y al menos a una columna de $SC2$ (en este caso la columna y e forman un bloque de orden 1).
- $e \in HR$ si está conectada a más de una columna de HC .

Una vez que todos los bloques de N_1 hayan superado en forma exitosa la comprobación, se realiza el Test de Admisión sobre el bloque $N_2 = (SR2, SC2)$. A diferencia del test efectuado sobre el bloque N_1 , los bloques rechazados de N_2 no pueden reasignarse; si se detecta un bloque N_{2i} que no supera el test, el algoritmo avanza directamente a la etapa de Reducción. Esto se ve claramente al observar la figura 3.6.a, puesto que las filas redundantes en el patrón FTiB sólo tienen elementos no nulos en las columnas correspondientes a $SR1$. Es por ello que la reasignación carece de sentido para N_2 .

Existen casos particulares, en donde el hecho de eliminar una única fila especial no resulta suficiente para hallar nuevos bloques admisibles. En estos casos, se deben eliminar temporalmente varias filas especiales hasta lograr un avance en la descomposición. Una vez que se encuentra un nuevo pareamiento maximal, se reincorporan al bigrafo las filas especiales eliminadas y luego se clasifican.

El algoritmo finaliza cuando se aceptaron todos los bloques de N_2 por el test de admisión o cuando los conjuntos $SR1$ y $SR2$ retornados por la Descomposición Gruesa están vacíos.

3.3.5 ETAPA 6: REORDENAMIENTO FINAL

El algoritmo reordena a la matriz N de la forma adecuada para alcanzar la FTiB.

3.4 ALGORITMO DEL MÉTODO DIRECTO

A continuación se incluye el algoritmo desarrollado por Ponzoni y *co.* (2004) para llevar a cabo la implementación del Método Directo (figura 3.7):

Entrada.: N (matriz de incidencia) y R (restricciones iniciales).
 Salida.: N reordenada a FTiB.

Etapa 0 Inicialización

- 0.1. Construir el bigrafo $G(N)=(R, C, E)$ asociado a N .

Etapa 1. Descomposición Gruesa.

- 1.1. Obtener un pareamiento maximal P_m de $G(N)$.
- 1.2. Clasificar las filas en $SR1, SR2, VR, HR$ según P_m .
- 1.3. Clasificar las columnas en $SC1, SC2$ y HC , según P_m .
- 1.4. Para cada fila especial que exista, clasificarla como:
 - VR** si todas sus aristas conducen a columnas de $SC1$ ó $SC2$.
 - SR1** si está conectada a una única columna de HC y a ninguna columna de $SC2$ (la columna y la fila especial forman un bloque de 1×1).
 - SR2** si está conectada a una única columna de HC y al menos a una columna de $SC2$ (la columna y la fila especial forman un bloque de 1×1).
 - HR** si está conectada a más de una columna de HC .

Etapa 2. Descomposición Fina.

- 2.1. Asociar el digrafo $G(N_1) = (V, E)$ a la matriz N_1 correspondiente al bloque ($SR1, SC1$).
 - 2.2. Descomponer $G(N)$ en sus componentes fuertes $N_{11}, N_{12}, \dots, N_{1q}$. Cada N_{1i} corresponde a un bloque de la diagonal.
 - 2.3. Asociar el digrafo $G(N_2) = (V, E)$ a la matriz N_2 correspondiente al bloque ($SR2, SC2$).
 - 2.4. Descomponer $G(N)$ en sus componentes fuertes $N_{21}, N_{22}, \dots, N_{2p}$. Cada componente N_{2j} corresponde a un bloque de la diagonal.
-

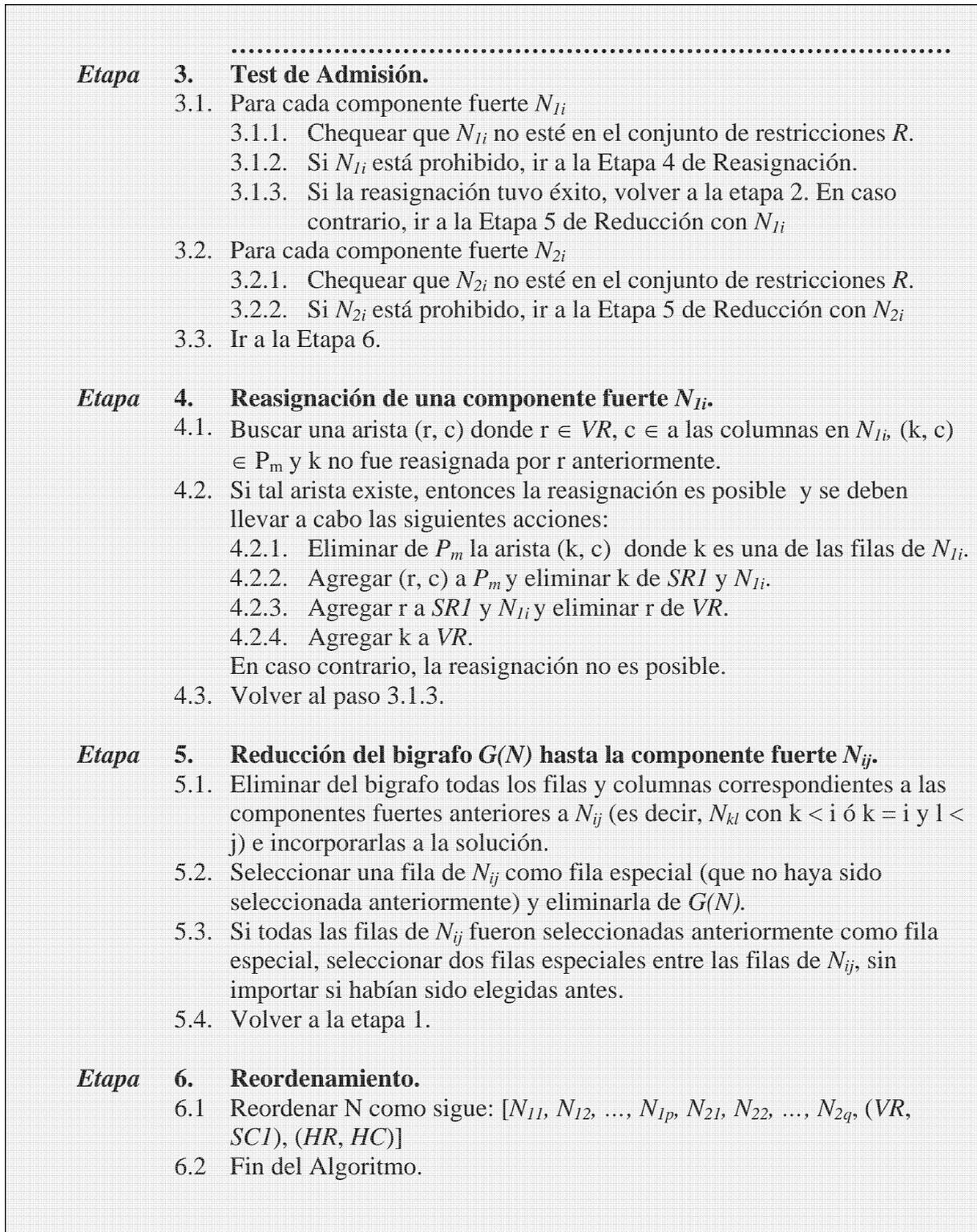


Figura 3.7. Algoritmo del Método Directo

3.5 EJEMPLO DE APLICACIÓN DEL MÉTODO DIRECTO

A continuación se presenta un ejemplo con los pasos completos que ejecuta el Método Directo para partir de una matriz general (figura 3.8) y llegar a la descomposición en bloques deseada.

Como se explicó anteriormente, la primera etapa es una etapa de inicialización, en la cual a partir de la matriz de incidencia N correspondiente al modelo matemático de la planta se genera su bigrafo asociado $G(N)$ (figura 3.9). Dicho bigrafo contiene dos conjuntos de nodos R y C , que representan a las filas y columnas de N respectivamente.

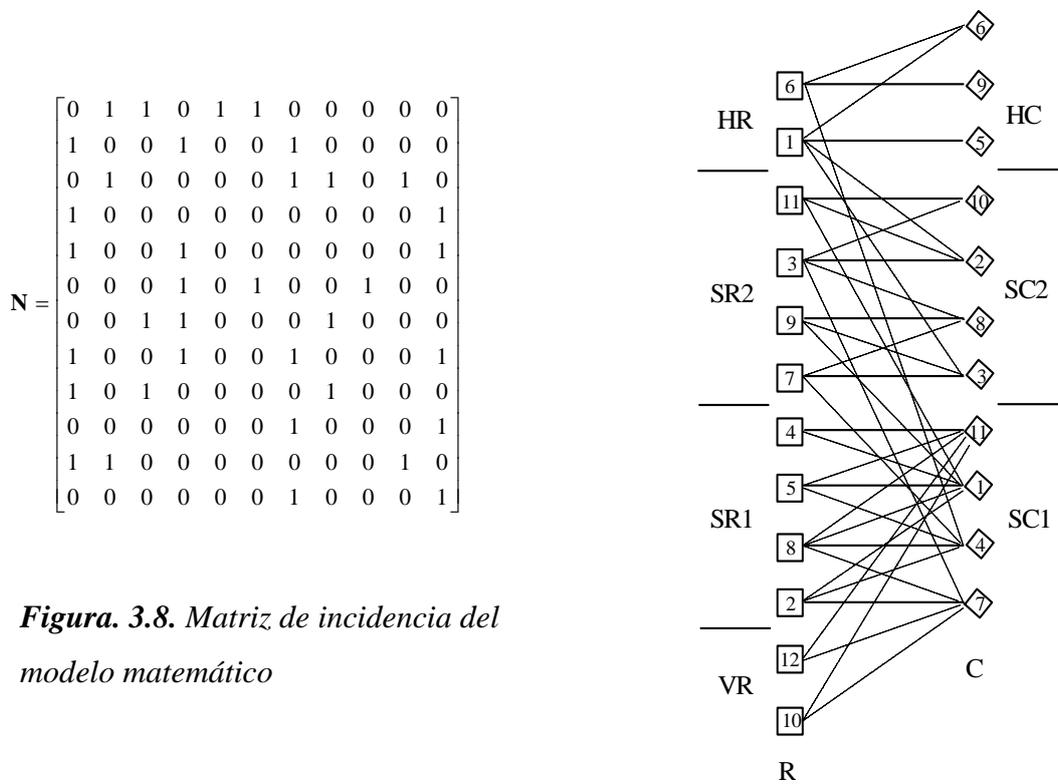


Figura 3.8. Matriz de incidencia del modelo matemático

Figura 3.9. Pareamiento Maximal del bigrafo $G(N)$

		SC1				SC2				HC		
		11	1	4	7	10	2	8	3	6	9	5
SR1	4	1	1	0	0	0	0	0	0	0	0	0
	5	1	1	1	0	0	0	0	0	0	0	0
	8	1	1	1	1	0	0	0	0	0	0	0
	2	1	0	0	1	0	0	0	0	0	0	0
SR2	11	0	1	0	0	1	1	0	0	0	0	0
	3	0	0	0	1	1	1	1	0	0	0	0
	9	0	1	0	0	0	0	1	1	0	0	0
	7	0	0	1	0	0	0	1	1	0	0	0
	12	1	0	0	1	0	0	0	0	0	0	0
10	0	1	1	1	0	0	0	0	0	0	0	
HC	6	0	0	1	0	0	0	0	0	1	1	0
	1	0	0	0	0	0	1	0	1	1	0	1

Figura 3.10. Matriz reordenada a partir de la descomposición gruesa

En el siguiente paso, etapa 1, se obtiene un pareamiento maximal P de $G(N)$ (figura 3.9). Un pareamiento está formado por un conjunto de aristas disjuntas (sin extremos en común) que conectan nodos de R con nodos de C (Gibbons, 1994). En particular, un pareamiento P es maximal si no existe otro pareamiento P' tal que P' posea más aristas que P . Las aristas de $G(N)$ indicadas en la figura 9 con trazo completo, corresponden al pareamiento maximal P encontrado mediante la aplicación del algoritmo PMB. Sobre la base de este pareamiento, se define una clasificación de los nodos R y C , la cual permite efectuar un reordenamiento de la matriz N a la forma mostrada en la figura 3.10. Esta clasificación de nodos, particiona a C en $SC1$, $SC2$ y HC , y a R en SRI , $SR2$, HR y VR . Los nodos de $SC1$ y $SC2$ están asociados a las variables observables, mientras que los de HC corresponden a las variables indeterminables. De forma similar, los nodos de SRI y $SR2$ representan las ecuaciones asignadas, los nodos de VR representan las ecuaciones redundantes y los de HR corresponden a las ecuaciones con variables indeterminables.

La siguiente fase (etapa 2) efectúa un proceso de sub-descomposición de los bloques $(SRI, SC1)$ y $(SR2, SC2)$ de la figura 3.10 empleando el clásico algoritmo de Tarjan (Tarjan, 1972). En particular, la figura 3.11 muestra la descomposición del bloque $N_2 = (SR2, SC2)$ (figura 3.11.a). En primera instancia se construye el digrafo $G(N_2)$ correspondiente al bloque (figura 3.11.b). Luego se particiona el digrafo en sus componentes fuertes (figura 3.11.c), y por

último se reordena N_2 de acuerdo a las componentes fuertes (figura 3.11.d). La descomposición fina del bloque $N_1 = (SRI, SCI)$ genera una única componente fuerte.

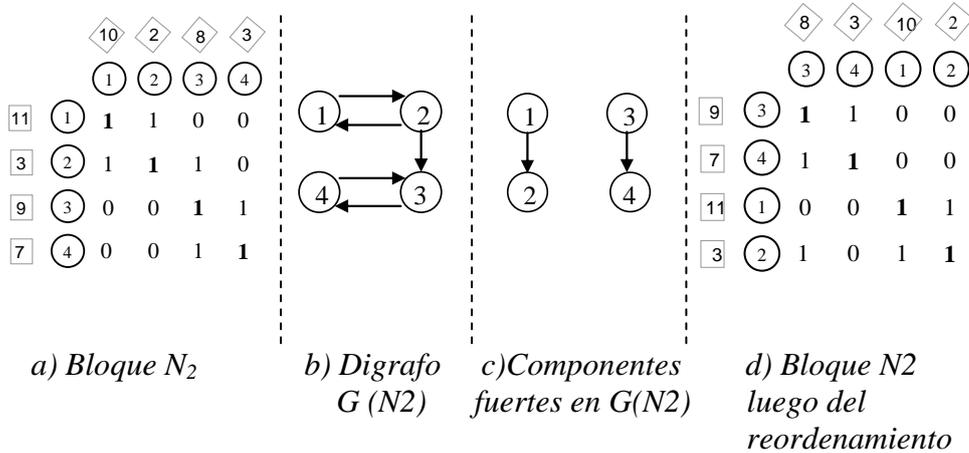


Figura 3.11. Descomposición Fina de $N_2 = (SC2, SR2)$

Puede suceder que algunos de los bloques hallados en la descomposición fina correspondan a un subsistema de ecuaciones numéricamente singular. En dicho casos, se genera una restricción que prohíbe la conformación de dicho subsistema (etapa 3). Sea T_1 por ejemplo una restricción correspondiente al bloque formado por los nodos fila {4, 5, 8, 2} y los nodos columna {11, 1, 4, 7}, el cual representa un subsistema de N numéricamente singular. En dicho caso, el bloque (SRI,SCI) no superaría el test de admisión, y por ende debe ser reasignado (etapa 4). Suponiendo que el algoritmo de reasignación selecciona para su permutación a los nodos fila 2 de SRI y 10 de VR , el bloque N_1 estará ahora compuesto por los nodos fila {4, 5, 8, 10}, como se puede ver en la figura 3.12, mientras que el nodo 2 corresponde a una ecuación redundante.

Si se vuelve a realizar la descomposición fina, se encuentra entonces una única componente fuerte integrada por todo el bloque N_1 . Supongamos ahora que el nuevo bloque, formado por los nodos filas {4, 5, 8, 10} y los nodos columna {11, 1, 4, 7}, es analizado por el test de admisión, y rechazado debido a que existe otra restricción T_2 formada por los nodos fila {4, 5, 10} y los nodos columna {11, 1, 4}. Esto provoca una nueva reasignación, en la cual se intercambian los nodos fila 4 y 12. El bloque N_1 resultante queda ahora formado por

los nodos filas {5, 8, 10, 12} y los nodos columna {7, 11, 1, 4}. Se repite la ejecución de la descomposición fina, y se obtienen ahora dos componentes fuertes $N_{11} = \{\{10, 12\}, \{7, 11\}\}$ y $N_{12} = \{\{5, 8\}, \{1, 4\}\}$. Ambos bloques superan la comprobación, y el test de admisión finaliza para N_j . Con esto termina el análisis de observabilidad para este ejemplo, quedando la matriz N reordenada tal como se muestra en la figura 3.13. Los elementos de la matriz resaltados en negrita corresponden a las aristas encontradas previamente, mediante la aplicación del algoritmo PMB.

		SC1				SC2				HC		
		11	1	4	7	8	3	10	2	6	9	5
SR1	4	1	1	0	0	0	0	0	0	0	0	0
	5	1	1	1	0	0	0	0	0	0	0	0
	8	1	1	1	1	0	0	0	0	0	0	0
	10	1	0	0	1	0	0	0	0	0	0	0
SR2	9	0	1	0	0	1	1	0	0	0	0	0
	7	0	0	0	1	1	1	0	0	0	0	0
	11	0	1	0	0	0	0	1	1	0	0	0
VR	3	0	0	1	0	1	0	1	1	0	0	0
	12	1	0	0	1	0	0	0	0	0	0	0
HR	2	0	1	1	1	0	0	0	0	0	0	0
	6	0	0	1	0	0	0	0	0	1	1	0
	1	0	0	0	0	0	1	0	1	1	0	1

Figura 3.12. Matriz resultante de la primera reasignación.

		SC1				SC2				HC		
		7	11	1	4	8	3	10	2	6	9	5
SR1	10	1	1	0	0	0	0	0	0	0	0	0
	12	1	1	0	0	0	0	0	0	0	0	0
	5	0	1	1	1	0	0	0	0	0	0	0
	8	1	1	1	1	0	0	0	0	0	0	0
SR2	9	0	0	1	0	1	1	0	0	0	0	0
	7	1	0	0	0	1	1	0	0	0	0	0
	11	0	0	1	0	0	0	1	1	0	0	0
VR	3	0	0	0	1	1	0	1	1	0	0	0
	4	0	1	1	0	0	0	0	0	0	0	0
HR	2	1	0	1	1	0	0	0	0	0	0	0
	6	0	0	0	1	0	0	0	0	1	1	0
	1	0	0	0	0	0	1	0	1	1	0	1

Figura 3.13. Matriz resultante de la segunda reasignación.

3.6 CONCLUSIONES

En este capítulo se presentaron los aspectos más generales de la teoría de grafos. Partiendo de dichos conceptos, se realizó una descripción detallada del módulo para análisis de observabilidad conocido como Método Directo. Este método es un algoritmo de particionamiento que efectúa un reordenamiento estructural de la matriz de incidencia correspondiente al modelo de estado estacionario de un proceso, para llevarla a una forma triangular inferior en bloques que establece una descomposición en subsistemas.

Esta técnica de partición, basada en descomposición de grafos, no efectúa búsquedas combinatorias ni emplea reglas heurísticas, sino que efectúa el reordenamiento de la matriz de incidencia en dos etapas, mediante el uso de bigrafos y digrafos. El algoritmo basa su funcionamiento en una nueva clasificación de nodos en pareamientos maximales de bigrafos, derivada de la propuesta por Dulmage y Mendelsohn para descomposición de matrices estructuralmente singulares.

El MD dio excelentes resultados al ser aplicado a la resolución de problemas de monitoreo de procesos, incluyendo casos académicos e industriales de mediano y gran tamaño. La técnica es extremadamente robusta y eficiente en tiempo de cómputo, y su eficacia aumenta a medida que los problemas crecen en tamaño y complejidad.

Hasta el momento, los estudios de eficacia del MD habían sido limitados al área de diseño de instrumentación. Sin embargo, el corazón de esta metodología es básicamente una nueva estrategia para analizar y / o resolver sistemas de ecuaciones algebraicas, no necesariamente lineales. En particular otros clásicos problemas complejos de ingeniería de procesos podrían beneficiarse de esta metodología. Por este motivo, en el capítulo 6 se propone realizar una extensión en el rango de aplicabilidad del Método Directo, haciendo una evaluación de su desempeño como preprocesamiento para la resolución de cualquier sistema de ecuaciones algebraicas, con especial énfasis en aquellos que surgen en otras aplicaciones del campo de la ingeniería de sistemas de procesos.

3.7 NOMENCLATURA DEL CAPÍTULO

A = Aristas del grafo

AR = Análisis de redundancia

C = Conjunto de nodos correspondiente a las columnas de la matriz *N*

FTiB = Matriz triangular inferior en bloques

G = Grafo

HC = Variables indeterminables

HR = Ecuaciones con variables indeterminables

MD = Método Directo

MGM = Módulo generador de modelos

N = Matriz de incidencia

P_M = Pareamiento Maximal

R = Conjunto de nodos correspondiente a las filas de la matriz *N*

SC = Variables observables

SR = Ecuaciones asignadas

VR = Ecuaciones redundantes

MÉTODO DIRECTO EXTENDIDO

4.1 INTRODUCCIÓN

En el capítulo 3 de esta Tesis se presentó en detalle el funcionamiento del módulo de análisis de observabilidad del DSS, el cual es llevado a cabo mediante la aplicación del algoritmo conocido como Método Directo (Ponzoni y *co.*, 2004).

Generalmente las técnicas propuestas para asistir en el análisis de observabilidad (AO), buscan determinar un diseño de instrumentación que maximice la cantidad de variables calculables de una planta. Esto se logra a partir de un procedimiento que descompone el modelo matemático en una secuencia de subsistemas de ecuaciones (ver capítulo 3). Dichos subsistemas permiten determinar qué variables pueden ser calculadas a partir de los datos arrojados por un conjunto de sensores.

Una característica de los modelos que representan los procesos químicos es su fuerte carácter no lineal. El grado de no linealidad (GNL) presente en los subsistemas obtenidos por análisis de observabilidad posee un fuerte impacto en el desempeño de las técnicas utilizadas para efectuar el análisis de redundancia (Ferraro y *co.*, 2002). Estas técnicas reducen significativamente sus tiempos y esfuerzos de cómputo cuanto mayor es la cantidad de subsistemas lineales que debe procesar respecto del número de sistemas no lineales. Otra característica sobresaliente es que las matrices de incidencia asociadas a los sistemas de ecuaciones que conforman los modelos de los procesos industriales son generalmente deficientes en rango. Esto se debe a la presencia de ecuaciones redundantes, cuyo propósito es permitir la reconciliación de datos a fin de detectar errores de medición en los sensores ubicados en la planta (Romagnoli y Sánchez, 1999).

Como consecuencia de esto, no existe una única forma de descomponer el sistemas de ecuaciones, y por ende, distintas particiones pueden conducir a subsistemas con distintos grados de no linealidad en función de las ecuaciones y variables que conforman cada subsistema. En este contexto, resultaría deseable encontrar aquella descomposición que tiende a disminuir la no linealidad de los subsistemas obtenidos, moviendo las ecuaciones no lineales al conjunto de ecuaciones redundantes o no asignadas, de acuerdo a la categoría detallada en la sección 2.3 de esta tesis. En la actualidad, ninguno de los algoritmos de búsqueda empleados en AO tiene en cuenta este factor, lo cual motivó a realizar un análisis de la factibilidad de extender alguna de las técnicas existentes a fin de incorporar este aspecto al proceso de descomposición.

En particular, en este capítulo se presenta una versión modificada del Método Directo. Esta nueva versión, denominada Método Directo Extendido (MDE) (Domancich y *co.*, 2009b), favorece la reducción del grado de no linealidad en la descomposición obtenida, sin afectar el grado de conocimiento alcanzado con el MD, lo cual constituye la principal contribución de la nueva propuesta. Esta nueva metodología de análisis de observabilidad surge de la incorporación del análisis del GNL al MD original, permitiendo así mejorar su desempeño y rango de aplicación. El objetivo principal de incorporar el análisis del GNL es reducir la cantidad de subsistemas no lineales obtenidos por la descomposición del MD. Al respecto, el primer paso para alcanzar esta meta consiste en definir cómo cuantificar el grado de no linealidad que aporta cada ecuación y variable del sistema al modelo matemático. Esta información resulta fundamental para guiar el proceso de descomposición del modelo, a fin de favorecer la conformación de subsistemas lineales frente a los no lineales.

4.2 DEFINICIÓN DEL GRADO DE NO LINEALIDAD

4.2.1 GRADO DE NO-LINEALIDAD EN LAS ECUACIONES

En todas las ecuaciones existen términos lineales, bilineales o no lineales, dependiendo de las operaciones algebraicas que incluya. Por ejemplo, en la ecuación $a + b^3 + c.d = 0$, el primer término es lineal, el segundo no lineal y el tercero bilineal. El GNL de una ecuación está determinado por una suma de factores de peso asociados a la linealidad o no linealidad de los términos que en ella aparecen.

4.2.2 GRADO DE NO-LINEALIDAD EN LAS VARIABLES:

El GNL de una variable está definido como la combinación de los factores que definen las linealidades o no linealidades de todos los términos en donde dicha variable aparece a lo largo de todas las ecuaciones del sistema. Dicho de otra forma, para cuantificar el GNL de una variable se recorre la matriz de incidencia de manera vertical, siendo la columna recorrida aquella perteneciente a la variable en cuestión.

4.3 CUANTIFICACIÓN DEL GRADO DE NO LINEALIDAD

Para calcular el GNL fue necesario definir un sistema de factores de peso o puntajes, a fin de asignar un peso específico a las linealidades o no linealidades de los términos. De esta manera, cada término tendrá un puntaje diferente, de acuerdo a las operaciones algebraicas que incluya. El criterio utilizado asigna un peso menor a los términos lineales, incrementando este valor a medida que aumenta la no linealidad. La tabla 4.1 es completamente general y especifica los pesos utilizados para la determinación del GNL para las ecuaciones y variables del sistema. Es importante destacar que es fundamental elegir factores de peso para la evaluación que incrementen su valor a medida que aumenta la no linealidad del término a evaluar, sin importar el valor específico que adopten siempre que se mantenga este criterio.

Término	Lineal	Bilineal	No Lineal				
			1 var.	2 var.	3 var.	4 var.	5 o + var.
Pesos	0	1	2.2	2.4	2.6	2.8	3.0

Tabla 4.1. Ponderación del tipo de términos según grado de no linealidad

Como se verá en la sección 4.9.1, es fundamental respetar el orden de ponderación mostrado en la tabla 4.1 para lograr obtener el mínimo grado de no linealidad en la descomposición resultante. A su vez, en dicha sección se puede apreciar como, para un mismo orden de ponderación, el escalar utilizado como peso no tiene influencia sobre el número de bloques no lineales obtenidos, sino que solamente influye en la cantidad de bloques de mínimo tamaño obtenido. Es importante destacar que no es factible efectuar un seteo universal de esos valores para cualquier tipo de plantas industriales, dado que la estructura algebraica de las ecuaciones involucradas en distintos tipos de proceso puede ser significativamente diferente. Lo aconsejable es que, dado un proceso específico, se realice una experimentación preliminar para ajustar de manera empírica estos valores. Un ejemplo de este

tipo de análisis experimental es reportado en la sección 4.9.1.2 para el estudio de una columna de destilación.

A continuación se detalla y ejemplifica cómo se realiza el cálculo del GNL para las ecuaciones y variables de un modelo matemático.

4.3.1 CUANTIFICACIÓN EN ECUACIONES

Para cada ecuación se realiza una sumatoria del puntaje correspondiente a cada uno de los términos involucrados en ella, y se divide por la cantidad de términos, de manera tal de normalizar el valor.

4.3.2 CUANTIFICACIÓN EN VARIABLES:

Para cada variable se realiza una sumatoria del puntaje correspondiente a cada uno de los términos (en todas las ecuaciones del sistema) en los que dicha variable interviene, y se divide por la cantidad de términos para normalizar el valor.

Para ayudar a la comprensión de este mecanismo, a continuación se presentan dos ejemplos donde se puede apreciar en casos concretos cómo funciona la metodología de cálculo propuesta.

4.3.3 EJEMPLOS DE CUANTIFICACIÓN DEL GRADO DE NO LINEALIDAD

EJEMPLO 4.1: Dada la ecuación $E_1: a + b^3 + c.d = 0$, calcular el GNL tanto para la ecuación como para las variables que la componen.

El cálculo debe ser realizado en dos etapas:

a) GNL para la ecuación:

La ecuación tiene $t=3$ términos. El puntaje w_i dado en la tabla 4.1 se aplica utilizando la siguiente fórmula:

$$GNL(E) = \frac{\sum_{i=1}^t w_i}{t} \quad (E 4.1)$$

Aplicando la fórmula anterior a la ecuación E1, se obtiene:

$$GNL(E_1) = \frac{0 + 2.2 + 1}{3} = 1.07 \quad (\text{E 4.2})$$

b) GNL para las variables:

Llamando t_v al número de términos donde la variable v participa, se puede aplicar la siguiente fórmula para el cálculo del GNL para cada una de las variables que integran la ecuación:

$$GNL(v) = \frac{\sum_{i=1}^{t_v} w_i}{t_v} \quad (\text{E 4.3})$$

Para el ejemplo 4.2 se obtiene,

Variable a:

$$GNL(a) = \frac{0}{1} = 0 \quad (\text{E 4.4})$$

Variable b:

$$GNL(b) = \frac{2.2}{1} = 2.2 \quad (\text{E 4.5})$$

Variable c:

$$GNL(c) = \frac{1.1}{1} = 1.1 \quad (\text{E 4.6})$$

Variable d:

$$GNL(d) = \frac{1}{1} = 1 \quad (\text{E 4.7})$$

EJEMPLO 4.2: Dado el siguiente sistema de ecuaciones de 3x3 (figura 4.1), con **a**, **b** y **c** como variables, calcular el GNL para las ecuaciones y las variables que conforman el sistema.

$$\left\{ \begin{array}{l} a^2 + b + b.c = 0 \\ \frac{3c^2}{a} + b.c = 0 \\ a + 3b + 5c = 0 \end{array} \right.$$

Figura 4.1. Sistema de ecuaciones que representa un proceso industrial.

a) GNL para las ecuaciones:

Ecuación 1:

$$GNL(E_1) = \frac{2.2 + 0 + 1}{3} = 1.07 \quad (\text{E 4.8})$$

Ecuación 2:

$$GNL(E_2) = \frac{2.4 + 1}{2} = 1.7 \quad (\text{E 4.9})$$

Ecuación 3:

$$GNL(E_3) = \frac{0 + 0 + 0}{3} = 0 \quad (\text{E 4.10})$$

Como puede apreciarse en los resultados obtenidos, la ecuación 3 es la “menos no lineal” dado que presenta el menor GNL mientras que la ecuación 2 es la “más no lineal” por ser la de mayor GNL.

b) GNL para las variables:

Variable a:

$$GNL(a) = \frac{2.2 + 2.4 + 0}{3} = 1.53 \quad (\text{E 4.11})$$

Variable b:

$$GNL(b) = \frac{0+1+1+0}{4} = 0.5 \quad (\text{E 4.12})$$

Variable c:

$$GNL(c) = \frac{1+2.4+1+0}{4} = 1.1 \quad (\text{E 4.13})$$

Continuando con el razonamiento anterior, la variable **b** es la “menos no lineal” (GNL=0.5) y la variable **a** es la “más no lineal” (GNL=1.53)

4.4 INTEGRACIÓN DEL GRADO DE NO LINEALIDAD

Una vez calculados los coeficientes de ponderación del GNL de variables y ecuaciones, estos valores deben ser integrados al proceso de descomposición gruesa que realiza el MD, de forma tal que los nodos seleccionados para el pareamiento maximal del bigrafo resulten ser preferentemente aquellos con el menor valor de GNL. De este modo, se obtiene un MDE que favorece la conformación de bloques de descomposición con menor grado de no linealidad.

En la figura 4.2 se presenta el pseudo-código del algoritmo PMB (Pareamiento Maximal de un Bigrafo) modificado, utilizado para calcular el pareamiento maximal de un bigrafo incorporando el análisis de GNL. La entrada de este algoritmo está conformada por la matriz de incidencia correspondiente al modelo matemático analizado (N), el conjunto de nodos del bigrafo correspondiente a las columnas de la matriz N (C) y el conjunto de nodos del bigrafo correspondiente a las filas de la matriz N (R). Completan el conjunto de datos de entrada al algoritmo las estructuras adicionales NVR y NVC, las cuales representan los vectores que almacenan los coeficientes de GNL correspondientes a las ecuaciones y variables del sistema respectivamente. Básicamente, la heurística introducida en el proceso de búsqueda del pareamiento maximal consiste en dar prioridad a la exploración de los nodos adyacentes con menor GNL sobre aquellos con mayor GNL.

Datos de entrada: N, R, C, NVR, NVC

Datos de salida: P_M

$P_M = \emptyset$
 $C_U = \emptyset$

// Reordenamiento de los nodos del bigrafo según sus coeficientes de GNL
 Reordenar la numeración de los nodos fila y columna de menor a mayor en función de sus coeficientes de GNL (NVR y NVC).

// Construcción de un pareamiento inicial

Para cada nodo $c \in C$, siguiendo la numeración del paso anterior, hacer:
 Aparear c con un nodo no apareado $r \in R$, si hay varios tomar el de menor numeración.
 Si no existe tal nodo r entonces $C = C \cup \{c\}$

fin-para

// Búsqueda de caminos aumentados
 $C_{UN} = \emptyset$

Repetir
 Buscar un camino aumentado A_u desde c , visitando únicamente nodos de R que no han sido visitados antes durante este paso.
 Marcar todos los nodos alcanzados como visitados.
Si se ha encontrado un camino aumentado A_u
entonces
 Aumentar P_M con A_u .

Sino
 Incluir c en C_{UN}

fin-si.
 $C = C_{UN}$.
 $C_{UN} = \emptyset$

hasta que no se encuentre ningún camino aumentado en un paso.

Figura 4.2. Versión modificada del algoritmo PMB.

Como se explicará a continuación, el hecho de que cada nodo deba estar apareado con el adyacente de menor GNL implica que el algoritmo deba realizar una búsqueda secuencial, con la consecuente disminución en el rendimiento y pérdida de escalabilidad. Es por ello que la implementación efectiva difiere del algoritmo, aunque mantiene la misma idea: aparear los nodos con el menor GNL posible de manera tal de obtener una descomposición en bloques con la mayor cantidad posible de bloques lineales.

4.5 MÓDULOS Y ENCABEZADOS QUE CONFORMAN EL MDE

El MDE, desarrollado en lenguaje C, contiene los siguientes módulos:

- **MetDirExtRec.c:** Es el programa principal. Desde aquí se invocan a las funciones que calculan el GNL de variables y ecuaciones, y desarrollan las etapas de Descomposición Gruesa, Descomposición Fina, Test de admisión, Reasignación y Reducción.
- **GNL_Parser.c:** Contiene las rutinas que se encargan de calcular el GNL de variables y ecuaciones, siguiendo la metodología explicada en la sección 4.3. Para que el MDE sea capaz de utilizar los GNL de ecuaciones y variables en su descomposición, resultó necesario especificar el mecanismo por el cual dichos valores son incorporados al algoritmo del Método Directo. Para realizar esta tarea se decidió hacer uso de una herramienta conocida como analizador sintáctico (*parser*). Un *parser* es un algoritmo que busca y reconoce cadenas de texto con un cierto patrón dentro de un archivo de texto. Para realizar el cálculo del GNL el analizador sintáctico fue programado de manera tal de operar en dos etapas. En una primera etapa se aplica un nuevo formato a los términos de las ecuaciones generados por el Módulo Generador de Modelos, generando un nuevo archivo de texto. En una segunda etapa, se recorre cada ecuación del nuevo archivo, calculando el GNL correspondiente a las ecuaciones y variables. Debido a que programar un *parser* no es una tarea sencilla, se decidió hacer uso de una herramienta ya disponible como es Flex 2.5 (1995). Flex es un generador rápido de *parsers*, es decir, una herramienta para “programar” *parsers*. Su comportamiento es similar a un programa escrito en algún lenguaje compilable: en un archivo fuente se indican los patrones o expresiones regulares a reconocer y las acciones asociadas escritas en lenguaje C. Al compilar el archivo fuente, Flex genera como resultado el código C correspondiente al *parser* que identifica dichas expresiones.
- **Fparser.l:** Contiene las expresiones regulares a reconocer por el analizador sintáctico, y el código C asociado a cada una. Es el archivo fuente del *parser* que genera Flex.

- **Flex_GNL_Parser.c:** Contiene el parser (generado por Flex al compilar el archivo fuente `fparser.l`). Incluye las rutinas para reconocer las cadenas de texto del sistema.
- **Bigrafo.c:** Guarda las rutinas que operan sobre bigrafos. Se encarga de cargar el bigrafo desde el archivo correspondiente, calcular los pareamientos maximales y ordenar los nodos adyacentes en función del GNL.
- **Digrafo.c:** Contiene el código que opera dentro de la Descomposición Fina. Obtiene los dígrafos asociados y calcula sus componentes fuertes, entre otras cosas.
- **Restric.c:** Guarda las rutinas que administran las restricciones externas que se aplican sobre el sistema de ecuaciones, y que se utilizan dentro del Test de Admisión.
- **Subset.c:** Administra los subconjuntos de asignación que incluyen los bloques aceptados por el Test de Admisión del método.

Los encabezados que conforman el programa son los siguientes:

- **_bigrafo.h:** Declara las estructuras para trabajar con bigrafos.
- **_digrafo.h:** Declara las estructuras para trabajar con digrafos.
- **_dimensiones.h:** Declara las dimensiones máximas de cada estructura.
- **_GNL.h:** Declara las estructuras para guardar el GNL asociado.
- **_GNLcoef.h:** Declara los coeficientes que indican la linealidad de cada término.
- **_GNLterminos.h:** Declara los valores internos para identificar cada tipo de término. Se utiliza en `Fparser.l` y en `GNL_Parser.c`.
- **_namefiles.h:** Declara el tamaño máximo de los nombres de archivos, así como las extensiones predefinidas.
- **_restric.h:** Declara las estructuras de datos para trabajar con las restricciones del sistema.

- **_subsets.h:** Declara las estructuras necesarias para trabajar con los subconjuntos de asignaciones.

4.6 ARCHIVOS DE ENTRADA Y SALIDA DEL MDE

A continuación se detalla la lista de archivos de entrada necesarios para utilizar el programa y los archivos de salida obtenidos a partir de su ejecución.

4.6.1 ARCHIVOS DE ENTRADA

Estos archivos son generados de manera automática por el MGM y cada uno contiene un tipo de información diferente de acuerdo a la extensión que poseen.

- **.rtf:** Contiene el sistema de ecuaciones que representa al proceso industrial.
- **.bgf:** Contiene el bigrafo inicial, indicando los nodos fila (ecuaciones) y sus columnas (variables) adyacentes.
- **.equ:** Guarda una lista de las ecuaciones del sistema, con el número y nombre cada una. También guarda la cantidad total de ecuaciones.
- **.var:** Es similar al anterior, solo que la información está referida a las variables del sistema.
- **.cts:** Contiene las restricciones externas del sistema.

4.6.2 ARCHIVOS DE SALIDA

Los archivos obtenidos mediante la ejecución del MDE son los siguientes:

- **.BGFequ:** Contiene el bigrafo ordenado por nodo fila y los nodos columna adyacentes ordenados según su correspondiente GNL. No tiene peso en la aplicación del método, sino que se utiliza únicamente para controlar el funcionamiento en la etapa de desarrollo y depuración.
- **.BGFvar:** Análogo al anterior, pero en este caso el bigrafo se encuentra ordenado por nodo columna, con los nodos fila adyacentes ordenados por su GNL asociado.
- **.GNLequ:** Guarda la lista de las ecuaciones del sistema, y su GNL.
- **.GNLvar:** Guarda la lista de las variables del sistema, y su correspondiente GNL.

- **.rsp:** Contiene la lista de los subconjuntos de asignación aceptados.
- **.sol:** Contiene una versión intermedia de los subconjuntos de asignación, y se utiliza como interfase con otros sistemas de simulación.
- **.system:** Guarda el sistema de ecuaciones resultante de aplicar el *parser* al sistema de ecuaciones del modelo.

4.7 COMPARACIÓN DE DESEMPEÑO ENTRE EL MDE Y EL MD ORIGINAL. PRESENTACIÓN DE CASOS DE ESTUDIO.

Dado que el Método Directo Extendido representa una evolución del Método Directo original, se realizó una comparación del funcionamiento de ambos para tres casos de estudio diferentes, los cuales representan procesos industriales reales con diferentes niveles de complejidad. De esta manera se pudo evaluar en forma concreta la magnitud de las mejoras obtenidas con respecto al módulo original. Los resultados obtenidos son comparados mediante la utilización de la siguiente fórmula:

$$\Delta = \frac{MDE_0 - MD_0}{MD_0} * 100 \quad (E 4.14)$$

donde Δ es un porcentaje que cuantifica la diferencia entre ambos métodos. MD_0 y MDE_0 representan diferentes variables de salida de interés, obtenidas mediante la aplicación de ambos métodos, las cuales son detalladas a continuación,.

Particularmente, para el primer caso de estudio, se presenta un análisis comparativo del resultado de emplear distintos conjuntos de pesos (tabla 4.1) para evaluar el grado de no linealidad de los términos matemáticos, a fin de fundamentar la elección de los valores empleados para el desarrollo del algoritmo. Adicionalmente, se incluye para este caso de estudio un análisis del impacto que posee la reducción del grado de no linealidad sobre la fase de análisis de redundancias. De esta manera se logró cuantificar la mejora en el desempeño global del DSS a partir de la incorporación al mismo del método directo modificado.

4.7.1 COLUMNA DE DESTILACIÓN

4.7.1.1 DESCRIPCIÓN DEL PROCESO INDUSTRIAL

El primer ejemplo corresponde a la sección de purificación mediante destilación de una planta de síntesis de amoníaco. En esta sección se utiliza una columna de destilación de dos etapas, la cual recibe en su primera etapa una corriente compuesta por amoníaco y agua. Para recuperar el amoníaco como producto por el tope de la columna (99.5% de pureza) se utiliza un condensador total, mientras que por el fondo el producto es agua en más de un 99%. A continuación se incluye un esquema de la columna (figura 4.3)

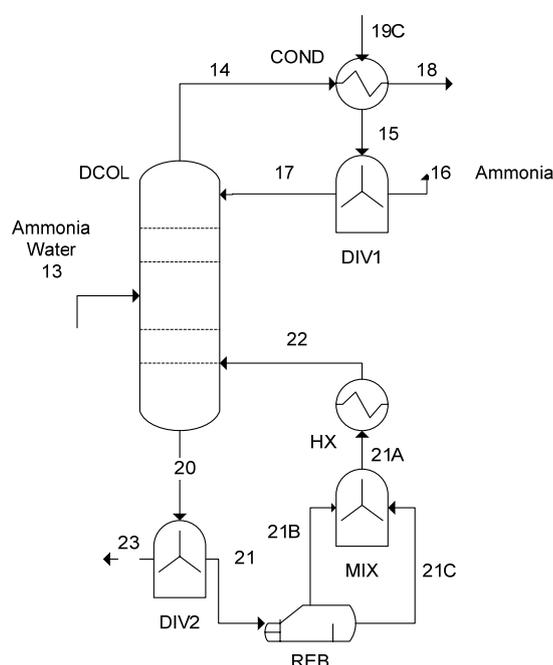


Figura 4.3. Esquema de la columna de destilación

El modelo matemático está compuesto por 104 ecuaciones y 110 variables, de las cuales 25 son variables medidas y 85 son no medidas.

4.7.1.2 EVALUACIÓN EXPERIMENTAL DE LOS PESOS A UTILIZAR

Para definir la heurística a utilizar por el Método Directo Extendido se realizaron diferentes pruebas, las cuales pueden ser agrupadas en 3 casos diferentes. Se realizó este estudio completo sobre sólo uno de los procesos industriales presentados en este capítulo debido a que el mismo resulta totalmente análogo para cualquier otro caso de estudio. El proceso analizado corresponde al modelo matemático de una columna de destilación,

Como se mencionó anteriormente, el estudio realizado consta de 3 casos. Cada uno de los casos presentados sigue un criterio de puntuación general diferente, de manera tal de lograr analizar el comportamiento del algoritmo de descomposición para diferentes escenarios. A su vez, para cada uno de los 3 casos se fueron variando los escalares utilizados para puntuar, siempre manteniendo el criterio de puntuación correspondiente al caso. De esta manera se logró analizar también la influencia de emplear distintos valores siguiendo un mismo criterio general de puntuación.

En el primero de los casos (caso A) se invirtió completamente la lógica planificada originalmente, dando entonces la mayor penalización (puntaje) a los términos lineales, continuando con los términos bilineales, y finalmente, se asignó con el menor puntaje a los términos no lineales. Esto, como era de esperarse, condujo a la descomposición con mayor cantidad de bloques no lineales, y a su vez, menor cantidad de bloques de 1×1 , lo cual constituye el peor resultado obtenido. Este resultado resulta lógico, dado que dentro del algoritmo del Método Directo, se está minimizando una función compuesta por varios pesos, con el propósito de representar el grado de no linealidad. Si se asigna el valor más alto a los bloques lineales, resulta obvio que no se va a obtener el resultado esperado. Si en cambio, el objetivo fuese maximizar la función (GNL), en ese caso se daría que asignar el valor más alto a los términos lineales ayudaría a cumplir con ese objetivo.

En segundo lugar, se asignó mayor penalización a los subsistemas bilineales por sobre los lineales (caso B), manteniendo, al igual que en el caso A, la menor puntuación para los bloques no lineales, continuando así con la preferencia de estos subsistemas por sobre el resto. El hecho de asignar un menor puntaje a los bloques lineales respecto de los bilineales, condujo a una mejora en cuanto a la cantidad de bloques no lineales obtenida, así como también en los subsistemas de mínimo tamaño. Es claro que, al continuar favoreciendo a los bloques no lineales por sobre el resto, la partición obtenida continúa sin ser la mejor posible.

Por último, se realizaron las pruebas asignando los puntajes que seguían el criterio recomendado en esta Tesis, es decir, dando el mayor puntaje a los subsistemas no lineales, seguido por los bilineales, asignando entonces la menor penalización a los subconjuntos de ecuaciones lineales (caso C). En este caso, teniendo en cuenta la cantidad de bloques no lineales obtenidos, se generó la mejor partición posible. A su vez, la variación en los escalares utilizados, permitió obtener diferentes descomposiciones, las cuales solo difirieron en el número de subsistemas de 1×1 , pero mantienen el mismo nivel de linealidad. De este análisis

surge que es fundamental mantener el orden de precedencia establecido en el caso C entre los distintos tipos de términos (lineal, bilineal, no lineales), más allá de los valores específicos utilizados para la ponderación.

A continuación se presenta una tabla (tabla 4.2) que resume los resultados obtenidos para cada uno de los casos estudiados. En la misma, se incluye en la primera fila la descomposición obtenida mediante la aplicación del MD. Luego se puede apreciar como, a medida que se desciende por la tabla (analizando desde el caso A al C), el número de bloques no lineales va disminuyendo, mientras que el número de bloques de 1x1 se incrementa.

	PARÁMETROS							Bloques No lineales	Bloques de 1x1
	Lineal	Bilineal	No lineal						
			1 V	2 V	3 V	4 V	5+V		
MD	0	0	0	0	0	0	0	4	48
CASO A	4	0	2	2.2	2.4	2.6	2.8	4	48
	4	3	2	2.2	2.4	2.6	2.8	4	48
	6	3	1.8	1.6	1.4	1.2	0	4	50
CASO B	2	3	1.8	1.6	1.4	1.2	0	3	55
	3	5	1.8	1.6	1.4	1.2	0	3	55
	6	7	1.8	1.6	1.4	1.2	0	3	56
CASO C	0	1	7	7.2	7.4	7.6	7.8	2	56
	0	1	2	3	3.2	3.4	3.6	2	58
	0	1	2.2	2.4	2.6	2.8	3	2	58

Tabla 4.2. Resultados de observabilidad para el modelo de la columna de destilación con el MDE, para diferentes conjuntos de pesos.

Es importante destacar la influencia que los escalares elegidos poseen sobre la descomposición obtenida, para un mismo criterio de ponderación. Analizando la tabla 4.2, se puede ver que, para un mismo criterio de ponderación, el escalar utilizado como peso no tiene influencia sobre el número de bloques no lineales alcanzado, pero sí posee influencia sobre el número de bloques de mínimo tamaño resultante en la descomposición. Si se analiza cada caso de la tabla de manera individual, se puede apreciar que el número de subsistemas no lineales se mantiene constante, mientras que el número de bloques de mínimo tamaño difiere dependiendo del set de valores elegido para llevar a cabo el cálculo de los grados de no linealidad.

Es importante aclarar que en los tres casos, y para cualquier conjunto de valores, se obtuvo el máximo grado de observabilidad posible, siendo a su vez este igual al valor obtenido mediante la aplicación del MD original.

4.7.1.3 ANÁLISIS DE OBSERVABILIDAD UTILIZANDO EL MD Y EL MDE

A continuación se incluye una tabla (tabla 4.3) con los resultados de observabilidad obtenidos mediante la aplicación del MD y el MDE, para el modelo matemático de la columna de destilación.

Cantidad de	MD	MDE	Δ
Variables Observables	63	63	0
Variables No Observables	22	22	0
Subsistemas 1x1	48	58	+ 21%
Subsistemas Lineales	47	57	+ 21%
Subsistemas No Lineales	4	2	- 50%

Tabla 4.3. Resultados de observabilidad obtenidos para el modelo de la columna de destilación aplicando el MD y el MDE.

Ambos métodos logran determinar la misma cantidad de variables observables, pero en la aplicación del MDE se aprecia una notable mejoría, puesto que se alcanza un incremento del 21% tanto en la producción de subsistemas de dimensión más pequeña (1x1) como en la generación de subsistemas lineales. El resultado más relevante está dado por la reducción en un 50% en la cantidad de subsistemas no lineales.

4.7.1.4 ANÁLISIS DE REDUNDANCIA DEL PROCESO A PARTIR DE LAS DESCOMPOSICIONES OBTENIDAS POR EL MD Y EL MDE

El MDE genera un reacomodamiento del sistema de ecuaciones que simplifica la tarea del algoritmo de análisis de redundancia (AR) (Ferraro y *co.*, 2002), acelerando su ejecución y optimizando su capacidad de análisis de las descomposiciones alimentadas al mismo. Para mostrar esto, es necesario explicar brevemente la principal idea detrás del algoritmo de AR. Para lograr analizar si una determinada variable medida m es redundante, el algoritmo AR toma cada ecuación redundante y computa de manera simbólica la derivada de su miembro izquierdo con respecto a m . Esto se realiza por medio de una interfase en Visual C, la cual interactúa con el programa Maple (Waterloo, 2009), quien tiene la capacidad de realizar diferenciación analítica de ecuaciones. Las derivadas calculadas deben además tener en cuenta la influencia indirecta de m en la expresión a través de las variables observables, cuyos valores podrían depender de esa medición. El algoritmo utiliza la regla de la cadena para computar esta influencia indirecta. Esto involucra resolver una secuencia de sistemas lineales cuya matriz de coeficientes está conformada por los Jacobianos simbólicos de los subsistemas

generados por medio del algoritmo de observabilidad. Los pasos subsecuentes en el algoritmo de AR requieren el cómputo adicional de derivadas simbólicas utilizando la regla de la cadena.

El punto fundamental en el que se focaliza la mejora implementada al algoritmo de observabilidad es que los subsistemas no lineales generalmente poseen un Jacobiano cuya inversa tiene una expresión complicada. Tener esos subsistemas en el tope de la descomposición tiene un efecto adverso en la eficiencia del análisis de redundancia, ya que los bloques siguientes tendrán alguna dependencia indirecta de esos subsistemas. El MDE favorece la formación de bloques lineales, de manera tal que los subsistemas no lineales tienden a aparecer al final de la descomposición, minimizando su impacto. Esto puede mejorar de manera significativa el proceso de análisis de redundancia.

Otro punto importante es que, si las expresiones simbólicas son tan complejas que el AR no puede decidir si la medición m es redundante (Red) o no redundante (N-Red) luego de un tiempo determinado de ejecución o un cierto número de iteraciones, entonces la variable m es etiquetada como “no categorizada” (N-Cat). De esta manera, la obtención de expresiones más simples dentro de los bloques generalmente reducen el número de mediciones no categorizadas, incrementando así el nivel de conocimiento de la planta provisto por el algoritmo de análisis de redundancia.

Adicionalmente, la preferencia que brinda el MDE a las ecuaciones con bajo grado de no linealidad para construir los subsistemas, tiende a producir una partición con un número mayor de ecuaciones lineales y bilineales asignadas, y mayor número de ecuaciones no lineales redundantes. Esto también es beneficioso para el AR, dado que tener una ecuación no lineal como dentro del grupo de las redundantes es menos perjudicial que tenerla formando parte de uno de los subsistemas formados por el algoritmo de observabilidad.

A continuación se presenta la tabla 4.4 con el resultado del análisis de redundancia ejecutado para el caso de estudio de la columna de destilación. Se compara el desempeño del algoritmo de AR para las descomposiciones obtenidas mediante la aplicación del MD y el MDE.

En esta tabla se resumen los resultados obtenidos para la etapa del diseño de instrumentación conocida como análisis de redundancia. El análisis fue realizado tanto para la

descomposición de bloques obtenida mediante la aplicación del Método Directo original (MD), como para la que se obtuvo por el Método Directo Extendido (MDE).

VARIABLE	CATEGORIZACIÓN	
	MD	MDE
F(20)	N-Cat	Red
X(22,20)	N-Cat	Red
P(21)	Red	Red
T(20)	N-Cat	N-Cat
P(23)	Red	Red
F(22)	N-Cat	Red
T(22)	N-Red	N-Red
P(HX)	N-Red	N-Red
P(21A)	N-Red	N-Red
T(21A)	N-Red	N-Red
T(21C)	N-Cat	Red
F(14)	N-Cat	N-Cat
P(13)	N-Red	N-Red
T(13)	N-Red	N-Red
X(22,13)	N-Red	N-Red
P(14)	N-Red	N-Red
T(14)	N-Red	N-Red
T(15)	N-Cat	N-Cat
P(15)	N-Red	N-Red
T(19C)	N-Red	N-Red
P(18)	N-Red	N-Red
F(16)	N-Cat	Red
X(20,16)	Red	Red
X(22,16)	Red	Red
P(REB)	N-Cat	Red
Total Uncat. Var.	9 (36%)	3 (12%)
Total Cat. Var.	16 (64%)	22 (88%)
Total Time	7001	135

Tabla 4.4. Resultado del análisis de redundancias de las variables medidas para las descomposiciones obtenidas por el MD y el MDE. Las variables $P(A)$, $T(A)$ y $F(A)$ representan la presión, temperatura y caudal de la corriente A respectivamente. La variable $X(B,A)$ representa la composición del componente B en la corriente A.

En la primera columna de la tabla, se incluye la lista de variables medidas para el caso de estudio de la columna de destilación. En la segunda columna, se puede apreciar cuál es la categorización que le asignó el software de redundancia a cada una de las variables medidas. Como se explicó anteriormente, esa categorización se divide en: redundantes (Red), no redundantes (N-Red) y no categorizadas (N-Cat). A su vez, esa columna se subdivide en dos,

donde se puede ver cómo varía la categorización obtenida dependiendo del algoritmo utilizado para realizar el análisis de observabilidad. Finalmente, en las últimas tres filas de la tabla, se incluyen estadísticas globales que indican el grado total de conocimiento alcanzado y el tiempo de ejecución para las descomposiciones surgidas a partir de cada algoritmo de observabilidad.

Los resultados exhibidos en la tabla 4.4 muestran que la aplicación del MDE lleva a una clara mejoría en cuanto al grado de conocimiento alcanzado de las variables medidas. Esto se puede apreciar en el número de variables medidas que resultan no categorizadas al aplicar uno u otro algoritmo de observabilidad. En tal sentido, para la descomposición obtenida con el MD sólo se alcanzó un grado de conocimiento del 64% frente al 88% obtenido empleando el MDE.

Otro parámetro importante para analizar es el tiempo de ejecución empleado por el algoritmo de redundancia para realizar el análisis de ambas descomposiciones. Como se puede apreciar, el tiempo requerido para completar el análisis a partir de la partición obtenida con el MD (7001 segundos) es significativamente mayor que el requerido para el tratamiento de la descomposición hallada por el MDE (135 segundos). Esta notable diferencia entre ambas ejecuciones está relacionada directamente con el número de variables no categorizadas para uno y otro caso. Dada una variable medida, si el algoritmo de redundancia alcanza un límite máximo de iteraciones sin lograr clasificarla, asigna a esa variable la etiqueta de N-Cat, continuando el análisis con una nueva variable. Luego, el tiempo utilizado para realizar el análisis frustrado de la variable N-Cat impacta significativamente en el tiempo total de ejecución del algoritmo. Por ende, al triplicarse el número de variables N-Cat de un caso al otro, el algoritmo de redundancia termina consumiendo mucho más tiempo.

Es importante destacar que las clasificaciones obtenidas para ambos casos resultan consistentes. Es decir, siempre que una variable fue categorizada en ambos análisis, la categoría fue siempre la misma (Red o N-Red). Esto refleja el hecho más general de que la categoría de una variable medida es independiente del método de observabilidad utilizado previo al análisis de redundancia (Mah, 1990; Madron, 1992), la única diferencia entre emplear diferentes métodos está en el número de variables que se lograrán categorizar.

4.7.1.5 RESUMEN DEL IMPACTO DEL MDE EN EL PROCESO GLOBAL DE DISEÑO Y ANÁLISIS DE INSTRUMENTACIÓN

Analizando los resultados obtenidos, se puede comprobar la importancia y el impacto directo que las modificaciones implementadas al algoritmo de observabilidad tienen sobre el desempeño global del DSS para diseño de instrumentación. Las principales mejoras se pueden resumir en los siguientes ítems:

- **Mayor conocimiento del grado de redundancia del proceso:** Al finalizar el proceso de diseño de instrumentación, el usuario tiene la posibilidad de contar con un grado mayor de conocimiento acerca del nivel de redundancia que el conjunto de mediciones impuesto le da al sistema en estudio. Esto resulta fundamental para la toma de decisiones relacionadas con la remoción o la incorporación de nuevos sensores. La remoción de sensores implica quitar mediciones redundantes que pueden resultar innecesarias, disminuyendo así los costos necesarios para monitorear la planta. El agregado de sensores involucra la medición de nuevas variables, de manera tal de lograr que variables críticas del proceso se vuelvan redundantes cuando antes no lo eran.
- **Reducción de tiempos en el proceso global de diseño de instrumentación:** Como se explicó en la Sección 2.7 de esta Tesis, el proceso completo de diseño de instrumentación es una tarea iterativa conformada por varios pasos, lo que obliga al usuario a ejecutar en más de una oportunidad las distintas etapas, requiriéndose mucho tiempo de ejecución y análisis para alcanzar un diseño satisfactorio. El análisis de redundancia, por el tiempo y esfuerzo computacional que requiere para ser llevado a cabo, constituye uno de los cuellos de botella del diseño de instrumentación. La aplicación del algoritmo de observabilidad propuesto en este capítulo permite entonces lograr una reducción significativa en el tiempo necesario para la ejecución del análisis de redundancias de una planta de procesos.
- **Simplificación del proceso de análisis de bloques prohibidos:** en la figura 3.5 de esta tesis se incluye un esquema que describe la arquitectura del MD. En ella se puede ver que el paso 3 del algoritmo consiste en un test de admisión de los bloques generados en la descomposición. Como se presentó en Ponzoni y *co.* (2004), en este paso el algoritmo comprueba automáticamente cada bloque formado a fin de

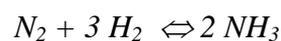
asegurar que no contenga alguno de los subconjuntos prohibidos especificados previamente por el usuario. Estos subconjuntos se encuentran constituidos por ecuaciones linealmente dependientes, como puede ser por ejemplo, un conjunto formado por balances de masa por componente de una corriente más el balance de masa global. Este tipo de análisis se ve simplificado mediante la aplicación del MDE, ya que se cuenta con mayor cantidad de bloques lineales y de mínimo tamaño (1x1), facilitándose así la detección de subsistemas singulares.

4.7.2 PLANTA DE SÍNTESIS DE AMONÍACO

4.7.2.1 DESCRIPCIÓN DEL PROCESO INDUSTRIAL

El segundo ejemplo corresponde a una planta completa de síntesis de amoníaco basada en el proceso Haber-Bosch. Esta planta fue diseñada por *Bike* (1985) para producir 1500 toneladas por día de amoníaco a 240K y 450 KPa. La planta está diseñada para obtener amoníaco con una pureza no menor al 99.5%. Las corrientes de alimentación de hidrógeno y nitrógeno son proveídas por un gasoducto, el cual también contiene inertes como argón y metano.

La cinética de la reacción de síntesis de amoníaco puede ser descrita mediante la siguiente ecuación de equilibrio:



El producto obtenido en el reactor ingresa a una columna de absorción donde el amoníaco es separado mediante la utilización de una corriente de agua. La corriente líquida que abandona la torre es alimentada a una columna de destilación de manera tal de obtener amoníaco puro como producto de tope. Existe también una sección de recuperación compuesta por membranas, donde el hidrógeno que no reaccionó es recuperado y reciclado al reactor. A continuación se incluye un esquema del proceso completo (figura 4.4)

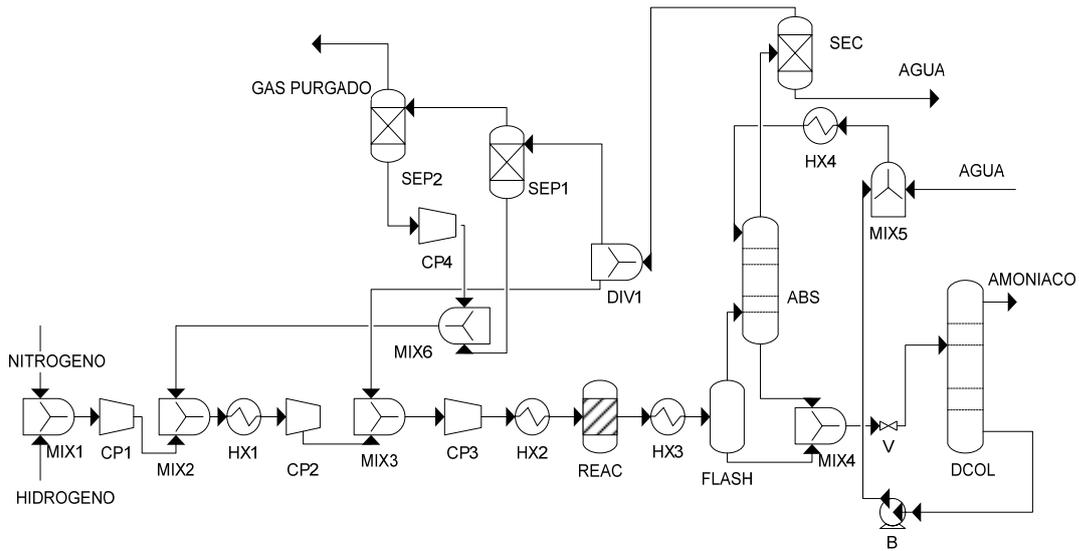


Figura 4.4. Esquema de la planta de síntesis de amoníaco

En este caso, el modelo matemático está formado por 557 ecuaciones y 587 variables, 74 medidas y 513 no medidas.

4.7.2.2 ANÁLISIS DE OBSERVABILIDAD UTILIZANDO EL MD Y EL MDE

A continuación se incluye una tabla (tabla 4.5) con los resultados de observabilidad obtenidos mediante la aplicación del MD y el MDE, para el modelo matemático de la planta de síntesis de amoníaco.

Cantidad de:	MD	MDE	Δ
Variables Observables	216	216	0
Variables No Observables	297	297	0
Subsistemas 1 x 1	144	164	+ 14%
Subsistemas Lineales	143	164	+ 15%
Subsistemas No Lineales	8	4	- 50%

Tabla 4.5. Resultados de observabilidad obtenidos para el modelo de la planta de síntesis de amoníaco aplicando el MD y el MDE.

Al igual que en el caso anterior, en ambos métodos se asigna la misma cantidad de variables observables. Los resultados obtenidos son similares a los del caso anterior. Aquí se evidencia un incremento del 14 % en la cantidad de subsistemas de 1x1, y un aumento del 15 % en la cantidad de subsistemas lineales. También, al igual que en el caso anterior, se reduce en un 50% la cantidad de subsistemas no lineales.

4.7.3 PLANTA DE PRODUCCIÓN DE ETANO

4.7.3.1 DESCRIPCIÓN DEL PROCESO INDUSTRIAL

El último caso de estudio corresponde a una planta real de etano, la cual se alimenta de gas de yacimientos, los cuales contienen C2, C3, C4 y C5+. La planta recibe hasta 36 millones de m³ estándar diarios de gas natural de múltiples pozos con objeto de recuperar la fracción líquida (C2+).

El proceso puede ser dividido en tres secciones principales (figura 4.5): compresión y deshidratación del gas, separación criogénica y fraccionamiento. En primer lugar, el gas alimentado es filtrado y comprimido en tres compresores en paralelo. De manera tal de evitar la formación de hielo e hidratos, el gas es enfriado con agua y deshidratado. Una vez finalizado este proceso, la alimentación es dividida en dos corrientes y cada una de ellas es enviada a un tren criogénico (figura 4.6), el cual está compuesto básicamente por un aeroenfriador, intercambiadores de calor, válvulas Joule-Thompson y una torre demetanizadora. La torre demetanizadora es una columna de destilación de baja temperatura donde se realiza la separación entre metano y etano. El producto de fondo de la columna es enviado a un proceso convencional de separación donde se obtiene etano puro, propano, butano y gasolina natural. El gas residual (producto de tope) es comprimido y comercializado.

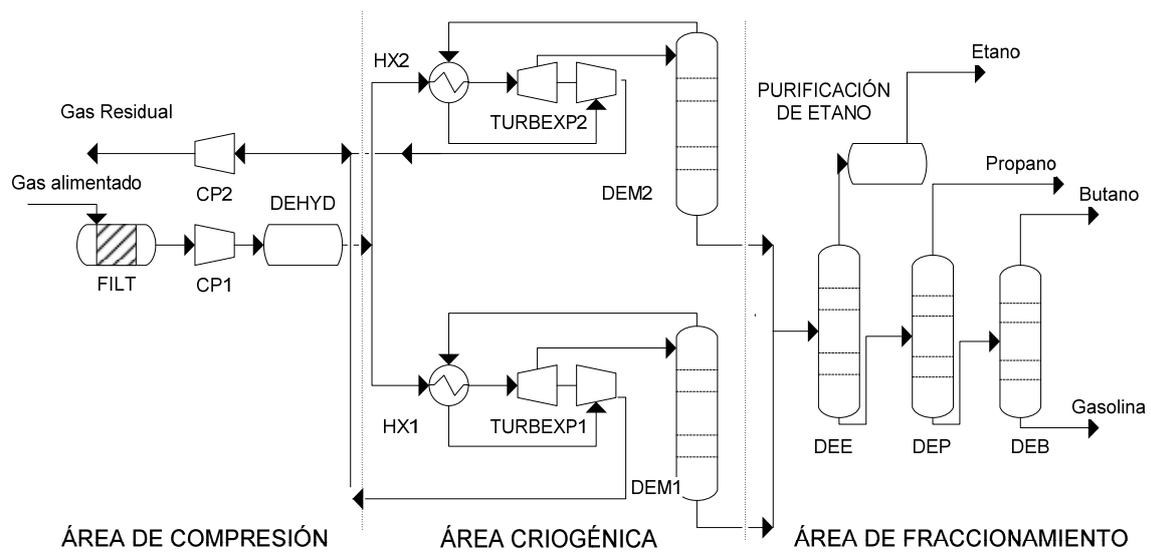


Figura 4.5. Esquema simplificado de la planta de Etano.

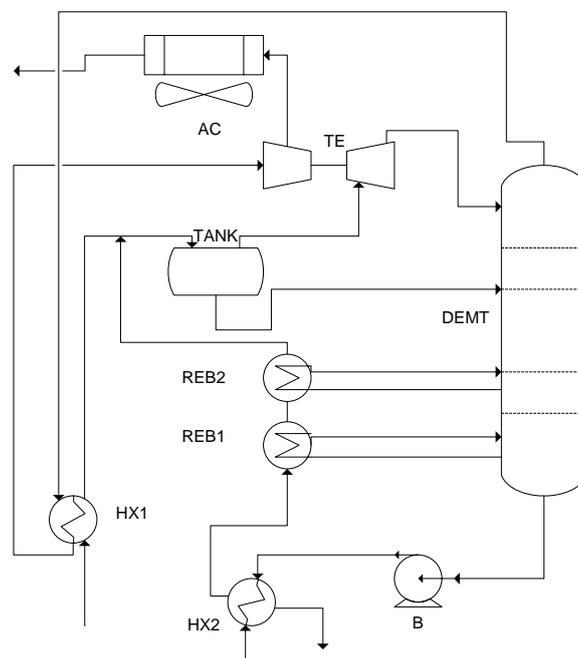


Figura 4.6. Sector criogénico de la planta de Etano.

El modelo matemático está compuesto por 1830 ecuaciones y 1932 variables, 330 medidas y 1602 no medidas.

4.7.3.2 ANÁLISIS DE OBSERVABILIDAD UTILIZANDO EL MD Y EL MDE

A continuación se incluye una tabla (tabla 4.6) con los resultados de observabilidad obtenidos mediante la aplicación del MD y el MDE, para el modelo matemático de la planta de etano.

Cantidad de	MD	MDE	Δ
Variables Observables	929	929	
Variables No Observables	673	673	
Subsistemas 1 x 1	767	777	+ 1.3%
Subsistemas Lineales	758	776	+ 2.3%
Subsistemas No Lineales	26	20	- 23%

Tabla 4.6. Resultados de observabilidad obtenidos para el modelo de la planta de producción de etano aplicando el MD y el MDE.

En este caso se aprecia un incremento del 1.3% en la cantidad de subsistemas de dimensión mínima, mientras que para la cantidad de subsistemas lineales el incremento es del 2.3%. Entretanto, la cantidad de subsistemas no lineales se ve disminuida en un 23%.

Si se realiza un análisis integral de los tres casos, se obtiene un promedio aproximado del 12% en la generación de subsistemas de dimensión 1x1 y en la generación de subsistemas lineales, lo que implica un aumento en la velocidad de resolución de estos sistemas, puesto que el cálculo se reduce a una simple operación aritmética. Por otro lado, y aún más importante, se comprueba la hipótesis planteada al verificar una disminución en promedio del 43% en la generación de subsistemas no lineales.

4.8 CONCLUSIONES

En este capítulo se propone una mejora al algoritmo de observabilidad conocido como Método Directo (MD). Esta nueva variante, llamada Método Directo Extendido (MDE), consiste en la implementación de un módulo que cuantifica la no linealidad inherente a un sistema de ecuaciones. Este módulo, guía al MD tradicional hacia la descomposición del modelo matemático en subsistemas con el menor grado de no linealidad posible.

El nuevo algoritmo fue probado en diferentes casos de estudio. El resultado indica que en todos los casos, el MDE fue capaz de alcanzar un ordenamiento que reduce la no linealidad de los subsistemas obtenidos a través del proceso de análisis de observabilidad. Al mismo

tiempo, el algoritmo es capaz de incrementar la cantidad de bloques generados de mínimo tamaño.

Como fue explicado anteriormente, una de las etapas del diseño de instrumentación es el análisis de redundancia, el cual está basado en la derivación simbólica de las funciones. Como era de esperarse, y fue verificado en la sección 4.7.1.4, los cambios implementados en el algoritmo de particionamiento produjeron mejoras en el proceso de análisis de redundancia. En primer lugar, se logró contar con un grado mayor de conocimiento acerca del nivel de redundancia que el obtenido a partir de la descomposición hallada con el DM original. Esto resulta fundamental para la toma de decisiones relacionadas con la remoción o incorporación de nuevos sensores. En segundo lugar, se logró una reducción de tiempos del proceso global de diseño de instrumentación llevado a cabo mediante la utilización del DSS. Eso se logra gracias a que la aplicación del MDE permite lograr una disminución significativa en el tiempo necesario para la ejecución del análisis de redundancia.

Existen también otros tipos de análisis sobre plantas de procesos que pueden verse favorecidos a futuro con el empleo del MDE. Tal es el caso de la reconciliación de datos. Esta tarea, posterior al diseño de instrumentación, implica comparar datos obtenidos mediante mediciones en campo, con valores estimados por medio de las ecuaciones del modelo, a fin de detectar posibles fallas en los instrumentos de medición. Para que pueda ser llevada a cabo, se requiere únicamente la resolución matemática de los subsistemas de ecuaciones generados mediante el algoritmo de observabilidad, sin utilizar las ecuaciones que no fueron incluidas en estos subsistemas - ecuaciones redundantes o no asignadas. Esto se debe a que una reducción en el grado de no linealidad de la descomposición obtenida, implica resolver mayor cantidad de subsistemas lineales, los cuales pueden ser resueltos por algún método de resolución directa, disminuyendo así los requerimientos computacionales para llevar a cabo la tarea de reconciliación.

Otra tarea que se puede ver significativamente favorecida por la estrategia de particionamiento aplicada por el MDE es la simulación de plantas de procesos. La misma requiere la resolución numérica del modelo matemático que representa el proceso en cuestión. Para sistemas complejos y de grandes dimensiones, la resolución involucra un gran esfuerzo computacional. A su vez, el usuario debe llevar a cabo la ardua tarea de inicializar el total de las variables de un modelo, de manera tal de poder llevar a cabo su resolución mediante algún método numérico iterativo. En ese sentido, el MDE podría ser utilizado como un paso previo

a la simulación de procesos, particionando previamente el modelo completo de la planta. De este modo, la descomposición del modelo en subsistemas proporciona dos ventajas principales: por un lado, reduce los costos computacionales de llevar a cabo la resolución del sistema, y esto es posible porque el usuario tiene la posibilidad de aplicar un método de resolución directa para los sistemas lineales y uno iterativo para los no lineales. Por otro lado, este tratamiento previo permite al usuario visualizar cuáles son los bloques que deberían necesariamente ser inicializados y en qué casos esto no es necesario (bloques pequeños o lineales), simplificando la tarea de inicialización de las variables del modelo.

De este modo, se puede apreciar que el impacto favorable de incorporar el tratamiento del grado de no linealidad al Método Directo puede redundar en beneficios que van más allá de la tarea específica de diseño de instrumentación, lo cual amplía el rango de aplicación de esta metodología de descomposición de sistemas.

4.9 NOMENCLATURA DEL CAPÍTULO

ABS = Absorbedor	HX = Intercambiador de calor
AC= Aeroenfriador	MD= Método Directo
AO= Análisis de observabilidad	MDE= Método Directo Extendido
AR= Análisis de redundancia	MGM= Módulo generador de modelos
B= Bomba	MIX = Mezclador
C = Conjunto de nodos correspondiente a las columnas de la matriz N	N = Matriz de incidencia
COND = Condensador	N-Cat= Variables no categorizadas
CP = Compresor	N-Red= Variables no redundantes
DCOL = Columna de destilación	NLD = Nonlinearity degree
DEB = Debutanizadora	NVR= Grado de no linealidad correspondiente a las ecuaciones
DEE = Deetanizadora	NVC= Grado de no linealidad correspondiente a las variables
DEHYD = Deshidratador	P _M = Pareamiento Maximal
DEMT = Demetanizadora	R = Conjunto de nodos correspondiente a las columnas de la matriz N
DIV = Divisor	REAC = Reactor
FILT = Filtro	
GNL= Grado de no linealidad	

REB = Rebullidor

Red= Variables redundantes

SEP = Separador

SEC= Secador

T = Tanque

t = Número de términos de una ecuación

t_v = Número de términos donde la variable

v forma parte

TE = Turbo expansor

V = Válvula

v = Variable

w = Puntaje de los términos

PARTE 2

ANÁLISIS CUANTITATIVO DE PROCESOS

GENERADOR AUTOMÁTICO DE MODELOS PARA SIMULACIÓN Y OPTIMIZACIÓN DE PROCESOS

5.1 INTRODUCCIÓN

En el capítulo 1 de esta Tesis, se explicó que la misma se encuentra dividida principalmente en dos partes. La primera de ellas (capítulos 2, 3 y 4), comprende temas que se pueden agrupar bajo el concepto de “análisis cualitativo de procesos” (ver sección 1.2). A partir de este capítulo, comienza la segunda sección de la Tesis (capítulos 5 y 6), la cual está dedicada a lo que hemos denominado “análisis cuantitativo” de procesos. Esta segunda sección incluye herramientas desarrolladas para facilitar la generación y resolución matemática de modelos matemáticos que representan el comportamiento de un proceso en particular.

Como se verá en este capítulo, el modelo matemático que representa el comportamiento de una planta o proceso químico, puede ser simulado u optimizado, dependiendo de las características del mismo. Las herramientas de modelado comúnmente reportadas en la literatura intentan asistir a usuarios con diversas necesidades. Marquardt (1994) presentó un trabajo de revisión acerca del modelado asistido por computadora. Más tarde, Bogusch y Marquardt (1997) resaltaron la importancia de llevar a cabo un adecuado modelado estructurado, desarrollando VeDa, un modelo de datos orientado a objetos. De acuerdo a Vazquez-Roman y *co* (1996), las técnicas de modelado pueden ser clasificadas en dos categorías. La primera de ellas corresponde a paquetes que contienen una librería de modelos para la descripción de unidades individuales, como son SpeedUp (Perkins y Sargent,

1982) o gPROMS (gPROMS home page, 2009, Pantelides y Barton, 1992). La segunda categoría incluye generadores basados en descripciones físicas del proceso, por medio de las leyes fundamentales. Siguiendo esta metodología, Vazquez-Roman y *co* (1996) desarrollaron KBMoSS, un sistema de soporte de modelado con una interfaz flexible implementada a través de la programación orientada a objetos, mientras que Dietrich y Eigenberg (1995) desarrollaron Bimap, el cual emplea computación simbólica para manipular un set de ecuaciones básico, una vez que el usuario ha definido las suposiciones y simplificaciones deseadas. Es importante destacar que una buena interfaz resulta una parte indispensable de un generador de modelos, sobre todo cuando se trata de modelos complejos y de grandes dimensiones (Berrais, 1998).

A pesar de que los principios que gobiernan los diferentes procesos son relativamente semejantes en todos ellos (se basan principalmente en balances de masa y de energía), la demanda computacional es diferente para cada aplicación específica a la que se quiera asignar el generador de modelos (diseño de instrumentación, reconciliación de datos, simulación, optimización, etc). Entre las diferentes necesidades con las que el usuario puede enfrentarse se encuentran, entre otras, el diferente grado de detalle en el modelo matemático a obtener, las facilidades en la incorporación de nuevos equipos si el software pretende ser flexible a diversas aplicaciones y la adecuada sintaxis del sistema de ecuaciones generado, el cual dependerá de los algoritmos de resolución utilizados una vez obtenido el modelo matemático.

En este capítulo se presenta un software para la generación automática de modelos matemáticos de plantas de procesos químicos. Este software, permite al usuario modelar cualquier planta, en el sentido de obtener el modelo matemático que representa el proceso en cuestión. Dichos modelos matemáticos pueden ser luego utilizados para simular u optimizar el funcionamiento de esta planta química, resolviendo el sistema de ecuaciones obtenido, mediante la utilización de un simulador orientado a ecuaciones. De esta manera, se puede disponer del sistema completo de ecuaciones que representa el proceso en cuestión, sin tener la necesidad de ingresar manualmente uno a uno los modelos de los equipos que conforman la planta. El programa se encarga de generar el modelo completo de la planta a partir del ingreso, por parte del usuario, de la topología de la misma. Esto resulta de suma utilidad ya que, como se describe en este capítulo, el usuario puede combinar las ventajas de un simulador orientado a objetos con las que posee la metodología orientada a ecuaciones. Una vez obtenido automáticamente el sistema completo de ecuaciones que representa el modelo en

cuestión, el usuario podrá llevar a cabo su resolución matemática. Para esto, es posible utilizar otra de las herramientas desarrolladas en el marco de esta Tesis y presentada en el capítulo 6, denominada *MP4SO* (por sus siglas en inglés: Partición de Modelos para Simulación y Optimización).

Dado que los programas comerciales existentes deben ser tratados como “cajas negras”, pues generalmente es imposible tener acceso al código fuente de modo de poder adaptar el mismo a las necesidades de un usuario particular, fue necesario desarrollar un software completamente nuevo que cumpla con los requisitos establecidos previo a su desarrollo. De esta manera, se obtuvo una herramienta que posee una interfaz amigable, con facilidades para la incorporación de nuevos equipos por parte del usuario, y que resulta capaz de interactuar de manera directa con el *MP4SO* (ver capítulo 6), generando el modelo matemático con la sintaxis exacta como para ejecutar dicho programa sobre el modelo obtenido y luego simular u optimizar el mismo.

Este capítulo incluye, en primer lugar, una descripción teórica de lo que se conoce como simulación de procesos (sección 5.2), dando detalles de las etapas que conforman una simulación así como también los diferentes enfoques que pueden ser utilizados para llevar a cabo la simulación de un proceso industrial. Luego se describe brevemente el concepto de optimización de procesos (sección 5.3), mencionando los elementos que conforman una optimización, así como también los tipos de problema que se pueden presentar y las metodologías existentes en la literatura para llevar a cabo la resolución de los mismos. La sección 5.4 presenta en detalle el generador de modelos desarrollado, detallando sus principales características. También se presentan las diferencias existentes entre un modelo matemático utilizado para diseño de instrumentación y uno que debe ser utilizado para ser simulado u optimizado. En la sección 5.5 se describe la incorporación del modelo complejo de una columna reactiva a nuestro generador de modelos (Olivera y *co.*, 2006), el cual luego de ser automáticamente generado, fue simulado y optimizado (Domancich y *co.*, 2007; Domancich y *co.*, 2009a). Finalmente, en la sección 5.6 se incluyen las conclusiones respecto de los temas desarrollados en este capítulo.

5.2 SIMULACIÓN DE PROCESOS

La simulación de un determinado proceso permite predecir qué comportamiento tendrá el mismo bajo ciertas condiciones dadas. Para un determinado conjunto de datos de

entrada y parámetros fijos de un equipo en particular, la simulación permite obtener las condiciones de salida que ese proceso tendrá. Básicamente consiste en resolver un sistema de ecuaciones generado a partir del modelado matemático del equipo, teniendo ese modelo igual número de ecuaciones que de variables, es decir, conformando un sistema cuadrado sin grados de libertad.

5.2.1 ETAPAS EN EL DESARROLLO DE UN MODELO PARA SIMULACIÓN

Biegler (1989) define las siguientes etapas como las más comunes a seguir para realizar la simulación de una planta de procesos químicos:

- **Definición de la estructura del flowsheet:** En esta etapa es donde se comienza a formar el ambiente de simulación por medio de bloques que representan las unidades de operación y corrientes que representan flujos de materia y energía entre dichas unidades.
- **Desarrollo de modelos para cada unidad de proceso:** Como se explicó anteriormente, un modelo define el sistema de ecuaciones que será utilizado para calcular los balances de masa y energía para cada unidad de proceso.
- **Elección de un sistema de unidades consistentes:** Para comodidad del usuario se elige el sistema de unidades que mejor se ajuste a los datos que manejará el problema.
- **Especificación de los componentes (especies químicas) presentes en las corrientes de procesos:** Los simuladores poseen bancos de datos de propiedades de los diferentes compuestos, así como también paquetes para cálculo de propiedades termodinámicas y físicas.
- **Selección de modelos para cálculo de propiedades físicas:** Es necesario elegir algún modelo termodinámico adecuado para predecir propiedades de mezclas y fenómenos tales como entalpías molares de mezclas, equilibrio líquido-vapor, volumen molar de la mezcla, etc.
- **Satisfacción de los grados de libertad de un flowsheet:** Una vez definido el modelo se deben fijar los grados de libertad del sistema. Algunos ejemplos de variables típicas que pueden ser fijadas para satisfacer los grados de libertad son:

variables de corrientes de alimentación (flujos molares, composiciones, temperaturas y presiones), procesos químicos (grados de avance, conversión y coeficientes estequiométricos), especificaciones de diseño (caudales molares de productos y especificaciones de pureza) y parámetros de equipos (caídas de presión, eficiencia de los equipos, diámetro de los platos de una columna).

- **Selección de un método numérico para la resolución:** Dado que el modelo matemático resulta generalmente no lineal, se debe recurrir a algún método numérico para llevar a cabo su resolución. Para una correcta resolución se debe tener en cuenta la robustez y rapidez con que opera cada uno de los métodos.
- **Comprobación de resultados:** Esta etapa es fundamental, ya que la resolución numérica puede arrojar resultados sin sentido físico. Es importante comprobar los resultados obtenidos en la simulación y compararlos, de ser posible, con datos reales del proceso.

5.2.2 CLASIFICACIÓN DE LOS SIMULADORES DE PROCESOS

Existen principalmente dos clases de simuladores de procesos químicos: secuencial modular y orientado a ecuaciones. A continuación se incluye una explicación de cada tipo de simulador con sus ventajas y desventajas.

5.2.2.1 ENFOQUE SECUENCIAL MODULAR

El enfoque secuencial modular es una aproximación para resolver problemas de simulación siguiendo la estructura del flowsheet del proceso. Esta estrategia sugiere un particionamiento y un orden de precedencia, resolviendo entonces el problema en partes, de a un equipo por vez hasta completar el proceso. Es básicamente la tecnología seleccionada para la simulación de procesos en estado estacionario. Las ventajas y desventajas que presenta esta estrategia son:

Ventajas

- El programa ejecutivo resulta muy simple, ya que los cálculos se realizan equipo a equipo y existe un módulo de cálculo para cada equipo que se ejecuta independientemente de los demás módulos, una vez que se conocen las variables de entrada a cada uno de ellos.

- Bajos requisitos de memoria de computadora, dado que se resuelve cada uno de los equipos de manera individual.
- Cuentan con una completa librería que modela gran cantidad de operaciones unitarias, por lo que casi cualquier planta de procesos puede ser simulada con algún software basado en este enfoque, salvo que incluya algún equipo especializado que no se encuentre disponible en la librería.
- Pueden aplicarse diferentes estrategias de resolución numérica para cada uno de los equipos existentes según sea la conveniencia.
- La estrategia de resolución que sigue este enfoque es intuitiva para el usuario, ya que se resuelve equipo por equipo en el orden en que el flowsheet fue creado.
- Los mensajes de error son más fáciles de corregir, ya que en general se informan para cada equipo de manera específica.

Desventajas

- Tiene dificultad para converger en procesos altamente integrados, es decir, con gran cantidad de corrientes de recicló.
- No se tiene acceso a los modelos matemáticos de los equipos, y puede resultar un trabajo considerable el agregado de modelos programados por el usuario (ver apéndice B).
- Resulta dificultoso y de gran costo computacional realizar cálculos de diseño, donde la entrada del equipo no es dato sino una variable a determinar.
- Pueden converger problemas mal especificados, por ejemplo, especificaciones redundantes.
- Debido a la metodología de cálculo que emplea, es un enfoque difícil de extender a cálculos de tipo dinámico y de optimización.

5.2.2.2 ENFOQUE ORIENTADO A ECUACIONES

El enfoque orientado a ecuaciones, también conocido como simultáneo o basado en ecuaciones, es el principal competidor de la estrategia secuencial modular. Es más fácil describir esta estrategia que llevarla a cabo en una implementación de software: en lugar de resolver los equipos de manera modular, de a uno por vez, un simulador orientado a ecuaciones agrupa todas las ecuaciones que representan el modelo matemático de la planta completa para resolverlas de manera simultánea. Más específicamente, la estrategia orientada a ecuaciones está conformada por los siguientes pasos:

- Tanto las ecuaciones como las variables del modelo se encuentran definidas de manera individual para cada uno de los equipos de la planta. Es decir, el modelo se descompone de manera modular al igual que en la estrategia anterior, sin ser aprovechado esto a la hora de llevar a cabo la resolución del sistema de ecuaciones.
- Estas ecuaciones y variables son luego ensambladas de manera tal de formar un gran sistema no lineal de ecuaciones.
- Se incorporan al modelo especificaciones adicionales de manera tal de llevar a cero los grados de libertad del problema, obteniendo así un problema cuadrado de solución única.
- El sistema se resuelve de manera simultánea por medio de la utilización de alguna rutina de cálculo para sistemas no lineales.

A continuación se detallan las ventajas y desventajas que esta estrategia presenta:

Ventajas

- Esta estrategia resulta mucho más eficiente que la secuencial modular para flowsheets con reciclos.
- Resulta muy eficiente para problemas de diseño, donde se deben determinar parámetros de equipos que en una simulación serían datos. Para resolver este tipo de problema de manera secuencial es necesario suponer los parámetros tomando diferentes valores hasta que se cumplan las condiciones de salida que el problema real tiene. En el enfoque simultáneo esto no es necesario, ya que el sistema de ecuaciones sigue siendo el mismo, por lo que la resolución no cambia si se trata de simulación de un proceso o de diseño.

- Debido a que se trabaja directamente con ecuaciones y variables y no con modelos ocultos, resulta sencillo el agregado de nuevos modelos a la simulación, o la modificación de los modelos existentes.
- Es fácilmente extensible a otro tipo de cálculos como son el estado dinámico y la optimización, los cuales pueden ser llevados a cabo por diferentes rutinas de cálculo.
- Es posible aplicar estrategias de particionamiento que agilicen la resolución de estos grandes sistemas de ecuaciones, solucionando así con una de las grandes desventajas de esta estrategia. Como se verá en el capítulo 6, en el marco de esta Tesis se amplió el rango de aplicación del Método Directo (Ponzoni y *co.*, 2004) de manera tal de ser utilizado como estrategia de preprocesamiento, descomponiendo el modelo matemático en bloques de resolución más sencilla.
- Es más sencillo de realizar el diagnóstico para cierto tipo de errores en la formulación del problema. Por ejemplo, si existe un problema en la especificación del modelo, un simulador orientado a ecuaciones tiene la posibilidad de verificar el sistema completo de ecuaciones en búsqueda del mismo. Un ejemplo sería la singularidad del modelo.

Desventajas

- Generalmente los métodos numéricos desarrollados para sistemas no lineales no son tan confiables y robustos para sistemas de gran tamaño, lo cual no sucede en el enfoque secuencial. Más aún, para grandes sistemas de ecuaciones es necesario conocer previamente una aproximación a la solución del sistema, de manera tal de brindar adecuados puntos iniciales al método numérico para que este converja hacia la solución.
- El enfoque necesita de mayores recursos computacionales para llevar a cabo los cálculos de los grandes sistemas de ecuaciones.
- Tanto los modelos matemáticos como las rutinas para cálculo de propiedades termodinámicas deben ser ingresados por el usuario, resultando esto una tarea compleja y que debe ser llevada a cabo con sumo cuidado, de manera tal de ingresar en forma correcta todos los modelos matemáticos.

5.3 OPTIMIZACIÓN DE PROCESOS

Llevar a cabo la resolución de un problema de optimización implica encontrar el valor que deben tomar las variables que forman parte del modelo matemático, para hacer óptima una determinada función. A diferencia de los problemas de simulación, los problemas de optimización cuentan con un número mayor de variables que de ecuaciones, las cuales son fijadas mediante el cumplimiento de ciertos requerimientos para encontrar el óptimo buscado. Los problemas de optimización se componen básicamente de tres elementos:

- **Función objetivo:** Es la medida cuantitativa del funcionamiento del modelo que se desea optimizar (maximizar o minimizar). Como ejemplo de funciones objetivo se pueden mencionar: la minimización de costos operativos de una planta, la maximización de los beneficios netos de venta, la minimización de la materia prima utilizada en la fabricación de un producto, etc.
- **Variables:** Representan las decisiones que se pueden tomar para afectar el valor de la función objetivo. Desde un punto de vista funcional se pueden clasificar en variables *independientes o de control* y variables *dependientes o de estado*, aunque matemáticamente todas son iguales.
- **Restricciones:** Representan el conjunto de relaciones (expresadas mediante ecuaciones e inecuaciones) que ciertas variables está obligadas a satisfacer.

Existen algunos problemas de optimización que se desvían ligeramente del esquema presentado anteriormente. Ellos son:

- **Sistemas de ecuaciones lineales/no lineales:** No existe una función objetivo como tal. Únicamente interesa encontrar una solución factible a un problema con un conjunto dado de restricciones.
- **Optimización sin restricciones:** Se trata de encontrar el conjunto de valores de las variables que determinan el mínimo/máximo de una función objetivo, sin tener que cumplir con restricciones impuestas.
- **Optimización multiobjetivo:** Se trata de problemas con más de una función objetivo. El problema que se plantea es cómo trabajar con varias funciones objetivo a la vez. Teniendo en cuenta que el óptimo para un objetivo no lo es para otro, son

objetivos en conflicto entre sí, que no tienen solución única sino una solución de compromiso, generalmente elegida a partir de una función Pareto.

5.3.1 MÉTODOS DE OPTIMIZACIÓN

A continuación se incluye la clasificación de los métodos utilizados para llevar a cabo la optimización de un determinado modelo matemático:

- **Métodos determinísticos:** Son los métodos que habitualmente se pueden encontrar en los libros de optimización (Hillier y Lieberman, 1997; Edgar y Himmelblau, 1988). Entre ellos se encuentran la optimización lineal, lineal mixta entera, no lineal, estocástica, dinámica, etc.
- **Métodos metaheurísticos:** Dentro de este grupo se encuentran los algoritmos evolutivos genéticos y las búsquedas heurísticas (método tabú, búsqueda aleatoria, etc).

De manera general, se puede afirmar que los métodos clásicos buscan y garantizan la obtención de un óptimo local mientras que los métodos metaheurísticos tienen mecanismos específicos para alcanzar un óptimo global aunque no garantizan que se pueda alcanzar.

5.4 GENERADOR DE MODELOS PARA SIMULACIÓN Y OPTIMIZACIÓN DE PROCESOS INDUSTRIALES

A partir del desarrollo del MGM para diseño de instrumentación (Vazquez y *co.* 2001), se planteó en el marco de esta Tesis desarrollar un software con la misma filosofía del MGM, es decir, la de generar el modelo matemático de cualquier planta, pero orientado a la simulación y optimización de procesos.

5.4.1 OBJETIVOS DE DISEÑO DEL SOFTWARE Y CARACTERÍSTICAS PRINCIPALES DEL MISMO

En cuanto a los objetivos específicos de diseño del software generador de modelos, los mismos se definieron de manera tal que el programa contara con:

- Visualización completa de la planta de procesos, de manera tal que resulte natural y sencillo el ingreso al programa de una nueva planta.

- Utilización de programación orientada a objetos, cuya filosofía permite una representación natural del tipo de información que se maneja en el armado de una planta de procesos (cada equipo representa un objeto independiente).
- Modelado matemático riguroso y preciso de plantas reales así como también de las propiedades físicas y químicas de las corrientes que integran el proceso.
- Adecuada sintaxis en el modelo generado, de manera tal que el mismo pueda ser ingresado a un directamente simulador orientado a ecuaciones.
- Flexibilidad en las opciones de modelado, de manera tal de facilitar la incorporación de nuevos modelos de equipos y ecuaciones termodinámicas.
- Chequeos de consistencia y validación de datos.
- Herramientas inteligentes que permitan facilitar el ingreso de la topología de la planta, así como también realizar chequeos en corrientes y equipos.
- Capacidad para el tratamiento de plantas de dimensión industrial
- Guía para desarrolladores, la cual detalle la metodología a seguir para incorporar nuevos equipos o paquetes termodinámicos, así como también modificar los modelos ya existentes.

En base al cumplimiento de estos objetivos, se desarrolló entonces un software que posee una interfaz amigable y permite al usuario ingresar la topología completa de una planta química de manera secuencial, seleccionando de un menú los equipos que forman parte de la misma (Domancich y *co*, 2009e). El programa tiene incorporada una etapa de verificación, la cual se encarga de verificar de manera automática que el flowsheet ingresado no contenga inconsistencias en los compuestos de las corrientes que ingresan o salen de los equipos, así como también en las fases de las mismas.

Una vez llevadas a cabo exitosamente las comprobaciones mencionadas anteriormente, el programa genera automáticamente el modelo matemático completo que representa el proceso en cuestión, en un formato compatible con el simulador orientado a ecuaciones GAMS. De esta manera, dicho modelo pueda ser simulado u optimizado dependiendo de las necesidades del usuario. El programa informa al usuario las dimensiones

del modelo de manera tal que el mismo pueda analizar si el modelo resultante es cuadrado o si posee grados de libertad que permitan llevar a cabo la optimización del mismo.

El enfoque utilizado para llevar a cabo la programación del paquete se denomina orientado a objetos. El mismo permite darle la suficiente flexibilidad al programa como para que el usuario pueda incorporar cualquier tipo de equipo con mínimo esfuerzo. Para ello, debe seguirse una determinada sintaxis desarrollada en Lenguaje Extensible de Marcas (XML) (Raik, E, 2001; Harold, E. y Means. W., 2004), la cual se explica en detalle en el anexo C de esta Tesis. Cada uno de los equipos ingresados por el usuario, así como también los modelos termodinámicos, son tratados como bloques independientes, los cuales una vez programados son incorporados de manera directa al módulo principal, resultando así disponibles para su utilización en cualquier caso que quiera ingresarse.

Es importante destacar que la sintaxis con que se generan los modelos matemáticos puede ser modificada fácilmente, de manera tal de adaptar los modelos a los diferentes programas de resolución que quieran ser utilizados. En nuestro caso en particular se eligió generar una salida totalmente compatible con GAMS de manera tal de poder aprovechar la gran cantidad de algoritmos de resolución con los que cuenta dicho software. De esta manera, el usuario genera de manera externa y mediante la utilización de nuestro software el modelo de la planta en cuestión, pudiendo introducir luego el mismo a GAMS para realizar la simulación u optimización del mismo

5.4.2 MODELADO RIGUROSO PARA SIMULACIÓN Y OPTIMIZACIÓN

Para llevar a cabo el desarrollo del nuevo software, uno de los objetivos fundamentales a cumplir fue el de incorporar modelos matemáticos completos y rigurosos al software. Es importante destacar que para el desarrollo del MGM esto no resultó necesario, ya que el diseño de instrumentación es una tarea cualitativa, en la cual se realiza una clasificación de las variables y ecuaciones (ver sección 2.2 de esta Tesis) sin realizar cálculos con las mismas. Es por ese motivo que el MGM no cuenta con este tipo de modelos matemáticos. Se realizó entonces el modelado riguroso de varios de los equipos disponibles en cualquier planta química, los cuales fueron incorporados al software de manera tal de tener un conjunto completo de equipos que puedan ser seleccionados por el usuario para realizar la simulación de cualquier planta de procesos. En cuanto a las propiedades físicas, químicas y termodinámicas, se realizó un modelado ideal de las mismas, de manera tal de sentar las bases

como para que el usuario pueda incorporar los modelos necesarios de acuerdo al tipo de problema a resolver. A continuación se presenta un ejemplo de sistema de ecuaciones generado para un análisis de observabilidad (figura 5.1), el cual fue generado por el MGM y uno generado por el nuevo software generador de modelos (figura 5.2),

```

ComprBMG0.. F_S1-F_S2 =e= 0;
ComprBMC0.. F_S1*y_S1_H20-F_S2*y_S2_H20 =e= 0;
ComprBMC1.. F_S1*y_S1_H3N-F_S2*y_S2_H3N =e= 0;
ComprBMC2.. F_S1*y_S1_N2-F_S2*y_S2_N2 =e= 0;
ComprBE0.. F_S1*h_S1-F_S2*h_S2+Q_Compresor =e= 0;
ComprPRES.. P_S1+deltaP_Compresor-P_S2 =e= 0;
ComprPOT0.. Q_Cmpr-f(deltaP_Cmpr,T_S1,F_S1,Ef_Cmpr) =e= 0;
ComprTsal.. T_S2-f(deltaP_Cmpr,T_S1,Ef_Cmpr) =e= 0;
S1DENS0.. Rho_S1-f(P_S1,T_S1,y_S1_H20)-f(P_S1,T_S1,y_S1_H3N)-f(P_S1,T_S1,y_S1_N2) =e= 0
S1ENTP0.. h_S1-f(P_S1,T_S1,y_S1_H20)-f(P_S1,T_S1,y_S1_H3N)-f(P_S1,T_S1,y_S1_N2) =e= 0;
S1NORMY0.. y_S1_H20+y_S1_H3N+y_S1_N2-1 =e= 0;
S2NORMY0.. y_S2_H20+y_S2_H3N+y_S2_N2-1 =e= 0;
S2DENS0.. Rho_S2-f(P_S2,T_S2,y_S2_H20)-f(P_S2,T_S2,y_S2_H3N)-f(P_S2,T_S2,y_S2_N2) =e= 0
S2ENTP0.. h_S2-f(P_S2,T_S2,y_S2_H20)-f(P_S2,T_S2,y_S2_H3N)-f(P_S2,T_S2,y_S2_N2) =e= 0;

```

Figura. 5.1. Ejemplo de sistema de ecuaciones generado por el MGM para llevar a cabo el análisis de observabilidad de una planta de procesos

```

ComprBMG0.. F_S1-F_S2 =e= 0;
ComprBMC0.. F_S1*y_S1_H20-F_S2*y_S2_H20 =e= 0;
ComprBMC1.. F_S1*y_S1_H3N-F_S2*y_S2_H3N =e= 0;
ComprBMC2.. F_S1*y_S1_N2-F_S2*y_S2_N2 =e= 0;
ComprBE0.. F_S1*h_S1-F_S2*h_S2+Q_Compresor =e= 0;
ComprPRES.. P_S1+deltaP_Compresor-P_S2 =e= 0;
ComprWKO.. Q_Cmpr-((Z_Cmpr*8.31*T_S1)/MW_S1*k_Cmpr/(k_Cmpr-1))*((P_S2/P_S1)**((k_Cmpr-1)/k_Cmpr-1))/Ef_Cmpr =e=0;
ComprTsal.. T_S2-T_S1*(P_S2/P_S1)**((n_Cmpr-1)/n_Cmpr) =e= 0;
ComprEf.. Ef_Cmpr-((k_Cmpr-1)/k_Cmpr)/((n_Cmpr-1)/n_Cmpr) =e= 0;
S1DENS0.. Rho_S1-P_S1*MW_S1/(0.082*T_S1) =e= 0;
S1ENTP0.. hv_S1_H20-hf_H20-cpa_H20*(T_S1-298)-cpb_H20/2*(T_S1**2-298**2)-cpc_H20/3*(T_S1**3-298**3)-cpd_S1_H20/4*(T_S1**4-298**4) =e= 0;
S1ENTP1.. hv_S1_H3N-hf_H3N-cpa_H3N*(T_S1-298)-cpb_H3N/2*(T_S1**2-298**2)-cpc_H3N/3*(T_S1**3-298**3)-cpd_S1_H3N/4*(T_S1**4-298**4) =e= 0;
S1ENTP2.. hv_S1_N2-hf_N2-cpa_N2*(T_S1-298)-cpb_N2/2*(T_S1**2-298**2)-cpc_N2/3*(T_S1**3-298**3)-cpd_S1_N2/4*(T_S1**4-298**4) =e= 0;
S1ENTP0.. hv_S1-hv_S1_H20*y_S1_H20-hv_S1_H3N*y_S1_H3N-hv_S1_N2*y_S1_N2 =e= 0;
S1MW0.. MW_S1-MW_H20*y_S1_H20-MW_H3N*y_S1_H3N-MW_N2*y_S1_N2 =e= 0;
S1NORMY0.. y_S1_H20+y_S1_H3N+y_S1_N2-1 =e= 0;
S2NORMY0.. y_S2_H20+y_S2_H3N+y_S2_N2-1 =e= 0;
S2MW0.. MW_S2-MW_H20*y_S2_H20-MW_H3N*y_S2_H3N-MW_N2*y_S2_N2 =e= 0;
S2DENS0.. Rho_S2-P_S2*MW_S2/(0.082*T_S2) =e= 0;
S2ENTP0.. hv_S2_H3N-hf_H3N-cpa_H3N*(T_S2-298)-cpb_H3N/2*(T_S2**2-298**2)-cpc_H3N/3*(T_S2**3-298**3)-cpd_S2_H3N/4*(T_S2**4-298**4) =e= 0;
S2ENTP1.. hv_S2_N2-hf_N2-cpa_N2*(T_S2-298)-cpb_N2/2*(T_S2**2-298**2)-cpc_N2/3*(T_S2**3-298**3)-cpd_S2_N2/4*(T_S2**4-298**4) =e= 0;
S2ENTP2.. hv_S2_H20-hf_H20-cpa_H20*(T_S2-298)-cpb_H20/2*(T_S2**2-298**2)-cpc_H20/3*(T_S2**3-298**3)-cpd_S2_H20/4*(T_S2**4-298**4) =e= 0;
S2ENTP0.. hv_S2-hv_S2_H20*y_S2_H20-hv_S2_H3N*y_S2_H3N-hv_S2_N2*y_S2_N2 =e= 0;

```

Figura. 5.2. Ejemplo de sistema de ecuaciones generado por el nuevo generador de modelos para simulación y optimización de procesos

Los ejemplos mostrados anteriormente, corresponden al modelo matemático de un compresor de simple etapa, al cual ingresa una determinada corriente S1 y sale de él una corriente S2. Las ecuaciones resaltadas corresponden al modelado del compresor en sí mismo, así como también de las propiedades de la corriente de entrada S1 (entalpía y densidad). Como se puede apreciar en el primero de los casos, tanto el modelado del compresor como el de las propiedades de la corriente se llevan a cabo con la utilización de una funcionalidad que incluye las variables de las cuales cada propiedad o variable de operación del equipo dependen. Esto resulta suficiente para llevar a cabo el diseño de instrumentación de una planta, donde el objetivo es implementar una configuración adecuada de instrumentos de control que permita utilizar el mínimo número de sensores necesarios para asegurar el conocimiento necesario del comportamiento del proceso. Por otro lado, en la figura 5.2 se muestra cómo deben reformularse esas ecuaciones si se desea simular u optimizar el proceso en cuestión, reemplazando las funcionalidades por su forma matemática rigurosa.

5.4.3 DESCRIPCIÓN DEL SOFTWARE

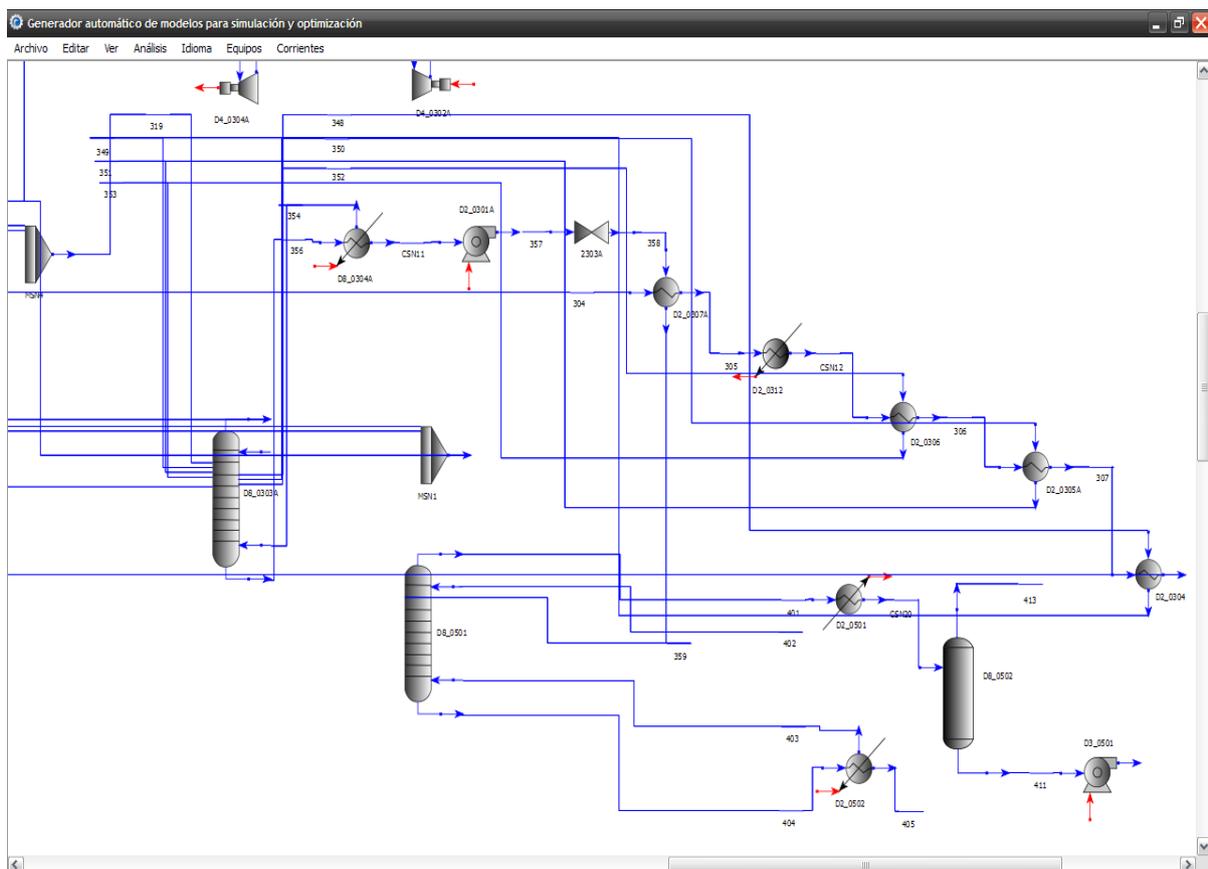


Figura 5.3. Captura de pantalla del área principal del software para un caso de estudio

A continuación se describen los principales componentes del software. La figura 5.3 muestra una captura de pantalla del mismo, donde se visualiza un caso de estudio incorporado.

Área principal

Se trata de la parte del programa destinada al ingreso de la topología del proceso a estudiar. Sobre ella el usuario dibuja el flowsheet del proceso del cual se generará el modelo matemático.

Archivos

Permite generar un nuevo archivo y guardar los cambios generados en él, así como también abrir modelos previamente desarrollados con el software. También desde este menú es posible imprimir el flowsheet que haya sido ingresado.

Editar

Este menú permite modificar en todo momento los compuestos que forman parte del modelo, así como también seleccionar la fuente con que son nombrados los equipos en el flowsheet.

Ver

Permite al usuario incrementar o reducir el tamaño del flowsheet.

Idioma

Brinda la opción de utilizar el software en castellano o inglés.

Análisis

Este menú se encarga de llevar a cabo el análisis de consistencias del modelo, así como también de generar el modelo matemático final que representa el proceso en cuestión. El procedimiento por el cual se lleva a cabo está dividido en dos etapas:

- **Comprobación de consistencias:** Una vez ingresada la topología completa de la planta, se lleva a cabo esta primera etapa, la cual se encarga de chequear que no existan inconsistencias entre los equipos y las corrientes que ingresan o abandonan los mismos. Se verifica que las fases de las corrientes sean las

correctas de acuerdo al equipo a las que están conectadas, así como también se verifica la correcta definición de los compuestos que forman parte de ellas.

- **Generación del modelo matemático:** Una vez realizados con éxito los chequeos respectivos, se genera el modelo matemático completo que representa el comportamiento de la planta en cuestión. El modelo luego puede ser guardado en formato HTML (Lenguaje de Marcas de Hipertexto) de manera tal de poder ser manipulado con mayor facilidad.

Equipos

Contiene la librería de equipos de la que dispone el usuario para construir la topología de la planta. Una vez seleccionado y ubicado un equipo en el área principal, el usuario debe ingresar dentro de él de manera tal de especificar el mismo de manera completa. Al ingresar al equipo se abre una ventana del mismo la cual permite al usuario definir el nombre, así como también sus conexiones y parámetros conocidos. Contiene los siguientes ítems:

- **General:** Ventana reservada para que el usuario ingrese el nombre del equipo, así como también cualquier nota que desee referida al mismo. La figura 5.4 muestra la ventana “General” para una bomba centrífuga.

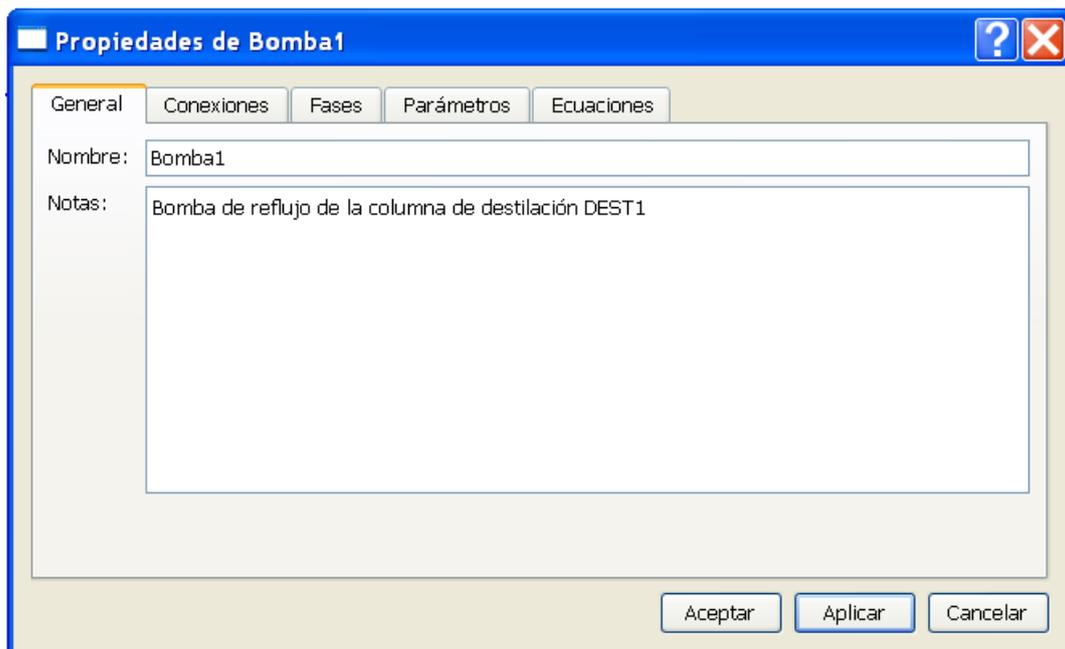


Figura 5.4. Ventana “General” del equipo para definición de nombre del mismo y notas.

- **Conexiones:** En esta ventana se definen las corrientes de materia que ingresan y egresan del equipo (ver figura 5.5).

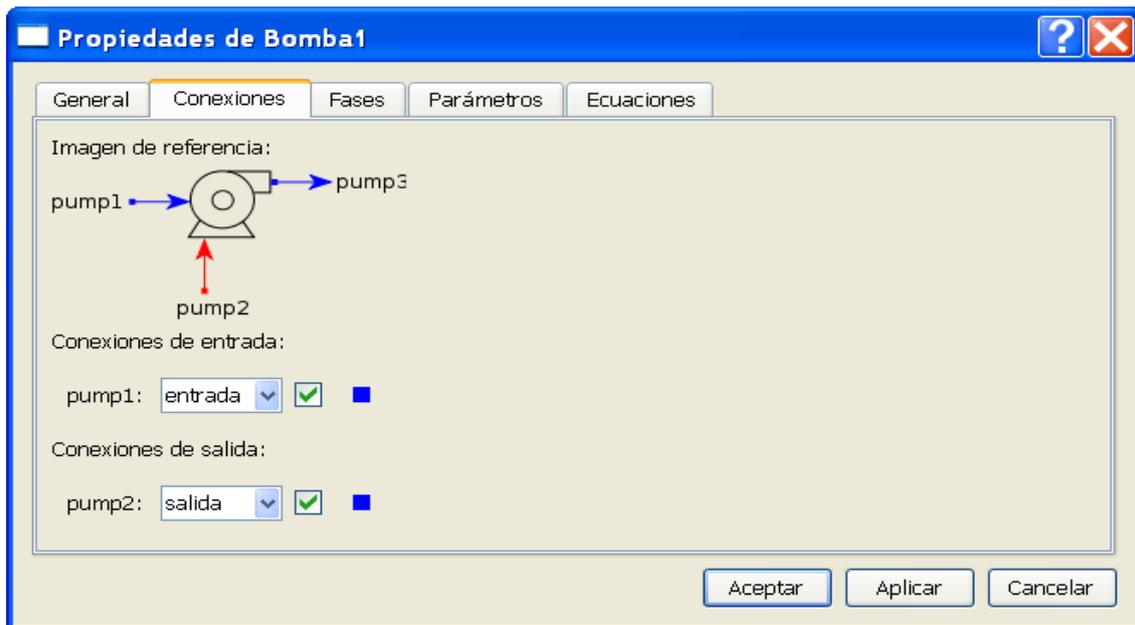


Figura 5.5. Definición de las corrientes de entrada y salida del equipo.

- **Fases:** Informa, de acuerdo al equipo con que se está trabajando, en qué fase se deben encontrar las corrientes que sean conectadas al mismo (ver figura 5.6).

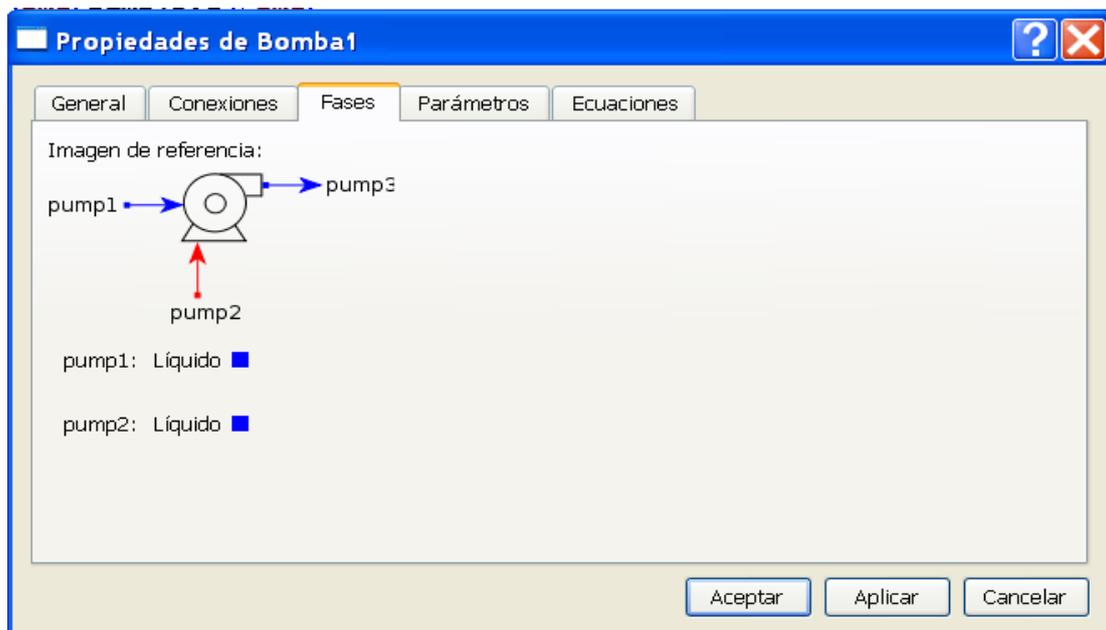


Figura 5.6. Pantalla de información sobre las fases que permite el equipo.

- **Parámetros:** Desde esta ventana se ingresan al programa los valores de los parámetros característicos del equipo (ver figura 5.7), como por ejemplo: caída de presión en el mismo, potencia consumida, eficiencia, etc. Dichos parámetros deben ser ingresados si son valores conocidos, ya que si no se ingresa número alguno (siendo 0.00 el valor por defecto) se está indicando al programa que dichos parámetros son en realidad variables cuyo valor será luego calculado mediante la simulación u optimización del modelo matemático.

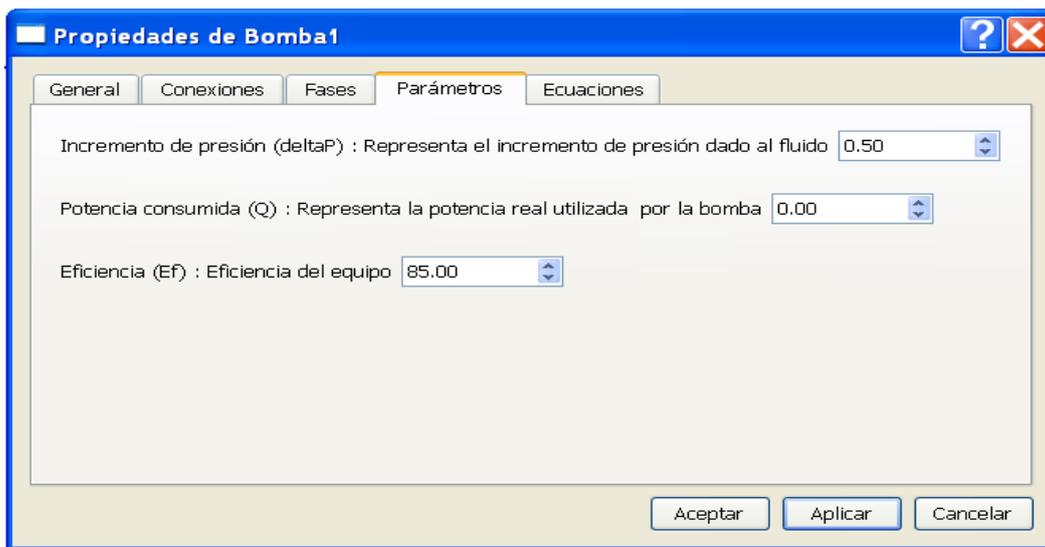


Figura 5.7. Definición de los parámetros del equipo.

- **Ecuaciones:** Muestra la lista total de ecuaciones del equipo (figura 5.8). Por defecto se encuentra seleccionado un determinado grupo de ecuaciones, teniendo el usuario la posibilidad de seleccionar las que desee de acuerdo al nivel de detalle con que se quiera modelar el equipo.

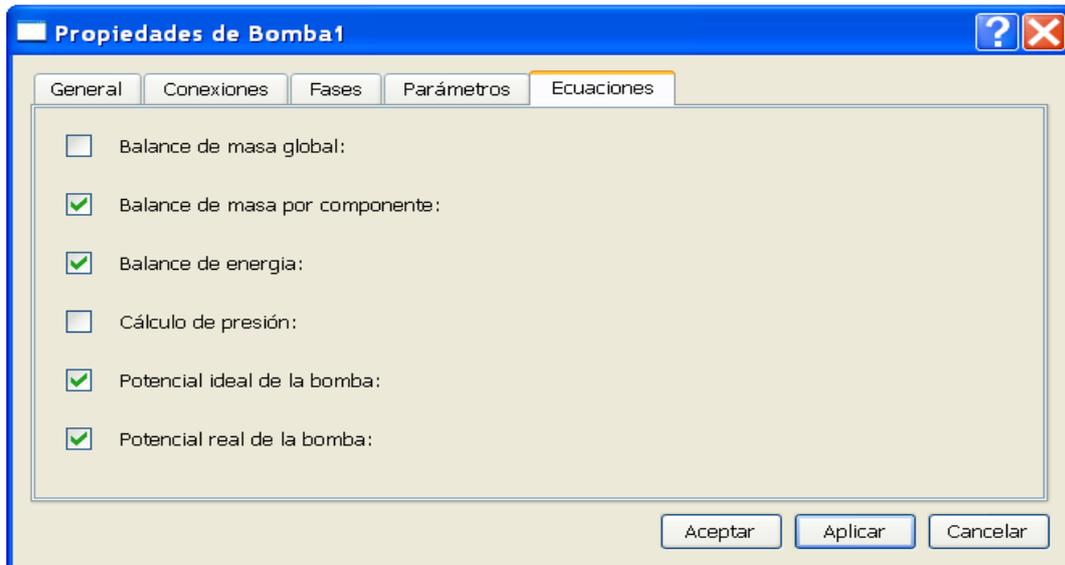


Figura 5.8. Definición de las ecuaciones que forman parte del modelo matemático del equipo.

Corrientes

Contiene las corrientes materiales que utiliza el usuario para realizar las conexiones entre equipos. Una vez ubicada la corriente en el área principal, el usuario debe ingresar dentro de ella de manera tal de completar las especificaciones necesarias de la misma. Al ingresar a la corriente se abre una ventana que contiene los siguientes elementos:

- **General:** Ventana reservada para que el usuario ingrese el nombre de la corriente, así como también cualquier nota que desee referida a la misma. También se debe definir la fase en que se encuentra la misma, es decir, líquido, vapor o mezcla. La figura 5.9 muestra la ventana “General” para una determinada corriente.

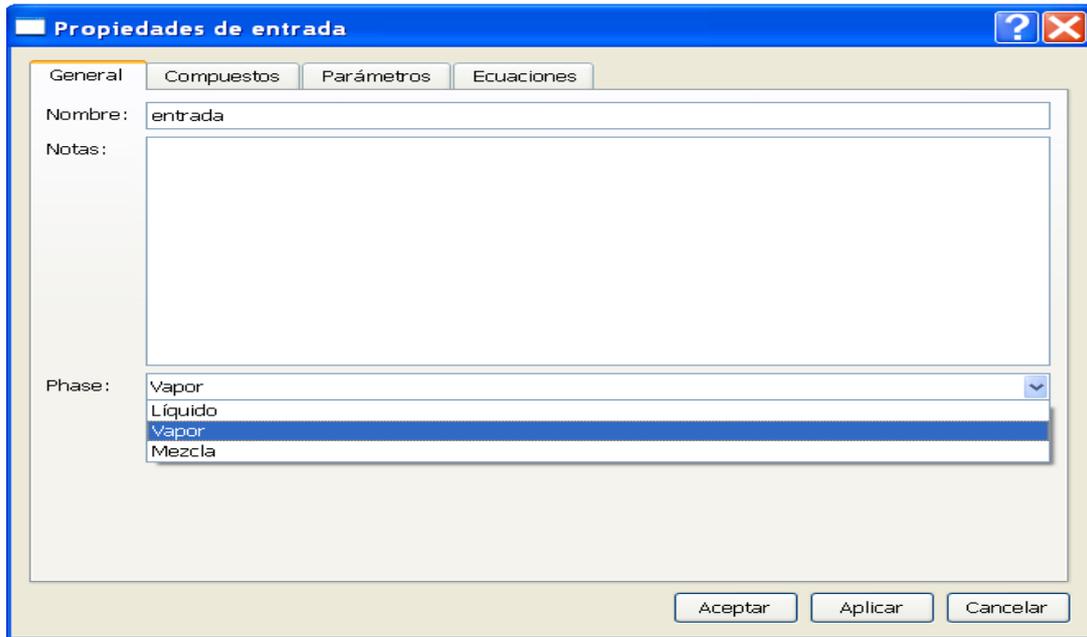


Figura 5.9. Definición de las propiedades de la corriente. Pantalla “General”.

- Compuestos:** En esta ventana el usuario define los compuestos que forman parte de la corriente que está siendo definida. Para ello se debe seleccionar de la lista total de compuestos que forman parte de la simulación (definida al comenzar un nuevo caso de estudio) aquellos que integran dicha corriente. La figura 5.10 muestra la ventana donde se realiza la selección de compuestos.

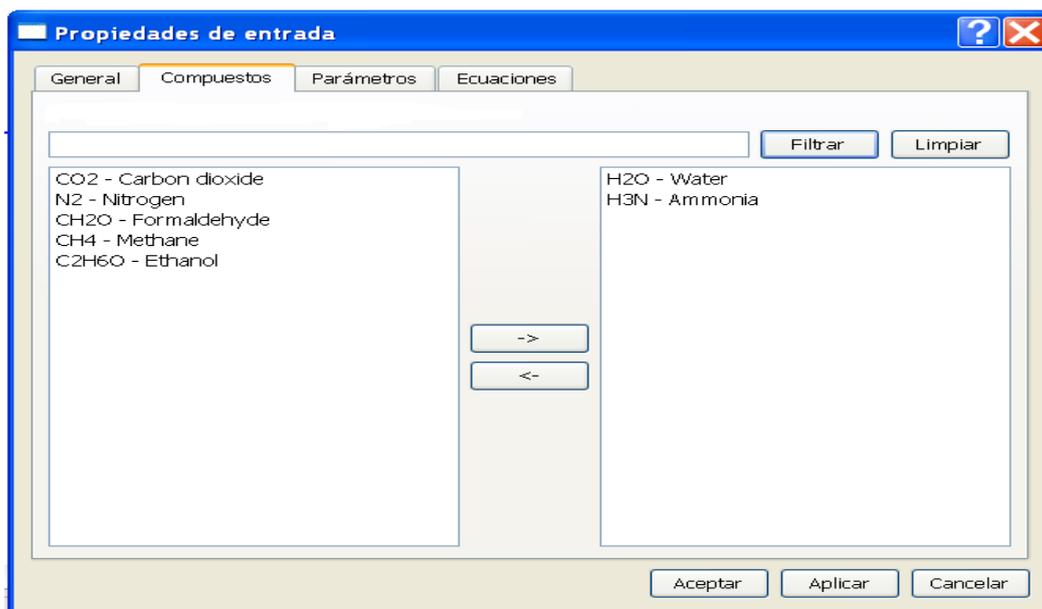


Figura 5.10. Definición de los compuestos que integran la corriente.

- **Parámetros:** Esta ventana contiene una lista de las variables medibles que forman parte de la corriente, es decir: caudal, composición, temperatura y presión (ver figura 5.11). El usuario debe ingresar los valores de las variables que sean conocidas, y dejar vacíos los campos de aquellas que quedarán como incógnitas y serán calculadas con la simulación u optimización del modelo.

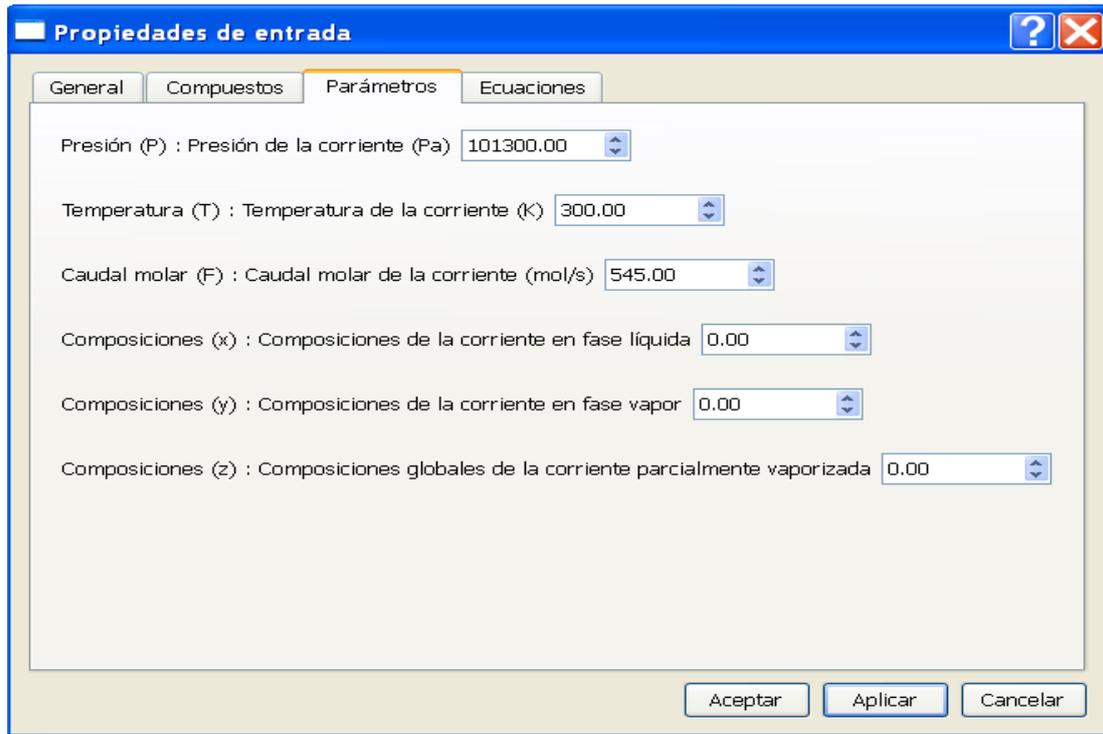


Figura 5.11. Definición de las variables medidas de la corriente.

- **Ecuaciones:** Muestra la lista total de ecuaciones que modelan las propiedades físicas, químicas y de equilibrio de la corriente en cuestión (ver figura 5.12). La ventana posee una función denominada “Tildar ecuaciones” la cual, una vez definida la fase de la corriente, selecciona de manera automática solamente las ecuaciones que correspondan a la fase de la corriente, no incluyendo en el modelo aquellas que no debieran formar parte del mismo.

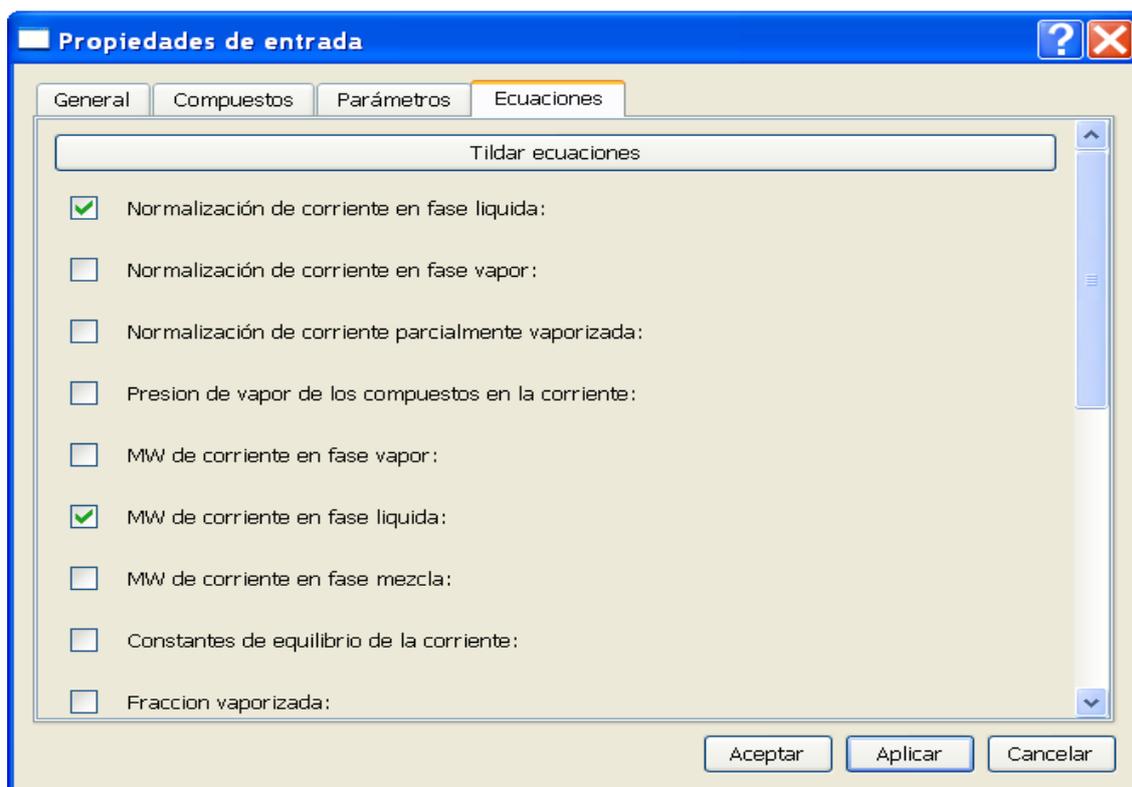


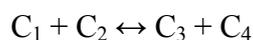
Figura 5.12. Selección de las ecuaciones que modelan las propiedades físicas y químicas de la corriente.

5.4.4 CASO DE ESTUDIO: COLUMNA DE DESTILACIÓN REACTIVA

Para sacar provecho del software desarrollado, y realizar las pruebas necesarias del mismo, se decidió incorporar al generador de modelos matemáticos una columna de destilación reactiva (Olivera y co., 2006), la cual fue posteriormente simulada y optimizada mediante la utilización de GAMS. El modelo matemático fue obtenido a partir del programa anteriormente descrito. A continuación se presenta una breve descripción del proceso en cuestión.

5.4.4.1 DESCRIPCIÓN DEL PROCESO DE DESTILACIÓN REACTIVA

La destilación reactiva es una operación híbrida que combina dos procesos claves en ingeniería química: la reacción y la separación (Almeyda y co., 2005). Dado el siguiente esquema de reacción reversible:



donde los puntos de ebullición siguen la secuencia C_1 , C_3 , C_4 y C_2 , un flowsheet tradicional para llevar a cabo el proceso constaría de un reactor o secuencia de reactores seguido por un tren de

procesos de separación y reacción combinados son mayores que si se tratan por separado. El acoplado de sistemas no lineales conformados por reacciones químicas, equilibrio de fases y transporte de masa y energía puede derivar en efectos no deseados, como por ejemplo la aparición de azeótropos reactivos y multiplicidad de estados estacionarios (Hauan y *co.*, 1995).

Dejando de lado el modelado matemático, la combinación de reacción y separación en un mismo equipo ofrece ciertas ventajas que el proceso convencional no posee (Podrebarac y *co.*, 1998). Como ejemplos de las ventajas de esta integración, se pueden mencionar:

- Se logra superar las limitaciones que impone el equilibrio químico.
- Se alcanzan mayores selectividades.
- El calor de reacción puede ser utilizado en el proceso de destilación.
- Se puede evitar el uso de solventes auxiliares.
- Las mezclas azeotrópicas pueden ser separadas con mayor facilidad.
- Se logra una enorme reducción de costos de inversión y operación. (Domancich y *co.*, 2007).

Algunos procesos industriales donde se utiliza el concepto de destilación reactiva son:

- Síntesis de Metil-acetato (Agreda y *co.*, 1990).
- Preparación de éteres como MTBE (Metil-Tert-Butil Éter) y ETBE (Etil-Tert-Butil Éter) (Jacobs and Krishna, 1993), utilizados como adhesivos para gasolinas.

En Taylor y Krishna (2000) se puede encontrar una completa revisión acerca de los aspectos fundamentales del modelado de columnas de destilación reactiva. Allí se pone especial énfasis en la diferencia entre modelos basados en el equilibrio y aquellos basados en el no-equilibrio, con sus respectivas ventajas y desventajas. Doherty y Malone (2001) brindan una serie de comentarios valiosos acerca de la tendencia futura y desafíos que propone el proceso, y finalmente, en Almeida-Rivera y *co.* (2004) se puede encontrar una revisión de los métodos de diseño para destilación reactiva.

5.4.4.2 SIMULACIÓN Y OPTIMIZACIÓN DE LA COLUMNA REACTIVA

El modelo de la columna fue incorporado al generador automático de modelos y posteriormente testeado mediante la simulación y optimización del mismo. Para ello se utilizó como caso de estudio la producción de MTBE a partir de metanol e isobuteno. En el anexo D de esta Tesis se presentan diversos resultados obtenidos a partir de este modelo, para determinar su fiabilidad y posibilidad de realización de numerosos cálculos. Si bien dichos cálculos no están directamente relacionados con el objetivo de esta Tesis, muestran la flexibilidad de las representaciones matemáticas de equipos obtenidas mediante el generador de modelos.

Tanto los balances de masa y energía del equipo como las relaciones de equilibrio utilizadas se describen en el anexo D. También se incluyen en dicho anexo todas las ecuaciones para los cálculos geométricos y económicos utilizadas en la optimización del equipo.

5.5 CONCLUSIONES

En este capítulo se presentó un software para la generación automática de modelos matemáticos de plantas de procesos. El mismo permite obtener el sistema de ecuaciones que representa el funcionamiento de una planta de procesos a partir del ingreso al programa de la topología de la misma. Una vez obtenido el sistema de ecuaciones, se puede llevar a cabo la resolución del mismo por medio de alguno de los algoritmos disponibles para ello, teniendo en cuenta las características del sistema obtenido.

El software posee una interfaz amigable que permite al usuario ingresar la topología completa de una planta química de manera secuencial, seleccionando de un menú los equipos que forman parte de la misma. El programa tiene incorporada una etapa de verificación, la cual se encarga de chequear de manera automática que el flowsheet ingresado no contenga inconsistencias en los compuestos de las corrientes que ingresan o salen de los equipos, así como también en las fases de las mismas.

Una vez realizados los chequeos automáticos, se genera el modelo matemático completo que representa el proceso en cuestión, en un formato compatible con GAMS, de manera tal que dicho modelo pueda ser simulado u optimizado dependiendo de las necesidades del usuario. Se eligió en particular la sintaxis de GAMS poder aprovechar la gran cantidad de algoritmos de resolución con los que cuenta dicho software. Es importante

destacar que no resulta dificultoso modificar dicha sintaxis de manera tal de adaptarlo a cualquier otro programa o algoritmo de resolución que quiera ser utilizado.

El generador de modelos fue desarrollado utilizando un enfoque de programación orientado a objetos. De esta manera resulta flexible a la incorporación de nuevos equipos o paquetes termodinámicos de manera relativamente sencilla, siguiendo una metodología desarrollada en lenguaje XML (ver apéndice C). Para realizar los tests necesarios del programa, se incorporó al mismo el modelo de una columna de destilación reactiva, el cual fue posteriormente simulado y optimizado mediante la utilización de GAMS.

Mediante la utilización de este generador de modelos, el usuario obtiene de manera externa al programa resolvidor el modelo de la planta, pudiendo introducir luego el mismo a GAMS para realizar la simulación u optimización del mismo, sin tener necesidad de conocer la sintaxis específica de dicho programa. Por otro lado, como se verá a continuación en el capítulo 6, otra de las herramientas desarrolladas en el marco de esta Tesis consiste en un software, denominado *MP4SO* para llevar a cabo la simulación y optimización de procesos industriales, basado en la estrategia orientada a ecuaciones. Por este motivo, resultó de suma utilidad el desarrollo del generador automático de modelos como paso previo a la posterior aplicación del *MP4SO* para llevar a cabo la resolución del sistema de ecuaciones generado.

5.6 NOMENCLATURA DEL CAPÍTULO

CTA= Costo total anualizado

MGM= Módulo generador de modelos

MTBE= Metil-tert butil éter

PUR= Pureza

***MP4SO:
UN SOFTWARE
DE PARTICIÓN DE MODELOS
PARA SIMULACIÓN Y OPTIMIZACIÓN
DE PROCESOS INDUSTRIALES***

6.1 INTRODUCCIÓN

La complejidad inherente a los modelos matemáticos empleados para la simulación y optimización de procesos industriales dificulta las tareas de análisis y resolución computacional de los mismos. Una estrategia empleada habitualmente para reducir la complejidad de sistemas algebraicos es su descomposición en subsistemas más simples de ecuaciones (Duff y *co.*, 1986). Sin embargo, la mayoría de estas metodologías está orientada a la resolución de sistemas cuadrados o lineales, con lo cual se limita su rango de aplicación. En particular, la simulación de procesos industriales requiere la resolución matemática de sistemas cuadrados de ecuaciones (sistemas no singulares), donde el modelo contiene igual número de ecuaciones que de variables, siendo única la solución que satisface el modelo planteado (Perkins, 1983). Por otro lado, los modelos de optimización son rectangulares, siendo el número de variables superior al de ecuaciones, dando lugar a múltiples soluciones posibles. Es importante destacar que generalmente, en ambos casos, se está en presencia de modelos matemáticos fuertemente no lineales. Estas son las razones por las que si uno quisiera realizar una descomposición de este tipo de modelos, debería contar con algoritmos que permitan trabajar con modelos cuadrados o rectangulares, y de carácter lineal o no lineal.

En el capítulo 4 de esta Tesis se describe en detalle un nuevo método de descomposición estructural de sistemas de ecuaciones para análisis de observabilidad que posee características que superan las limitaciones presentadas por otros algoritmos de descomposición (ver capítulo 3). Este algoritmo, conocido como Método Directo Extendido (MDE) (Domancich y *co.*, 2009b), se encuentra basado en teoría de grafos y efectúa un reordenamiento estructural de la matriz de incidencia correspondiente al modelo de estado estacionario de un proceso. Esta reestructuración conduce a una forma triangular inferior en bloques, la cual define implícitamente una descomposición en bloques y el orden de precedencia entre ellos, para la resolución eficiente del sistema de ecuaciones asociado a dicha matriz. A diferencia de otros métodos de particionamiento de matrices (Pothen y Fan, 1990; Gustavson, 1976; Madron, 1992, Duff y Reid, 1978), el MDE permite descomponer tanto matrices estructuralmente singulares como no singulares, correspondientes tanto a modelos matemáticos lineales como no lineales. De este modo, mediante la aplicación de este algoritmo de particionamiento, es posible llevar a cabo el tratamiento de casos de estudio de gran dimensión, tal como los que generalmente surgen en ingeniería de procesos.

Por otra parte, como se mencionó en la presente Tesis, existe una herramienta comercial orientada a ecuaciones para modelado, simulación y optimización de procesos, conocida como GAMS (por sus siglas en inglés: General Algebraic Modeling System, Sistema General de Modelado Algebraico) (Brooke y *co.*, 2004). Este software cuenta con gran variedad de *solvers* matemáticos y brinda facilidades para resolver sistemas en bloques, aunque no posee una técnica integrada para efectuar la descomposición en subsistemas. Por ende, resultaría valioso contar con un software que integre las potencialidades del Método Directo Extendido con el software comercial GAMS.

En este capítulo se presenta un software, denominado *MP4SO* (por sus siglas en inglés: Modeling Partitioning for Simulation and Optimization, Particionamiento de Modelos para Simulación y Optimización) (Domancich y *co.*, 2009) para simulación y optimización de procesos industriales, basado en la estrategia orientada a ecuaciones. *MP4SO* es una herramienta que trabaja de manera interoperativa con GAMS y basa su funcionamiento en el Método Directo Extendido. Su aplicación a problemas de grandes dimensiones, como son normalmente los procesos industriales, permite reducir las desventajas que naturalmente tiene la estrategia orientada a ecuaciones. Por un lado, la partición de un sistema complejo en subsistemas logra disminuir los recursos computacionales necesarios para resolverlo. Por otra

parte, la subdivisión del modelo facilita el análisis del proceso por parte del usuario. Por ejemplo, permite identificar de manera más efectiva las variables que deberían ser inicializadas, en estudio de simulación u optimización, a partir de los bloques encontrados, y sus correspondientes órdenes de precedencia para el cálculo. Esta característica resulta de gran utilidad para modelos de grandes dimensiones, ya que se ve facilitado el arduo trabajo de inicialización mediante prueba y error de todas las variables del modelo.

En la próxima sección se realiza una breve descripción del Método Directo Extendido, tratado en detalle en el capítulo 4 de esta Tesis así como también su integración con GAMS. La sección 6.3 describe los componentes principales de *MP4SO*. En la sección 6.4 se presenta el análisis de performance del software aplicado a diferentes casos de estudio, así como también una discusión de los resultados obtenidos. Finalmente, en la sección 6.5 se incluyen las conclusiones generales.

6.2 UTILIZACIÓN DEL MDE Y GAMS PARA SIMULACIÓN Y OPTIMIZACIÓN DE PROCESOS INDUSTRIALES

En esta sección se incluye una breve descripción del Método Directo Extendido, y se detalla cómo fue diseñada la integración con GAMS a fin de usar las potencialidades de ambos para conformar *MP4SO*.

6.2.1 DESCOMPOSICIÓN DE MODELOS MATEMÁTICOS MEDIANTE LA UTILIZACIÓN DEL MÉTODO DIRECTO EXTENDIDO

Como se describió en el capítulo 4 de esta Tesis, el Método Directo Extendido (Domancich y *co.*, 2009b) efectúa un reordenamiento estructural de la matriz de incidencia correspondiente al modelo de estado estacionario de un proceso, llevándola a una forma triangular inferior en bloques. De este modo se obtiene una descomposición en subsistemas y un orden de precedencia para la resolución eficiente del mismo. El MDE dio excelentes resultados al ser aplicado en la resolución de problemas de monitoreo de procesos reales. La técnica resulta sumamente eficiente en términos de tiempos de ejecución, así como también aplicable sobre cualquier tipo de matrices, sin importar su patrón estructural, incrementando su eficacia a medida que los problemas crecen en tamaño y complejidad. Por estos motivos, es posible afirmar que la técnica de descomposición llevada a cabo por el Método Directo Extendido constituye una base sólida para el desarrollo de una metodología para la resolución de sistemas de ecuaciones no lineales de gran tamaño. La descomposición en subsistemas y

posterior orden de resolución que se lleva a cabo simplifica la resolución del sistema original de ecuaciones. Más aún, como se verá en este capítulo, es también aplicable a sistemas rectangulares.

Como se detalló anteriormente, el Método Directo Extendido detecta dos tipos diferentes de variables, dependiendo de la factibilidad de cálculo de las mismas:

- **Variables Observables:** variables no medidas que pueden obtenerse a partir de las variables medidas utilizando las ecuaciones del modelo.
- **Variables No Observables:** variables no medidas que no pueden calcularse a partir de las variables medidas ni a partir de las ecuaciones del modelo.

Trasladando esta clasificación del análisis de observabilidad a la resolución matemática de modelos, se puede afirmar que las variables observables son aquellas calculadas a partir de los bloques cuadrados detectados por el algoritmo (simulación), mientras que las variables no observables pertenecen a bloques rectangulares con mayor número de variables que ecuaciones (optimización).

6.2.2 SISTEMA GENERAL DE MODELADO ALGEBRAICO (GAMS)

GAMS (Brooke y *co.*, 2004) es un software de uso general que permite el modelado, análisis y resolución de diversos problemas de simulación y optimización. Dado su nivel de abstracción y flexibilidad de uso, es utilizado tanto en ingeniería de procesos como en otras áreas de la ingeniería y ciencias exactas. En su sitio Web (GAMS, página Web, 2009) se encuentra una completa serie de documentos que ayudan al usuario en las etapas de modelado, corrección de errores, análisis de sensibilidad, generación de reportes de salida, etc. Entre sus características principales se encuentran:

- Su capacidad para pasar de resolver problemas de pequeñas dimensiones (docenas de variables y restricciones) a problemas mayores (miles de variables y restricciones) sin variar significativamente el código fuente. El manejo eficiente de subíndices permite escribir de manera compacta restricciones similares mediante una sola línea de comando.
- Separa el proceso de modelado del proceso de resolución del problema. De esta manera, el usuario de GAMS debe ser capaz de conseguir una formulación

consistente del problema, y una vez la expresada en la notación correcta, este lenguaje hace uso de alguno de los *solvers* disponibles para obtener su solución. De esta manera, el usuario sólo ha de centrarse en obtener un modelo del problema y puede ignorar el funcionamiento interno del algoritmo que se necesita para resolverlo. La separación de estas dos tareas permite cambiar el modelo para mejorarlo o completarlo de manera más cómoda. El conocimiento de los métodos empleados para la solución de los modelos es importante para poder plantear adecuadamente los mismos, pero aunque es deseable, no es imprescindible en una primera etapa.

- La forma en que GAMS representa un problema de optimización prácticamente coincide con la descripción matemática de ese problema. Por lo tanto, el código GAMS es sencillo de comprender para aquellos lectores familiarizados con el modelado de problemas de simulación y optimización.
- GAMS proporciona los mecanismos necesarios para resolver problemas de optimización con estructuras similares, como son aquellos que se derivan de las técnicas de descomposición. Esta característica resultó fundamental a la hora de seleccionar el software de resolución que forme parte de *MP4SO*.

6.2.3 RESOLUCIÓN EN BLOQUES LLEVADA A CABO MEDIANTE LA UTILIZACIÓN DEL MÉTODO DIRECTO Y GAMS

MP4SO permite resolver problemas de simulación y optimización. El programa basa su funcionamiento en la aplicación del Método Directo Extendido, seguido por la resolución utilizando alguno de los *solvers* provistos por GAMS, dependiendo del tipo de problema a resolver. En un primer paso el usuario alimenta el modelo matemático de interés al programa, usando la notación de GAMS. Dicho modelo es particionado utilizando el MDE, obteniéndose así una descomposición en subsistemas junto con el orden de resolución de los mismos. En una segunda etapa, *MP4SO* invoca internamente a GAMS, de forma transparente para el usuario, con la descomposición previamente obtenida. Esta invocación puede ser efectuada para simular el proceso en estudio, en el caso de sistemas cuadrados, u optimizar, en el caso de sistemas rectangulares. Finalmente, la resolución de los distintos subsistemas es llevada a cabo por los *solvers* de GAMS siguiendo el orden de resolución indicado por *MP4SO*.

Como se explicó anteriormente, las diferencias desde el punto de vista matemático entre un problema de simulación y uno de optimización radican en la estructura del sistema de ecuaciones obtenido en uno u otro caso. La simulación de procesos implica llevar a cabo la resolución de un sistema cuadrado de ecuaciones con solución única. Desde el punto de vista del análisis estructural efectuado por el algoritmo de observabilidad, el mismo da como resultado una serie de subsistemas cuadrados conformados por las variables del modelo, las cuales son clasificadas en su totalidad como variables observables. La simulación de dicho modelo implica entonces llevar a cabo la resolución bloque a bloque de todos esos subsistemas, llegando así a la solución final.

Por otro lado la optimización de un modelo, al tratarse de un problema con más variables que ecuaciones, arroja un análisis estructural diferente. Al analizar la observabilidad en estos casos, el resultado obtenido es una serie de bloques cuadrados conformados por las denominadas variables observables (independientes de las variables de optimización), seguidos por un bloque rectangular, que se encuentra integrado por variables observables pertenecientes a bloques anteriores, pero también por variables no observables. Esto da como resultado un bloque, generalmente de mayor tamaño que los bloques cuadrados, con múltiples soluciones posibles. La optimización de un modelo será entonces llevada a cabo en dos etapas. En una primera etapa se resuelven los bloques cuadrados que no dependen de las variables de optimización, los cuales poseen solución única. En una segunda etapa se lleva a cabo la resolución del bloque rectangular, fijando el optimizador las variables de manera tal de minimizar la función objetivo del problema en cuestión.

Como se me mencionó anteriormente, existen diversas técnicas para permutar matrices. En el capítulo 4 de esta Tesis se presentó la descomposición de Dulmage-Mendelsohn, la cual es considerada una de las técnicas más conocidas para permutar una matriz general a una forma triangular en bloques. En dicho capítulo se explicó, mediante la presentación de esquemas comparativos, cómo la descomposición alcanzada mediante la aplicación del MDE resulta más conveniente para diseño de instrumentación que la de Dulmage-Mendelsohn (ver figura 3.6). Si ambos métodos son aplicados a la resolución de sistemas de ecuaciones, también se pueden encontrar ventajas en la aplicación del MDE. La forma en que el MDE descompone la matriz de incidencia permite obtener en primer término todos los bloques cuadrados posibles (bloques (SC1, SR1) y (SC2, SR2)), relegando al final de la descomposición los bloques rectangulares. Esto, para llevar a cabo la optimización de un

modelo (sistema de ecuaciones rectangular), resulta de suma importancia, ya que como se verá a continuación, permite llevar a cabo la resolución de dichos bloques cuadrados en una primera instancia, dado que las variables que los conforman no tienen dependencia alguna con el resto de las variables del modelo.

Desde el punto de vista de la simulación de un proceso (sistemas cuadrados), se puede afirmar que mientras la descomposición de Dulmage-Mendelsohn no es aplicable a este tipo de sistemas, el MDE puede ser utilizado, y su costo de empleo es, en términos de orden de complejidad algorítmica, igual al método de Tarjan, el cual resulta ser el más eficiente de los métodos desarrollados para resolver sistemas cuadrados.

6.3 MP4SO: UN SOFTWARE PARA LA INTEGRACIÓN DEL MÉTODO DIRECTO Y GAMS

Como se explicó anteriormente, *MP4SO* surgió a partir de la idea de ampliar el campo de aplicación del Método Directo Extendido, de manera tal de sacar provecho de esta nueva técnica para partición de matrices, aplicando la misma a los campos de simulación y optimización de procesos. El objetivo fue desarrollar un software capaz de integrar bajo una misma interfase todos los algoritmos necesarios para llevar a cabo la resolución de cualquier sistema, realizando previamente la partición del mismo.

6.3.1 ETAPAS QUE CONFORMAN MP4SO

Para cumplir con el objetivo planteado, el primer paso fue llevar a cabo un diseño general del software. Para esto, fue necesario definir las etapas que integran el mismo, desde que el usuario ingresa el sistema original de ecuaciones, hasta que se obtienen los resultados de la resolución del mismo. También se llevó a cabo la identificación de qué tareas deben ser llevadas a cabo por el usuario y cuáles debía el programa realizar de manera automática. La secuencia de etapas quedó definida de la siguiente manera (figura 6.1):

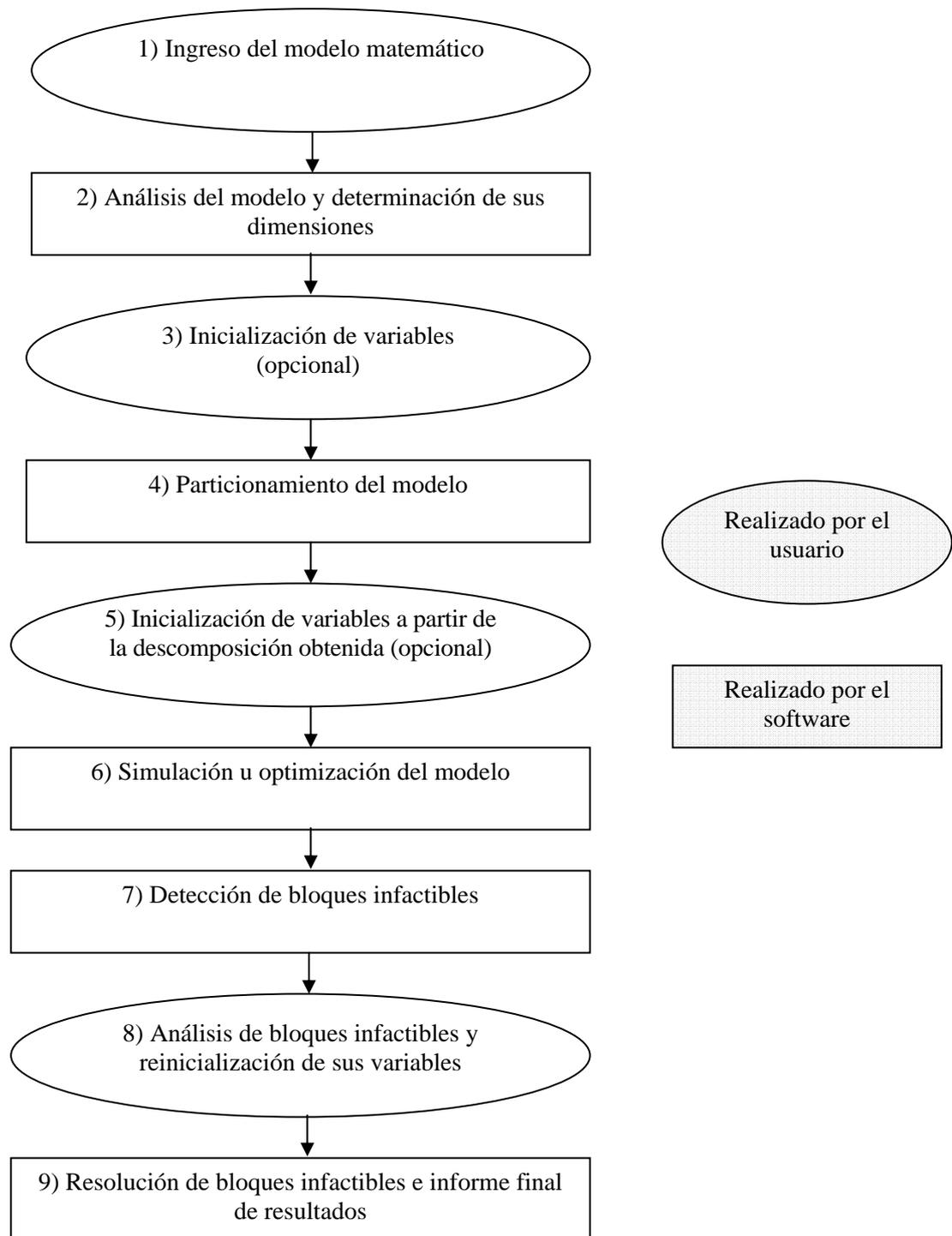


Figura 6.1. Etapas que conforman MP4SO. Las tareas descritas en un rectángulo corresponden a acciones que el software debe realizar de manera automática, mientras que las elipses describen tareas a realizar por el usuario.

Una vez definidas las etapas generales, y habiendo establecido cuáles de esas tareas deben ser llevadas a cabo de manera automática por el software, se realizó un análisis para determinar qué herramientas se encontraban disponibles para ser utilizadas y qué nuevo código hacía falta desarrollar para integrar todo bajo el mismo programa. Para llevar a cabo la ejecución de las diferentes etapas de manera directa y bajo un mismo entorno, se decidió desarrollar un programa base en lenguaje Java (Java, 2009), de manera tal de agrupar los diferentes códigos a utilizar. Esto permite al usuario ejecutar cada uno de los códigos desde una única interfase amigable, a través de los diferentes menús que conforman *MP4SO*.

Etapas 1: Ingreso del modelo matemático

Para realizar el ingreso del modelo, se debió generar en el programa base un área principal de texto. El mismo debía contar con la posibilidad de ingresar el modelo a mano o bien, pegar un modelo desarrollado previamente. Esto permitiría al usuario incorporar a *MP4SO* modelos ya existentes en lenguaje GAMS, así como también los modelos desarrollados de manera automática con el Generador de Modelos presentado en el capítulo 5 de esta Tesis.

Etapas 2: Análisis del modelo y determinación de sus dimensiones

En esta etapa, el software debe ser capaz de recibir un sistema de ecuaciones en lenguaje GAMS, analizarlo y generar los archivos necesarios para ingresar los mismo al Método Directo Extendido. Para ello fue necesario desarrollar un *parser* que leyera cada una de las ecuaciones y extrajera del sistema la información requerida por el algoritmo de observabilidad para llevar a cabo la descomposición del modelo. Los archivos generados por el *parser*, dado un sistema de ecuaciones son:

- **.bgf:** Contiene el bigrafo inicial, indicando los nodos fila (ecuaciones) y sus columnas (variables) adyacentes.
- **.equ:** Contiene la lista de las ecuaciones del sistema, con el número y nombre cada una. También guarda la cantidad total de ecuaciones.
- **.var:** Es similar al anterior, solo que la información está referida a las variables del sistema.

El *parser* fue realizado mediante la herramienta Flex 2.5 (1995), y el usuario lo ejecuta desde la interfase general del programa.

Etapa 3: Inicialización de variables

MP4SO debe permitir al usuario inicializar las variables del sistema de manera sencilla y directa, sin necesidad de modificar el código original ingresado por el usuario en la primera etapa. Para esto, debió programarse la interfase de manera tal que la inicialización de variables realizada por el usuario en el menú destinado para esto, sea incorporada automáticamente al modelo original que será ingresado luego en GAMS.

Etapa 4: Particionamiento del modelo

En esta etapa, *MP4SO* ejecuta el Método Directo Extendido de manera transparente para el usuario. Para ello, el software ingresa al MDE los archivos generados en la etapa 2, obteniendo como resultado la descomposición realizada por el algoritmo de observabilidad. Desde el punto de vista de la programación, esta etapa no requirió del desarrollo de nuevos algoritmos, sino solamente indicar a la interfase que lleve a cabo la ejecución del MDE cuando el usuario así lo solicite.

Etapa 5: Inicialización de variables

Al igual que en la etapa 3, esta etapa permite al usuario inicializar fácilmente las variables del sistema. La diferencia fundamental es que ahora se cuenta con la descomposición del sistema generada por el MDE. Esto ayuda al usuario a identificar los bloques que, ya sea por tamaño como por complejidad de sus ecuaciones, pueden generar problemas en su resolución. Se programó entonces la interfase general de manera tal de mostrar al usuario claramente los bloques conformados, ayudándolo en la toma de decisiones sobre qué variables deberían ser inicializadas.

Etapa 6: Simulación u optimización del modelo

Como se explicó anteriormente, se decidió utilizar GAMS para llevar a cabo la resolución de los subsistemas. La decisión surgió debido a que se encontraron en dicho software características que cuadraban con las necesidades que surgieron en el diseño de esta etapa. Dichas características son:

- Variedad de *solvers*, los cuales pueden ser utilizados dependiendo del tipo de modelo a resolver.
- Posibilidad de resolver tanto sistemas cuadrados como rectangulares. Esta característica de GAMS, sumada a la posibilidad del MDE de descomponer ambos tipos de matrices, permite a *MP4SO* llevar a cabo tanto la simulación como la optimización de un proceso.
- Resolución de un modelo en subsistemas. Esta propiedad de GAMS resultó fundamental para su elección, ya que permite resolver un modelo completo en partes, siempre que el usuario le ingrese al programa una secuencia de resolución. Esa secuencia, en nuestro caso, es el orden de resolución obtenido con el MDE.

Desde el punto de vista de programación, la ejecución de esta etapa requirió el desarrollo de un nuevo *parser*. El mismo tiene la función de tomar la descomposición informada por el MDE e ingresar la misma en GAMS. Para esto el *parser* genera un nuevo archivo de entrada a GAMS (extensión *gms*), con las modificaciones necesarias para que el modelo sea resuelto bloque a bloque. Dicho archivo indica a GAMS cómo realizar la conformación de cada bloque (ecuaciones que los integran) así como también la secuencia de resolución de los mismos.

Etapa 7: Detección de bloques infactibles

Para llevar a cabo esta etapa, se incorporó al programa base la posibilidad de buscar en el archivo de salida de GAMS (.lst) cuáles de los bloques resueltos resultaron infactibles. Dicha información es mostrada al usuario de manera clara, resaltando dichos bloques. Esta es la única incorporación a nivel código que debió realizarse para llevar a cabo esta etapa.

Etapa 8: Análisis de bloques infactibles y reinicialización de sus variables

En esta etapa el usuario puede, a partir del análisis de la información recibida, volver a inicializar aquellas variables que no pudieron ser resueltas por GAMS. Nuevamente, el código base recibe los cambios realizados por el usuario y los traslada al modelo matemático, de manera tal que al volver ejecutar GAMS se tengan en cuenta los nuevos valores de inicialización.

Etapa 9: Resolución de bloques infactibles e informe final de resultados

Una vez ejecutado GAMS, y habiendo el mismo detectado los bloques infactibles, el programa base genera 2 archivos con extensión gms diferentes.

- **Archivo1.gms:** contiene el modelo completo, con todos los parámetros, variables y ecuaciones que lo conforman. Este archivo contiene también el modelado de todos los bloques y las sentencias para resolverlo hasta el último bloque factible que GAMS pudo resolver.
- **Archivo2.gms:** contiene solamente la sentencia para resolver el bloque infactible. También se incluyen en este archivo los nuevos puntos iniciales y límites de las variables que forman el bloque en cuestión.

MP4SO ejecuta el primero de los archivos y almacena los resultados para luego utilizarlos como datos para las corridas del bloque con problemas de convergencia. Luego ejecuta el segundo archivo cargando los resultados obtenidos anteriormente. Este archivo puede ser corrido tantas veces como sea necesarios, sin necesidad de volver a correr todo el modelo. Esto se debe a que la primera parte del modelo ya fue resuelta correctamente, y sus resultados no se verán modificados. Los puntos iniciales y límites para resolver el bloque infactible se incorporan directamente al archivo 2, de manera tal que se puedan modificar todas las veces que resulte necesario sin afectar al archivo 1.

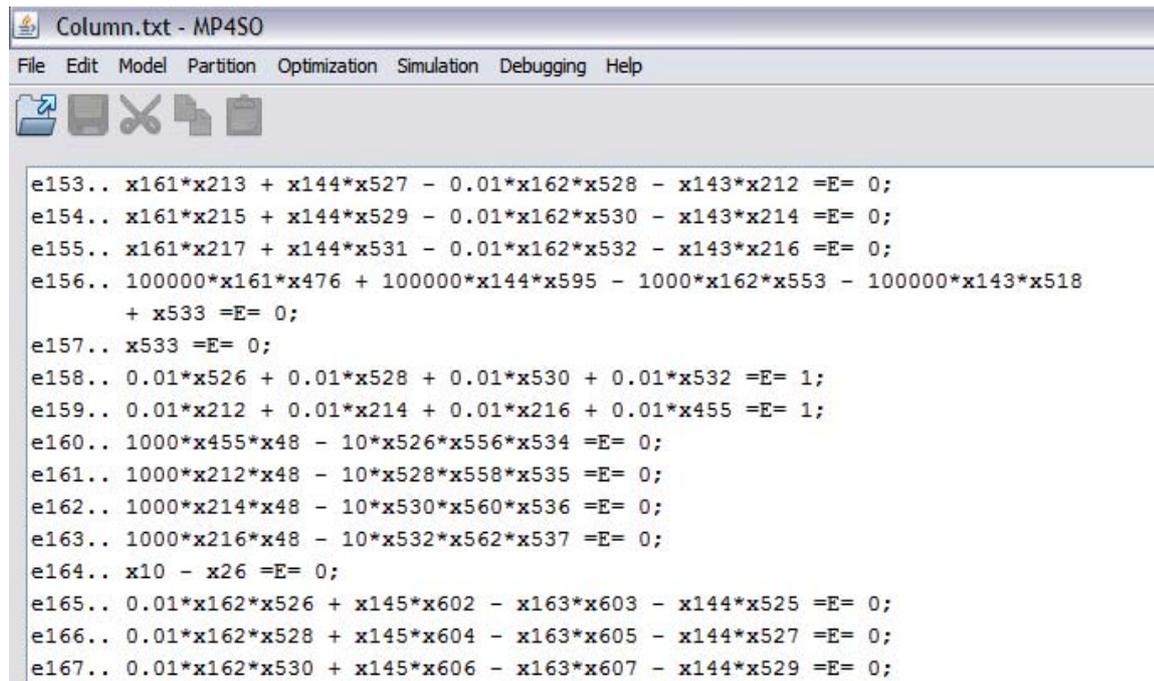
Una vez resuelto el bloque en cuestión, se corre el modelo completo buscando nuevos bloques infactibles en lo que resta del sistema.

6.3.2 MP4SO: PRESENTACIÓN DEL SOFTWARE

A continuación se mencionan los diferentes menús que conforman *MP4SO*, realizando una breve descripción de los ítems que los conforman.

Área de texto

El usuario puede ingresar el modelo matemático en esta hoja, ya sea escribiéndolo o pegándolo de otro archivo. La figura 6.2 muestra el área de texto con las ecuaciones ingresadas en él.



The screenshot shows a window titled 'Column.txt - MP4SO' with a menu bar (File, Edit, Model, Partition, Optimization, Simulation, Debugging, Help) and a toolbar. The main text area contains the following equations:

```

e153.. x161*x213 + x144*x527 - 0.01*x162*x528 - x143*x212 =E= 0;
e154.. x161*x215 + x144*x529 - 0.01*x162*x530 - x143*x214 =E= 0;
e155.. x161*x217 + x144*x531 - 0.01*x162*x532 - x143*x216 =E= 0;
e156.. 100000*x161*x476 + 100000*x144*x595 - 1000*x162*x553 - 100000*x143*x518
      + x533 =E= 0;
e157.. x533 =E= 0;
e158.. 0.01*x526 + 0.01*x528 + 0.01*x530 + 0.01*x532 =E= 1;
e159.. 0.01*x212 + 0.01*x214 + 0.01*x216 + 0.01*x455 =E= 1;
e160.. 1000*x455*x48 - 10*x526*x556*x534 =E= 0;
e161.. 1000*x212*x48 - 10*x528*x558*x535 =E= 0;
e162.. 1000*x214*x48 - 10*x530*x560*x536 =E= 0;
e163.. 1000*x216*x48 - 10*x532*x562*x537 =E= 0;
e164.. x10 - x26 =E= 0;
e165.. 0.01*x162*x526 + x145*x602 - x163*x603 - x144*x525 =E= 0;
e166.. 0.01*x162*x528 + x145*x604 - x163*x605 - x144*x527 =E= 0;
e167.. 0.01*x162*x530 + x145*x606 - x163*x607 - x144*x529 =E= 0;
    
```

Figura 6.2. Captura de pantalla del área de texto de MP4SO. En la figura se visualiza cómo fue ingresado el sistema de ecuaciones de un determinado modelo matemático.

File

Permite al usuario importar modelos previamente desarrollados en GAMS, así como también guardar modelos desarrollados con el software y los cambios realizados.

Edit

Permite al usuario copiar y pegar partes del modelo.

Model

Este ítem se encarga de realizar un análisis del modelo en cuestión y mostrar la información del mismo de manera clara (ver figura 6.3). Está compuesta por las siguientes ventanas:

- *Preprocessing*: Analiza el modelo matemático especificado con notación de GAMS, extrayendo los datos requeridos por el MDE para efectuar la descomposición.
- *Dimension*: Muestra las dimensiones totales del modelo.

- *Parameters*: Muestra la lista de parámetros del modelo, así como también sus valores.
- *Variables*: Muestra la lista de variables del modelo, así como también sus valores límites y valores iniciales.

Es importante destacar que el usuario tiene la posibilidad de modificar los valores mostrados directamente desde este menú, sin tener la necesidad de realizarlo sobre el modelo ingresado.

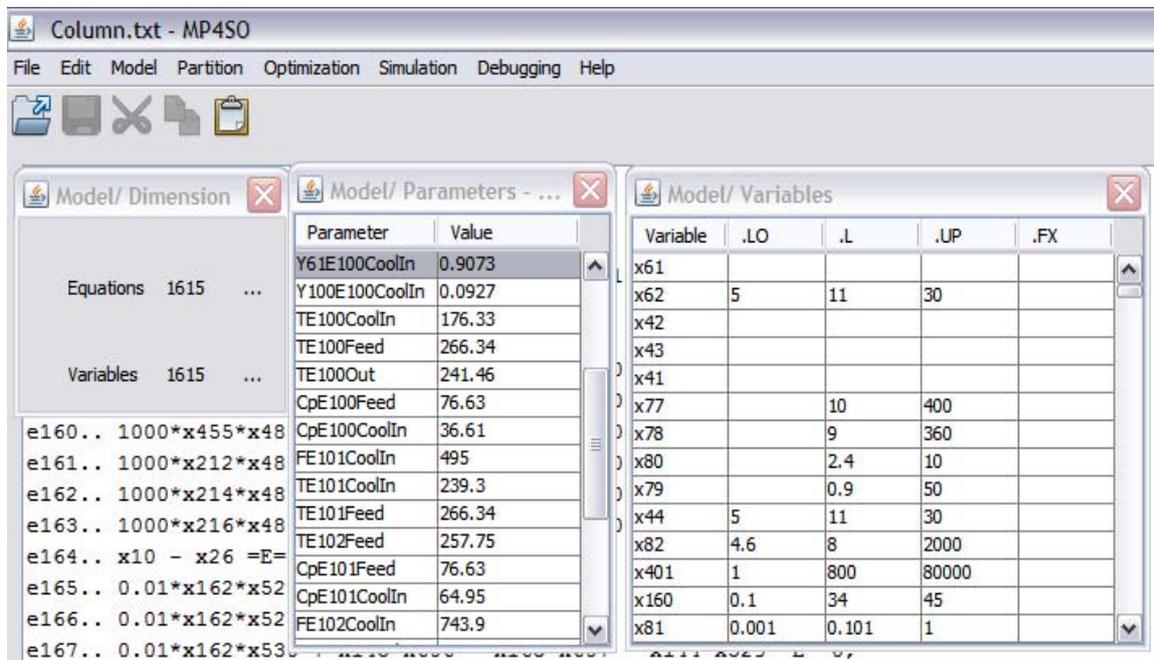


Figura 6.3. Captura de pantalla del menú “Model”. En la figura se puede visualizar la ventana que informa las dimensiones del modelo en cuestión, así como también las ventanas de parámetros y variables. Tanto los valores de los parámetros como los límites y valores iniciales de las variables pueden ser modificados en dichas ventanas.

Partition

Este menú permite al usuario ejecutar el MDE y llevar a cabo la partición del modelo, mostrando de manera clara los resultados obtenidos (ver figura 6.4). Está compuesto por los siguientes ítems:

- *Run*: Inicia la partición usando MDE.
- *Subsystems*: Informa de manera sencilla qué ecuaciones y variables integran cada uno de los bloques obtenidos.
- *Equations*: Muestra la lista completa de ecuaciones ordenadas de acuerdo a los bloques obtenidos con el MDE.
- *Variables*: Muestra la lista de variables del modelo ordenadas de acuerdo a la partición obtenida, así como también sus valores límites y valores iniciales. El usuario tiene la posibilidad de cambiar estos valores directamente desde esta ventana

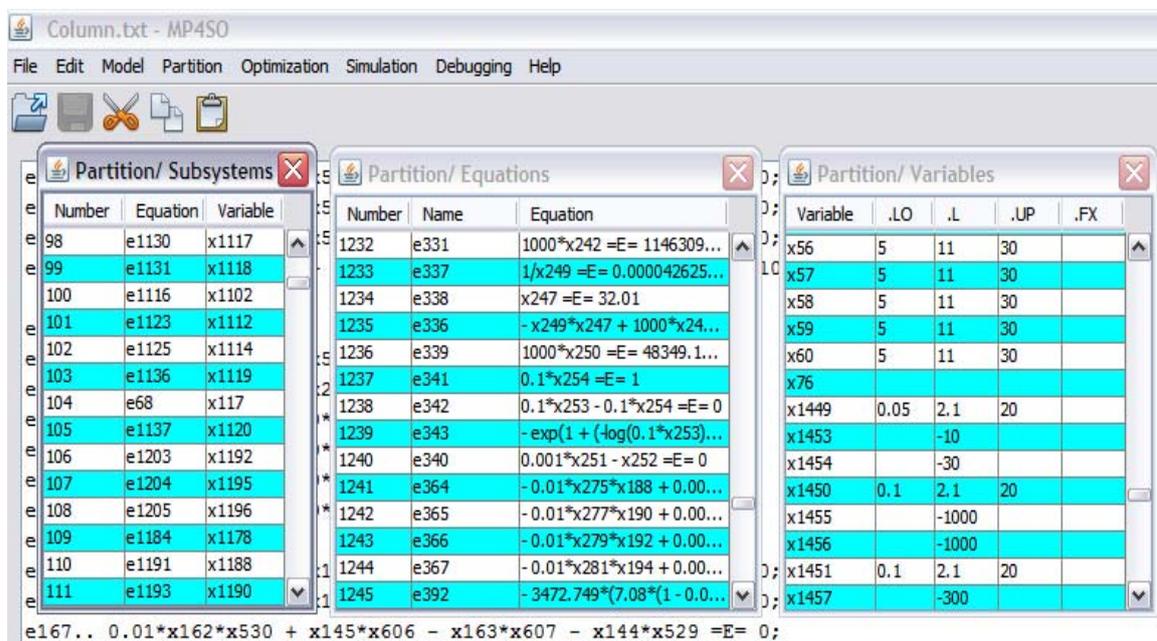


Figura 6.4. Captura de pantalla del menú “Partition”. En la figura se puede visualizar cómo MP4SO muestra de manera clara, en colores blanco y celeste, la conformación de los diferentes subsistemas obtenidos a partir de la aplicación del MDE. Cada fila corresponde a una ecuación y un conjunto de ecuaciones pintadas del mismo color conforman un bloque.

Optimization/Simulation

Dependiendo de si se trata de una simulación u optimización, el usuario deberá utilizar uno u otro menú desplegable (ver figura 6.5). Desde este menú el usuario ejecuta GAMS, de manera tal de llevar a cabo la resolución en bloques del modelo, de acuerdo al orden establecido por el MDE. Este menú está compuesto por los siguientes ítems:

- *Run*: Ejecuta GAMS para resolver el modelo.
- *Results*: Muestra el archivo de salida generado por GAMS (extensión .lst).
- *Subsystems*: Muestra la totalidad de los bloques del modelo, resaltando aquellos que por algún motivo no pudieron ser resueltos por GAMS (bloques infactibles).
- *Variables*: Muestra los valores calculados por GAMS para cada variable. Nuevamente, el usuario tiene la posibilidad de cambiar valores límites e iniciales de las variables directamente desde esta ventana. La variable cuyo valor no pudo ser calculado correctamente es resaltada de manera de ser fácilmente identificada.

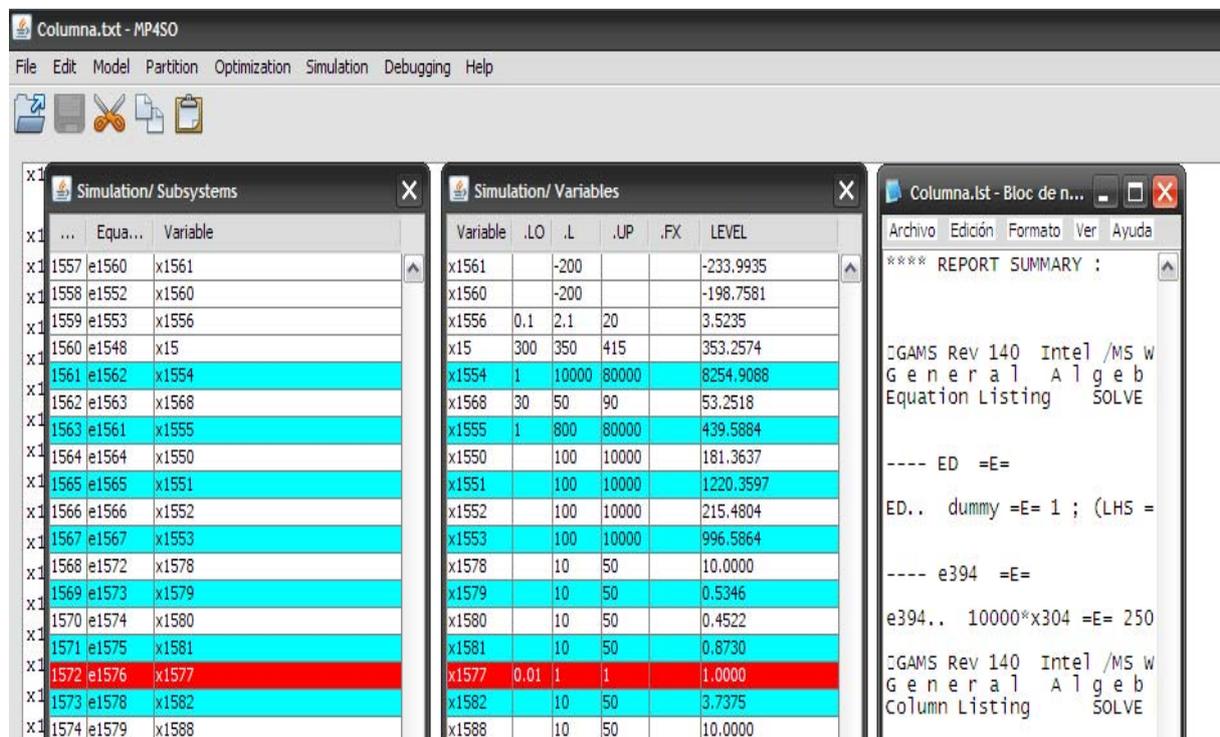


Figura 6.5. Captura de pantalla del menú “Simulation”. En la figura se puede visualizar cómo el programa destaca en color rojo aquellos bloques que no pudieron ser resueltos correctamente. Esto permite al usuario modificar directamente desde aquí los valores límites e iniciales de las variables problemáticas, de manera tal de volver a resolver el bloque. Para este caso en particular, el bloque infactible es de tamaño 1X1.

Debugging:

Este menú está orientado al tratamiento de subsistemas no factibles del modelo bajo estudio (bloques en rojo), permitiendo al usuario llevar a cabo correcciones sobre el modelo

focalizándose únicamente en los bloques no factibles. Para esto, *MP4SO* resuelve parcialmente el modelo hasta el bloque no factible, almacenando los resultados obtenidos. Luego, el usuario tiene la posibilidad de correr el bloque conflictivo dando diferentes valores iniciales o límites a sus variables hasta que el bloque converja. Una vez lograda la convergencia, se ejecuta nuevamente el modelo en búsqueda de nuevos bloques infactibles hasta llegar a la solución total del sistema. Este menú está compuesto por los siguientes ítems:

- *Start Debugging*: Almacena el modelo en dos archivos diferentes. Un archivo contiene la parte del modelo que converge mientras que el otro está formado por el primero de los bloques infactibles del sistema de ecuaciones.
- *Run Feasible Part*: Ejecuta la parte factible del modelo en GAMS, almacenando los resultados obtenidos.
- *Variables*: Muestra al usuario una tabla con las variables que forman el bloque infactible, así como también sus límites y puntos iniciales. El usuario puede cambiar esos valores antes de intentar resolver el bloque en GAMS.
- *Run Infeasible Part*: Ejecuta el bloque infactible en GAMS, utilizando para las variables ya calculadas que forman parte del mismo los resultados obtenidos anteriormente.
- *Results*: Muestra el archivo de salida generado por GAMS para el bloque infactible.

Help

Contiene un manual detallado acerca de la utilización del software.

6.4 ANÁLISIS DE PERFORMANCE DEL SOFTWARE

Para verificar el funcionamiento del programa se corrieron casos de estudio de diferentes dimensiones y complejidad. Los mismos corresponden a modelos matemáticos de procesos reales, los cuales en algunos casos fueron simulados y en otros optimizados. Para cada uno de ellos se incluye una breve descripción del proceso y tamaño del modelo, seguido por el detalle de los resultados obtenidos. Para los modelos simulados se detalla la descomposición obtenida por el programa, así como también el porcentaje de variables que deben ser inicializadas para llevar a cabo la resolución en bloques. Para los modelos

optimizados se detallan los bloques cuadrados obtenidos, así como también el tamaño del bloque rectangular asignado a optimización. También se incluye la información de qué porcentaje del modelo fue asignado a simulación (bloques cuadrados) y qué porcentaje de las variables debieron ser inicializadas para lograr la convergencia del modelo completo.

6.4.1 SIMULACIÓN DE PROCESOS

6.4.1.1 CRAQUEO CATALÍTICO DE GASOIL

En este proceso nafta de bajo octanaje y gas oil fresco son craqueados catalíticamente mediante la utilización de un catalizador compuesto por zeolita, obteniéndose una nafta con mayor grado de octanaje. La recuperación de una fracción de nafta del producto craqueado, y la combinación del mismo con la alimentación fresca de nafta, incrementa aún más el grado de octanaje del producto final. El siguiente caso de estudio tiene como objetivo la determinación de los coeficientes de reacción para el craqueo catalítico del gas oil. El modelo fue extraído del sitio Web COPS (COPS, página Web, 2009) el cual contiene una amplia librería de problemas de optimización de gran escala. En la figura 6.6 se puede visualizar un esquema simplificado del proceso en cuestión.

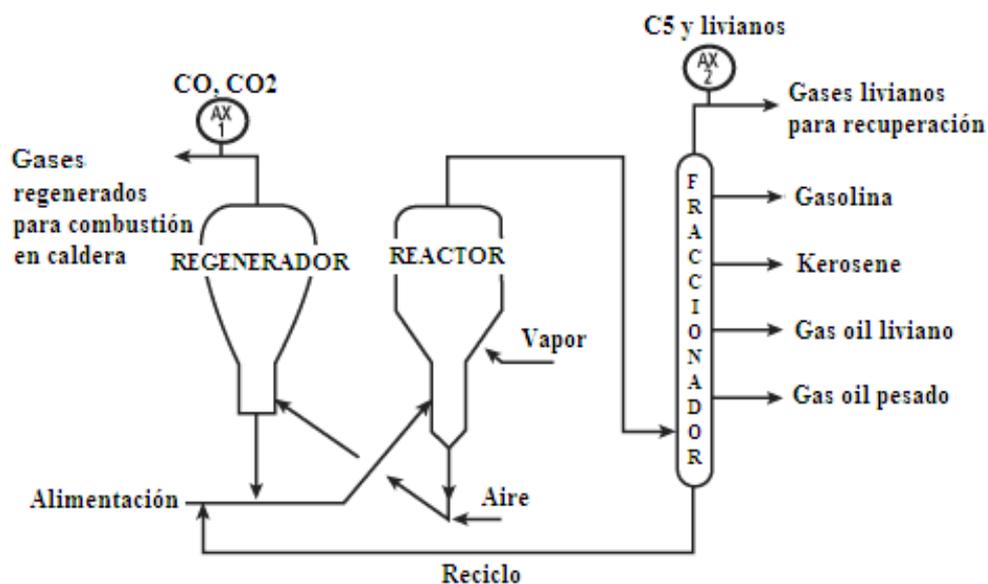


Figura 6.6. Esquema del proceso de craqueo catalítico de gas oil.

El modelo matemático elegido para representar el proceso consiste en un sistema cuadrado de 1299 ecuaciones y variables. La tabla 6.1 muestra los resultados obtenidos para este caso.

N° Ec.	N° Var.	Bloques			% Var Inicializ.
		1x1	12x12	285x285	
1299	1299	102	76	1	21

Tabla. 6.1. Descomposición obtenida para la simulación del craqueo catalítico de gasoil.

En este caso, se puede ver que la descomposición lograda requiere solamente la inicialización del 21% de las variables del sistema, para que el mismo pueda ser resuelto. Dicho porcentaje está dado por las variables que conforman el bloque de 285x285, el cual no pudo ser resuelto sin tener sus variables inicializadas. El resto de los bloques (de tamaño 1x1 y 12x12) no tuvieron necesidad de ser inicializados.

6.4.1.2 COLUMNA DE DESTILACIÓN REACTIVA

La destilación reactiva constituye un claro ejemplo de lo que se conoce como “integración de procesos”. Puede ser presentada como una serie de reactores multifuncionales, donde tanto los procesos de reacción como de separación son combinados en una única unidad, reduciendo de manera notable los costos inversión (ver anexo D de esta Tesis). La columna de destilación reactiva es modelada como una columna de platos, utilizando etapas reactivas y no reactivas dependiendo del proceso que esté ocurriendo dentro de ella. En el capítulo 5 de esta Tesis se puede encontrar una descripción más detallada del proceso, mientras que en el apéndice A se incluye el modelo matemático completo de la misma.

El modelo de estado estacionario utilizado para este caso de estudio esta formado por un sistema no lineal cuadrado de 1615 ecuaciones y variables. La tabla 6.2 incluye los resultados obtenidos mediante la aplicación de *MP4SO*.

N° Ec.	N° Var.	Bloques				% Var Inicializ.
		1x1	5x5	13x13	997x997	
1615	1615	600	1	1	1	61

Tabla. 6.2. Descomposición obtenida para la simulación de la columna reactiva.

A partir de la tabla 6.2 se puede observar que se trata de un caso con mayor grado de acople entre ecuaciones que en el primero de los ejemplos. Esto puede resultar lógico viendo

que se trata de una columna de destilación reactiva. El mayor grado de acople en el sistema da como resultado una descomposición más pobre, y por lo tanto, la necesidad de incrementar el porcentaje de variables inicializadas para llevar a cabo la resolución del modelo. En este caso se debe inicializar el 61% del modelo, porcentaje que está dado por el bloque de 997x997 que no pudo ser desacoplado.

6.4.2 OPTIMIZACIÓN DE PROCESOS

6.4.2.1 RED DE INTERCAMBIADORES DE CALOR

Este caso representa una red de intercambiadores de calor, en los que una corriente principal compuesta por metano, etano, propano y butano, es enfriada de 20°F a -24°F. Para lograr esto se utiliza una red de intercambiadores de calor con servicios externos, dividida en dos partes en paralelo, una que contiene un intercambiador y la otra que contiene dos. La alimentación principal se divide entonces en dos corrientes que se combinan luego para obtener una corriente a la temperatura deseada. El esquema del proceso se muestra en la figura 6.7.

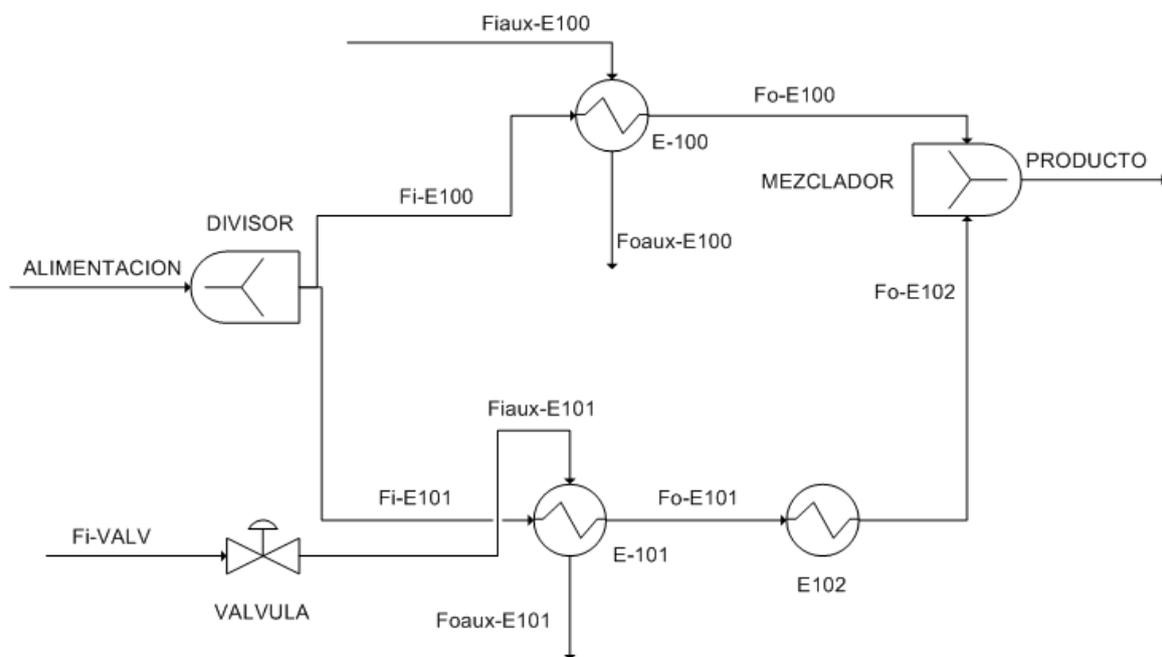


Figura 6.7. Esquema del proceso de optimización de la red de intercambiadores de calor.

El modelo matemático está compuesto por 58 ecuaciones y 59 variables. El mismo resulta de realizar los balances de masa en los mezcladores y divisores, así como también los balances de masa y energía en cada uno de los tres intercambiadores de calor. La función objetivo del caso de estudio es la sumatoria de los UA (área de intercambio de los equipos), la cual debe ser minimizada en orden de reducir al máximo los costos de instalación de los intercambiadores. La variable de optimización es Fi-E101 el caudal que se hace circular por la serie de dos intercambiadores, el cual es calculado luego de realizada la optimización del proceso. La tabla 6.3 resume los resultados obtenidos para el proceso en cuestión.

N° Ec.	N° Var.	Bloq. Simulación	Bloq. Optimización		% Ec. Simul.	% Var Inicializ.
		1x1	Ec.	Var.		
58	59	40	18	19	69	31

Tabla. 6.3. Descomposición obtenida para la simulación de la red de intercambiadores

Este ejemplo de muestra claramente cómo normalmente en una optimización, una parte del modelo es independiente de las variables de optimización y función objetivo. Para este caso en particular, un 69% de las ecuaciones del modelo pudieron independizarse y descomponerse en subsistemas. Esto implica que sólo el 31% del modelo original fue tratado como un problema de optimización, mientras que el resto de las ecuaciones conformaron un sistema cuadrado que fue simulado. A su vez, la parte del modelo simulada fue descompuesta en 40 bloques de 1x1, por lo que las variables correspondientes a dichos subsistemas no debieron ser inicializadas. Sí debieron ser inicializadas las variables correspondientes al bloque de optimización, lo que conforma el 31% de variables del modelo.

6.4.2.2 UBICACIÓN ÓPTIMA DE CATALIZADOR EN UN REACTOR TUBULAR

El modelo de reactor flujo-pistón (RFP) es utilizado para estimar las variables claves de operación cuando se utiliza un reactor tubular para alcanzar un producto específico. En un RFP el fluido atraviesa la unidad de manera uniforme, de manera tal que el tiempo de residencia es el mismo para todos los elementos del fluido. Una suposición necesaria para el modelado de este tipo de reactores es que el flujo que atraviesa un RFP se encuentra perfectamente mezclado en la dirección radial, pero no en dirección axial (hacia adelante o hacia atrás). Para su modelado la unidad es dividida en diferentes unidades diferenciales de volumen (ver figura 6.8), los cuales son considerados como entidades separadas, ubicadas una a continuación de otra. La suposición de mezclado perfecto contempla que no existen

gradientes espaciales en una propiedad determinada como la concentración, temperatura, potencial químico, etc.

Un reactor flujo pistón se encuentra típicamente conformado por un tubo empacado con algún material sólido en su interior (frecuentemente un catalizador). En algunos casos se trata de un tubo dentro de un casco con el cual, a medida que la reacción ocurre dentro de él, también intercambia calor con el casco mencionado.

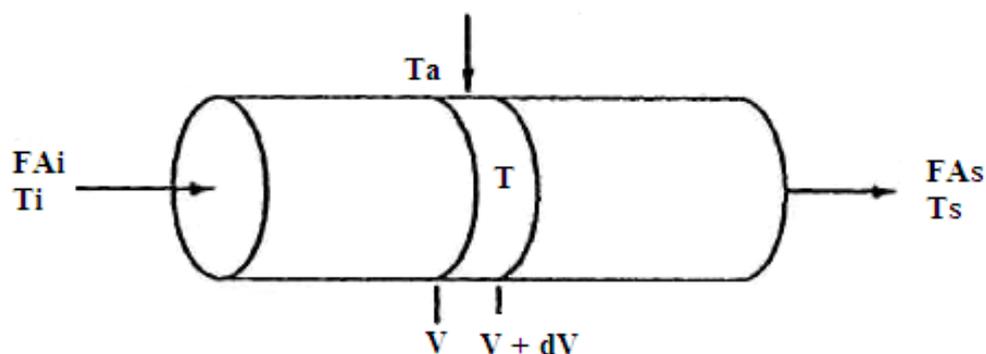


Figura 6.8. Esquema de un reactor tubular flujo pistón

El caso de estudio presentado en esta sección consiste en una optimización que tiene como objetivo la determinación de la política óptima de distribución de dos catalizadores a lo largo de un reactor tubular flujo-pistón, en el cual ocurren varias reacciones diferentes. El modelo matemático utilizado consta de 201 ecuaciones con 227 variables y fue extraído del sitio Web COPS (COPS, página Web, 2009). La tabla 6.4 resume los resultados obtenidos de la ejecución de MP4SO para el proceso en cuestión.

N° Ec.	N° Var.	Bloq. Simulación				Bloq. Optimización		% Ec. Simul.	% Var. Inicial.
		2x2	3x3	4x4	6x6	Ec.	Var.		
201	227	14	1	2	6	126	152	38	66

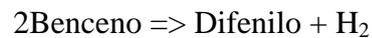
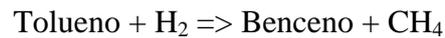
Tabla. 6.4. Descomposición obtenida para la optimización de la ubicación de un catalizador en un reactor tubular

En este caso, se puede ver que existe un mayor grado de acople entre la función objetivo y el resto de las ecuaciones del modelo. Esto se debe a que existe un mayor número de grados de libertad que el caso anterior. Para este caso sólo el 38% de las ecuaciones pudieron agruparse en bloques cuadrados, mientras que el resto integran el bloque de la optimización. En cuanto a la inicialización de variables, se puede apreciar en la tabla 6.4 que el 66% de las variables del modelo debieron ser inicializadas. Ese valor corresponde a las variables que conforman el bloque

de optimización, dado que los bloques cuadrados asignados a simulación pudieron ser descompuestos de manera tal de no requerir inicialización de sus variables. En la tabla 6.4 se puede apreciar que el máximo tamaño de bloque obtenido para simulación es de 6x6.

6.4.2.3 PROCESO DE HDA PARA PRODUCCIÓN DE BENCENO

En este ejemplo se considera la hidrodealquilación del tolueno (HDA) para producir benceno (Douglas 1988). Las reacciones de interés son:



Ambas reacciones se llevan a cabo en un rango de temperaturas que va desde 1150°F (debajo de esta temperatura la velocidad de reacción es muy lenta) hasta 1300°F (por encima de esta temperatura se produce una significativa cantidad de hidrocraqueo), y a una presión de 500 psia. Se requiere un exceso de hidrógeno (relación 5 a 1) para prevenir la formación de coque, y los gases efluentes del reactor deben ser rápidamente llevados a 1150°F para que no ocurra el mismo efecto en el intercambiador de calor que sigue al reactor. Un esquema simplificado del proceso se muestra en la figura 6.9. Las corrientes de alimentación del proceso de tolueno e hidrógeno son calentadas y mezcladas con las corrientes de reciclaje, antes de ser alimentadas al reactor. La corriente de producto que abandona el reactor contiene hidrógeno, metano, benceno, tolueno y difenilo. Luego, mediante un sistema de separación se logra obtener el benceno (producto principal) y el difenilo (producto secundario). Es importante resaltar que se debe realizar una purga de gases, antes de que estos ingresen nuevamente al reactor como reciclaje, ya que la acumulación de inertes (CH₄) complicaría el correcto funcionamiento del reactor.

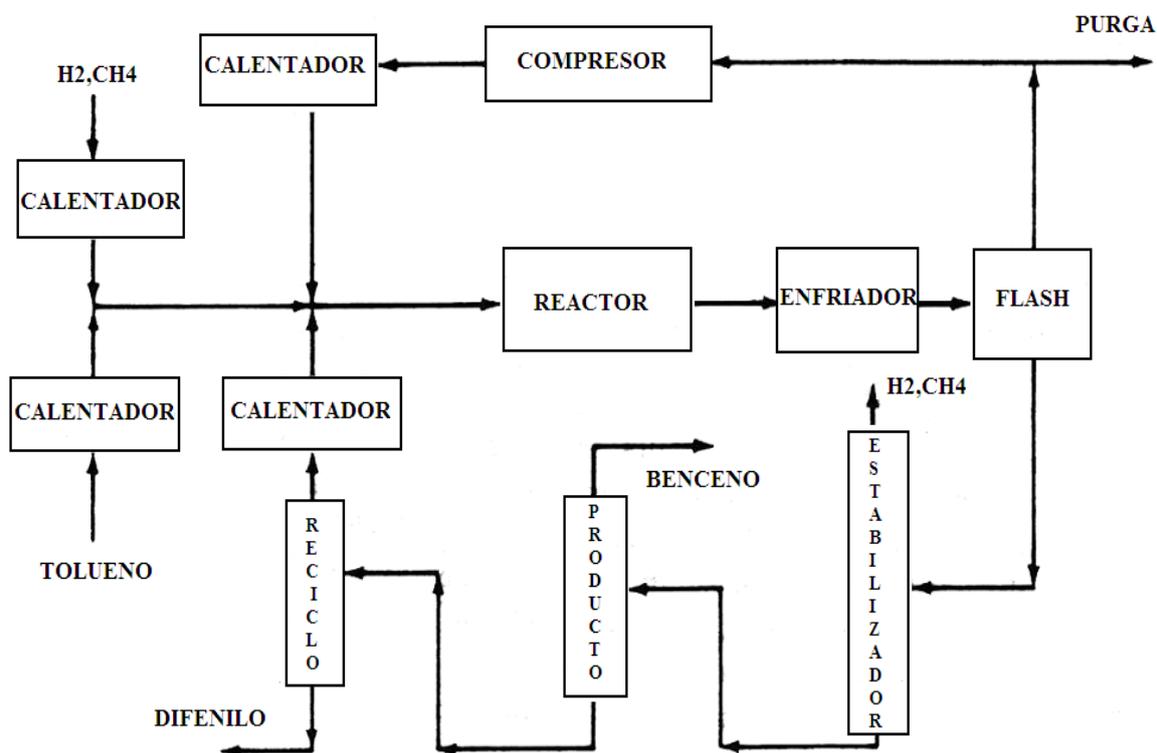


Figura 6.9. Esquema del proceso de HDA para producción de benceno

En este problema de optimización, se fijó como función objetivo el potencial económico obtenido de la venta del benceno, difenilo y los gases purgados (combustible). Las variables de optimización son la conversión que se obtiene en el reactor y la fracción de hidrógeno en la purga. Estas variables son calculadas por el algoritmo de optimización. El sistema resultante es de 634 ecuaciones y 695 variables. La tabla 6.5 incluye los resultados obtenidos mediante la aplicación de MP4SO al caso de estudio en cuestión.

Nº Ec.	Nº Var.	Bloq. Simulación				Bloq. Optimización		% Ec. Simul.	% Var Inicial.
		2x2	3x3	4x4	6x6	Ec.	Var.		
634	695	184	2	1	1	435	496	32	71

Tabla. 6.5. Descomposición obtenida para la optimización de la producción de HDA

Este caso arroja una descomposición similar al caso anterior. El 32% de las ecuaciones pudieron agruparse en bloques cuadrados, mientras que el resto integran el bloque de la optimización. A su vez, se puede ver en la tabla 6.5 que el 71% de las variables del modelo debieron ser inicializadas. Nuevamente, ese valor corresponde a las variables que conforman el

bloque de optimización. Las variables asignadas a simulación se encuentran formando bloques pequeños, por lo que no resultó necesario inicializar dichas variables.

6.4.3 ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Como se puede apreciar, las descomposiciones obtenidas dependen del grado de acople que exista entre las variables del modelo en cuestión. También influye si se trata de un problema de simulación u optimización. En un problema de optimización, se busca minimizar una función objetivo fijando las diferentes variables de optimización, por lo que el grado de acople es mayor que en un problema de simulación. Aplicando *MP4SO* a un problema de optimización, uno puede detectar qué parte del modelo se encuentra desacoplada de las variables de optimización y la función objetivo, y esa parte es la que el programa está en condiciones de descomponer, generando entonces un bloque final con todas las ecuaciones que contengan variables de optimización y la función objetivo. Los tiempos de ejecución de los ejemplos, tanto para el modelo completo como para el particionado se encontraron entre los 10 y los 120 segundos ejecutados sobre un procesador Pentium 4 (3.00 Ghz).

Es importante resaltar que muchos de los bloques obtenidos en los ejemplos son de mínimo tamaño y resolución directa. Por lo tanto una vez resueltos, el usuario puede ya dejarlos de lado para focalizarse en los bloques de mayor tamaño. Estos últimos son los que generalmente requieren de un proceso de prueba y error, cambiando límites y valores iniciales de variables hasta lograr la convergencia de los mismos.

6.5 CONCLUSIONES

En este capítulo se presentó un nuevo software para simulación y optimización de procesos industriales denominado *MP4SO*, el cual se encuentra basado en la estrategia orientada a ecuaciones. *MP4SO* consiste en una herramienta que trabaja de manera interoperativa con el paquete comercial de modelado matemático GAMS. Este software basa su funcionamiento en el algoritmo de análisis de observabilidad conocido como Método Directo Extendido. El MDE es un algoritmo de particionamiento basado en teoría de grafos que efectúa un reordenamiento estructural de la matriz de incidencia correspondiente al modelo de estado estacionario de un proceso. Esta restructuración conduce a una forma triangular inferior en bloques, la cual define implícitamente una descomposición en subsistemas y un orden de precedencia para la resolución eficiente del sistema de ecuaciones asociado a dicha matriz.

MP4SO puede ser utilizado tanto en modelos matemáticos estructuralmente singulares como no singulares, correspondientes tanto a sistemas de ecuaciones lineales como no lineales. La aplicación de este software a problemas de grandes dimensiones, como son normalmente los procesos industriales, permite reducir las desventajas que naturalmente tiene la estrategia orientada a ecuaciones.

El software posee una interfase amigable que permite su empleo tanto en simulación, como en optimización. En un primer paso, se particiona el modelo usando el MDE, obteniendo una descomposición en subsistemas. En una segunda etapa, se alimenta a GAMS con la descomposición obtenida, a fin de realizar la simulación (sistemas no singulares) u optimización (sistemas singulares) del proceso bajo estudio, siguiendo el orden de resolución propuesto por nuestro algoritmo de descomposición. Finalmente, el sistema es resuelto usando los *solvers* de GAMS, mediante la resolución de cada uno de los subsistemas.

Para verificar el funcionamiento del software, se modelaron procesos industriales de diferente complejidad, entre los que se encuentra la columna de destilación reactiva tratada en detalle en el capítulo 5 de esta Tesis. Algunos de los casos de estudio fueron simulados y otros optimizados. La diversidad de acoplamiento de los modelos permitió obtener mejores o peores descomposiciones, dependiendo de si se trataba de simulaciones u optimizaciones, y del caso de estudio resuelto.

Comparando cualquier modelo particionado con el modelo original se pueden resaltar las siguientes ventajas:

- Un modelo particionado siempre tendrá menos requerimientos computacionales para llevar a cabo su resolución.
- El número de variables a inicializar para lograr la convergencia disminuye notablemente en modelos particionados, por lo que la ardua tarea de inicialización se ve simplificada.
- Gracias a *MP4SO* se facilita notablemente la etapa de revisión del modelo ante la aparición de errores en la resolución matemática. El software informa al usuario claramente cuál es el bloque infactible, de manera tal que sólo se deba realizar el procedimiento de prueba y error de valores iniciales con el bloque en cuestión.

- En términos de tiempos de resolución, la mayor ganancia se da en la etapa de corrección de errores. Para un modelo particionado, esta etapa debe realizarse sólo para bloques específicos, mientras que para el modelo sin descomponer es necesario ejecutar el sistema completo innumerable cantidad de veces hasta lograr depurar todos los errores.

CONCLUSIONES Y TRABAJOS FUTUROS

El objetivo global de esta Tesis Doctoral ha sido desarrollar métodos para llevar a cabo el análisis estructural de modelos matemáticos, de modo que se facilite la resolución de problemas en ingeniería de procesos. Dichos métodos fueron aplicados a los campos de diseño de instrumentación, simulación y optimización de procesos químicos. Para cumplir con dicho objetivo, se utilizó como herramienta principal un nuevo algoritmo de particionamiento de matrices denominado Método Directo, el cual fue desarrollado previamente por nuestro grupo de investigación.

En el campo de diseño de instrumentación se logró desarrollar una nueva versión del Método Directo, que a diferencia de los algoritmos para descomposición ya existentes, cuantifica la no linealidad inherente a un sistema de ecuaciones. De esta manera, se puede llevar a cabo la descomposición de un modelo matemático obteniendo subsistemas con el menor grado de no linealidad posible. Como se verificó en esta Tesis, la aplicación de esta nueva técnica de descomposición favorece el proceso global de diseño de instrumentación en dos aspectos principales: incremento en la velocidad de procesamiento de datos e incremento en el conocimiento acerca del nivel de redundancia del modelo. .

En los campos de simulación y optimización de procesos, se desarrollaron herramientas que integradas entre sí conforman un paquete completo para generación, partición y resolución de modelos matemáticos pertenecientes a plantas químicas, sean éstos cuadrados o no. Dicho paquete resulta sencillo de utilizar, muy flexible, y además combina las ventajas que naturalmente posee el enfoque de simulación secuencial modular con las del enfoque orientado a ecuaciones.

En este capítulo se resumen conclusiones obtenidas a partir de la aplicación de las diversas técnicas desarrolladas en el marco de esta Tesis para los diferentes campos de la

ingeniería de procesos, así como también los resultados principales obtenidos mediante la aplicación de cada una de ellas. También se presentan recomendaciones y lineamientos generales para trabajos futuros relacionados con el tema, así como también sugerencias sobre extensiones naturales del trabajo desarrollado en esta Tesis.

7.1 CONCLUSIONES

7.1.1 SECCIÓN I: ANÁLISIS CUALITATIVO DE PROCESOS

El objetivo principal del análisis de observabilidad para diseño de instrumentación es identificar en el modelo matemático de una planta de procesos qué variables no medidas pueden calcularse a partir de las variables medidas y de las ecuaciones que integran el sistema. A partir de la ejecución de dicho análisis se puede determinar cuáles de las variables no medidas se ubican en el grupo de las observables (calculables) y cuáles en el grupo de las no observables (no calculables). En particular, el DSS desarrollado por nuestro grupo de investigación emplea para llevar a cabo esta tarea el algoritmo conocido como Método Directo.

El Método Directo constituye una técnica de particionamiento de matrices muy robusta y eficiente. Entre sus características principales se encuentra la posibilidad de aplicarlo sobre cualquier tipo de matrices (cuadradas o rectangulares); su facilidad para la incorporación de restricciones en la conformación de los bloques para guiar el ordenamiento según pautas especificadas previamente; y, finalmente, su flexibilidad a la hora de generar la descomposición de la matriz en bloques, siendo posible generar diversos reordenamientos para un mismo problema hasta encontrar el más satisfactorio. El resultado de la aplicación del algoritmo es la obtención de una matriz triangular inferior en bloques donde los bloques de la diagonal son cuadrados y con transversal llena. En el peor caso posible de descomposición, es decir, cuando el sistema no pueda ser descompuesto en subsistemas, el método arroja como resultado una matriz idéntica a la suministrada para ser particionada.

Generalmente, con las técnicas propuestas para realizar el análisis de observabilidad, se busca determinar un diseño de instrumentación que maximice la cantidad de variables calculables de una planta. Una característica de los modelos que representan los procesos químicos es su fuerte carácter no lineal, cuya principal desventaja es que este carácter posee un fuerte impacto en el desempeño de las técnicas utilizadas para efectuar el análisis de redundancia. Estas técnicas reducen significativamente el tiempo y esfuerzo de cómputo

requeridos cuanto mayor es la cantidad de subsistemas lineales que debe procesar respecto del número de sistemas no lineales. Otra característica de los modelos es que las matrices de incidencia asociadas a los sistemas de ecuaciones son generalmente deficientes en rango. Esto se debe a la presencia de ecuaciones redundantes, cuyo propósito es permitir la reconciliación de datos a fin de detectar errores de medición en los sensores ubicados en la planta.

Como consecuencia de esto, no existe una única forma de descomponer el sistemas de ecuaciones, pudiendo obtenerse diversas particiones con distintos grados de no linealidad, en función de las ecuaciones y variables que conforman cada subsistema. Esta característica dio lugar a uno de los objetivos planteados en esta Tesis, el cual consistió en encontrar aquella descomposición disminuya efectivamente la no linealidad de los subsistemas obtenidos, ubicando las ecuaciones no lineales en el conjunto de ecuaciones redundantes o no asignadas. Actualmente, ninguno de los algoritmos para análisis de observabilidad tiene en cuenta este factor, lo cual motivó a realizar un análisis de la factibilidad de extender el Método Directo a fin de incorporar este aspecto al proceso de descomposición.

Se implementó entonces una mejora al Método Directo, dando lugar a un nuevo algoritmo para análisis de observabilidad denominado Método Directo Extendido. En este método se desarrolló un módulo específico que cuantifica la no linealidad inherente a un sistema de ecuaciones. Este módulo permite alcanzar un ordenamiento que reduce la no linealidad de los subsistemas obtenidos a través del proceso de análisis de observabilidad. De la misma manera, el algoritmo es capaz de incrementar la cantidad de bloques de mínimo tamaño generados en el procesamiento.

Se analizaron los resultados obtenidos y se pudo verificar que la característica de las descomposiciones obtenidas produjo mejoras en la eficiencia y los tiempos de cómputo requeridos para realizar el análisis de redundancia, que es otra de las etapas del diseño de instrumentación. Por un lado, se logró incrementar el conocimiento acerca del nivel de redundancia del proceso en cuestión, lo cual facilita la toma de decisiones por parte del usuario. Por otro lado, se logró una reducción de tiempos del proceso global de diseño de instrumentación, dado que el Método Directo Extendido permite lograr una disminución significativa en el tiempo necesario para la ejecución del análisis de redundancia.

7.1.2 SECCIÓN II: ANÁLISIS CUANTITATIVO DE PROCESOS

La segunda parte de esta Tesis se encuentra dedicada a lo que hemos denominado “análisis cuantitativo de procesos”. El objetivo principal de esta sección consistió en ampliar el rango de aplicación del Método Directo, siendo ya conocida su gran utilidad en el área de diseño de instrumentación. Para cumplir con dicho objetivo, se desarrollaron técnicas y herramientas que asistieran al usuario en la resolución de modelos matemáticos, es decir, en las tareas de simulación y optimización de procesos.

En primer lugar, se presentó un software para la generación automática de modelos matemáticos de plantas de procesos. El mismo permite obtener el sistema de ecuaciones que representa el funcionamiento de una planta de procesos a partir del ingreso al programa de la topología de la misma. Una vez obtenido el sistema de ecuaciones, se puede llevar a cabo la resolución del mismo por medio de alguno de los algoritmos disponibles para ello, teniendo en cuenta las características del sistema obtenido.

El generador posee una interfaz amigable que permite al usuario ingresar fácilmente la topología de la planta y obtener el sistema completo de ecuaciones que representa el comportamiento de la misma. Otra característica de interés es la etapa de verificación, en la cual se lleva a cabo la comprobación de compuestos pertenecientes a las corrientes que ingresan o salen de los equipos, así como también la verificación de las fases de las mismas. El software resulta flexible a la incorporación de nuevos equipos o paquetes termodinámicos de manera relativamente sencilla, siguiendo una metodología desarrollada en lenguaje XML. El formato de los modelos matemáticos generados es fácilmente adaptable a los diferentes programas de resolución que quieran ser ejecutados para simular u optimizar los mismos. Actualmente, el formato elegido para ello es compatible con GAMS, de manera tal de poder sacar provecho de la gran cantidad de algoritmos de resolución que este posee. Para probar el funcionamiento del generador, se incorporó al mismo el modelo de una columna de destilación reactiva, el cual fue posteriormente simulado y optimizado mediante la utilización de GAMS.

Una vez descrito el generador de modelos, se presentó en esta Tesis un nuevo desarrollo de software denominado *MP4SO* para simulación y optimización de procesos industriales, el cual se encuentra basado en la estrategia orientada a ecuaciones. El mismo consiste en una herramienta de intuitivo manejo, que trabaja de manera interoperativa con

GAMS y basa su funcionamiento en las descomposiciones obtenidas a partir de la aplicación del Método Directo Extendido.

Su aplicación a problemas de grandes dimensiones permite reducir las desventajas que naturalmente posee la estrategia orientada a ecuaciones. La partición de un sistema complejo permite, por un lado, disminuir los recursos computacionales necesarios para resolverlo. Por otra parte, la subdivisión del modelo facilita el análisis del proceso por parte del usuario en lo que respecta a la inicialización de variables y detección de errores entre otras tareas.

El software puede ser utilizado tanto para simulación de procesos (sistemas cuadrados) como para optimización (sistemas rectangulares). *MP4SO* funciona en dos etapas. En un primer paso, llama al Método Directo Extendido para particionar el modelo matemático, obteniendo la descomposición en bloques. En segundo lugar, *MP4SO* alimenta los bloques generados a GAMS, así como también el orden de resolución de los mismos, a fin de llevar a cabo la simulación u optimización del proceso, dependiendo del tipo de sistema en estudio. Cada uno de los subsistemas es resuelto en GAMS, haciendo uso de la gran variedad de *solvers* que dicho software posee para ello.

Se analizó el impacto de utilizar *MP4SO* sobre diferentes casos de estudio, en los que algunos de ellos fueron simulados y otros optimizados. La diversidad de acoplamiento de los modelos permitió obtener mejores o peores descomposiciones, dependiendo de si se trataba de simulaciones u optimizaciones, y del caso de estudio resuelto. En todos los casos se obtuvieron particiones del modelo original, lo cual frente a la resolución del modelo completo trae numerosas ventajas, como por ejemplo menores requerimientos computacionales, disminución en el número de variables que deben ser inicializadas y facilidades en la detección de errores del modelo.

A modo de conclusión final de esta sección, se puede afirmar que se amplió con éxito el rango de aplicación del Método Directo. Habiendo sido largamente probado en el campo de diseño de instrumentación, quedó demostrada ahora su utilidad en las áreas de simulación y optimización de procesos. La utilización de manera conjunta del generador de modelos y el software para análisis y resolución de procesos (*MP4SO*), presentados en el marco de esta Tesis Doctoral, constituyen un completo paquete para modelado, simulación y optimización de procesos. Dicho paquete combina las ventajas de la resolución secuencial modular con las

ventajas de la resolución orientada a ecuaciones, constituyendo así una herramienta de suma utilidad por su sencillez y flexibilidad de utilización

7.2 TRABAJOS FUTUROS

A futuro se vislumbran interesantes perspectivas para continuar las investigaciones realizadas en esta Tesis. Desde el punto de vista del análisis cualitativo de procesos, es necesario llevar a cabo la integración del nuevo módulo para análisis de observabilidad (Método Directo Extendido) al paquete completo que conforma el DSS para diseño de instrumentación. Actualmente se cuenta en nuestro grupo de investigación con una nueva versión del DSS, el cual fue rediseñado de manera tal de facilitar su uso, integrando los módulos que lo conforman bajo una única interfaz. En estos momentos, el único algoritmo disponible para realizar el análisis de observabilidad de los modelos es el Método Directo. Debido a que la extensión a este método presentada en el marco de esta Tesis fue ampliamente probada y su aplicación resulta claramente beneficiosa para el proceso global de diseño de instrumentación, resultaría deseable incorporar dicho algoritmo a la nueva interfaz. De esta manera, el usuario tendrá la posibilidad de seleccionar uno u otro algoritmo de descomposición, e incluso realizar comparaciones entre el desempeño de ambos para cualquier caso de estudio que se ingrese al software.

Es importante mencionar que uno de los trabajos futuros sobre el Método Directo Extendido (MDE) es la automatización en la generación de valores de pesos para el cálculo del grado de no linealidad. De momento se utiliza una tabla provista por el usuario, donde los valores siguen una regla básica de valor creciente a medida que aumenta la no linealidad, pero son sólo una decisión del usuario. Se vio en la sección 4.7.1.2 que estos pesos no afectan al número de bloques no lineales encontrados por el algoritmo, siempre que se respete el orden creciente a medida que aumenta la no linealidad. Sin embargo, pueden ciertamente afectar el tamaño de dichos bloques. Por este motivo, sería deseable contar con valores generados automáticamente dentro del mismo método en el marco del algoritmo utilizado para la minimización del grado de no linealidad. Para esto, el algoritmo debería incluir un procedimiento interno iterativo que realice diferentes descomposiciones de un mismo problema variando los pesos seleccionados, utilizando como descomposición final aquella para la cual se obtengan los bloques de ecuaciones de menor tamaño.

Otra tarea interesante es investigar el aporte del MDE como técnica de análisis numérico, comparable con las herramientas existentes. En otras palabras, evaluar la clasificación y el ordenamiento de ecuaciones que brinda el MDE como potencial herramienta matemática para el preprocesamiento de sistemas de ecuaciones grandes, no asociados necesariamente a problemas ingenieriles, de manera tal de posicionarlo adecuadamente entre los métodos generales de resolución más reconocidos.

Debido a que el diseño de instrumentación no implica la resolución matemática de sistemas de ecuaciones, se hace referencia a él como una metodología meramente estructural. Sin embargo, una tarea de tipo numérico, fundamental en el monitoreo de procesos, es la reconciliación de datos. La misma implica comparar datos obtenidos mediante mediciones en campo, con valores estimados por medio de las ecuaciones del modelo, de manera tal de detectar posibles fallas en los instrumentos de medición. Esta tarea requiere, en una de sus etapas de ejecución, la resolución matemática de los subsistemas de ecuaciones generados mediante el algoritmo de observabilidad, sin utilizar las ecuaciones que no fueron incluidas en los subsistemas (ecuaciones redundantes o no asignadas). Por este motivo, una reducción en el grado de no linealidad de la descomposición obtenida, implicaría resolver mayor cantidad de subsistemas lineales, las cuales pueden ser computadas por algún método de resolución directa. Resulta entonces, de sumo interés, realizar el testeo del MDE aplicado al campo de reconciliación de datos. De esta manera, se podría evaluar si, mejorando una de las etapas de la reconciliación de datos, se logran obtener resultados positivos sobre el funcionamiento global del proceso.

Con respecto al generador de modelos matemáticos, resultaría valioso que el software contara con una completa librería de modelos termodinámicos, del mismo modo que incluye actualmente una librería de equipos. De esta manera, se podría seleccionar, dependiendo del proceso a modelar, el paquete termodinámico adecuado, incorporándose las ecuaciones directamente desde el programa al sistema completo de ecuaciones del proceso. Actualmente el generador cuenta con las herramientas necesarias como para incorporar los modelos termodinámicos, sin embargo, dichos modelos no forman parte de la librería estándar del mismo.

Con respecto al software para resolución de modelos, se puede afirmar que *MP4SO* fue desarrollado como una herramienta para simulación y optimización de procesos en estado estacionario. Aprovechando el gran potencial que el programa posee, sería deseable extender

su aplicación a sistemas de ecuaciones diferenciales, de manera tal de poder incorporar la condición dinámica al modelado de los procesos. Debería entonces incorporarse al software un paso previo a la descomposición de los modelos, el cual consistiría en llevar a cabo la discretización de los mismos. Los problemas que incluyen sistemas diferenciales pueden ser clasificados en familias, de acuerdo a la estructura particular del sistema bajo estudio o las técnicas necesarias para llevar a cabo la optimización. Sería entonces de gran utilidad contar con un conjunto de métodos de discretización, de manera tal que cada problema que se ingrese al software pueda, en una primera etapa, ser discretizado, de manera tal de transformar el sistema de ecuaciones diferenciales en un sistema no lineal de ecuaciones algebraicas. Luego, una vez obtenido el sistema algebraico de ecuaciones, se aplicarían las etapas del software ya descritas, es decir, la partición del sistema y la posterior simulación u optimización del mismo, resolviendo el modelo bloque a bloque de acuerdo a la descomposición obtenida anteriormente.

Dado el advenimiento de los procesadores *multicore* al mercado, otra forma de aprovechar al máximo el potencial de *MP4SO* sería incorporar técnicas para resolución de sistemas en paralelo. La descomposición de un sistema de ecuaciones en subsistemas también constituye una herramienta útil cuando se desea paralelizar la resolución del mismo. La idea fundamental es que cada uno de los subsistemas resultantes de la descomposición podría ser resuelto de manera simultánea, siempre y cuando se realice un estudio acerca de la dependencia de los bloques encontrados entre sí.

Dicho estudio se lleva a cabo a través de la construcción de la correspondiente estructura de dependencias, la cual puede ser representada por medio de grafos acíclicos direccionados (GAD). En dichos grafos, cada nodo representa un bloque diagonal de la matriz previamente llevada a su forma triangular inferior en bloques, mientras que las flechas reflejan la dependencia entre los diferentes subsistemas que integran la partición. De esta manera, mediante la incorporación a *MP4SO* de un algoritmo que realice este estudio, se podría conocer para cada una de las descomposiciones realizadas, cuáles de los subsistemas pueden ser resueltos de manera simultánea en diferentes computadoras. A partir de dicha información, se podría acelerar de manera significativa la resolución del sistema completo de ecuaciones.

APÉNDICES

METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA INGENIERÍA

A.1 INTRODUCCIÓN

El desarrollo de software para ingeniería es una tarea multidisciplinaria compleja, en la cual participan tanto especialistas en programación de software como ingenieros que comprenden el aspecto físico del problema. Dicha tarea requiere tener en cuenta algunos aspectos para poder ser llevada a cabo con éxito, los cuales se detallan a continuación:

- La definición de una metodología de desarrollo de software apropiado que permita una implementación eficiente, que satisfice los requerimientos de calidad y tiempos de entrega del usuario.
- La elección de un lenguaje de implementación que soporte la metodología de desarrollo del software.

Es importante destacar que la metodología que se describe a continuación es aplicable al desarrollo de cualquier tipo de software de ingeniería. Como primera medida, para lograr familiarizarse con esta metodología se decidió trabajar en el diseño de un algoritmo de optimización de plantas de procesos, el cual fue aplicado con éxito a tres casos de estudio reales. Tanto la descripción del algoritmo como los resultados obtenidos pueden ser encontrados en detalle en Domancich y *co.*, 2004.

A.2 DEFINICIÓN DE LA METODOLOGÍA APLICADA EN LA TESIS PARA DESARROLLO DE SOFTWARE

Las metodologías tradicionales para software de ingeniería son el ciclo de vida clásico (“*classic life cycle*”), la construcción de prototipos (“*prototype building*”) y el modelo evolutivo (“*evolutive model*”). Todas ellas se basan en la definición de una serie de etapas, las cuales son llevadas a cabo en forma secuencial. En la metodología de ciclo de vida clásico, las fases siguen un modelo de cascada. Este enfoque tiende a desdibujar algunos aspectos importantes como la inevitable necesidad de iterar a través de las diferentes tareas involucradas, lo cual lleva a la corrección de errores y la modificación de los requerimientos definidos originalmente.

La metodología de construcción de prototipos consiste en desarrollar un modelo “juguete” de software como una versión de la implementación final. El modelo preliminar se enfoca en ciertos aspectos del sistema de manera tal de satisfacer lo antes posible los requerimientos del usuario con el mínimo costo posible. En este caso, el lazo de realimentación es el refinamiento del prototipo, el cual es creado como una manera de capturar todos los requerimientos del proyecto. Como resultado de esto, las incertidumbres de especificación son notablemente reducidas. Paradójicamente, el prototipo en sí mismo constituye el principal inconveniente de este enfoque, debido a que el usuario lo ve como una versión final, sin darse cuenta de que el modelo “juguete” no incluye ni los algoritmos óptimos ni la posibilidad de realizar un mantenimiento del mismo a mediano y largo plazo. Como resultado, no se dispone, por razones económicas y de plazos de entrega, de tiempo para desarrollar el software definitivo, haciendo que el prototipo incompleto se convierta en el producto final.

La metodología de modelo evolutivo está constituida por cuatro fases que son llevadas a cabo en varias oportunidades, siguiendo un camino en forma de espiral. Estas cuatro etapas son planeamiento, análisis de riesgo, ingeniería y evaluación del usuario. En la etapa de ingeniería, la técnica combina características de las otras dos metodologías, beneficiándose de sus ventajas, que son precisamente que las primeras iteraciones involucran la utilización de prototipos, mientras que la filosofía del ciclo de vida es utilizada para la implementación final. En contraste con otras metodologías, el modelo evolutivo también incluye una etapa de

análisis de riesgo que continuamente monitorea parámetros económicos y de calidad como son el costo, el tiempo de ejecución y el criterio de evaluación de su funcionamiento.

Las metodologías clásicas para desarrollo de software ingenieril no resultan apropiadas para el desarrollo del software llevado a cabo a lo largo de los años. Las metodologías de ciclo de vida y construcción de prototipos tienen las desventajas mencionadas anteriormente. En un principio, el modelo evolutivo podría haber resultado útil para nuestro propósito. No obstante, se observó que era conveniente introducir algunas modificaciones en esta metodología. Se consideró importante introducir los siguientes aspectos:

- La presencia continua del usuario final, quien asiste al programador controlando en todo momento la calidad del producto elimina la necesidad de tener una etapa de análisis de riesgo.
- Debido a que las especificaciones del usuario varían a lo largo del desarrollo del proyecto, resultaba ideal contar con prototipos en cada nivel de evolución.

Teniendo en cuenta los aspectos mencionados anteriormente, se definió una nueva metodología para desarrollo de software (Vazquez y *co.*, 2001), la cual fue aplicada para los desarrollos de software presentados en esta tesis. Dicha metodología contiene los siguientes pasos: planeamiento, ingeniería de software, evaluación del programador y evaluación del usuario. Estos pasos se repiten hasta que se logran obtener el producto final. La figura A.1 muestra un esquema de la metodología propuesta. El significado de los números ubicados dentro de los círculos se encuentra en la tabla A.1.

1	Análisis inicial de requerimientos
2	Desarrollo del prototipo inicial del software
3	Primera evaluación y revisión de los requerimientos basados en el prototipo inicial
4,8,12	Evaluación del usuario
5,9,13	Planeamiento de las mejoras a realizar al prototipo basadas en las evaluaciones del usuario y el desarrollador
6,10,14	Implementación del prototipo
7,11	Evaluación del desarrollador

Tabla A.1. Pasos que constituyen las iteraciones de la metodología aplicada.

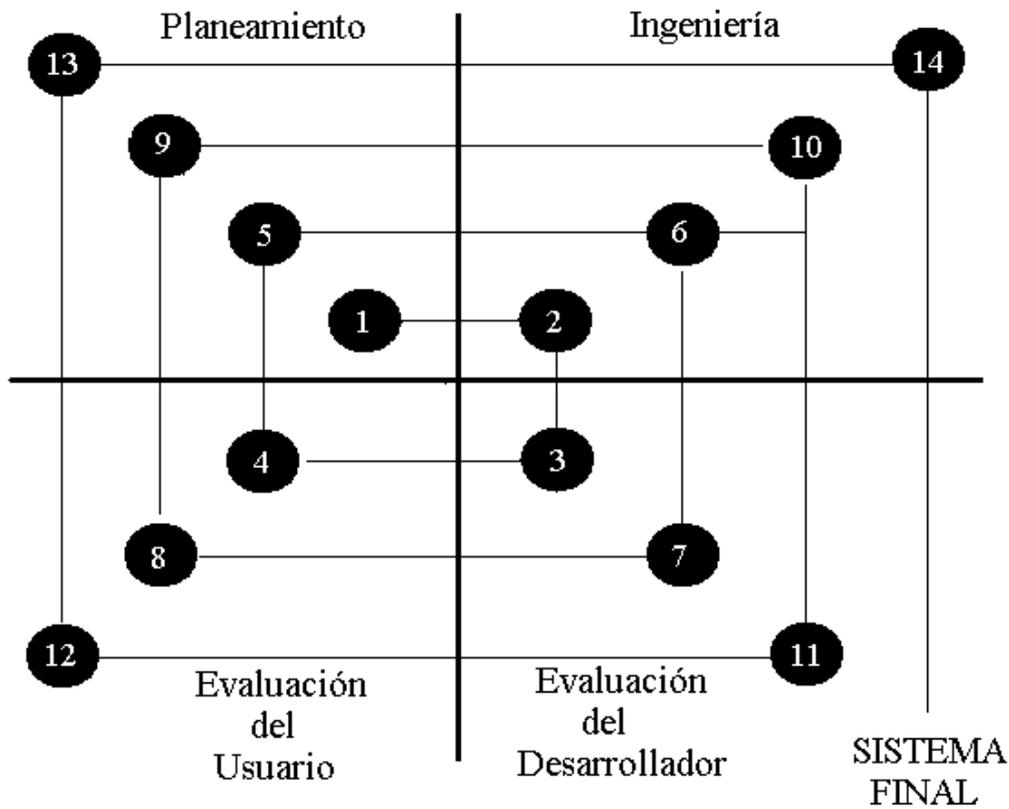


Figura A.1. Esquema de la metodología para desarrollo de software aplicada en esta tesis.

La parte fundamental de esta metodología es la etapa de ingeniería. En todos los niveles de evolución del proyecto se construye un prototipo; en este sentido la propuesta difiere del modelo evolutivo, donde la construcción del prototipo se realiza solo en las etapas iniciales. Un prototipo construido de manera temprana implica una implementación parcial o total de las funcionalidades requeridas por el software, por medio de algoritmos y/o estructuras de datos simples, prestando poca atención a su eficiencia. A medida que el proyecto evoluciona, los requerimientos de buena calidad ganan importancia. En la etapa final, cuando todas las funcionalidades fueron implementadas, la principal preocupación es la de realizar el diseño óptimo de algoritmos y estructuras de datos. Queda claro que se mantiene la idea de utilizar un prototipo para reducir las incertidumbres de especificaciones, el cual se utiliza hasta alcanzar la convergencia definitiva del producto. De manera opuesta a lo que sucede en el enfoque clásico, el último prototipo no se descarta, ya que constituye la implementación final.

Esta metodología es aplicable debido a que el prototipo progresa hacia un software que cumple con los requerimientos de calidad pero que también resulta eficiente, flexible, de fácil mantenimiento y amigable para el usuario. Para evitar una evolución caótica del software, cada iteración de la metodología incluye dos etapas: la evaluación del usuario y la del programador. La evaluación del usuario está enfocada en el análisis de los requerimientos de funcionalidad, mientras que la evaluación del programador consiste en chequear aspectos técnicos, y debe ser incluida dentro de la descripción de la metodología utilizada, de manera tal de asegurar que la calidad del prototipo mejore continuamente. Finalmente, la etapa de planeamiento le da forma al nuevo prototipo sobre la base de los resultados obtenidos en ambas evaluaciones.

***EXTENSIÓN DE LAS
FACILIDADES DE SIMULACIÓN
DE HYSYS:
INCORPORACIÓN DE MÓDULOS
DEFINIDOS BAJO
EL ESTÁNDAR CAPE-OPEN***

B.1 INTRODUCCIÓN

En el área de la ingeniería de procesos, existe una gran cantidad de modelos matemáticos representando diferentes equipos de procesos programados mediante la utilización de diferentes herramientas comerciales como son: Modelado de procesos a partir de principios básicos en lenguaje FORTRAN (FORTRAN Company, página Web, 2009), GAMS (Brooke y *co.*, 2004), gPROMS (gPROMS, página Web, 2009), etc. Con la aparición del simulador secuencial modular (HYSYS (Hyprotech, 2002), ASPEN (Aspentech, página Web, 2009)) muchos de estos modelos resultaron de poca utilidad, ya que los programas mencionados contienen una gran librería de modelos de equipos químicos, los cuales en general permiten realizar una simulación completa de cualquier planta de procesos químicos con poca necesidad de conocimiento de principios básicos de parte del usuario.

El inconveniente surge cuando alguna de las plantas que se desea simular contiene algún equipo específico (por ejemplo un reactor) donde su funcionamiento no puede ser

representado en forma certera por alguno de los modelos generales ofrecidos por los programas comerciales. Es aquí cuando se torna deseable disponer de un mecanismo por el cual uno pueda convertir los modelos específicos programados por el usuario en un “*plug-in*”, un accesorio que, como la palabra lo dice, pueda ser instalado o enchufado en algún simulador secuencial modular. De esta manera, el usuario puede aprovechar las bondades de dichos programas haciendo uso de su completa librería de modelos y paquetes termodinámicos, pero sin perder la posibilidad de utilizar para algún equipo en particular, un modelo desarrollado por él mismo.

El estándar CAPE-OPEN (CO) provee un ambiente ideal para este tipo de tareas, ya que permite transformar esos modelos de equipos programados por el usuario en módulos enchufables independientes, los cuales pueden ser fácilmente instalados en diferentes plataformas de simulación y posteriormente utilizados como un equipo más de esa plataforma. CAPE es actualmente una herramienta de integración muy utilizada en la industria moderna, la cual muchas veces se nutre de módulos específicos (modelos matemáticos de equipos, paquetes termodinámicos) desarrollados por instituciones académicas y de investigación para simular en forma rigurosa sus plantas de procesos, maximizando así sus ganancias y optimizando la seguridad y control de sus procesos (Mayer and Shoenmakers, 1998).

Este apéndice es el resumen de un trabajo publicado en el año 2009 (Domancich y *co.*, 2009) que brinda, en primer lugar, una descripción del estándar CAPE-OPEN, describiendo su utilidad, software disponible y metodología general de trabajo. Luego se tratan en detalle las estrategias de migración para dos casos diferentes: FORTRAN-HYSYS y GAMS-HYSYS. Para ello se tomaron dos casos de estudio concretos.

- Para la primera de las metodologías (migración FORTRAN-HYSYS), se utilizó un modelo matemático específico de un reactor de amoníaco, el cual se encontraba ya programado en FORTRAN. Para testear su funcionamiento, se realizó en HYSYS una simulación de la planta completa de síntesis de amoníaco a la que ese reactor pertenece, instalando allí el módulo del reactor y verificando los resultados con información real disponible.
- Para desarrollar la segunda metodología de migración (GAMS-HYSYS) se decidió seleccionar un equipo complejo como es una columna de destilación reactiva, el cual debió ser previamente modelado y simulado en GAMS (ver Apéndice D para

una descripción completa del modelo y los resultados obtenidos). Este equipo resultó de gran interés ya que debido a la combinación de procesos que se dan en su interior (reacción –separación) el modelo matemático resultante es un sistema de ecuaciones no lineales complejo y de gran tamaño. Se describe entonces la metodología a seguir para lograr generar, a partir de un equipo modelado en GAMS, un módulo enchufable en cualquier simulador secuencial modular. Para verificar su correcto funcionamiento, el módulo fue instalado y corrido en HYSYS.

B.2 PROYECTO CAPE OPEN

B.2.1 ALCANCES DEL PROYECTO CAPE-OPEN

CO-LaN, es una organización internacionalmente conocida y dirigida por los propios usuarios, cuyo propósito es facilitar la implementación de las interfaces estándares CO en software comercial para hacer de la simulación abierta una realidad (CO-LaN 2003 Leaflet). Las misiones de CO-LaN son: presentar a los vendedores de software las prioridades que los usuarios tienen para los estándares CO, asegurarse de que se dé la explotación y distribución de los estándares, proveer entrenamiento y facilidades para la realización de la migración y brindar herramientas confiables para verificar el correcto funcionamiento de los módulos desarrollados. El proyecto CAPE-OPEN se focaliza principalmente en generar herramientas para el modelado de procesos y su utilización en simulaciones de estado estacionario y dinámico (CAPE-OPEN Synthesis Report,1999). El proyecto reconoce dos tipos de herramientas: modulares y orientadas a ecuaciones.

En una arquitectura para herramientas de modelado modular de procesos existe lo que se conoce como Ejecutivo de Modelado de Procesos (PME). El PME es el responsable de la construcción del modelo y llevar a cabo todos los cálculos necesarios para su resolución. Para poder cumplir con su propósito, se comunica con otros módulos, donde cada uno de ellos representa una operación unitaria. Cada módulo necesita comunicarse con los paquetes de propiedades para calcular las propiedades físicas que ocurren dentro del modelo de la unidad de operación (UO). Debido a que cada módulo de operación unitaria necesita resolver el sistema de ecuaciones que tiene asociado, deben contener algoritmos especializados para efectuar los cálculos o bien llamar a rutinas externas de resolución. Una característica importante de las herramientas de modelado modular es la necesidad de que el Ejecutivo

organice y coordine todos los cálculos llevados a cabo por los módulos individuales de operaciones unitarias.

La estructura básica en una arquitectura para un paquete de procesos orientado a ecuaciones es la misma que la presentada anteriormente. La principal diferencia es que los módulos de operaciones unitarias no necesitan resolver sus propios sistemas de ecuaciones. En lugar de esto, lo que hacen es enviar la información al Ejecutivo, el cual ensambla toda la información en un gran sistema de ecuaciones y lo resuelve por medio de la aplicación de una o más rutinas de resolución.

La última visión de CAPE-OPEN es permitir tareas complejas de modelado de procesos y aplicaciones basadas en los modelos, las cuales son llevadas a cabo en forma exitosa y económica mediante la utilización de diversos componentes de software provenientes de varias fuentes diferentes y, de ser posible, ejecutados en diferentes computadoras. En la figura B.1 se muestra un ejemplo de esto. En este caso, el PME es proporcionado por un proveedor, mientras que los Componentes de Modelado de Procesos (CMP) provienen de diferentes proveedores.

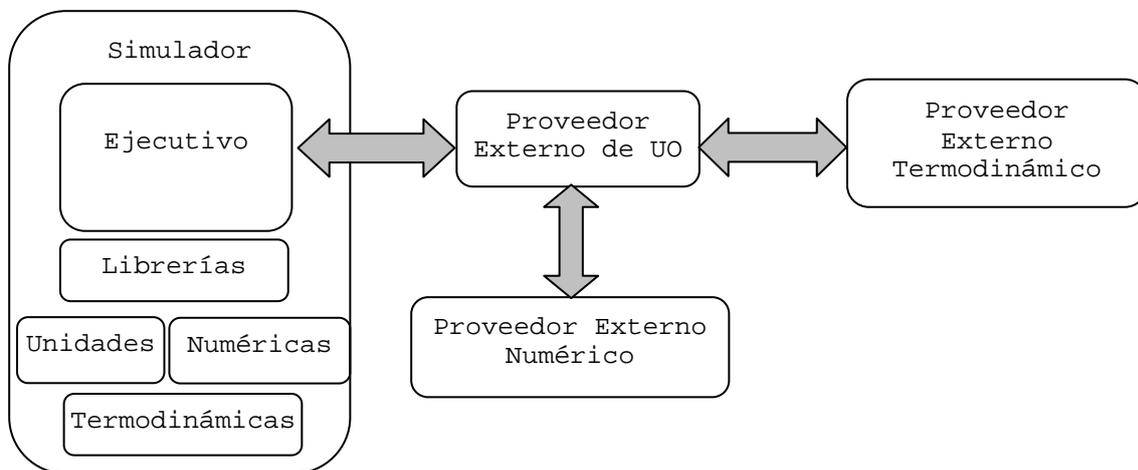


Figura B.1. Integración de las diferentes herramientas de CAPE-OPEN

Este análisis conduce a la identificación de las siguientes clases para estandarización:

- Termodinámicas: proveen servicios relacionados con las propiedades físicas.

- Módulos de Unidades de Operación.
- Numéricas: provee rutinas de resolución numérica.
- Herramientas de análisis de flowsheets.

Las interfaces mencionadas anteriormente pueden ser implementadas de diferentes maneras., por ejemplo como una subrutina llamada desde un lenguaje estándar como FORTRAN o C (CAPE-OPEN Synthesis Report, 1999). Sin embargo, CAPE-OPEN prefirió adoptar un enfoque orientado a objetos, el cual toma cada CMP como un objeto independiente. Cada objeto de software es capaz de interactuar con otros basados en una definición formal de interfaz expresada en lenguajes estándar de programación.

B.2.2 INTERFACES DE LAS UNIDADES DE OPERACIÓN

Una UO secuencial modular está constituida por (CAPE-OPEN,2001):

- **Puertos:** son necesarios para permitir al módulo conectarse a otros módulos y así intercambiar masa, energía e información. Cada puerto de masa tiene asociado a él un objeto termodinámico. Utilizando este objeto la UO es capaz de obtener la información termodinámica y de propiedades de las corrientes de entrada y salida. Estas propiedades forman parte del Sistema de Paquete de Propiedades.
- **Parámetros:** representan la información que no está asociada a los puertos de entrada pero, si embargo, resulta de interés para el usuario.
- **Interfaz de usuario:** permite al usuario configurar cada instancia del módulo de acuerdo a sus gustos y requerimientos. Esta interfaz debe ser proveída por el desarrollador y está fuera del alcance del estándar.
- **Reporte:** se encarga de presentar al usuario los resultados de los cálculos.

B.3 DESARROLLO DE EXTENSIONES DE FORTRAN A HYSYS

A continuación se describe el diseño e implementación de dos Unidades de Operación, ambas desarrolladas bajo los estándares de CO (Domancich y *co.*, 2009d), La primera de ellas corresponde a un reactor para síntesis de amoníaco, programado en FORTRAN y testado en HYSYS. La segunda aplicación es el modelo de una columna de destilación reactiva, la cual fue programada en GAMS y cuyo módulo también fue testado en HYSYS.

|

B.3.1 ESTRATEGIA DE MIGRACIÓN DE MODELOS DE FORTRAN A HYSYS

Existen dos caminos para llevar a cabo un proceso de migración.: la re-implementación del código original desde cero y el “encapsulado” del código existente (Köller, 1999):

- **Re-implementación del código original:** parece ser la manera más sencilla de llevar a cabo la migración. Sin embargo, se debe tener mucho cuidado a la hora de crear el nuevo sistema. A pesar de que el código original puede haber sido testeado y utilizado en reiteradas oportunidades, el software creado desde cero puede contener gran cantidad de errores, dando como resultado errores al intentar ejecutar el módulo, o inconsistencias entre el comportamiento original del modelo y la nueva implementación. Encontrar y corregir estos errores puede resultar una tarea ardua y que requiera mucho tiempo, por este motivo esta estrategia puede resultar en definitiva más peligrosa y costosa de lo que parece en un principio.
- **Encapsulado del código original:** consiste en realizar la migración del código sin realizar cambios en el código fuente. En este caso, el código escrito en FORTRAN es tratado como una “caja negra” cuya funcionalidad es utilizada, pero no resulta de interés cómo es implementada. Para se provee una cápsula orientada a objetos alrededor de esa caja negra de manera tal de lograr integrar el código programado en FORTRAN al módulo que se quiere desarrollar. Esta cápsula se genera por medio de la utilización de un lenguaje orientado a objetos como Java o C++.

Mantener el código fuente intacto es una manera de evitar la introducción de nuevos errores al sistema. Otra ventaja de esta estrategia es que no es necesario optimizar el funcionamiento del módulo, ya que esta tarea ha sido realizada por desarrollador original del modelo. Teniendo en cuenta estas ventajas, fue este el método por el que se optó para llevar a cabo el desarrollo del módulo del reactor de amoníaco.

En el proyecto Global de CAPE-OPEN, se desarrolló una gran cantidad de herramientas de manera tal de brindar soporte en la implementación de los componentes basados en los estándares CO (Köller y Töbermann, 2002). Todas estas herramientas producen el esqueleto del código fuente para los componentes CO, evitando que el desarrollador deba implementar una serie de pasos que resulta repetitivo para cada uno de los módulos que se deban desarrollar. La herramienta utilizada para el desarrollo de este proyecto se encarga de la generación mecánica del Paquete de UO. Genera un proyecto en Visual Basic

que contiene el código fuente de la UO y provee el paquete de instalación para permitir al usuario instalar el módulo en cualquier computadora. Tanto el código que genera la rutina de cálculo como el de la interfaz gráfica de la unidad deben ser provistos por el desarrollador, ya que no son generados por la herramienta de CO.

Como se explicó anteriormente, el proceso de migración requiere la integración de diferentes partes de código, las cuales en general se encuentran desarrolladas bajo diversos lenguajes de programación. Cada uno de estos lenguajes posee su propia estructura y tipo de datos, así como también diferentes convenciones acerca del llamado a procedimientos, subrutinas o funciones (Köller y Töbermann, 2002). Para sobrellevar estas diferencias, es necesaria la utilización de una tecnología de puente. Por medio de la misma, un componente de un determinado software (un ejecutable, una librería o una parte del código fuente) es utilizado para llamar a una rutina que reside en una librería compilada en el nuevo código fuente. Esta metodología permite separar claramente el código nuevo del viejo, facilitando entonces el mantenimiento de ambos subsistemas. Para la realización de este proyecto se utilizó una de estas tecnologías, la cual es conocida como “Librería de enlace dinámica” (DLL). Se define como DLL a los archivos con código ejecutable que se ejecutan cuando son llamados desde alguna aplicación en particular u otra DLL. Las ventajas del uso son:

- **Reducen el tamaño de los archivos ejecutables:** gran parte del código puede estar almacenado en bibliotecas y no en el propio ejecutable.
- **Pueden estar compartidas entre varias aplicaciones:** si el código es suficientemente genérico, puede resultar de utilidad para múltiples aplicaciones.
- **Facilitan la gestión y aprovechamiento de la memoria del sistema:** la carga dinámica permite al sistema operativo aplicar algoritmos que mejoren el rendimiento del sistema cuando se carguen estas bibliotecas.
- **Brindan mayor flexibilidad frente a cambios:** mejora el rendimiento o soluciona errores distribuyendo únicamente una nueva versión de la biblioteca dinámica. Esta mejora es aprovechada por todas las aplicaciones que compartan la misma DLL.

Para este proyecto, la creación de una DLL permite llamar a una subrutina de FORTRAN desde Visual Basic, sin la necesidad de incorporar código adicional. La figura B.2 presenta un esquema básico del proceso descrito. En la misma, se pueden ver las distintas

capas que integran el proceso de migración. En primer lugar, el usuario debe desarrollar o contar con algún modelo matemático ya implementado en FORTRAN. En segundo lugar, se genera la DLL que encapsula al código original, la cual está integrada por código ejecutable del modelo de FORTRAN. Por último se genera, con la herramienta de CO el esqueleto de la UO, el cual contiene código fuente general para cualquier UO. Dicho esqueleto llamará a la DLL las veces que sea necesario para lograr el correcto funcionamiento del nuevo módulo.

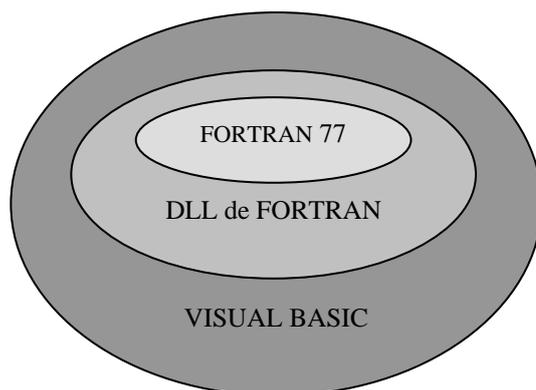


Figura B.2. Proceso de encapsulado del código de FORTRAN

B.3.2 CASO DE ESTUDIO: PLANTA DE AMONÍACO

El reactor utilizado como caso de estudio es una unidad flujo-pistón, isotérmico y con conversión fija (Byke and Grossman, 1985). La reacción de síntesis de amoníaco se produce sobre un catalizador de hierro y es descrita a continuación:



El reactor debe ser enfriado para evitar que la temperatura del catalizador alcance su temperatura de desactivación, es decir, 800K. Para cumplir esto, una temperatura de entrada de 703K y un coeficiente de transferencia de calor de 19644 KJ/h.K son suficientes.

La conversión en el reactor varía directamente con el volumen de catalizador y, para un dado volumen de catalizador, existe un valor de coeficiente de transferencia de calor (UA) para el cual la conversión obtenida resulta máxima. El valor de UA determina la velocidad a la cual puede ser extraído el calor del equipo. Para valores bajos de UA, el calor generado dentro del reactor no extraído en suficiente cantidad, haciendo que la temperatura dentro de la unidad se incremente rápidamente. Esto provoca la aparición de “puntos calientes” cerca de la alimentación, desactivando gran parte del catalizador. Como resultado de esto, la conversión

decrece notoriamente. Para valores muy altos de UA, la velocidad a la que se remueve el calor es muy alta, disminuyendo la velocidad a la que ocurre la reacción.

B.3.3 RESULTADOS

Una vez que el código de FORTRAN fue probado y encapsulado en una DLL, se creó la Unidad de Operación . Esta UO presenta las siguientes características:

- Posee dos puertos de materia disponibles, INPORT (puerto de entrada) y OUTPORT (puerto de salida). La corriente conectada al puerto INPORT tiene la información del flujo de entrada, el cual contiene hidrógeno, amoníaco, argon, nitrógeno y agua. También contiene información de la temperatura y presión de la corriente de entrada. Por otro lado, la corriente que sale del puerto OUTPORT contiene información de caudales, temperatura y presión a la salida del reactor, pero con la diferencia de que estos valores son calculados por la UO.
- El equipo también incluye una serie de parámetros (volumen de catalizador, dimensiones del reactor, “rebanadas” en las que se dividirá la unidad para efectuar la integración numérica, etc) que deben ser especificados por el usuario para poder llevar a cabo los cálculos correspondientes.

Un detalle a tener en cuenta es el de las unidades a utilizar en el proyecto. Debido a que CO especifica cierto tipo de unidades para definir la temperatura, presión, caudales, etc, y dado que el código original de FORTRAN utilizaba otras unidades, fue necesario agregar una subrutina de conversión de unidades dentro de la rutina de cálculo de la UO. Otro punto con el que se debió tener especial cuidado es con el tema de la nomenclatura de los componentes, debiendo estos ser escritos de manera tal de que coincidieran con los que se incluyen en el estándar (CAPE-OPEN - Thermodynamics and Physical Properties document, 2002).

B.3.3.1 CÓDIGO EN VISUAL BASIC PARA EL REACTOR DE AMONÍACO

En la figura B.3 se incluye el núcleo del pseudo código desarrollado para llevar a cabo la rutina de cálculo de la UO del reactor estudiado. El mismo relaciona FORTRAN con HYSYS.

```
Private Sub ICapeUnit_Calculate()  
  
'Declare private variables  
  
...  
  
'Gets connected ports  
  
Set PORTIN = GetPort("PORTIN")  
  
Set PORTOUT = GetPort("PORTOUT")  
  
'Gets material objects associated to the ports  
  
Set materialIn = PORTIN.connectedObject  
  
Set materialOut = PORTOUT.connectedObject  
  
'Gets components ids and number of components  
  
Components = materialIn.ComponentIds  
  
NumComponents = materialIn.GetNumComponents  
  
'Gets the total flow and partial flows of the input  
  
flowIN = materialIn.GetProp("flow", "Overall", Components, vbNullString, "mole")  
totalIN = materialIn.GetProp("totalFlow", "Overall", Empty, vbNullString, "mole")  
  
Gets the temperature and pressure of the input  
  
T = materialIn.GetProp("temperature", "Overall", Empty, vbNullString, Empty)  
P = materialIn.GetProp("pressure", "Overall", Empty, vbNullString, Empty)  
  
'Gets the values of the parameters  
  
...  
  
'Calculates input parameters for the DLL routine  
  
...  
  
'Calls the DLL routine
```

Figura B.3. Código desarrollado en Visual Basic para implementar la Unidad de Operación correspondiente al reactor de amoníaco.

B.3.3.2 REACTOR DE AMONÍACO PROBADO EN HYSYS

Para verificar de manera estricta, tanto el funcionamiento del equipo, como su correcta interacción con otros equipos dentro de una simulación, se decidió realizar en HYSYS la simulación de la planta de síntesis de amoníaco diseñada por Byke y Grossmann (1985). Esta fue realizada utilizando, en principio, un reactor que el mismo simulador contiene en su librería de equipos. Debido a que el modelo difiere de aquel propuesto por Bike y Grossman, los resultados obtenidos no son los mismos en la simulación que el caso de estudio publicado.

Una vez que se logró hacer funcionar la simulación de la planta, se realizó la instalación de la UO en HYSYS, de manera tal de que la misma sea incorporada al software como una extensión dentro de la lista de Unidades de Operación (figura B.4)

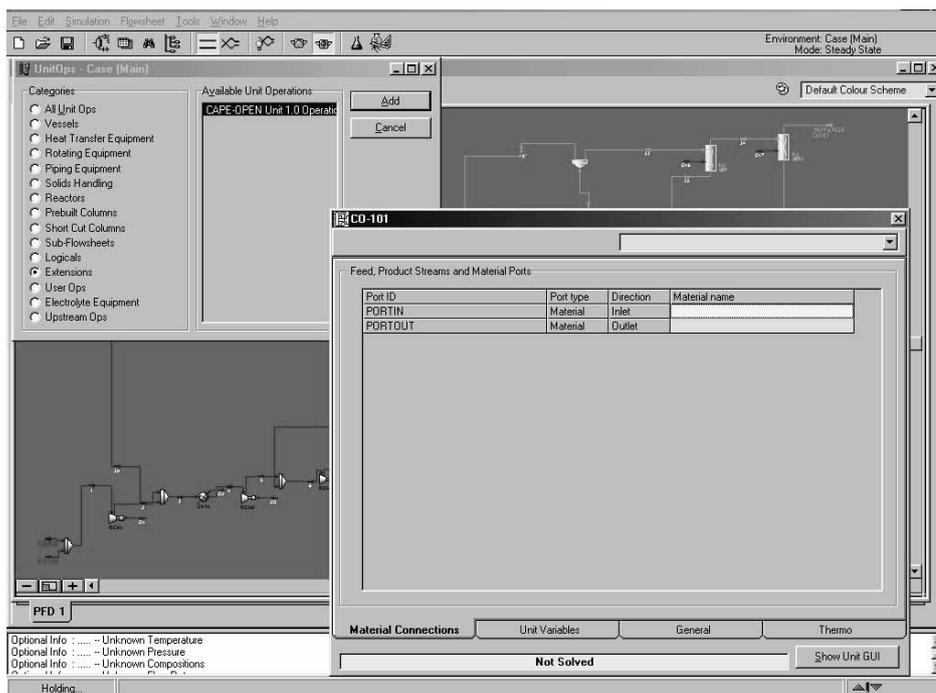


Figura B.4. Unidad de Operación CO disponible en HYSYS

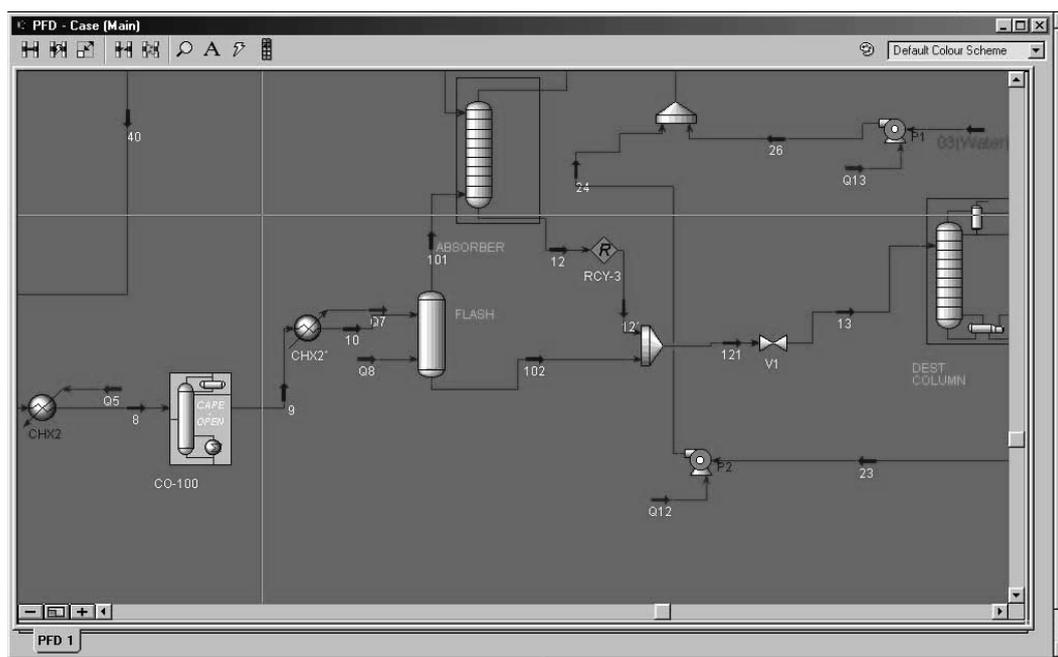


Figura B.5. Unidad de Operación CO incorporada a la simulación de la planta

Una vez incorporada la UO a la lista de equipos de HYSYS, se reemplazó el reactor utilizado inicialmente para realizar la simulación por el equipo incorporado, simulando nuevamente la planta completa (figura B.5). Los resultados obtenidos fueron similares a los publicados por Bike y Grossman, verificando así el correcto funcionamiento de la unidad CO dentro de la simulación completa de un proceso químico.

B.4 DESARROLLO DE EXTENSIONES DE GAMS A HYSYS

A continuación se describe la estrategia utilizada para llevar a cabo el desarrollo de un módulo que cumple con los estándares CO, a partir del modelo de una columna de destilación reactiva desarrollado y testeado previamente en GAMS. Al igual que en el inciso B.3, el módulo desarrollado fue instalado en HYSYS y su correcto funcionamiento fue verificado mediante la creación de un caso de estudio en ese simulador comercial.

B.4.1 ESTRATEGIA DE MIGRACIÓN DE MODELOS DE GAMS A HYSYS

B.4.1.1 DESARROLLO DE MÓDULOS COMPATIBLES CON CAPE-OPEN

Para llevar a cabo el desarrollo de un módulo compatible con CAPE-OPEN para una columna de destilación reactiva, se utilizó nuevamente la herramienta para Unidades de Operación desarrollado por el Proyecto CO (Köller and Töbermann, 2002). Como se vio

anteriormente, la misma genera un proyecto en Visual Basic que contiene el código fuente de la UO y el paquete de instalación para ser instalado en otras computadoras.

El primer paso para el desarrollo del módulo fue el modelado y programación de la columna en GAMS. Para el correcto funcionamiento del módulo, es imprescindible realizar una verificación rigurosa del correcto funcionamiento del modelo en forma independiente, tarea que fue realizada satisfactoriamente (Domancich y *co.*,2007; Domancich y *co.*,2009a) y descrita en detalle en el capítulo 5 de esta Tesis.

Debido a que el código fuente es generado por CO en lenguaje Visual Basic, se hizo necesario realizar una migración del mismo, de manera tal de integrar diferentes partes de código, implementadas en diversos lenguajes de programación. Para esto se debieron realizar al modelo de GAMS alguna modificaciones, permitiendo la obtención de datos para realizar la simulación desde una fuente externa, y a su vez transfiriendo resultados obtenidos a partir de la ejecución de la misma. Con esta finalidad, se hizo uso de las propiedades de los archivos con extensión GDX (*GAMS Data Exchange*). Un archivo GDX tiene como función grabar los valores de uno o más símbolos como son sets, parámetros, variables y ecuaciones.

GAMS posee una herramienta llamada GDXXRW, la cual tiene como función permitir la lectura de información desde un archivo de Excel, o la escritura en el mismo. GDXXRW tiene la posibilidad de leer múltiples hojas de un archivo Excel y escribir los datos en un archivo GDX. Por otro lado, también es posible recorrer el camino contrario. El código de la UO generado crea entonces un archivo Excel con todos los datos necesarios y llama entonces a la rutina GDXXRW antes de resolver el modelo en GAMS, el cual al ser ejecutado utiliza como datos de entrada los que lee del archivo de Excel. Una vez que el modelo es resuelto, los resultados son grabados en un nuevo archivo GDX, el cual puede ser leído por el código Visual Basic de la Unidad de Operación. En la figura B.6 se incluye un esquema del proceso de lectura y transferencia de datos desde Visual Basic a GAMS.

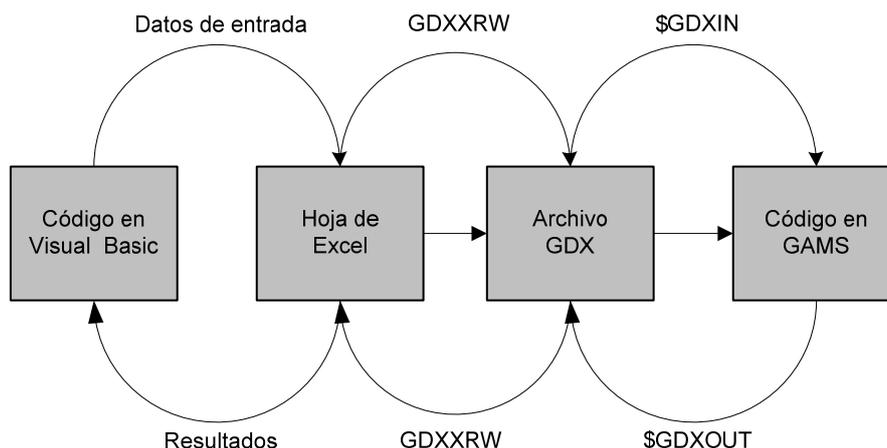


Figura B.6. Unidad de Operación CO incorporada a la simulación de la planta

En resumen, la UO de HYSYS, la cual fue programa en Visual Basic, llama a GAMS, donde el modelo matemático debe ser previamente introducido y testeado. Es importante resaltar que GAMS es el encargado de llevar a cabo la resolución matemática del modelo, y envía los resultados de la simulación nuevamente a HYSYS. Estos resultados son utilizados por HYSYS como entrada a cualquier otro equipo que forme parte del proceso industrial, de manera tal de continuar con la simulación completa de la planta.

B.4.1.2 DEFINICIÓN DE LOS PASOS A SEGUIR PARA DESARROLLAR MÓDULOS COMPATIBLES CON CAPE OPEN A PARTIR DE MODELOS DESARROLLADOS EN GAMS

Para desarrollar una Unidad de Operación compatible con CO a partir de un modelo programado en GAMS, es necesario realizar algunas modificaciones al código original de GAMS. Una de ellas es lograr que todos los datos definidos por el usuario sean leídos por el modelo desde una fuente externa, siguiendo la metodología explicada anteriormente. Se creó entonces un archivo de Excel con todos los datos necesarios para llevar a cabo la simulación del modelo, generando luego un archivo GDX utilizando la herramienta GDXXRW. El código original de GAMS debe incluir las siguientes sentencias:

`$GDXIN datain.gdx` (carga el archivo GDX)

`SET VAR1 (*)` (define una variable)

`$LOAD VAR1` (carga el valor de la variable definida desde una fuente externa)

Una vez que el código de GAMS fue testeado con las modificaciones impuestas, el paso siguiente fue el desarrollo de la Unidad de Operación CO mediante la utilización de la

herramienta provista por CO. El código de Visual Basic generado fue modificado de manera tal de crear el archivo Excel con los datos de entrada y realizar la llamada a GAMS para llevar a cabo la resolución matemática del modelo. El código debe incluir las siguientes sentencias:

```
x = CreateProcessA(0&, "gams.exe model.gms", 0&, 0&, 1&,
NORMAL_PRIORITY_CLASS, 0&, 0&, NameStart, NameOfProc)
( llamada al modelo de GAMS )
x = WaitForSingleObject(NameOfProc.hProcess, INFINITE)
x = CloseHandle(NameOfProc.hProcess)
```

En resumen, los pasos a seguir para desarrollar un modulo instalable en HYSYS a partir de un modelo matemático desarrollado en GAMS son:

- A) *Desarrollar y testear en GAMS el modelo matemático..*
- B) *Utilizar la herramienta de CO para Unidades de Operación para generar un nuevo proyecto en Visual Basic que contenga:*
 - el código fuente de la Unidad de Operación*
 - el paquete de instalación necesario para llevar a cabo la instalación del mismo en cualquier máquina.*
- C) *Modificar el código original de GAMS de manera tal de lograr obtener los datos para llevar a cabo la simulación del modelo desde una fuente externa. Para lograr esto debe ser utilizada la función GDX provista por GAMS.*
- D) *La función GDX de GAMS provee una utilidad llamada GDXXRW que permite la lectura y escritura de datos de una hoja de datos contenida en un archivo Excel.*

B.4.2 CASO DE ESTUDIO: COLUMNA DE DESTILACIÓN REACTIVA

Para llevar a cabo el testeo del procedimiento detallado anteriormente, se decidió hacer uso del modelo matemático perteneciente a una columna de destilación reactiva, el cual había sido previamente desarrollado y testado en GAMS (Apéndice D).

B.4.3 RESULTADOS

Una vez que el código de Visual Basic fue desarrollado y testado, la UO fue instalada en HYSYS, siendo disponible para su utilización como una Unidad Externa. La figura B.7

esquematiza cómo se realiza el ciclo de transferencia de datos entre GAMS y HYSYS una vez instalado el módulo.

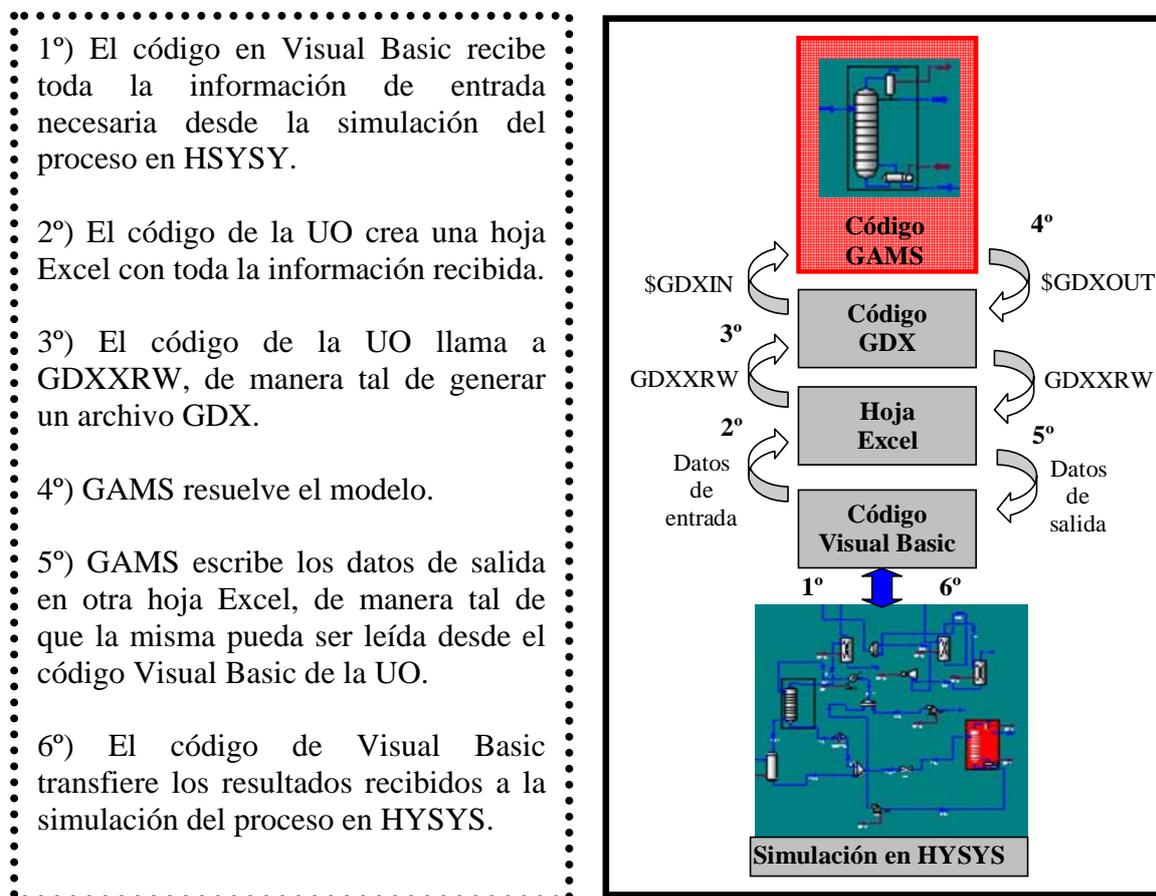


Figura B.7. Ciclo de transferencia de datos entre GAMS y HYSYS

En resumen, la UO de HYSYS, programada en Visual Basic, llama a GAMS, donde el modelo matemático debe previamente haber sido desarrollado y testeado. GAMS es el encargado de llevar a cabo la resolución matemática del modelo, y manda luego los resultados a HYSYS. Estos resultados son empleados por HYSYS como entrada a cualquier otro equipo que forme parte del mismo proceso, de manera tal de continuar con la simulación de la planta.

B.4.3.1 CÓDIGO DE VISUAL BASIC PARA LA COLUMNA REACTIVA

En la figura B.8 se incluye la parte más relevante del pseudo código desarrollado en Visual Basic para la rutina que relaciona el modelo desarrollado en GAMS con la simulación implementada en HYSYS para la UO de la columna reactiva.

```
// Proceso de inicialización de GAMS

NameStart.lpTitle = "VB GAMS"

NameStart.dwFlags = STARTF_USESHOWWINDOW

NameStart.wShowWindow = SW_SHOWDEFAULT

NameStart.cb = Len(NameStart)

// Creación del archivo datain.gdx

x = CreateProcessA(0&, "C:\Program Files\GAMS21.5\GDXXRW datain.xls

set=i

rng=etapas!B1:R1 cdim=1 par=etapas rng=etapas!A1:R5 cdim=1 rdim=1

set=val

rng=valores!A1:R1 cdim=1 par=valores rng=valores!A1:R2 cdim=1 par=portin

rng=portin!A1:J5 rdim=1 cdim=1 set=j rng=portin!A2:A5 rdim=1 par=aw

rng=a!A1:E5 cdim=1 rdim=1 par=bw rng=b!A1:E5 cdim=1 rdim=1 par=antoine

rng=antoine!A1:D5 cdim=1 rdim=1 par=cp rng=cp!A1:E5 cdim=1 rdim=1", 0&,

0&,

1&, NORMAL_PRIORITY_CLASS, 0&, 0&, NameStart, NameOfProc)

If x <> 1 Then

MsgBox "CreateProcess GDXXRW in failed. Error: " &

Err.LastDllError

End If

x = WaitForSingleObject(NameOfProc.hProcess, INFINITE)

x = CloseHandle(NameOfProc.hProcess)
```

```
.....  
  
// Creación del archive dataout.xls  
  
x = CreateProcessA(0&, "C:\Program Files\GAMS21.5\GDXXRW dataout.gdx  
  
var=x  
  
rng=x! var=y rng=y! var=l rng=l! var=v rng=v! var=FD rng=FD! var=T rng=T!  
  
var=rx rng=rx! var=Qcond rng=Qcond! var=Qreb rng=Qreb! var=conversion  
  
rng=conversion! ", 0&, 0&, 1&, NORMAL_PRIORITY_CLASS, 0&, 0&,  
  
NameStart,  
  
NameOfProc)  
  
If x <> 1 Then  
  
MsgBox "CreateProcess GDXXRW out failed. Error: " &
```

Figura B.8. Código de Visual Basic para la implementación de la Unidad de Operaciones perteneciente a la columna de destilación reactiva

B.5 CONCLUSIONES

En este apéndice se presentaron los conceptos y herramientas primarias que deberían ser utilizadas a la hora de desarrollar un módulo compatible con CAPE-OPEN. Las propuestas de desarrollo de módulos presentadas en este apéndice son útiles para la industria, donde simuladores comerciales como HYSYS se han vuelto de suma utilidad. Para propósitos de simulación, HYSYS provee una gran librería de modelos generales. En contraste, hay modelos específicos, utilizados en plantas particulares que el ingeniero de procesos debería incluir en su simulación para llevar a cabo una representación correcta del proceso en estudio. Gracias a la metodología desarrollada, el ingeniero puede aprovechar sus modelos específicos, los cuales seguramente demandaron grandes esfuerzos para ser desarrollados.

Se desarrollaron metodologías para incorporar dos unidades diferentes definidas por el usuario. La primera de ellas fue modelada y simulada utilizando FORTRAN, mientras que la segunda de las unidades fue implementada en GAMS. Siguiendo la metodología presentada en este apéndice, es posible lograr que cualquier modelo desarrollado por medio de ambos programas resulte compatible con HYSYS. Más aún, a través de esta metodología se logra sacar provecho de los completos paquetes de propiedades químicas y termodinámicas que posee HYSYS, los cuales no se encuentran presentes en los lenguajes de programación mencionados. Esto permite al usuario utilizar datos precisos de propiedades químicas y de equilibrio, los cuales son directamente calculados por HYSYS, evitando así el trabajo de incorporar al modelo matemático del equipo ecuaciones de termodinámica, correlaciones y datos de las propiedades de los compuestos.

Desde un punto de vista académico, CO permite nuevos desarrollos teóricos, los cuales pueden ser implementados y testeados mediante la utilización de componentes de software calificado. Los nuevos módulos son capaces de utilizar componentes externos, como por ejemplo bases de datos de un simulador con propiedades químicas y termodinámicas. Dichos componentes pueden provenir de diferentes fuentes, mejorando así la calidad del producto final y reduciendo tiempo y costos de implementación.

Desde un punto de vista comercial, CO permite el desarrollo competitivo de productos. Las compañías que desarrollan componentes de software para industrias de procesos pueden expandir el número de aplicaciones donde sus productos pueden ser utilizados. Adicionalmente, dichas compañías e instituciones académicas y de investigación tendrán la posibilidad de desarrollar componentes y luego transferirlos a la industria de procesos de una manera más cómoda y directa.

El desarrollo de software compatible con CAPE-OPEN involucra principalmente la interacción de dos disciplinas: Ciencias de la Computación e Ingeniería de Procesos. Esta complementación de conocimientos es necesaria, dado que por un lado, es necesario tener un importante conocimiento acerca de manejo de software (DLLs, programación orientada a objetos, etc), y por otro, es fundamental tener un conocimiento profundo del equipo que va a ser modelado y el proceso que el mismo va a llevar a cabo. Por estos motivos y la diversidad de casos existentes, debe resaltarse que el proceso de migración es una tarea de cierta

complejidad, siendo así prácticamente infactible llevar a cabo una automatización general de los procedimientos detallados en este apéndice.

***INCORPORACIÓN DE EQUIPOS
AL GENERADOR DE MODELOS
POR MEDIO DE SU DEFINICIÓN
EN LENGUAJE XML***

C.1 INTRODUCCIÓN

En el capítulo 5 de esta Tesis se presentó un software para la generación de modelos matemáticos de plantas de procesos. Dicha herramienta permite obtener de manera automática el sistema de ecuaciones que representa el comportamiento de una determinada planta, a partir del ingreso al software de la topología de la misma.

Como se detalló anteriormente (ver sección 5.4), entre las características principales del software se encuentra su flexibilidad en las opciones de modelado, lo cual facilita la incorporación de nuevos modelos de equipos y ecuaciones termodinámicas. Esta cualidad resulta de suma importancia, ya que permite que el programa sea de uso general y adaptable a cualquier tipo de proceso que quiera ser estudiado.

Para lograr la flexibilidad deseada se diseñó el generador de modelos de manera tal que trabaje con un enfoque orientado a objetos, donde el código fuente, el cual representa el esqueleto principal de la herramienta, resulta independiente de los equipos que forman parte del software. Dichos equipos pueden ser modificados o incorporados al programa con facilidad, sin que el usuario tenga la necesidad de comprender el lenguaje de programación y la sintaxis con la que se encuentra desarrollado el código fuente principal. Para lograr esta independencia se utilizó el lenguaje de programación conocido como XML (por sus siglas en inglés: Lenguaje Extensible de Marcas), el cual se describe a continuación.

C.2 LENGUAJE XML PARA INCORPORACIÓN DE EQUIPOS

C.2.1 INTRODUCCIÓN AL LENGUAJE XML

XML (Raik, E, 2001; Harold, E. y Means. W., 2004) es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados. Puede ser visto como un metalenguaje, dado que con él cada usuario puede definir su propio lenguaje de presentación. La particularidad más importante de XML es que no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea a su gusto, dependiendo del contenido del documento. De esta forma, si por ejemplo se está trabajando en el desarrollo de un software para inventariado de librerías, el documento XML a desarrollar contendrá información sobre libros, incluyendo entonces etiquetas como <AUTOR>, <EDITORIAL>, <Nº_DE_PÁGINAS>, <PRECIO>, etc. Por otro lado, un documento XML relacionado con educación incluirá etiquetas del tipo de <ASIGNATURA>, <ALUMNO>, <CURSO>, <NOTA>, etc.

El lenguaje XML permite definir un conjunto ilimitado de etiquetas, proporcionando un marco de trabajo para etiquetado de datos estructurados. Un elemento de XML puede declarar que sus datos asociados sean el precio de venta al público, un impuesto de venta, el título de un libro o cualquier otro elemento de datos deseado. Proporciona además una representación estructural de los datos que ha probado ser ampliamente implementable y fácil de distribuir. Diferentes implementaciones industriales han demostrado la calidad intrínseca y la fortaleza industrial del formato de datos con estructura de árbol del XML.

C.2.2 CARACTERÍSTICAS DEL LENGUAJE

A continuación se describen las principales características del lenguaje XML:

- Resulta fácilmente extensible. Luego de ser diseñado e implementado, es posible extender su utilización mediante la adición de nuevas etiquetas, de modo de poder continuar utilizándolo sin inconvenientes.
- Resulta sencillo para un tercero comprender y procesar la estructura de un archivo en lenguaje XML. De esta manera, se mejora la compatibilidad entre diferentes aplicaciones.

- Contiene un analizador estándar, compatible con cualquier versión de XML. Esto posibilita el empleo de cualquiera de las versiones disponibles, permitiendo evitar errores y acelerando el desarrollo de aplicaciones.

C.2.3 DOCUMENTOS XML BIEN FORMADOS

Se denomina documentos XML “bien formados” a aquellos que cumplen con todas las definiciones básicas de formato, pudiendo ser analizados de manera correcta por cualquiera de los analizadores sintácticos disponibles para el lenguaje. Un documento “bien formado” posee las siguientes características:

- Los documentos deben seguir una estructura estrictamente jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente incluida en otra, es decir, las etiquetas deben estar correctamente anidadas. Los elementos con contenido deben estar correctamente cerrados.
- Los documentos XML sólo permiten un elemento raíz del que todos los demás sean parte, es decir, solo pueden tener un elemento inicial.
- Los valores atributos en XML siempre deben estar encerrados entre comillas simples o dobles.
- El XML es sensible a mayúsculas y minúsculas. Existe un conjunto de caracteres llamados espacios en blanco (espacios, tabuladores, retornos de carro, saltos de línea) que los procesadores XML tratan de forma diferente en el marcado XML.
- Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan marcas; son partes del documento que el procesador XML espera entender. El resto del documento entre marcas son los datos "entendibles" por las personas.

C.2.4 APLICACIÓN DEL LENGUAJE XML PARA LA INCORPORACIÓN DE EQUIPOS AL GENERADOR AUTOMÁTICO DE MODELOS

El desarrollo de una aplicación que permita definir equipos de manera genérica permite lograr una independencia entre el usuario y el desarrollador del código principal del programa. Para alcanzar este objetivo, se utilizaron para el diseño del generador de modelos

las herramientas que los lenguajes de marcas proveen, y más específicamente, las que provee el lenguaje XML.

XML define un modelo de conexiones entre objetos, los cuales se relacionan entre sí siempre y cuando uno de ellos contenga a los otros. Aplicando este concepto a una planta química, se plantearon los diferentes objetos que puede contener la planta completa, de manera tal de poder desarrollar un código XML con esta información. Rápidamente, se pueden identificar como objetos dentro de ella las corrientes de entrada y salida, las ecuaciones que modelan el comportamiento de cada uno de los equipos que forman parte de la misma, las variables y parámetros que forman parte de dichas ecuaciones (presión, temperatura, caudal, caída de presión de un equipo, etc), una imagen que identifica cada uno de los equipos y corrientes que integran el flowsheet, y lo más trivial, un nombre para cada equipo y corriente que forme parte del proceso.

Para relacionar el lugar donde una corriente de entrada o salida se posiciona en la imagen que identifica al equipo, cada una tiene una coordenada local (x,y) que denota el punto dentro de la imagen donde se dibujará el enlace entre una corriente y el equipo. En el caso del sistema de ecuaciones el análisis ya no es tan simple, pero a grandes rasgos se pueden identificar variables, literales (constantes o simplemente un paréntesis), y sumatorias. La figura C.1 resume lo explicado anteriormente mediante la utilización de un diagrama de relaciones. En ella se pueden visualizar las relaciones existentes entre las diferentes etiquetas que definen un determinado equipo.

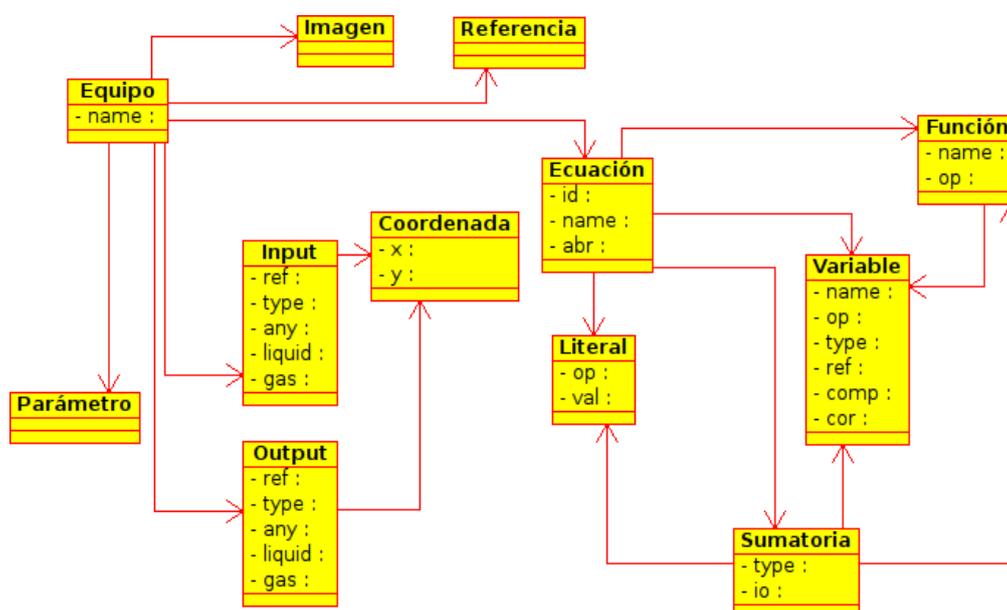


Figura C.1. Diagrama de relaciones entre las etiquetas que conforman un equipo

Como se explicó anteriormente, el lenguaje XML permite al usuario definir los elementos acorde al problema que se trate. Para nuestra aplicación en particular, los equipos fueron diseñados conteniendo la siguiente estructura:

```

<equipo repeat="[true|false]">
  <nombre>[cadena]</nombre>
  <reference>[cadena]</reference>
  <img>[cadena]</img>
  <entradas>
    <entrada ref="[cadena]" type="[mat|energ]" finitas="true"
any="[true|false]" liquid="[true|false]" gas="[true|false]">
      <x></x>
      <y></y>
    </entrada>
    ...
    <entrada ref="[cadena]" type="[mat|energ]"
finitas="[true|false]" any="[true|false]" liquid="[true|false]"
gas="[true|false]">
      <x></x>
      <y></y>
      <x></x>
      <y></y>
    </entrada>
  </entradas>
  <salidas>
    <salida ref="[cadena]" type="[mat|energ]" finitas="true"
any="[true|false]" liquid="[true|false]" gas="[true|false]">
      <x></x>
      <y></y>
    </salida>
    ...
    <salida ref="[cadena]" type="[mat|energ]"
finitas="[true|false]" any="[true|false]" liquid="[true|false]"
gas="[true|false]">
      <x></x>
      <y></y>
      <x></x>
      <y></y>
    </salida>
  </salidas>
  <ecuacion id="1" name="Work" abr="WK">
    <lit op="[pos|neg|div|mul|cadena]" val="[cadena]" />
    <var name="[cadena]" op="[pos|neg|div|mul|cadena]"
type="[eq|cio]" ref="[cadena]" comp="[false|compIn|compOut]"
cor="[false|compIn|compOut]" infbyref="[true|false]" />
    <sumat type="[cor|comp]" io="[in|out]">
      <var ... />
      <lit ... />
      <func ...>
      </func>
    </sumat>

```

```
</ecuacion>
...
<ecuacion id="n" ...>
...
</ecuacion>
<param abr="[cadena]" nombre="[cadena]">
  [Descripcion]
</param>
...
<param ...>
  ...
</param>
</equipo>
```

A continuación se detalla el significado de cada una de las etiquetas utilizadas en el código:

ecuación: engloba el conjunto de variables, literales, funciones y sumatorias que definen una ecuación que será igualada a cero.

entrada: define una posible conexión al equipo, donde puede que se inserte una corriente de entrada. El parámetro *ref* le da nombre a la conexión, mientras que *type* define si la corriente que se conecta deberá ser material o energética. En caso de que sea material, *any*, *liquid*, y *gas* serán los parámetros encargados de definir el estado de la corriente que se conecta. Si se trata de una corriente en fase líquida, se deberá fijar el parámetro *liquid* como verdadero (*true*). Lo mismo sucede con el estado gaseoso y el parámetro *gas*. En caso de ser mezcla tanto *liquid* como *gas* deberán ser definidos como verdaderos, y en caso de que acepte corrientes de cualquier estado se definirá entonces el parámetro *any* como verdadero. El parámetro *finitas* define si la cantidad de entradas de este tipo son finitas o no. Un ejemplo de entradas no finitas puede ser visto en el mezclador, el cual acepta N corrientes de entrada materiales.

entradas: engloba todas las definiciones de posible conexiones de entrada que puede contener el equipo.

img: nombre de la imagen vectorial que formará parte del flowsheet cuando se agregue el equipo en cuestión.

lit: son elementos cuyo valor definido a través del parámetro *val* se incluye de manera exacta en la ecuación generada.

reference: nombre de la imagen png o jpg que se utilizará como referencia para realizar las conexiones del equipo. La misma será mostrada cuando que el usuario se introduzca en un equipo en particular para definir las entradas y salidas del mismo.

salida: análogo a entrada.

salidas: análogo a entradas.

sumat: define una sumatoria que se desdobra dependiendo la cantidad de compuestos (*type=comp*) y corrientes de entrada o salida (*type=cor*) que contenga el equipo. Si una ecuación contiene términos que

param: denota una variable dentro de las ecuaciones perteneciente al equipo en particular, que puede transformarse en parámetro si su valor es conocido. Entre estos parámetros se encuentran la caída de presión dentro del equipo, el calor incorporado o extraído del mismo, la eficiencia, etc.

var: define una variable perteneciente a una corriente en particular, la cual formará parte de la ecuación en la que se la está definiendo. Si al definir la corriente en el flowsheet, se le asigna un valor a dicha variable, la misma pasará a formar parte de los parámetros del modelo, dejando la misma de ser una incógnita. El parámetro *name* denota el nombre de la variable, *op* denota la operación que precede a la variable, que puede ser alguna de las predefinidas (*pos,mul,neg,div*) o una cualquiera definida por el usuario. *Type* define si la variable hace referencia al objeto al que pertenece o a algún objeto al cual este está conectado. En caso de ser *type cio*, el parámetro *ref* define a qué conexión influencia. *Comp* y *cor* definen si la variable va a estar influenciada por los compuestos o por las corrientes respectivamente. Si están dentro de una sumatoria, y esta se encuentra definida correctamente, la sumatoria se desdoblará modificando esta variable. Si el parámetro *infbyref* es verdadero (*true*) cambiará el nombre de la variable por x, y o z según la corriente conectada a la que se hace referencia (*ref*) sea líquido, vapor o mezcla, respectivamente.

Para llevar a cabo la incorporación de un nuevo equipo al generador de modelos, se debe utilizar entonces la estructura base detallada anteriormente, adaptando la misma al equipo que se desee ingresar. Para esto se deberán definir características propias del mismo como son:

- una imagen de referencia del equipo

- el número de corrientes de entrada y salida
- la fase de las corrientes de entrada y salida (líquido, vapor o mezcla)
- la repetición o no de los compuestos de las corrientes de entrada a en las de salida (ej: un mezclador repite los compuestos de entrada en todas sus corrientes de salida mientras que un reactor o una columna no)
- el modelo matemático que representa el comportamiento del equipo

MODELO MATEMÁTICO DE LA COLUMNA DE DESTILACIÓN REACTIVA

D.1 MODELOS MATEMÁTICOS UTILIZADOS

D.1.1 MODELO MATEMÁTICO DEL EQUIPO

D.1.1.1 ETAPAS NO REACTIVAS

a) Balance de masa por componente

$$F_{Feed} * z_i + F_{in,L} * x_{i,in} + F_{in,V} * y_{i,in} - F_{out,L} * x_i - F_{out,V} * y_i = 0 \quad i=1..nc$$

b) Balance de energía

$$F_{Feed} * h_{Feed} + F_{in,L} * h_{in,L} + F_{in,V} * h_{in,V} - F_{out,L} * h_{out,L} - F_{out,V} * h_{out,V} = 0$$

Nota: El calor de reacción del balance de energía es omitido porque se utilizó el calor de formación como valor de referencia para el cálculo de las entalpías.

c) Normalización

$$\sum x_i = 1 \quad i=1..nc$$

$$\sum y_i = 1 \quad i=1..nc$$

d) Equilibrio entre fases

$$\phi^L(T, P, x)_i * x_i = \phi^V(T, P, y)_i * y_i \quad i=1..nc$$

D.1.1.2 ETAPAS REACTIVAS**a) Balance de masa por componente:**

$$F_{Feed} * z_i + F_{in, L} * x_{i, in} + F_{in, V} * y_{i, in} - F_{out, L} * x_i - F_{out, V} * y_i + r_{ij} = 0 \quad i=1..nc, j=1..nRx$$

b) Balance de energía

$$F_{Feed} * h_{Feed} + F_{in, L} * h_{in, L} + F_{in, V} * h_{in, V} - F_{out, L} * h_{out, L} - F_{out, V} * h_{out, V} = 0$$

c) Normalización

$$\sum x_i = 1 \quad i=1..nc$$

$$\sum y_i = 1 \quad i=1..nc$$

d) Equilibrio entre fases

$$\phi^L(T, P, x)_i * x_i = \phi^V(T, P, y)_i * y_i \quad i=1..nc$$

e) Fuerza impulsora de la reacción química (Sundmacher y Hoffmann, 1999)

$$g_j = \left(\frac{a_{i, C4}}{a_{i, MeOH}} - \frac{a_{i, MTBE}}{K_{eq} * a_{i, MeOH}^2} \right) \quad i=1..nc$$

f) Actividad

$$a_i = \gamma_i * x_i \quad i=1..nc$$

g) Constante de la reacción química (Thiel y co., 2002)

$$kf(T) = kf(T^0) * e^{\frac{Ea}{R} \left(\frac{1}{T} - \frac{1}{T^0} \right)}$$

$$kf(T^0) = 15.5 \cdot 10^{-3} \text{ mol} / (\text{s} * \text{eq}[\text{H}^+])$$

$$Ea = 92400 \text{ J/mol}$$

$$T^{\circ} = 333.15 \text{ K}$$

h) Constante de equilibrio químico (Thiel y co., 2002)

$$\ln K_{eq} = \ln K_{eq,0} + \alpha \left(\frac{1}{T} - \frac{1}{T^*} \right) + \beta \ln \left(\frac{T}{T^*} \right) + \sigma (T - T^*) + \delta (T^2 - T^{*2}) + \varepsilon (T^3 - T^{*3}) + \theta (T^4 - T^{*4})$$

$$\alpha = -1.49277 \cdot 10^3$$

$$\beta = -7.74002 \cdot 10^1$$

$$\sigma = 5.07563 \cdot 10^{-1}$$

$$\delta = -9.12739 \cdot 10^{-4}$$

$$\varepsilon = 1.10649 \cdot 10^{-6}$$

$$\theta = -6.27996 \cdot 10^{-10}$$

$$T^* = 298.315 \text{ K}$$

$$K_{eq,0} = 284$$

i) Velocidad de reacción

$$r_{i,j} = v_{i,j} * k_{f_j} * g_j * m_{cat} \quad i=1..nc, j=1..nRx$$

D.1.1.3 CONDENSADOR

a) Balance de masa por componente

$$F_{in,v} * y_{i,in} - (FD + F_{ref}) * x_i = 0 \quad i=1..nc, j=1..nRx$$

b) Relación de reflujo

$$RR = F_{ref} / FD$$

c) Balance de energía

$$F_{in,v} * h_{in,v} - (FD * h_D + F_{ref} * h_{ref}) - Q_{cond} = 0$$

d) Normalización

$$\sum x_i = 1 \quad i=1..nc$$

$$\sum y_i = 1 \quad i=1..nc$$

e) Equilibrio entre fases

$$\phi^L(T, P, x)_i * x_i = \phi^V(T, P, y)_i * y_i \quad i=1..nc$$

D.1.1.4 REBULLIDOR

a) Balance de masa por componente:

$$F_{in, L} * x_{i, in} - (F_{out, L} * x_i + F_{out, V} * y_i) = 0 \quad i=1..nc, j=1..nRx$$

b) Balance de energía

$$F_{in, L} * h_{in, L} - (F_{out, L} * h_{out, L} + F_{out, V} * h_{out, V}) + Q_{reb} = 0$$

c) Normalización

$$\sum x_i = 1 \quad i=1..nc$$

$$\sum y_i = 1 \quad i=1..nc$$

d) Equilibrio entre fases

$$\phi^L(T, P, x)_i * x_i = \phi^V(T, P, y)_i * y_i \quad i=1..nc$$

D.1.2 MODELO DE EQUILIBRIO

Para llevar a cabo el modelado entre fases se utilizó la formulación conocida como γ - θ , mediante el planteo de la siguiente expresión:

$$y_i * \phi * P = x_i * \gamma_i * p_i^V$$

La fase vapor fue asumida como ideal ($\theta=1$), mientras que la fase líquida fue modelada utilizando las ecuaciones de coeficiente de actividad de Wilson. Este método resulta adecuado para mezclas multicomponentes que no forman dos fases líquidas (Kooijman y Taylor, 2000). El conjunto de ecuaciones utilizado es el siguiente:

$$\ln(\Lambda_{ij}) = a_{ij} + \frac{b_{ij}}{T}$$

$$S_{w_i} = \sum_{j=1}^{j=nc} x_j * \Lambda_{ij}$$

$$\ln(\gamma_i) = 1 - \ln(S_{w_i}) - \sum_{k=1}^{k=nc} \frac{x_k * \Lambda_{ki}}{S_{w_k}}$$

Los valores utilizados para los parámetros a_{ij} and b_{ij} a una presión de 11 bar son listados a continuación (Espinosa y *co.*, 1995)

a_{ij}

	iC₄	MeOH	MTBE	nC₄
iC₄	0	-0.742	0.2413	0
MeOH	0.742	0	0.9833	0.81492
MTBE	-0.2413	-0.9833	0	0
nC₄	0	-0.81492	0	0

b_{ij}

	iC₄	MeOH	MTBE	nC₄
iC₄	0	-85.5447	30.2477	0.66
MeOH	-1296.719	0	-746.3971	-1628.2145
MTBE	-136.6574	204.5029	0	88.3592
nC₄	0.6538	-194.0354	37.1378	0

D.1.3 MODELO GEOMÉTRICO (Douglas, 1988)

D.1.3.1 ALTURA DE LA COLUMNA

$$H_c = \left(\frac{2.3N_t}{0.5}\right) * 0.348$$

D.1.3.2 DIÁMETRO DE LA COLUMNA

$$D_c = 5.707 * 10^{-3} * \sqrt{\frac{F_V}{0.126} \left(\frac{Mg}{Rho_v} * 16000\right)^{1/4}}$$

D.1.3.3 ÁREA DEL CONDENSADOR

$$A_{cond} = \frac{Q_{cond}}{U_{cond} \Delta T_{m_{log cond}}}$$

D.1.3.4 ÁREA DEL REBULLIDOR

$$A_{cond} = \frac{Q_{reb}}{U_{reb} \Delta T_{m_{logreb}}}$$

D.1.4 COSTOS DE INSTALACIÓN (Douglas, 1988)**D.1.4.1 CASCO DE LA COLUMNA**

$$C_{col} = \left(\frac{MS}{280}\right) * 101.9 * (D_{col} * f_{m-ft})^{1.066} * (H_c * f_{m-ft})^{0.802} * (2.18 + F_C)$$

D.1.4.2 PLATOS DE LA COLUMNA

$$C_{trays} = \left(\frac{MS}{280}\right) * 4.7 * (D_{col} * f_{m-ft})^{1.55} * (H_t * f_{m-ft}) * (F_C) * N_t$$

D.1.4.3 CATALIZADOR

$$C_{cat} = m_{t_{cat}} * P_{cat}$$

D.1.4.4 CONDENSADOR

$$C_{cond} = \left(\frac{MS}{280}\right) * 101.3 * (A_{cond} * f_{m2-ft2})^{0.65} * (2.29 + F_C)$$

D.1.4.5 REBULLIDOR

$$C_{reb} = \left(\frac{MS}{280}\right) * 101.3 * (A_{reb} * f_{m2-ft2})^{0.65} * (2.29 + F_C)$$

D.1.5 COSTO ANUAL DE OPERACIÓN**D.1.5.1 AGUA DE ENFRIAMIENTO**

$$C_w = FV_w * P_w * 2.88e7$$

D.1.5.2 VAPOR SATURADO

$$C_s = FM_s * P_s * 2.88e7$$

D.1.6 COSTOS TOTALES**D.1.6.1 COSTO TOTAL DE INSTALACIÓN**

$$C_{inst} = C_{col} + C_{trays} + C_{cat} + C_{cond} + C_{reb}$$

D.1.6.2 COSTO TOTAL ANUAL DE OPERACIÓN

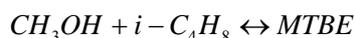
$$C_{op} = C_w + C_s$$

D.1.6.3 COSTO TOTAL ANUALIZADO

$$CTA = \frac{C_{inst}}{POT} + C_{op}$$

D.2 SIMULACIÓN DE LA COLUMNA: PRODUCCIÓN DE MTBE.

En esta sección se presenta como caso de estudio la producción de MTBE a partir de metanol e isobuteno, mediante la utilización de una columna de destilación reactiva (Domancich y *co.*, 2009a). La reacción global considerada dentro del equipo es:



Una de las corrientes de entrada al equipo está formada por una mezcla de isobuteno y n-butano, siendo este último un inerte para la reacción. La otra corriente se encuentra formada por metanol puro. Para el proceso convencional, el metanol normalmente se alimenta en exceso para mejorar la conversión. Esto causa problema a la hora de realizar la separación entre el producto (MTBE) y los reactivos que no reaccionaron. El inconveniente surge dado que el MTBE forma azeótropos con el metanol e isobuteno, haciendo que la tarea de separación se dificulte. Recientemente, se presentó un esquema híbrido para obtener MTBE puro a partir de un esquema reacción-separación (Daviou y *co.*, 2004), donde el metanol excedente es recuperado utilizando una membrana de pervaporación y reciclado luego al

reactor, mientras que el MTBE es obtenido como producto de fondo de una única columna de destilación, obteniéndose un producto con alto grado de pureza.

A continuación se presenta el esquema de la columna reactiva y la configuración utilizada (figura D.1) para la obtención de MTBE mediante el proceso de destilación reactiva.

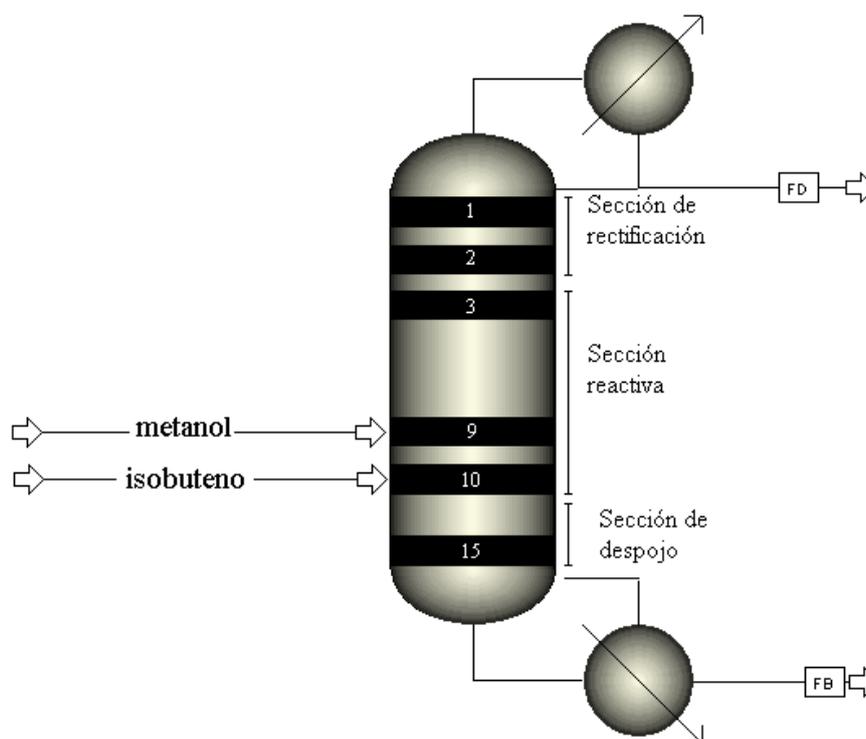


Figura D.1. Esquema de la columna de destilación reactiva

Se decidió alimentar las corrientes de metanol e isobuteno siguiendo la proporción estequiométrica, lográndose así la obtención de MTBE de alta pureza. De esta manera, el problema de los azeótropos entre mezclas se reduce, dado que el proceso de constante reacción-separación hace que no lleguen a formarse con la misma intensidad que en un proceso convencional (Taylor y Krishna, 2000).

Para el caso base se fijó el número de platos en 17. La zona reactiva se encuentra ubicada entre los platos 3 y 10. La tabla D.1 incluye todos los datos utilizados para la simulación del caso base. Los valores fueron extraídos de Seader y Henley (1998); y Almeida-Rivera (2005). La relación de reflujo fue fijada en 7 porque este valor garantiza la no aparición del fenómeno de multiplicidad de estados estacionarios (Singh y *co.*, 2005).

CONDICIONES DE OPERACIÓN	
Número de platos	15
Número de platos reactivos	8
Presión de operación	11 bar
Relación de reflujo	7
Caudal de fondo (FB)	203 mol/s
Masa total de catalizador	1632.8 kg (8000 eq _[H⁺])
DATOS DE ALIMENTACIÓN	
Alimentación 1	
Temperatura [K]	350
Presión [bar]	11
Caudal [mol/s]	455
Composición	
	n-butano: 0.63
	isobuteno: 0.37
Ubicación	Plato 10
Alimentación 2	
Temperatura [K]	320
Presión [bar]	11
Caudal [mol/s]	168
Composición	
	Metanol 1.0
Ubicación	Plato 9
PRODUCTOS	
Destilado (FD):	
Temperatura [K]	353
Composición	
	nC4: 0.764
	iC4: 0.12
	MeOH: 0.116
	MTBE: 0.0
Caudal [mol/s]	285
Fondo (FB):	
Temperatura [K]	387
Composición	
	nC4: 0.341
	iC4: 0.011
	MeOH: 0.0
	MTBE 0.648
CARGA TÉRMICA [W]	
Condensador	4.35.10 ⁷
Rebullidor	3.13.10 ⁷

Tabla D.1. Especificaciones de la columna utilizadas para el caso base

D.2.1 INFLUENCIA DEL NÚMERO DE PLATOS REACTIVOS

Tomando como base los datos de la tabla D.1 y sabiendo que la conversión obtenida en el equipo resulta función del número de platos reactivos, se decidió estudiar la variación

que sufre la misma a medida que se modifica el número de etapas reactivas. La figura D.2 muestra cómo varía la conversión a medida que se agregan etapas reactivas (número total de etapas constante). En ella se puede observar que para el número máximo propuesto de etapas reactivas, la conversión alcanza un 80%.

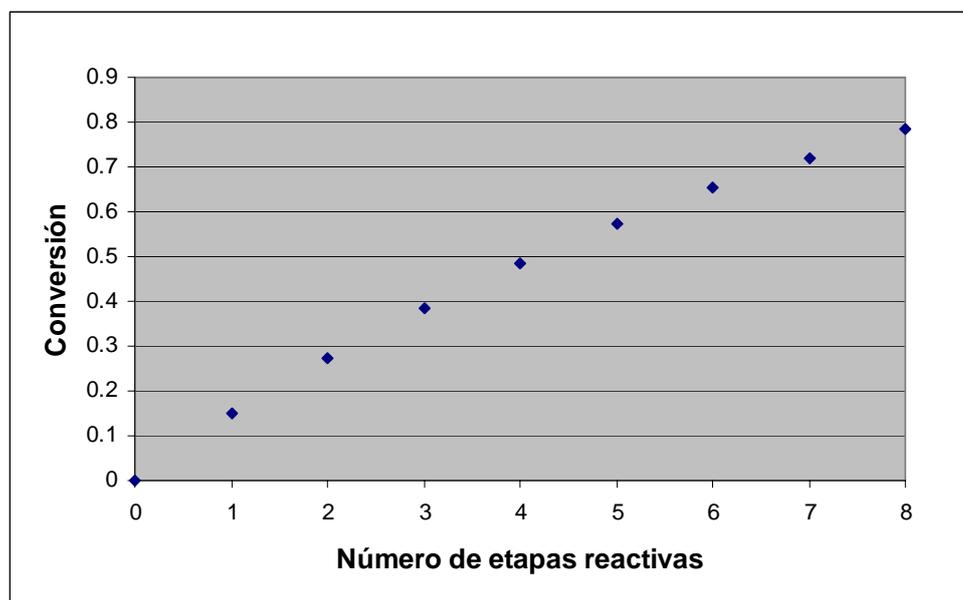


Figura D.2. Conversión obtenida en función del número de etapas reactivas

Es importante resaltar que, para la máxima conversión obtenida, no es posible obtener MTBE de alta pureza para el caso base, lo cual se puede visualizar en la figura D.2. La razón es que la cantidad de n-butano alimentada a la columna se distribuye a lo largo de la columna contaminando el MTBE de la corriente de fondo. La composición molar mostrada en la figura D.2 corresponde a una composición másica de 0.73, lo cual resulta menor al porcentaje requerido para ser utilizado como aditivo para gasolina (0.95).

Como se puede observar en la figura D.2, la utilización de más de 8 platos reactivos no mejora la conversión obtenida de manera significativa. Esto se debe a que la conversión no crece linealmente con el número de etapas reactiva, por lo que, el crecimiento se vuelve menor a medida que el número de platos reactivos aumenta. El efecto de agregar etapas reactivas es muy marcado para las primeras etapas, mientras que a partir de las 8 etapas el cambio ya no resulta significativo. En la figura D.3 se puede observar la variación de los perfiles de composición de MTBE a medida que se incrementa el número de etapas reactivas.

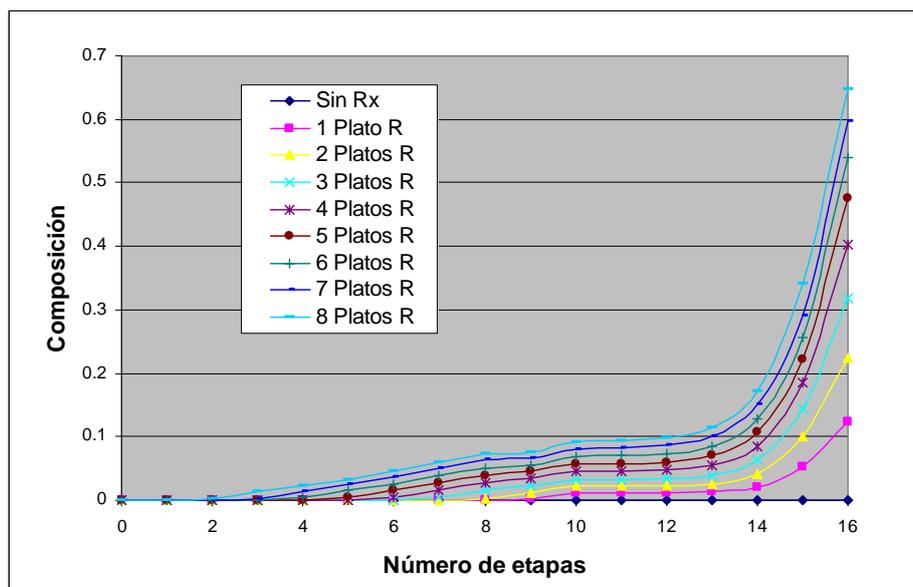


Figura D.3. Perfiles de composición molar de MTBE a lo largo de la columna en función del número de etapas reactivas

Si se realiza un análisis acerca de los requerimientos energéticos por parte del condensador y rebullidor de la columna para llevar a cabo la operación de la misma, se puede observar que los requerimientos decrecen a medida que las etapas reactivas se van incrementando (figura D.4). Esto evidencia claramente que el esquema de integración energética presente en este tipo de unidades permite disminuir notablemente los costos operativos.

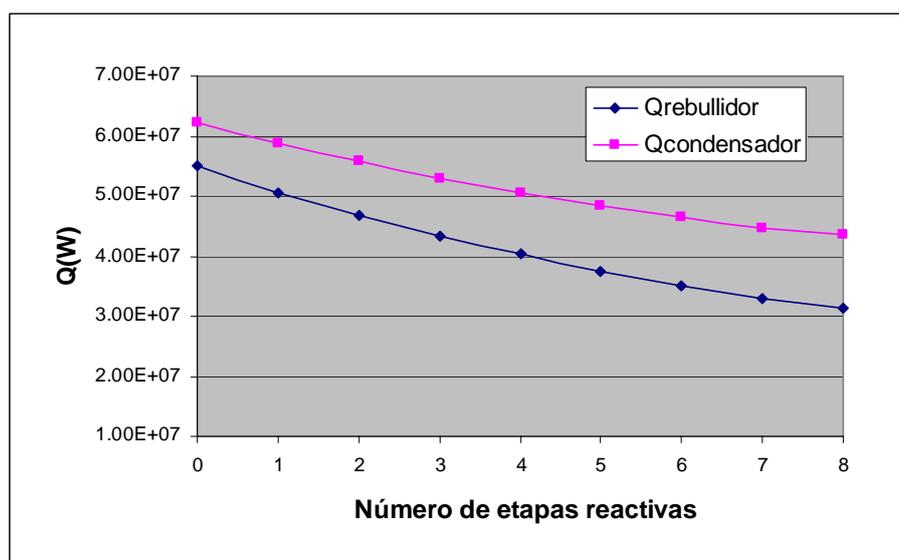


Figura D.4. Consumo energético del rebullidor y condensador de la columna en función del número de etapas reactivas

D.2.2 INFLUENCIA DEL CAUDAL DE FONDO

Una alternativa para mejorar la composición del MTBE en la corriente de fondo, sin modificar la masa de catalizador de los platos, es reducir el caudal de la corriente de fondo (figura D.5). Sin embargo esta acción tiene un efecto adverso sobre la reacción, dado que, cerrando el fondo de la columna se está desplazando el equilibrio de la misma hacia los reactivos, disminuyendo así la conversión obtenida. Este efecto puede ser visualizado en la figura D.6.

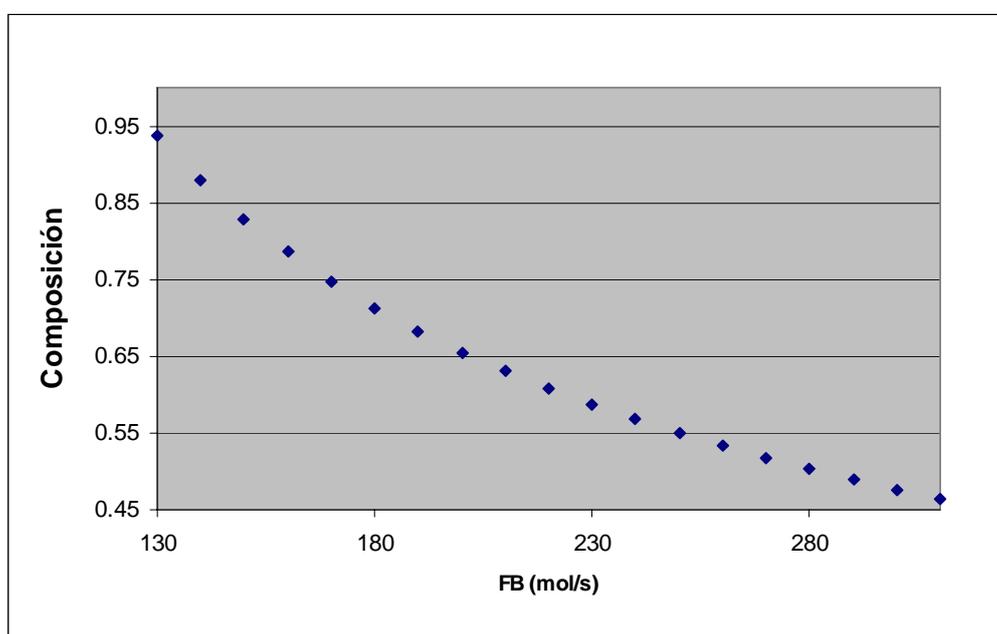


Figura D.5. Composición de MTBE en función del caudal de fondo (FB)

Otra variable de interés para analizar es la temperatura y su perfil a lo largo de la columna. La figura D.7 muestra los perfiles obtenidos de temperaturas en el equipo, a partir de realizar variaciones del caudal fondo para un número fijo de 8 etapas reactivas.

En la figura D.7 se pudo observar cómo la temperatura aumenta en la zona reactiva con el incremento del caudal de fondo, invirtiendo ese comportamiento en la zona de despojo. Esto se puede explicar teniendo en cuenta la naturaleza exotérmica de la reacción en cuestión. Gracias a la figura D.6 se puede apreciar cómo la conversión se ve incrementada con el aumento del caudal de fondo. Analizando la figura D.7, se puede ver que el mayor perfil de temperatura para la zona reactiva se obtiene con el mayor caudal de fondo. Eso significa que

tenemos la zona reactiva más caliente cuando obtenemos la mayor conversión, debido a que la reacción al ser exotérmica libera calor elevando la temperatura de la zona donde ella ocurre.

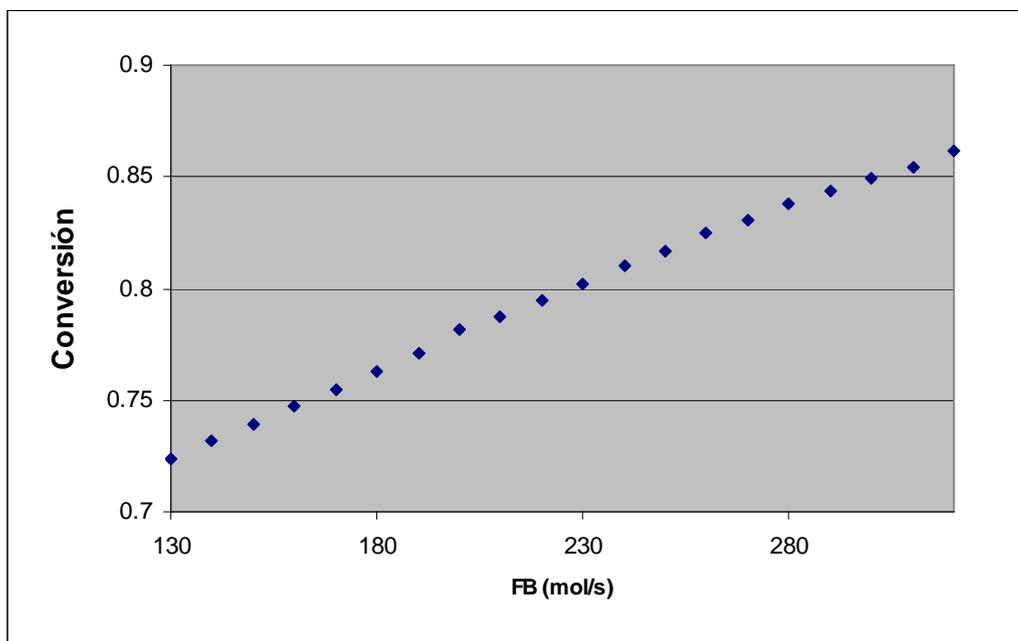


Figura D.6. Conversión obtenida en función del caudal de fondo (FB)

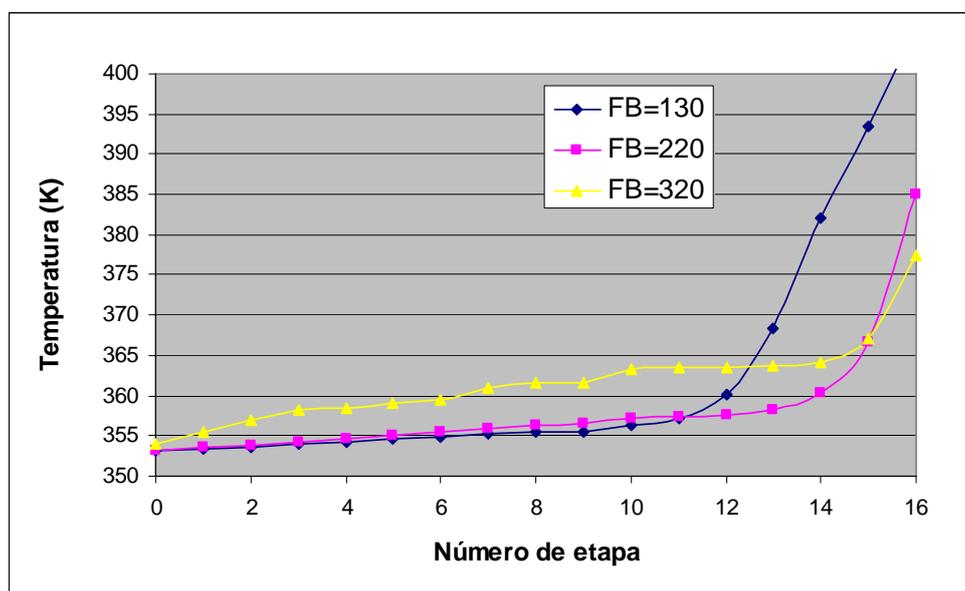


Figura D.7. Perfiles de temperatura obtenidos para diferentes caudales de fondo

D.3 OPTIMIZACIÓN DEL PROCESO DE PRODUCCIÓN DE MTBE

Una vez testado el modelo de la columna reactiva, se decidió llevar a cabo la optimización económica del mismo (Domancich y *co.*, 2007), por medio de la aplicación de

un algoritmo MINLP (Mezcla Entera No Lineal). El esquema de costos resultante fue comparado con el costo del proceso convencional de producción de MTBE, considerando para ello que la reacción y la separación son llevadas a cabo en diferentes equipos.

Las ecuaciones básicas utilizadas para llevar a cabo la optimización del modelo son las mismas que se incorporaron al generador automático de modelos, con la salvedad de que debieron realizarse algunas modificaciones en la formulación del problema. La primera de ellas consistió en convertir en variable el número y ubicación de las etapas reactivas (en el caso base resultaba constante), permitiendo de esta manera que el optimizador escoja los valores más convenientes. Otro cambio implementado fue la incorporación de ecuaciones geométricas y de costo, las cuales permiten obtener la configuración óptima de la columna reactiva (altura, diámetro, área de los intercambiadores de calor). Las ecuaciones de costo incluyen una serie de correlaciones que permiten obtener los costos de inversión y operación del equipo (Douglas, 1988). El costo de inversión fue calculado a partir de los costos individuales de casco, platos, catalizador, condensador y rebullidor. Los costos obtenidos, y publicados en Domancich y *co.*, 2007, se encuentran actualizados a Noviembre de 2006, momento en que fue publicado el trabajo. Para llevar a cabo una actualización de esos valores al día de hoy, simplemente se debe modificar en el modelo matemático los valores de los índices de costos M&S (Marshall & Swift). El costo total de inversión fue anualizado (CTA) considerando una tasa de interés del 10%, un tiempo de recuperación del capital de 5 años y 8000 horas anuales de operación de los equipos. Los servicios auxiliares para el condensador y rebullidor son agua y vapor de baja presión respectivamente. Todas las ecuaciones utilizadas para los cálculos geométricos y económicos se encuentran detalladas en el Anexo D de esta Tesis.

D.3.1 FORMULACIÓN DEL PROBLEMA DE OPTIMIZACIÓN

Para llevar a cabo la optimización del modelo, fue necesario realizar la formulación de dos problemas, debido a que se plantearon dos objetivos diferentes:

- a) Minimizar el CTA (optimización económica)
- b) Maximizar la pureza de MTBE en el fondo de la columna (optimización de pureza)

La variable binaria $y_{r,I}$ fue asignada a todos los platos, haciendo referencia a los platos reactivos con $y_{r,I} = 1$ y a los no reactivos con $y_{r,I} = 0$. Como restricción adicional, se impuso

que todos los platos reactivos deben ubicarse de forma consecutiva, y que al menos un plato no reactivo debe ubicarse sobre la sección reactiva y otro por debajo ($y_{r,l}=0$, $y_{r,n}=0$). Para evitar el producto de variables enteras-continuas, se implementó la formulación *Big-M* (Edgar y Himmelblau, 1988), utilizando los límites superiores como límites.

En todo momento se mantuvo fijo el número total de platos en 15. Este valor no fue elegido arbitrariamente, sino que fue seleccionado a partir de realizar diferentes optimizaciones variando dicho número. Se pudo verificar que el agregado de más platos no mejoraba el proceso de manera significativa, incrementando los costos de instalación y operación. Por otro lado, la utilización de menos de 15 platos en la columna empobrece el proceso. El valor de reflujo no fue considerado como variable de optimización, dado que se trata de una variable clave para evitar la aparición de múltiples estados estacionarios. De acuerdo a Singh *et al* (2005), el valor asignado a dicha variable fue de 7, valor que asegura la existencia de solo un estado estacionario. Ambos problema de optimización están conformados por 1173 variables continuas y 45 discretas. La resolución fue llevada a cabo utilizando GAMS.

$$\begin{array}{ll}
 \min \mathbf{CTA}(\mathbf{x},\mathbf{y}) & \max \mathbf{PUR}(\mathbf{x},\mathbf{y}) \\
 \text{s.t. } \mathbf{g}(\mathbf{x},\mathbf{y}) \leq 0 & \text{s.t. } \mathbf{g}(\mathbf{x},\mathbf{y}) \leq 0 \\
 \mathbf{h}(\mathbf{x},\mathbf{y}) = 0 & \mathbf{h}(\mathbf{x},\mathbf{y}) = 0 \\
 \mathbf{b}^L \leq \mathbf{x} \leq \mathbf{b}^U & \mathbf{b}^L \leq \mathbf{x} \leq \mathbf{b}^U \\
 \mathbf{y} \in \{0,1\} & \mathbf{y} \in \{0,1\}
 \end{array}
 \quad (\text{P1}) \qquad (\text{P2})$$

Donde

\mathbf{x} (**variables continuas**): caudal de fondo (F_B), masa de catalizador por plato (m_{cat}).

\mathbf{y} (**variables discretas**): variables asociadas a los factores de existencia o no de reacción en un plato (y_r) y la ubicación de las alimentaciones (y_{MeOH} para metanol y y_{iC4} para isobuteno e inertes)

$\mathbf{g}(\mathbf{x},\mathbf{y})$: restricciones en la composición deseada de producto (MTBE) y la conversión obtenida en el equipo. Conversión MTBE ≥ 0.95 ; pureza purity $\geq 88\%w$

$\mathbf{h}(\mathbf{x},\mathbf{y})$: ecuaciones MESH, modelo de equilibrio y cinético (Domancich *et al.* 2009b), ecuaciones geométricas y económicas (Domancich *et al.* 2007)

$\mathbf{b}^L, \mathbf{b}^U$: límites inferior y superior de la variables de optimización \mathbf{x} .

$180 \text{ mol/s} < F_B < 205 \text{ mol/s}$; $0 < m_{cat,j} < 1000 \text{ Kg}$, $j=1, N_t=15$

D.3.2 RESULTADOS DE LA OPTIMIZACIÓN

La tabla D.2 muestra los resultados obtenidos para ambas optimizaciones. Como era de esperarse, al llevar a cabo la optimización económica del modelo, la composición de fondo de MTBE alcanza su valor mínimo. Por otro lado, cuando se realiza la maximización de la pureza, el máximo valor alcanzado fue de 0.96 de pureza, valor requerido para ser utilizado como aditivo en gasolinas. En ambos casos la conversión alcanzó el 98%, un valor elevado si se compara con el obtenido en el proceso convencional de producción de MTBE, el cual se encuentra constituido por un reactor y 2 columnas de destilación operando a diferentes presiones (Hoch *et al*, 2003). En dicho esquema de proceso, la conversión sólo alcanza el 85%.

rr =7	Optimiz. de pureza	Optimiz. económica
Conversión	0.984	0.985
Pureza de MTBE (fracción másica)	0.96	0.88
Alimentación de Metanol	9	9
Alimentación de Isobuteno e Inertes	10	10
F_B [kgmol/s]	180	205
Q_{reb} [kW]	10140	8157
A_{reb} [m ²]	286	230
Q_{cond} [kW]	23800	22022
A_{cond} [m ²]	873	805
Masa de catalizador/plato [kg]		
plato 1	0	0
platos 2 to 8	1000	1000
plato 9	1000	436
plato 10	887	758
platos 11 to 15	0	0
Masa total de catalizador [kg]	8887	8194
Costo del casco de la columna [\$]	1283810	1229238
Costo de platos [\$]	53350	50085
Costo de catalizador [\$]	766761	707200
Costo de rebullidor [\$]	312840	271520
Costo de condensador [\$]	646404	613062
Costo total de instalación [\$]	3063165	2871105
Costo de agua [\$/año]	640702	590571
Costo de vapor [\$/año]	1477191	1187925
Costo de operación [\$/año]	2117893	1778496
CTA	2730526	2352717

Tabla D.2. Resultados de la optimización para ambas alternativas.

D.3.3 COMPARACIÓN CON PROCESO CONVENCIONAL

Haciendo uso de los valores obtenidos para las diferentes optimizaciones de la columna, se decidió llevar a cabo una comparación de costos entre el proceso convencional y la destilación reactiva, para la producción de MTBE. Para la destilación reactiva, se tomó como caso a comparar el obtenido a partir de la optimización de pureza, el cual da como resultado el grado de pureza requerido para poder utilizar el MTBE en la gasolina.

La tabla D.3 muestra los costos involucrados en el proceso convencional. Como se puede apreciar a partir de la comparación de ambos resultados, tanto los costos de inversión como los operativos son mayores para el proceso convencional que para la destilación reactiva. Es sencillo imaginar que los costos de instalación se ven reducidos por la diferencia en cuanto a cantidad de equipos que se necesitan en uno y otro caso.

En cuanto a los costos operativos, estos se ven reducidos en la destilación reactiva porque dicho proceso posee un esquema de integración energética favorable, que hace que el calor liberado por las reacciones pueda ser aprovechado para generar vapor y llevar a cabo la separación de los productos, disminuyendo así los requerimientos energéticos externos y por ende los costos operativos. Por otro lado, el aprovechamiento de ese calor hace que la zona reactiva de la columna nunca alcance los 95°C, temperatura a la cual el catalizador comienza a sufrir desactivación. En el proceso convencional, es necesario utilizar en el reactor una camisa refrigeradora, de manera tal de no superar esa temperatura límite, incrementando así los costos del proceso. Otro factor que ayuda a reducir los costos operativos es la alta conversión obtenida (cercana al 100%), la cual hace que el proceso de separación de productos y reactivos resulte más sencillo que el esquema convencional de producción.

Como se puede apreciar en la tabla D.3, mediante la utilización de la columna de destilación reactiva se pueden lograr reducciones que rondan el 20% en costos de inversión y hasta 50% en costos operativos, en comparación al proceso convencional. Esto resulta un incentivo importante para el estudio y desarrollo de sistemas de destilación reactiva.

Equipos	Costos Operativos		Costos de Inversión
Reactor	\$ 370000	Reactor	\$ 400000
Columna 1		Catalizador	\$766761
Condensador	\$500000	Columna 1	\$860000
		Casco	
Rebullidor	\$2900000	Platos	\$35000
Columna 2		Condensador	\$550000
Condensador	\$40000	Rebullidor	\$500000
Rebullidor	\$220000	Columna 2	\$450000
		Casco	
		Platos	\$35000
		Condensador	\$550000
		Rebullidor	\$500000
Total	\$4030000	Total	\$3765361

Tabla D.3. Costos del proceso convencional de producción de MTBE.

D.4 NOMENCLATURA

A: Área [m²]

F_B Caudal de fondo [kgmol/s]

P_s: Precio del vapor [\$/Kg]

C_{cat}: Costo total de catalizador [\$]

C_{col}: Costo del casco de la columna [\$]

C_{cond}: Costo del condensador [\$]

C_{inst}: Costo total de instalación [\$]

C_{op}: Costo de operación [\$/año]

C_{reb}: Costo del rebullidor [\$]

C_s: Costo de vapor [\$/año]

C_{trays}: Costo total de los platos [\$]

CTA: Costo total anualizado [\$]

C_w: Costo de agua [\$/año]

D_{col}: Diámetro de la columna [m]

F_c: Factor de corrección por presión y tipo de material

f_{m-ft}: Factor de conversión

F_v: Caudal de vapor [m³/s]

H_c: Altura de la columna [m]

H_t: Espaciado entre platos [m]

M_g: Peso molecular del vapor

M&S: Índice de costo de Marshall y Swift

m_{cat}: Masa de catalizador [Kg]

N_t: Número de etapas

P_{cat}: Precio por kilo del catalizador [\$/Kg]

POT: Período estimado de recuperación del capital invertido [años]

PUR: Pureza del MTBE (fracción másica)

P_w: Precio del agua [\$/m³]

Q_{cond}: Carga térmica del condensador [W]

Q_{reb}: Carga térmica del rebullidor [W]

Rho : Densidad molar [mol/m³]

rr: Relación de reflujo

U: Coeficiente global de transferencia de calor [W/m²K]

X_{ib}: Conversión de isobuteno

