



**UNIVERSIDAD DE CARABOBO**  
**ÁREA DE ESTUDIOS DE POSTGRADO**  
**FACULTAD DE INGENIERÍA**  
**DOCTORADO EN INGENIERÍA**  
**ÁREA QUÍMICA**



**CONTRIBUCIÓN AL DISEÑO INTEGRADO DE**  
**REACTORES MEZCLA COMPLETA**

**Autor: MSc. Ing. CARLOS EDUARDO SEGURA GARCÍA**

Bárbula, enero de 2016.

Título:    CONTRIBUCIÓN AL DISEÑO INTEGRADO DE REACTORES MEZCLA  
          COMPLETA.

Autor:    CARLOS EDUARDO SEGURA GARCÍA

1 era. Edición, 2014.

Derechos Reservados:

© 2014, Carlos Eduardo Segura García. Universidad de Carabobo, Venezuela.

**ISBN:**

Depósito Legal:

Portada: Carlos Eduardo Segura García

Montaje electrónico: Carlos Eduardo Segura García

Impreso en Venezuela por: Centro de Reproducción PEQUIVEN Morón

Printed in Venezuela.



UNIVERSIDAD DE CARABOBO  
ÁREA DE ESTUDIOS DE POSTGRADO  
FACULTAD DE INGENIERÍA  
DOCTORADO EN INGENIERÍA  
ÁREA QUÍMICA



## CONTRIBUCIÓN AL DISEÑO INTEGRADO DE REACTORES MEZCLA COMPLETA

Tesis Doctoral presentada  
ante la Dirección de  
Postgrado de la Facultad  
de Ingeniería de la  
Universidad de Carabobo,  
para optar al Grado de  
Doctor en Ingeniería.

**Autor: MSc. Ing. Carlos Eduardo Segura García**

Tutor:  
**Dr. Sergio Perez**

Bárbula, enero de 2016

**CONTRIBUCIÓN AL DISEÑO INTEGRADO DE REACTORES  
MEZCLA COMPLETA**

# DOCTORADO EN INGENIERÍA

Tesis Doctoral

## CONTRIBUCIÓN AL DISEÑO INTEGRADO DE REACTORES MEZCLA COMPLETA

Autor: MSc. Ing. Carlos Eduardo Segura García

### INSTITUCIONES FINANCIERAS:

#### UNIVERSIDAD DE CARABOBO



La Dirección de Postgrado de la Facultad de Ingeniería de la Universidad de Carabobo aprobó el Proyecto de Tesis Doctoral, que permitió la realización de este trabajo, según oficio S/N del 15-02-08.

#### CENIPRO, UNEXPO BARQUISIMETO



El Centro de Investigaciones de Procesos (CENIPRO) de la Universidad Nacional Experimental Politécnica de Barquisimeto (UNEXPO), aprobó la pasantía doctoral, la cual permitió cumplir con un requisito del plan doctoral y completar este trabajo de grado, según oficio N° S/N del 26-10-12.

#### PEQUIVEN



La Vice Presidencia de la Corporación Petroquímica de Venezuela, a través de la Gerencia General de Proyectos Internacionales, aprobó la subvención de este trabajo, lo cual permitió cumplir con un requisito del plan doctoral y completar este trabajo de grado, según oficio N° S/N del 05-13.

## **AGRADECIMIENTOS**

A Dios todopoderoso por ser el centro y motivación en la vida.

A mi esposa y a mis hijos, ya que su ayuda, paciencia y comprensión fueron el pilar fundamental para la culminación de este trabajo.

A mis padres y hermanos en quienes siempre puedo confiar.

A la Universidad de Carabobo, a CENIPRO y a todo su personal.

A la profesora Marisol y Alexander Peraza, sin ellos no hubiese sido posible la culminación de este trabajo.

A mi amigo Jesús, por estar siempre allí y ayudarme en todo lo que le es posible.

Y muy especialmente a mi tutor, el Profesor Sergio Rodríguez, el cual me apoyó en todas las fases de esta investigación hasta su culminación.

## **DEDICATORIA**

*Cada vez Él me dijo: “Mi gracia es todo lo que necesitas; mi poder actúa mejor en la debilidad”. Así que ahora me alegra jactarme de mis debilidades, para que el poder de Cristo pueda actuar a través de mi...*

*2 Corintios 12:9 NTV*

*Segura García, Carlos Eduardo. "Contribution to the Integrated Design of Continuous Stirred Tank Reactors". Thesis for the degree of Chemical Engineering Doctor, University of Carabobo, Engineering Postgraduate Research Area, Carabobo, Venezuela, January 2016.*

## ABSTRACT

This work provides a method for the complete stirred tank reactor design called "simultaneous or Integrated Design Approach" which considers the dynamics of the process and its control together, from the design phase, guaranteeing that the system will respond properly to any disturbances. First, it was set the algorithm for solving nonlinear programming problems with differential algebraic equations (NLP-DAE), then raise all technical aspects necessary to describe the system, such as criteria for the design of control, based on the response full system together with some points of the response. It was used process design material and energy balances in transient state, as well as in conjunction with restrictions on the maximum reactors temperature, minimum conversion degree of the chemical reaction and relationship between equipments height and diameter. The resulting integrated design problem has constraints and no linear equations with several optimum locals. In this context, the present work propose to achieve a methodological contribution, through a systematic applications of different optimization techniques as heuristic methods (genetic algorithm and simulated annealing) and classic methods (deterministic).

**Keywords:** integrated process design, heuristic and classic optimization techniques.



*Segura García, Carlos Eduardo. “Contribución al Diseño Integrado de Reactores Mezcla Completa”. Tesis Doctoral para optar al Grado de Doctor en Ingeniería Química, Universidad de Carabobo, Área de Estudios de Postgrado de la Facultad de Ingeniería, Carabobo, Venezuela, enero de 2016.*

## RESUMEN

En el presente trabajo se establece un método para el diseño de reactores mezcla completa denominada “enfoque simultaneo o diseño integrado” el cual considera la dinámica del proceso y su control, conjuntamente desde la fase de diseño, garantizando que el sistema responderá adecuadamente frente a posibles perturbaciones. Primeramente se establece el algoritmo para la solución de problemas de programación no lineal con ecuaciones algebraicas diferenciales (NLP-DAE), después se plantean todos los aspectos técnicos necesarios que describen el sistema, tales como criterios para el diseño de control, basados en la respuesta completa del sistema (“error cuadrático integral” criterio integral) en conjunto con algunos puntos de la respuesta (índice de disminución y sobremodulación). para el diseño de procesos se utilizaron balances de materia y energía en estado transitorio, así como en conjunto con restricciones en la máxima temperatura permitida en los reactores, mínimo grado de conversión posible y relación entre la altura y el diámetro de los equipos. puesto que los problemas de diseño integrado de procesos son problemas de optimización con restricciones y de naturaleza altamente no lineal con múltiples óptimos locales, es preciso utilizar para resolverlos apropiadamente los métodos de optimización existentes, ya sea los métodos clásicos o los métodos heurísticos modernos. dentro de este contexto, en el presente trabajo se realiza una contribución metodológica, mediante un estudio sistemático de la aplicación de diferentes técnicas de optimización, clásicas (determinísticas) y heurísticas (algoritmo genético y recocido simulado) al diseño integrado de reactores mezcla completa.

**Palabras claves:** diseño integrado, diseño tradicional y optimización, técnicas de optimización clásicas y heurísticas.

## ÍNDICE GENERAL

<b>ÍNDICE GENERAL</b>	<b>I</b>
<b>ÍNDICE DE FIGURAS</b>	<b>VIII</b>
<b>ÍNDICE DE TABLAS</b>	<b>X</b>
<b>INTRODUCCIÓN</b>	<b>16</b>
<b>CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA</b>	<b>21</b>
<b>OBJETIVO GENERAL</b>	23
<b>OBJETIVOS ESPECÍFICOS</b>	23
<b>JUSTIFICACIÓN</b>	24
<b>ALCANCE</b>	27
<b>CAPÍTULO II. MARCO TEÓRICO</b>	<b>29</b>
<b>SECCIÓN 1. ESTADO DEL ARTE</b>	29
<b>SECCIÓN 2. BASES TEÓRICAS</b>	<b>35</b>
2.1. <i>DISEÑO DE PROCESOS Y EL ENFOQUE DEL DISEÑO INTEGRADO</i>	35
2.1.1. Diseño de procesos	35
2.1.1.1. El enfoque tradicional de diseño de procesos (enfoque secuencial)	38
2.1.1.2. El Enfoque del Diseño Integrado	41
2.2. <i>DISEÑO DE REACTORES MEZCLA COMPLETA</i>	46
2.2.1. El Reactor de mezcla completa	46
2.3. <i>CONTROL DE PROCESOS</i>	51
2.3.1. Los Controladores Clásicos	52
2.4. <i>CONCEPTOS ASOCIADOS A LOS MODELOS MATEMÁTICOS DE OPTIMIZACIÓN</i>	54
2.4.1. Etapas para resolver el problema de optimización	57
2.4.2. Tipos de optimización. Optimización local y local	57
2.4.2.1. Optimización local	58
2.4.2.2. Optimización Global	59
2.4.3. Clasificación histórica de los métodos de optimización	59
2.4.4. Métodos Híbridos de Optimización	63
2.4.5. Método del Gradiente	64

2.4.5.1	Método QUASI Newton	70
2.4.6	Algoritmo DIRECT	71
2.4.7	Algoritmos Genéticos	75
2.4.8	Método Recocido Simulado o Simulated Annealing (SA)	83
2.5	<i>MATLAB R2012A EN EL PROBLEMA DE DISEÑO INTEGRADO.</i>	92
2.5.1	Método del Gradiente en MATLAB R2012a. La Función “fmincon”	95
2.5.1.1	Estructura de parámetros de optimización “options” para “fmincon”	100
2.5.1.2	Parametros de la estructura “options” para “fmincon”	100
2.5.2	Algoritmo DIRECT en MATLAB R2012a: “gblSolve”.	104
2.5.3	Algoritmos Genéticos en MATLAB R2012a	106
2.5.3.1	Opciones de población	106
2.5.3.2	Opciones de Reproducción	109
2.5.3.3	Opciones de Migración	109
2.5.3.4	Opciones de criterios de parada	110
2.5.3.5	Opciones de Gráfico.	111
2.5.3.6	Opciones de escalamiento de ajuste	112
2.5.3.7	Opciones de Selección	112
2.5.3.8	Opciones de Cruce	113
2.5.3.9	Opciones de Mutación	113
2.5.4	Recocido Simulado (SA) en MATLAB R2012a.	114
2.5.4.1	Minimización mediante el uso de SIMULANNEALBND	114
2.5.4.2	Opciones de Recocido Simulado. Configuración problema de optimización	115
<b>CAPÍTULO III. MARCO METODOLÓGICO</b>		<b>121</b>
3.1.	<b>TIPO DE INVESTIGACIÓN</b>	121
3.2.	<b>PROCEDIMIENTO DE TRABAJO</b>	123
	<b>EL SISTEMA</b>	129
	<b>TIPO DE UNIDADES. OPERACIÓN UNITARIA. TIPO DE REACCIÓN</b>	129
	<i>Variables y su clasificación.</i>	133
	<i>Objetivos en el problema de Diseño Integrado.</i>	135
	<i>Modelación matemática</i>	139
	<i>Estructura algorítmica para la obtención de los valores óptimos de diseño, configuración de los equipos y parámetros del controlador</i>	145
<b>CAPÍTULO IV. PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS</b>		<b>161</b>
4.1	<b>ALGORITMOS GENÉTICOS</b>	161
	<i>Implementación de la estructura algorítmica en MATLAB R2012a</i>	161

4.2 <i>RECOCIDO SIMULADO (SA)</i>	188
<i>Implementación de la estructura algorítmica en MATLAB R2012a</i>	188
4.3 <i>ALGORITMO DIRECT (GBLSOLVE)</i>	205
<i>Implementación de la estructura algorítmica en MATLAB R2012a</i>	205
4.4. <i>DESEMPEÑO DE LAS TÉCNICAS DE OPTIMIZACIÓN EN EL DISEÑO INTEGRADO DEL SISTEMA</i>	220
<b>CAPÍTULO V</b>	<b>226</b>
<b>CONCLUSIONES</b>	<b>226</b>
<b>RECOMENDACIONES</b>	<b>229</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>230</b>
<b>ANEXOS</b>	<b>237</b>

## ÍNDICE DE FIGURAS

Figura 2.1. Metodología del enfoque secuencial. ....	39
Figura 2.2. Metodología del enfoque simultáneo. ....	46
Figura 2.3. Nomenclatura utilizada en un reactor mezcla completa.....	47
Figura 2.4. Esquemas globales de procesos controlados y no controlados. ....	51
Figura 2.5. Etapas para resolver el problema de optimización. ....	57
Figura 2.6. Clasificación de los métodos de optimización más relevantes. ...	61
Figura 2.7. Clasificación métodos de búsqueda local. Métodos de Descenso. ....	65
Figura 2.8. Función unidimensional $hx,d$ . ....	67
Figura 2.9. Método del Gradiente.....	68
Figura 2.10. Esquema de iteración del método del gradiente. ....	68
Figura 2.11. Espacio de soluciones y el Método Quasi-Newton.....	70
Figura 2.12. Algoritmo Genético Simple.. ....	78
Figura 2.13. Operador cruce basado en un punto.....	80
Figura 2.14. Operador de mutación. ....	81
Figura 2.15. Selección de nuevos estados en Recocido Simulado o Simulated Annealing (SA).....	87
Figura 3.1 Diagrama de bloques del proceso con los reactores en serie. ....	130
Figura 3.2 Diagrama de bloques del proceso con los reactores en paralelo. ....	130
.....	146
Figura 3.3. Diagrama de bloques de la estructura algorítmica para la resolución del problema de diseño integrado de dos reactores mezcla completa. ....	148
Programacion no lineal (NLP) del planteamiento original. Topología de reactores en serie.....	148
Programacion no lineal (NLP) del planteamiento original. Topología de reactores en paralelo.....	149
Figura 3.4. Algoritmo para resolver el problema NLP-DAE del diseño integrado de reactores mezcla completa. Módulo I: optimización en estado estacionario...	156

Figura 3.5. Algoritmo para resolver el problema NLP-DAE del diseño integrado de reactores mezcla completa. Módulo II: optimización en estado transitorio. ...	157
Figura 4.1.1. Ilustración de la relación entre las dimensiones de los dos (2) reactores en el arreglo en paralelo .....	182

## ÍNDICE DE TABLAS

Tabla 2.1. Descripción de los parámetros para “fmincon” .....	101
Tabla 2.2. Parámetros de entrada “gblsolve” en MATLAB R2012a .....	105
Tabla 2.3. Opciones del Algoritmo Genético empleadas para el problema de Optimización del sistema en estudio .....	107
Tabla 3.1 Parámetros que se asumen constantes en el problema .....	133
Tabla 3.2. Tipos de Variables para la formulación matemática .....	134
Tabla 3.3. Variables Dinámicas.....	136
TABLA 4.1.1 Reporte de resultados del costo total para varias corridas del programa OptEStatPar.m (Anexo K) en el estado estacionario y probabilidad de mutación 0.1 y 0.3, topología de reactores en paralelo .....	164
TABLA 4.1.2. Reporte de resultados del costo total para varias corridas del programa OptEStatSer.m (Anexo L) en el estado estacionario y probabilidad de mutación 0.1 y 0.3, topología de reactores en serie.....	165
Tabla 4.1.3. Reporte de resultados del programa OptEstatPar_AG.m (Anexo K) para el estado estacionario. Topología de reactores en paralelo. Probabilidad de mutación 0.3 .....	166
Tabla 4.1.4. Reporte de resultados del programa OptEstatSer_AG.m (Anexo L) para el estado estacionario. Topología de reactores en serie. Probabilidad de mutación 0.3 .....	167
Tabla 4.1.5. Valores de $\varepsilon$ (ypsilon) en varias corridas del programa OnlyControlPar_AG.m del Anexo G, topología de reactores en paralelo. ....	169
Tabla 4.1.6. Valores de $\varepsilon$ (ypsilon) en varias corridas del programa OnlyControlSer_AG.m del Anexo H, topología de reactores en serie.....	169
Tabla 4.1.7. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de	

reactores en paralelo. Optimización de control mediante algoritmos genéticos. (Anexo G).....	172
Tabla 4.1.8. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. Optimización del control mediante algoritmos genéticos.....	173
Tabla 4.1.9. Resumen de los resultados obtenidos para los puntos de partida al diseño integrado. Topología de reactores en paralelo .....	174
Tabla 4.1.10. Resumen de los resultados obtenidos para los puntos de partida al diseño integrado. Topología de reactores en serie.....	174
Tabla 4.1.11. Reporte de resultados del programa OptDynPar.m.....	178
Tabla 4.1.12. Reporte de resultados del programa OptDynSer.m.....	179
Tabla 4.1.13. Tabla comparativa resultados de las topologías de reactores estudiadas.....	180
Tabla 4.1.14. Resumen de las dimensiones óptimas de los reactores.....	181
Tabla 4.1.15. Volumen de los reactores y área de transferencia de calor.....	182
Tabla 4.1.16. Comparación de los costos totales según el arreglo de los reactores .....	183
Tabla 4.1.17. Resultados finales de la síntesis y diseño integrado de reactores mezcla completa usando algoritmos genéticos.....	187
Tabla 4.2.1. Opciones del SA para el diseño integrado del sistema.....	189
Tabla 4.2.2. Reporte de resultados del programa OptEstatPar_SA.m (Anexo O) para el estado estacionario. Topología de reactores en paralelo.....	190
Tabla 4.2.3. Reporte de resultados del programa OptEstatSer_SA.m (Anexo P) para el estado estacionario. Topología de reactores en serie .....	190
Tabla 4.2.4. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo. Optimización de control mediante SA.....	193



Tabla 4.2.5. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. Optimización de control mediante SA.....	194
Tabla 4.2.6. Resumen de los resultados obtenidos para los puntos de partida al Diseño Integrado. Topología de reactores en paralelo .....	195
Tabla 4.2.7. Resumen de los resultados obtenidos para los puntos de partida al Diseño Integrado. Topología de reactores en serie.....	195
Tabla 4.2.8. Resultados del Diseño Integrado mediante método híbrido (SA + SQP). Comparación de topologías serie y paralelo .....	201
Tabla 4.2.9. Volumen de los reactores y área de transferencia de calor diseño integrado método híbrido (SA/SQP) .....	204
Tabla 4.2.10. Resultados finales de la síntesis y diseño integrado de reactores mezcla completa usando SA.....	205
Tabla 4.3.1. Opciones del GBL para el Diseño Integrado del Sistema .....	207
Tabla 4.3.2. Reporte de resultados del programa OptEstatPar_GBL.m (Anexo S) para el estado estacionario. Topología de reactores en paralelo.....	208
Tabla 4.3.3. Reporte de resultados del programa OptEstatSer_GBL.m (Anexo T) para el estado estacionario. Topología de reactores en serie .....	208
Tabla 4.3.4. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo. Optimización de control mediante GBL.....	211
Tabla 4.3.5. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. Optimización de control mediante GBL.....	212
Tabla 4.3.6. Resumen de los resultados obtenidos para los puntos de partida al Diseño Integrado. Topología de reactores en paralelo. GBL .....	213
Tabla 4.3.7. Resumen de los resultados obtenidos para los puntos de partida al Diseño Integrado. Topología de reactores en serie. GBL.....	213

Tabla 4.3.8. Resultados del Diseño Integrado mediante método híbrido (GBL + SQP). Comparación de topologías serie y paralelo .....	217
Tabla 4.3.9. Resultado final de la síntesis y diseño integrado de reactores mezcla completa usando GBL .....	219
Tabla 4.4.1. Resultados finales de la síntesis y diseño integrado de una red de reactores mezcla completa en serie.....	221
Tabla 4.4.2. Costo computacional asociado a los métodos estudiados .....	224
Tabla 5.1. Identificación de los Anexos .....	237

## ÍNDICE DE GRÁFICAS

Gráfica 4.1.1. Respuesta dinámica del error cuadrático integral (ISE) para topología en paralelo.....	170
Gráfica 4.1.2. Respuesta dinámica del error cuadrático integral (ISE) para topología en serie.....	171
Gráfica 4.1.3. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en paralelo .....	176
Gráfica 4.1.4. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en serie.....	177
Gráfica 4.1.5. Respuesta dinámica de la temperatura de salida del primer reactor. diseño integrado para el arreglo de reactores en paralelo.....	184
Gráfica 4.1.6. Respuesta dinámica de la temperatura de salida del segundo reactor. diseño integrado para el arreglo de reactores en paralelo.....	185
Gráfica 4.1.7. Respuesta dinámica de la temperatura final. diseño integrado para el arreglo de reactores en paralelo .....	186
Gráfica 4.1.8. Respuesta dinámica de la temperatura a la salida del segundo reactor. Diseño integrado para el arreglo de reactores en serie .....	186
Gráfica 4.2.1. Respuesta dinámica del error cuadrático integral (ISE) para topología en paralelo.....	191
Gráfica 4.2.2. respuesta dinámica del error cuadrático integral (ISE) para topología en serie .....	192
Gráfica 4.2.3. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en paralelo. Hibridación SA y SQP .....	198

Gráfica 4.2.4. Relación entre el costo y la controlabilidad del sistema para la Topología de reactores en serie. Hibridación SA y SQP .....	199
Gráfica 4.2.5. Respuesta dinámica de la temperatura a la salida del segundo reactor. Topología de reactores en paralelo. Hibridación SA y SQP .....	202
Gráfica 4.2.6. Respuesta dinámica de la temperatura a la salida del segundo reactor. Topología de reactores en serie. Hibridación SA y SQP .....	203
Gráfica 4.3.1. Respuesta dinámica del error cuadrático integral (ISE) para topología en paralelo. GBL.....	209
Gráfica 4.3.2. Respuesta dinámica del error cuadrático integral (ISE) para topología en serie. GBL.....	210
Gráfica 4.3.3. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en paralelo. Hibridación GBL y SQP .....	214
Gráfica 4.3.4. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en serie. Hibridación GBL y SQP .....	215
Gráfica 4.3.5. Respuesta dinámica de la temperatura a la salida de los reactores. Topología de reactores en paralelo. Hibridación GBL y SQP .....	218
Gráfica 4.3.6. Respuesta dinámica de la temperatura a la salida del segundo reactor. Topología de reactores en serie.....	218
Gráfica 4.4.1. Respuesta dinámica de la temperatura a la salida del segundo reactor. Diseño integrado para el arreglo de reactores en serie, (a)algoritmos genéticos, (b)recocido simulado, (c)GBL.....	222

## INTRODUCCIÓN

La necesidad de evaluar las características de operabilidad dinámica de los procesos durante las fases de diseño ha sido conocida desde hace tiempo, pero cuantitativamente las aproximaciones para dirigir este tipo de problemas son bastante novedosas, siendo la interacción entre diseño y controlabilidad uno de los temas más relevantes en el diseño de procesos reactivos. Cabe mencionar que en las últimas cinco décadas se ha venido realizando un esfuerzo significativo dirigido a los estudios algorítmicos y teorías de aplicaciones que surgen del problema de diseño de procesos y control. Por otra parte se sabe que los procesos son de naturaleza dinámica, en ellos siempre ocurrirán cambios y si no se emprenden las acciones pertinentes, las variables importantes del proceso, es decir, aquellas que se relacionan con la seguridad, la calidad del producto y los índices de producción, no cumplirán con las condiciones de diseño.

La metodología comúnmente empleada por los ingenieros de procesos para abordar el diseño y síntesis de los mismos, consiste en realizar el modelado matemático con el uso de balances de masa y energía en estado estacionario sin tener en cuenta su dinámica. Después de realizado el diseño, el ingeniero de control e instrumentación le añade el sistema de control que permita la correcta operación del proceso, lo cual a pesar de ser factible no necesariamente es el mejor, en buena parte debido a los aspectos de integración entre el diseño y la controlabilidad dinámica del proceso o también a los estrictos requisitos de calidad. Esta técnica para el diseño es denominado “enfoque secuencial o tradicional”.

Se conoce que un proceso químico no controlado carece de sentido para llevar a cabo una reacción y obtener un producto. Esto es tan fácil de visualizar, pero a la hora de diseñar el control del proceso, por lo general se sigue haciendo aparte del diseño y síntesis del proceso. Por lo tanto, es equivalente desde el punto de vista económico a dejar un proceso sin control, ya que los costos que se generan por obtener un diseño que no es equilibrado en cuanto a los parámetros de sus controlador y las dimensiones de sus unidades, generará pérdidas y no será rentable en su funcionamiento a lo largo de su vida útil.

Es por esto que ha surgido la disciplina del diseño integrado de procesos, que salva estas dificultades al considerar la dinámica del proceso y su control conjuntamente desde la fase misma de diseño, garantizando que el sistema será después operable frente a posibles perturbaciones.

La idea del diseño integrado se fundamenta en la posibilidad de diseñar procesos con criterios de optimización, incorporando condiciones que garanticen un buen desempeño dinámico del sistema desde las primeras etapas del diseño. De esta forma se obtienen tanto los parámetros de un equipo, un proceso o una planta así como también los del controlador, pero de forma simultánea.

Hoy en día, el uso generalizado de los procesadores en la programación matemática está muy extendido, y no queda por fuera el área de la Ingeniería Química. En este punto, el Ingeniero Químico se une a la rama de la Ingeniería de Control, y se eliminan las barreras que pudieron haber existido en la compaginación de ambas especialidades.

El empleo de algoritmos de programación no es una tarea sencilla a la hora de ser aplicada a un proceso químico particular. Esto acarrea un estudio sistemático y

metodológico para no obviar ningún factor que pueda influenciar tanto en la inversión inicial como en los flujos de caja posteriores inherentes a mantener la estabilidad del sistema.

Por todo esto, resulta crucial la existencia, desarrollo y documentación de trabajos como el que a continuación se presenta, ya que permiten disponer de estrategias metodológicas que despejen el camino de los diseñadores de procesos y de control.

A lo largo del estudio que se expone en las siguientes páginas, se ha hecho un esfuerzo significativo por explorar distintas técnicas de optimización para el diseño integrado de reactores mezcla completa para un proceso químico específico. se trata con los reactores mezcla completa, dada su importancia y presencia dentro de los procesos químicos industriales.

Todo este esfuerzo está encaminado a contribuir metodológicamente para brindar las mejores opciones al diseñador de procesos, suministrando el desglose a detalle de la metodología para la optimización simultánea del diseño y control a través de un algoritmo de solución matemática para el caso de estudio.

Se obtiene una estrategia de solución para un problema de este tipo que puede ser empleado a nivel industrial para minimizar el esfuerzo para crear, evaluar y optimizar el diseño y control de reactores mezcla completa.

Las diferentes estrategias metodológicas presentadas servirán para seleccionar la vía más útil para diseñar de forma integrada reactores mezcla completa, de acuerdo a las necesidades particulares y serán la base para futuras investigaciones en otros procesos y/o unidades de procesos.

El trabajo está estructurado en cinco capítulos. El primero de ellos efectúa el planteamiento del problema, se trazan los objetivos y se delimita el estudio. El capítulo segundo presenta el estado del arte que enmarca al diseño integrado de procesos, este capítulo también contiene los fundamentos teóricos necesarios para la comprensión del tema, así como una revisión de los métodos de optimización empleados y la manera en la que pueden ser efectivamente implementados en la herramienta computacional seleccionada. El tercer capítulo adentra al lector en una serie de pasos bien articulados para alcanzar el objetivo propuesto, a través del procedimiento de trabajo. Ya en el capítulo cuarto se implementa el algoritmo matemático de resolución del problema en MATLAB R2012a, usando los distintos métodos de optimización. El último capítulo recopila los hallazgos del estudio en forma de conclusiones y recomendaciones. Los anexos al trabajo, reúnen toda la modelación matemática y la programación efectuada en forma de “scripts”, sin obviar detalle.



# CAPÍTULO I

## EL PROBLEMA

## **CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA**

Una de las tareas del ingeniero químico y de procesos cuando se encuentra en el ámbito laboral es el diseño eficiente de plantas químicas y de sus procesos. Un caso muy común se refiere al diseño de reactores mezcla completa.

En la actualidad se conoce que el diseño eficiente de un proceso conjuga la optimización de los objetivos económicos y la estabilidad del proceso. Pero no siempre es fácil lograr ese diseño eficiente cuando se diseña tomando en cuenta estos aspectos por separado, ya que al obtener un proceso económicamente óptimo, es probable que no se haya tomado en consideración los mejores parámetros que mantengan la operación estable del mismo. Es aquí donde surge el dilema y la mayoría de los casos se deja a un lado el control del proceso, ignorando que a lo largo de la vida útil de la planta el control tendrá también un impacto económico.

Por lo antes expuesto, se plantea la posibilidad de obtener un diseño óptimo del proceso en el que los objetivos económicos y la estabilidad del proceso sean considerados simultáneamente al inicio del diseño. Al introducir el enfoque integrado del diseño de procesos, tomando el caso de la red de reactores mezcla completa, surge un planteamiento matemático en el que se origina un problema de optimización multiobjetivo, un objetivo es la minimización los costos totales y el otro es la minimización del error cuadrático integral. Para resolver el problema multiobjetivo se convierte uno de los objetivos en restricción, en este caso el error cuadrático integral pasa a formar parte de las restricciones no lineales del problema, lo que implica resolver el problema multiobjetivo de control óptimo

entero mixto (MIOCP) a diferentes valores, los cuales van desde el óptimo controlable hasta el óptimo económico, en este punto el problema multiobjetivo se convierte en un problema de programación no lineal entera mixta con ecuaciones algebraico diferenciales MINLP-DAE.

Para el desarrollo del trabajo de grado se efectuará el diseño integrado de una red de reactores conformada por dos reactores mezcla completa con chaquetas de enfriamiento en donde ocurre una reacción irreversible de primer orden elemental y exotérmica.

Se considera desde las primeras etapas del planteamiento metodológico la estructura del proceso, efectuando la síntesis del mismo en cuanto a su topología, esto traduce el problema en uno de programación no lineal entera con ecuaciones algebraico diferenciales (NLP/DAE).

Se implementará el algoritmo de solución matemática en una herramienta computacional accesible, la cual incluya resolvidores de ecuaciones algebraico diferenciales (DAE) y de problemas de optimización no lineal (NLP).

El objetivo de efectuar el diseño integrado para este proceso es determinar las dimensiones físicas de las unidades, condiciones de operación y parámetros de entonación del control que optimicen tanto los criterios económicos, como la controlabilidad del proceso y a la vez garantice una operación factible. Es así como se pretende resolver el diseño integrado de una red de reactores mediante un estudio sistemático de la aplicación de diferentes técnicas de optimización, clásicas y heurísticas.

Académicamente el estudio aportará una contribución metodológica al diseño integrado de reactores mezcla completa, mediante un estudio sistemático de la

aplicación de diferentes técnicas de optimización, con el objetivo de presentar las ventajas y limitaciones de los diferentes métodos, así como diferentes esquemas e implementaciones del algoritmo para, en último término, obtener el máximo rendimiento de los métodos de optimización tanto locales como globales al problema específico, como lo representa el caso del diseño integrado de una red de reactores homogéneos.

### **OBJETIVO GENERAL**

Contribuir metodológicamente, mediante un estudio sistemático de la aplicación de diferentes técnicas de optimización al diseño integrado de reactores mezcla completa.

### **OBJETIVOS ESPECÍFICOS**

- Plantear todos los aspectos técnicos necesarios que describan el proceso, definiendo el conjunto de variables y parámetros, así como los niveles deseados de las salidas y restricciones del caso específico a diseñar.
- Establecer la estructura del proceso con su respectiva modelación matemática para la resolución del problema de diseño integrado del sistema, empleando métodos de optimización clásica y heurística.
- Determinar la técnica de optimización de mejor desempeño en la aplicación del diseño integrado de reactores mezcla completa.

## JUSTIFICACIÓN

En las últimas dos décadas se han venido ideando formas de mejorar los procesos. El incremento de la demanda y la masificación de la producción requieren procesos más eficientes tanto en tiempo como en calidad y ganancias. Antes el problema era solamente producir con calidad a bajos costos, luego mantener esa calidad, ahora es el tiempo y los costos de mantenimiento, sin perder producción, sin perder tiempo en paradas imprevistas de planta y cumpliendo la normativa ambiental. Todo eso ha llevado no sólo a diseñar procesos de manera óptima sino también a mantener operando el proceso de manera estable a pesar de las perturbaciones, por lo que el control de procesos se hace cada vez más indispensable.

En las áreas del conocimiento de procesos y optimización se han probado y puesto en práctica programas de optimización y simulación de procesos altamente eficientes y se ha desarrollado aún más el diseño del sistema de control para el cual también existen numerosos estudios y lo que es más importante, la puesta en práctica de forma eficiente a nivel industrial, que al fin y al cabo es el objetivo de la ingeniería.

Ahora surge la necesidad de integrar las disciplinas de procesos y control. Mantener aislado o diseñar de forma separadas el proceso y la arquitectura del control, es algo que ha traído preocupaciones, debido a que esto a la larga no refleja el impacto del sistema de control en las operaciones económicas, así como tampoco sobre la relación entre el diseño y la controlabilidad, lo que niega el

hecho de que la controlabilidad dinámica del proceso es una característica inherente a su diseño.

Es decir, un proceso que es óptimo en el estado estacionario deja de serlo cuando en él ocurren pequeños cambios y debido a ello la planta no exhibirá buenas operaciones económicas a través del tiempo. Las razones que podrían causar éste efecto pueden situarse en los gastos necesarios para adaptar el proceso a los cambios. De esta forma, una planta puede quedar en desventaja en relación a la creación o en todo caso a la adaptación de sus procesos hacia unos más integrados y más complejos repercutiendo negativamente en la productividad, calidad del producto y competitividad en el mercado. Por lo tanto, la operabilidad del proceso o su capacidad de adaptarse a pequeños cambios externos es una característica muy importante que debe ser considerada con seriedad en el diseño.

Se tienen así dos objetivos que deben plantearse simultáneamente en la tarea de síntesis de procesos: por un lado están las metas u objetivos a alcanzar durante el diseño del proceso, como la determinación de la estructura, la determinación de las condiciones operativas del proceso, la productividad y rentabilidad, por otro lado las características de controlabilidad, como lo son las estrategias de control que aseguren un comportamiento dinámico estable y que se satisfagan los requerimientos de calidad del producto.

Surge así el enfoque sobre el cual se basa el presente trabajo, el Diseño Integrado, denominado así por Gutiérrez y de Prada en el año 2003. El diseño integrado es la técnica de diseño de procesos mediante la cual se obtienen los parámetros físicos de la planta que minimicen el coste de operación y construcción, al mismo tiempo que cumplen con las características de controlabilidad que se imponen al proceso, obteniendo de esta manera diseños de planta flexibles capaces de mantener sus

condiciones de controlabilidad frente a futuras perturbaciones que se puedan presentar.

Todo estudio relativo a desarrollar ampliamente el diseño integrado de procesos constituye un avance más hacia la eficiencia en la producción. Ensayar distintos métodos de resolución del problema hasta obtener el que más se adapte a un proceso específico contribuirá a la extrapolación del empleo de estas técnicas de diseño a nivel industrial y de esta manera poder salvar todas las dificultades generadas por el diseño secuencial del proceso y de los sistemas de control.

Es pues así que surge la importancia de la presente investigación, que se justifica primeramente desde el punto de vista académico, al generar los antecedentes y la base científica de conocimiento que permitan dar un respaldo a la implementación del diseño integrado de procesos y por supuesto a nivel industrial, ya que con este tipo de investigaciones se estudian posibles vías para garantizar que una vez diseñado el proceso se generen en la práctica procesos de producción más eficientes. La promoción y masificación de estos trabajos permitirá ensayar todas las posibles técnicas de optimización para resolver el problema mencionado.

Hasta ahora, el problema específico de diseño integrado de procesos del sistema descrito por Schweiger y Floudas (1997) se ha resuelto mediante métodos de optimización local. Los algoritmos desarrollados y la simulación implementada arrojan resultados que ponen de manifiesto las ventajas del enfoque simultáneo de diseño sobre el enfoque secuencial. Ya que este hecho está plenamente probado, queda experimentar distintos métodos que produzcan la mejor solución en el espacio de búsqueda, alejándose del punto de partida para obtener óptimos globales. Para evitar la situación de caer en óptimos locales, el presente trabajo pretende emplear las diferentes técnicas de optimización (clásicas y heurísticas) al

sistema de estudio y de esta manera realizar una contribución metodológica al diseño integrado de reactores mezcla completa.

### **ALCANCE**

El trabajo propuesto se limita a efectuar el diseño integrado de una red de reactores conformada por dos reactores mezcla completa con chaquetas de enfriamiento en donde ocurre una reacción de primer orden elemental irreversible y exotérmica, usando como herramienta métodos de optimización clásicos y heurísticos. Comprenderá la determinación de la topología del sistema (tipo de interconexión) las dimensiones físicas de las unidades de procesos, condiciones de operación y los parámetros de entonación, así como también la relación cuantitativa entre los objetivos económicos del estado estacionario y el objetivo de controlabilidad dinámica.

En este trabajo se resuelve el problema de diseño integrado de una red de reactores, específicamente para el sistema definido, mediante un estudio sistemático de la aplicación de diferentes técnicas de optimización. el estudio se limita a la utilización de las técnicas del método del gradiente, algoritmo DIRECT, algoritmos genéticos y recocido simulado.



# **CAPÍTULO II**

## **MARCO TEÓRICO**

## **CAPÍTULO II. MARCO TEÓRICO**

### **SECCIÓN 1. ESTADO DEL ARTE**

El efecto del diseño sobre la controlabilidad no es nuevo. El concepto se encuentra en la literatura al menos hace 50 años, y se ha reportado en la industria al menos hace 30, donde entonces se mencionaba su importancia.

Uno de los primeros informes de los problemas en la industria causados por la ausencia de controlabilidad fue realizado por Anderson (1996): “Se diseñó inadecuadamente un efluente de alimentación del sistema del intercambiador de calor. La energía de recirculación introducía una realimentación positiva que desestabilizaba el sistema para ciertas condiciones de operación. La planta no comenzó a funcionar a toda capacidad hasta que no fue rediseñado el efluente de alimentación del sistema de intercambiador de calor”.

Las primeras ideas de introducir la controlabilidad en el diseño fueron planteadas por Nishida e Ichikawa en 1975 y 1976. Ellos fueron los primeros que estudiaron sistemáticamente el problema de incluir consideraciones dinámicas y de control dentro del problema de síntesis de procesos que contienen incertidumbres en los parámetros.

Sobre la evaluación de la controlabilidad basado en modelos en estado estacionario, se puede citar el trabajo de Douglas y colaboradores (1988) que

describen un procedimiento sistemático para evaluar la controlabilidad del proceso en las etapas de diseño, así como algunos aspectos económicos asociados con el control. Para mejorar la controlabilidad en lazo abierto sugieren:

1. Modificar las líneas de flujos que adicionan variables manipuladas.
2. Rediseñar unidades y piezas del equipamiento, para conseguir que las restricciones sobre el proceso nunca se vuelvan activas para todo el rango de las perturbaciones.
3. Ignorar la optimización de las variables de operación menos importantes para ampliar los grados de libertad.

El objetivo del análisis de controlabilidad que ellos persiguen es determinar la alternativa del menor costo de penalidad.

Marselle (1982) fue el primero en definir el problema de controlabilidad para una red de intercambiadores de calor. En ese artículo el procedimiento que se sigue es obtener diseños que manipulen el flujo de vapor y las fluctuaciones de temperatura con un máximo de eficiencia energética. Este problema fue abordado más tarde por Linnhoff y Kotjabasakis (1986) dirigido a la compensación entre controlabilidad y coste económico de un modo más efectivo a través de “condensados” y “tablas de sensibilidad”. Georgiou y Floudas (1989) propusieron un procedimiento de síntesis automático basado en una técnica de superestructura que generará tanto una red de recuperación de calor con bajo coste de capital y un rechazo total de perturbaciones, en el caso de no conseguir un rechazo total, disminuiría su propagación.

Sheffield (1992) proporcionó una perspectiva industrial sobre la necesidad de integrar el diseño y el control de sistemas. Tanto Morari (1992) como Perkins

(1989), han realizado una recopilación de algunos de los resultados importantes y esfuerzos previos, concernientes a la interacción entre el diseño y el control.

Brengel y Seider (1992) presentan una técnica para determinar el diseño de procesos óptimo operacionalmente y en estado estacionario. La controlabilidad de los diseños se evalúa con su comportamiento económico incorporando un algoritmo de control predictivo basado en modelos dentro del algoritmo de optimización de diseño del proceso, esta técnica coordinada, utiliza una función objetivo que es una suma de los pesos de las magnitudes económicas y de controlabilidad.

Sobre la evaluación de la controlabilidad basado en modelos lineales dinámicos, puede decirse que, dado un modelo lineal dinámico del proceso y dados algunos objetivos relacionados al comportamiento deseado en lazo cerrado y algunas restricciones sobre la estructura del controlador, el objetivo es encontrar los parámetros óptimos del controlador y comparar a través de una magnitud de controlabilidad el comportamiento óptimo que se alcanza entre diferentes diseños.

Morari y Skogestad (Morari, 1983; Skogestad y Morari, 1987b); Morari y Zafiriou, (1989); Skogestad, (1991); Skogestad y Wolff (1992)), han realizado contribuciones significantes en el análisis de la controlabilidad y en el estudio de la capacidad de adaptación dinámica de los sistemas. Ellos han introducido y analizado magnitudes de controlabilidad para la interacción de las variables y el rechazo de perturbaciones.

También han investigado las características inherentes a un proceso que limitan su controlabilidad y han propuesto un procedimiento de síntesis de reguladores que consideran la estabilidad robusta en presencia de incertidumbres en los modelos y acciones en las perturbaciones.

La función objetivo tradicional en control óptimo es la integral del cuadrado del error (ISE) y se ha utilizado para distinguir diseños alternos, por ejemplo, Lenhof y Morari (1982) utilizaron un criterio de comportamiento múltiple: ISE, desviación máxima de la respuesta de salida, máxima magnitud de la variable de control y la magnitud de saturación.

Jacobsen (1991) utilizó controladores PID óptimos para evaluar varios diseños de una columna de destilación azeotrópico homogéneo, demostrando que el valor singular estructurado es una buena medida de la robustez del sistema resultante frente a incertidumbres en el modelo.

Morari (1983) identificó las siguientes características del proceso que limitan el comportamiento del controlador independientemente del diseño del mismo: comportamiento de fase no mínima, limitaciones de los actuadores y las incertidumbres del modelo. Ocasionalmente, la magnitud del ruido puede tener un efecto dominante. Algunos investigadores han estudiado el efecto de los retardos, los ceros en la parte derecha del plano y las incertidumbres del modelo de forma individual al comportamiento en lazo cerrado.

Holt (1984) analizó los retardos multivariables. Psarris y Floudas (1991) utilizan un procedimiento de optimización lineal mixta entera, identifican límites exactos sobre el comportamiento que se alcanza con desacoplamiento de restricciones. Las propiedades de las transmisiones de ceros en la parte derecha del plano han sido investigadas por Holt (1984) y Morari y Zafiriou (1989). Se han identificado las condiciones bajo las cuales el desacoplamiento es la integral del error al cuadrado. Este trabajo fue seguido por un tratamiento más completo por Psarris y Floudas

(1991), quienes trataron el caso de un número infinito de ceros que usualmente surgen en presencia de retardos multivariados.

Psarris y Floudas (1991) utilizaron como un indicador de la sensibilidad del lazo cerrado sobre el error del modelo, el número de condición de la matriz de transferencia de la planta como una función de la frecuencia.

Skogestad (1991) utiliza una combinación de la matriz de ganancias relativas de Bristol dependiente de la frecuencia (RGA) y de la ganancia de perturbación del sistema en lazo cerrado (CLDG) para enjuiciar la controlabilidad relativa de diseños alternativos. Estas herramientas han sido utilizadas en muchos estudios recientes. La RGA la introdujo Bristol (1966), y continúa siendo ampliamente utilizada en la industria como un indicador de controlabilidad.

Narraway y Perkins (1993), analizan el problema de selección de la estructura de control, donde utilizan un controlador perfecto junto con un modelo linealizado para formular un problema de programación lineal entera mixta donde las variables enteras indican los pares entre las variables manipuladas. Luyben y Floudas (1994) utilizaron técnicas de optimización multiobjetivo para considerar simultáneamente los aspectos económicos y de controlabilidad del diseño de columnas de destilación.

Rowe y colaboradores (2002) propusieron una metodología de diseño integrado, en la que se podría considerar la operación de un proceso como un grado de libertad en el diseño de la planta. Esta metodología, la aplicaron a un complejo químico, del tipo “respuesta instantánea”, es decir, entrega rápida del producto con las especificaciones demandadas por el consumidor, este tipo de planta cambia frecuentemente su punto de operación, lo que implica que la operación dinámica de la misma sea muy compleja y difícil.

Al trabajar con algoritmos de optimización, se sabe que no existe uno mejor que otro, que simplemente algunos son mejores a los demás en ciertas circunstancias, la mayoría tiene sus ventajas y desventajas claramente marcadas y por ello no es de extrañarse que se haya intentado con éxito desarrollar algoritmos híbridos, es decir, derivados de otros o mezclas que intentan sacarle el mayor provecho a las ventajas de cada uno de ellos.

Los algoritmos híbridos se han utilizado además para la reconstrucción de imágenes, de acuerdo con Liu y otros (2009), para el clásico problema del comerciante viajero de Zhou y otros (2009), y Pirona en el 2009 utilizó algoritmos híbridos para el diseño integrado de una red de reactores mezcla completa. En 2010, Elhossini y otros presentaron una eficiente estrategia basada en PSO (particle swarm optimization) u optimización por enjambre de partículas y sobre el “frente de Pareto” que generalmente se ve en algoritmos evolutivos, para resolver problemas de optimización multiobjetivo.

Quienes también utilizaron una estrategia híbrida son Martínez y Behar en el CIE (2011), basada en algoritmos genéticos y en el método simplex para la identificación experimental de sistemas de lazo abierto, obteniendo resultados muy superiores a los obtenidos por métodos convencionales, una estrategia muy similar llamada algoritmo simplex Genético realizada por Hongfeng (2009). Los fundamentos teóricos que sustentan estas investigaciones se exponen a continuación.

## SECCIÓN 2. BASES TEÓRICAS

### 2.1. DISEÑO DE PROCESOS Y EL ENFOQUE DEL DISEÑO INTEGRADO

#### 2.1.1. Diseño de procesos

Resulta complejo plantear en función de un esquema formal, la actividad de diseño en la ingeniería de procesos. En primer lugar y en forma simplificada, se podría afirmar que un proceso químico está vinculado al tratamiento de materiales, mediante transformaciones fisicoquímicas y/o biológicas, y/o procesos de separación física para obtener los productos deseados. Las operaciones destinadas a la transformación contienen equipos o unidades que realizan tareas específicas y se encuentran conectados entre sí por medio de las corrientes que los vinculan.

Existen dos grandes grupos de variables que deben ser diferenciadas. En este contexto están las variables estructurales, las cuales están íntimamente ligadas a la estructura del diagrama de flujo, es decir, que especifican la presencia de los distintos equipos y su diagrama de interconexiones. Por otro lado, están las variables de operación que representan condiciones operativas (temperatura, caudal, presión), y ciertas características funcionales de los equipos, como son áreas, número de etapas, entre otros, por lo general identificadas como parámetros de diseño. Existen numerosas variantes para el armado del diagrama de flujo de un proceso determinado, así como también, numerosas posibilidades para la



asignación/especificación de las variables de operación y parámetros de equipo. El diseñador deberá optar por algún conjunto de ellas, de manera tal de optimizar la estructura resultante en función de los distintos aspectos funcionales a considerar en la operación de plantas químicas.

Por consiguiente, la tarea de diseño lleva implícita la necesidad de seleccionar dentro de un espacio de posibilidades, aquellas alternativas que en base a un criterio predeterminado, cumplan con los objetivos deseados. En la búsqueda de lograr un planteamiento formal del problema de diseño, conviene establecer una función objetivo adecuada a ser optimizada. En este punto, conviene explorar los objetivos que se esperan de un proceso químico en general.

Es bien sabido que toda actividad productiva de bienes o servicios se encuentra permanentemente sujeta a un marco de referencia en función del cual se evalúa su factibilidad. El marco económico constituye la fuente cuyo veredicto resulta inevitable de contemplar en toda actividad de ingeniería. Por lo tanto, costos de operación, inversión y tasa de retorno, son criterios que generalmente se tienen en cuenta en la función objetivo a optimizar.

Ahora bien, no es el factor económico el único componente que define la operabilidad de un complejo o proceso químico. En efecto, existen una serie de aspectos que necesariamente deben ser contemplados, como por ejemplo, elasticidad que se refiere a la habilidad del proceso para tolerar condiciones adversas tales como perturbaciones o variación en los parámetros, o flexibilidad que se refiere a la capacidad estructural y operativa del proceso para mantenerse funcionando con el mejor desempeño cuando las condiciones operativas corresponden a un rango de condiciones de diseño, o controlabilidad, confiabilidad e impacto ambiental.

Consecuentemente, el problema a resolver es multiobjetivo (Takama y Umeda (1980). Estos problemas se caracterizan por no poseer una única solución. Por lo tanto, decidir entre varias soluciones, lleva implícito subjetivamente la preferencia del diseñador respecto de los objetivos.

Según Scenna (1999), el problema completo, si es que puede formalizarse, podría ser formulado desde el punto de vista de la programación matemática como uno de optimización multiobjetivo.

Optimice  $\{f_1(x), f_2(x), \dots, f_N(x)\}$

Sujeto a:

$$\mathbf{h(x)} = \mathbf{0}$$

$$\mathbf{g(x)} > \mathbf{0} \tag{1}$$

Donde:  $f_1(x)$ ,  $f_2(x)$ ,.....y  $f_N(x)$  representan las N funciones objetivos a optimizar (minimizar o maximizar),

$h(x)$  expresa las restricciones correspondientes a los balances de materia y energía,

$g(x)$  son las restricciones que expresan zonas de operación factible, por ejemplo composiciones positivas, temperaturas positivas y restricciones propias del proceso.

El vector  $x$  comprende variables o parámetros de operación y variables estructurales.

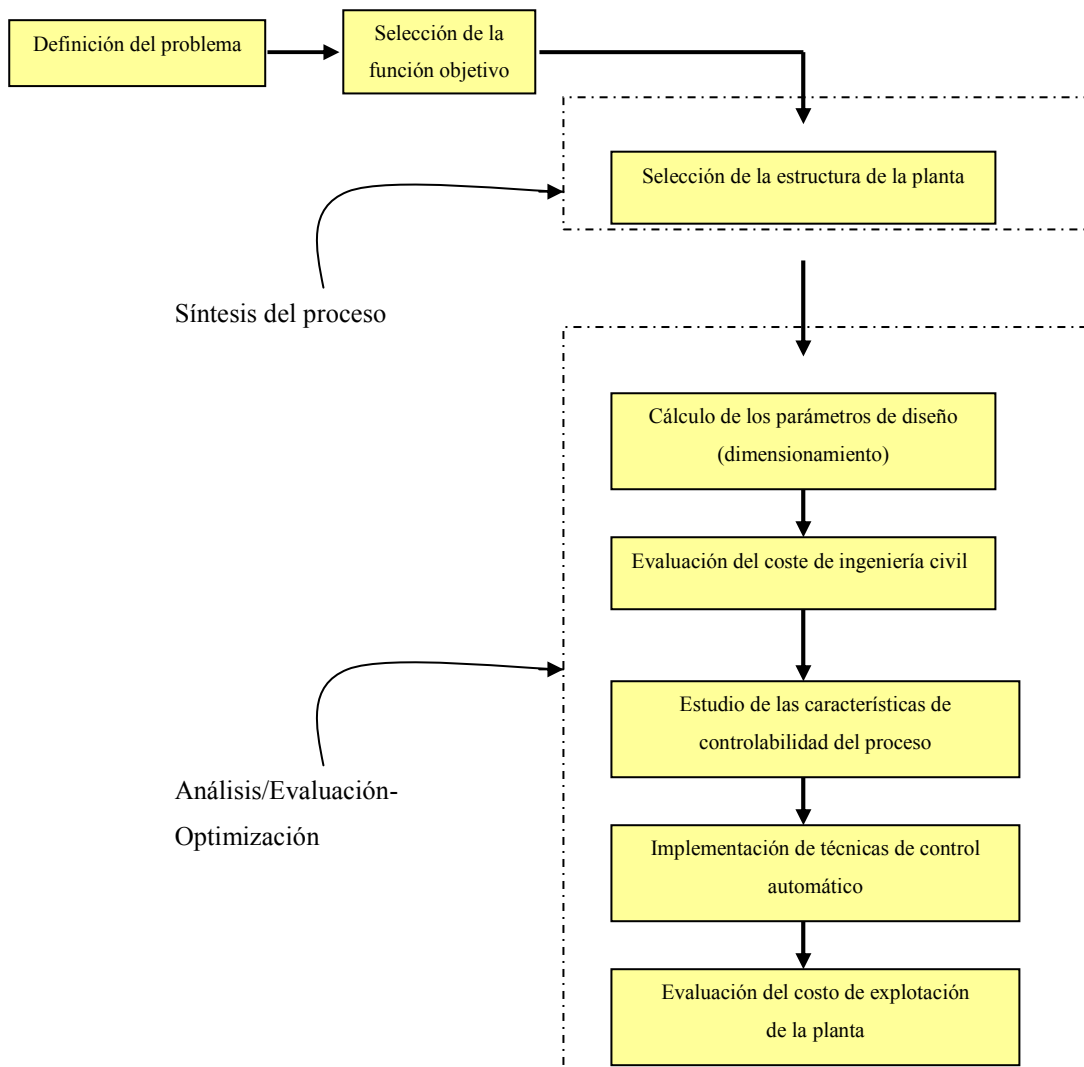
Una simplificación que se suele adoptar consiste en expresar todos los objetivos en una “única medida”, que generalmente es económica. Obviamente, este enfoque, además de ser una simplificación, no siempre es posible. Por último, aún con una única función objetivo, se deben resolver aún los otros problemas, por ejemplo, cómo manejar la variable estructural. En este contexto, debe encontrarse una metodología adecuada para resolver el problema matemático resultante.

El problema de diseño planteado previamente (esto es, optimización multiobjetivo no lineal, con variables de decisión tanto continuas como discretas, variables estructurales, variables aleatorias, funciones no lineales y funciones difícilmente expresables matemáticamente), resulta de muy difícil solución.

Para modelar un problema complejo como el descrito, que no es posible resolver basándose en una herramienta o algoritmo matemático que involucre todos los aspectos a considerar en forma simultánea, se plantean actualmente dos enfoques distintos: el enfoque tradicional de diseño de procesos o secuencial y el enfoque de Diseño Integrado.

#### **2.1.1.1. El enfoque tradicional de diseño de procesos (enfoque secuencial)**

Para efectuar el diseño de procesos químicos, se realiza tradicionalmente el diseño de la planta y el diseño y análisis del sistema de control de manera secuencial. Inicialmente el ingeniero de procesos determina la estructura del diagrama de flujo, las condiciones de operación en estado estacionario que se requieren para alcanzar los objetivos de producción y el cálculo de los parámetros físicos de la planta (tamaño de las unidades del proceso). Ver Figura 2.1.



**Figura 2.1. Metodología del enfoque secuencial.** Fuente: Gutiérrez. de Prada, (2003) y Quintana (2003)

Hasta ahora la tarea de diseño se ha centrado en optimizar un índice económico (minimizar el costo anual de mantenimiento, maximizar los beneficios o ganancias o minimizar el costo de construcción), considerando únicamente la operación de la planta en estado estacionario, evaluando un gran número de estructuras alternativas, de condiciones de diseños y parámetros que satisfagan los

requerimientos de operación establecidos, seleccionando finalmente la más económica tanto en gastos de costos de mantenimiento como de construcción de las mismas. En esta etapa se le ha brindado una menor atención a la controlabilidad dinámica del proceso. Una vez diseñada la planta, el ingeniero en control se encarga de establecer una estrategia de control que asegure un comportamiento dinámico estable y que satisfaga los requerimientos de calidad del producto.

Estos objetivos deben alcanzarse ante la presencia de grandes perturbaciones, de fallos de los equipos, variaciones de carga, entre otros. Frecuentemente esta es una tarea difícil puesto que la controlabilidad del proceso, que debería ser una parte integral en las etapas de diseño, no se tiene en cuenta previamente.

Dada la elevada magnitud y complejidad del problema, es muy útil adoptar la estrategia de subdividirlo en subproblemas que sean factibles de ser abordados y, a partir de las soluciones parciales generadas, encontrar la solución al problema global. Los postulados de la teoría general de sistemas según Umeda (1982) aconsejan una división del mismo en subproblemas más sencillos, con una determinada estrategia, para luego componer la solución a partir de las soluciones parciales.

La estrategia de descomposición óptima dependerá del problema de diseño a resolverse. Una de las estrategias más naturales y sencillas de comprender es la propuesta por Douglas (1988). El enfoque jerárquico descompone al problema completo en una serie de subproblemas usando una jerarquía de decisiones y se fija un orden de prioridades relativas al fijar la secuencia de subproblemas a resolverse.

Debido a la elevada dimensión de los problemas asociados a la tarea de diseño, su elevado índice combinatorio y la gran cantidad de información que debe manejarse, la única herramienta viable para tal fin resultan ser los métodos computacionales.

Además de no existir una herramienta computacional (programa o algoritmo) capaz de resolver el problema del ciclo completo de actividades en la tarea de diseño, debido a la gran variedad de alternativas que deben contemplarse, tampoco existe algoritmo efectivo para la síntesis de procesos genéricos completos. A medida que se simplifica (particiona) el problema, es decir, se adoptan subsistemas del complejo a diseñar, los algoritmos propuestos se caracterizan por su mayor contenido formal y su robustez.

#### **2.1.1.2. El Enfoque del Diseño Integrado**

Gutiérrez y de Prada (2003) señalan que el “Diseño Integrado” es una técnica o enfoque de diseño de plantas mediante el cual las características de controlabilidad son consideradas previamente en las etapas de diseño, lo que permite fijar especificaciones dinámicas del sistema que facilitará la operación y flexibilidad de las mismas que pueden ser gobernadas por un sistema de control automático.

La idea del Diseño Integrado se fundamenta en la posibilidad de diseñar procesos con criterios de optimización, incorporando condiciones que garanticen un buen desempeño dinámico del sistema en todas las primeras etapas del diseño. De ésta forma se obtienen tanto los parámetros de la planta como los del controlador en forma simultánea. La interacción entre el diseño de procesos y el control de

procesos surge porque el diseño de procesos químicos determina de un modo inherente su controlabilidad, que cualitativamente significa lo bien que un proceso es capaz de rechazar perturbaciones, cuán severamente interactúan las múltiples variables y cuán fácilmente el sistema se mueve de un punto de operación a otro.

Lyuben (1993) establece que la metodología de Diseño Integrado permite la evaluación de los parámetros de la planta y el sistema de control al mismo tiempo, haciendo que el sistema diseñado sea más fácilmente controlable. En la etapa de diseño se evalúan diversos índices de controlabilidad junto con consideraciones económicas para producir una planta óptima.

Esta metodología se plantea matemáticamente como un problema de optimización no lineal multiobjetivo con restricciones, donde los objetivos que se consideran son de tipo económico y de control, y las restricciones son los requerimientos de proceso, físicos y de controlabilidad que puede incluir: seguridad, flexibilidad, operabilidad, entre otros.

En el caso de la síntesis de reactores, si el número de reactores y la manera en que están interconectadas las unidades de proceso se encuentran establecido de antemano, la característica entera mixta no se presenta. El problema se traduce en un problema de optimización no lineal con ecuaciones algebraico diferenciales definidas estas últimas por la controlabilidad y los aspectos dinámicos del sistema. Este caso fue estudiado por Segura (2007).

En el diseño de plantas de procesos químicos se involucran fundamentalmente tres áreas: la síntesis de procesos, el diseño de procesos y la tarea del control de procesos.

De acuerdo con Nishida y otros (1981), en la tarea de síntesis de procesos se determina la interconexión óptima entre las unidades del proceso como también el tipo de diseño de las unidades dentro del sistema de procesos para alcanzar unos objetivos de producción específicos.

La tarea de diseño de procesos está enfocada a la determinación de las condiciones de operación de cada una de las unidades del proceso y los dimensionamientos que se requieren para la producción dicho material o producto específico.

Por otro lado la tarea del control de procesos se centra en establecer las pautas de operación que aseguren un comportamiento estable del proceso satisfaciendo sus requerimientos en condiciones desfavorables tales como la acción de perturbaciones e incertidumbres en los modelos empleados en el diseño del sistema de control.

La síntesis de procesos ha sido llevada a cabo tradicionalmente de manera manual e intuitiva, una vez que se proponen diferentes esquemas de diagramas de flujo, los ingenieros se centran en elegir el esquema de la planta económicamente óptimo desde el punto de vista de ingeniería civil, pero sin tener en cuenta la secuencia e interconexión óptima entre las unidades que lo componen, y sin considerar las características de controlabilidad en lazo abierto o en lazo cerrado que tendría la planta una vez diseñada.

Existen dos razones fundamentales que han condicionado el uso mayoritario de las técnicas de diseño de procesos tradicional antes mencionada, una ha sido la carencia de herramientas de diseño asistidas por ordenador que permitieran el estudio dinámico de plantas (en los últimos años, se evidencia un desarrollo pronunciado en esta área) y la segunda razón son las barreras histórica y tecnológica, existente entre el ingeniero de procesos y el ingeniero de control.



Actualmente, los avances tecnológicos en hardware y software permiten realizar el diseño simultáneo de procesos y sistemas de control teniendo en cuenta restricciones de operación, restricciones de tipo físico y que consideren la controlabilidad del conjunto.

La definición de controlabilidad no es única. Concretamente, Skogestad y Postlethwaite (1997), definieron el concepto de controlabilidad como “la mejor calidad de respuesta que puede obtenerse mediante un controlador de retroalimentación”, esta definición general proporciona un marco de trabajo conceptual con el cual se puede abordar el problema del Diseño Integrado.

Algunos investigadores han apuntado a esta idea como fundamental, en lo referente a incorporar la controlabilidad como una magnitud en las etapas de diseño de los procesos. Conceptualmente se puede distinguir cuatro tipos de requerimientos:

- Flexibilidad: habilidad de un sistema para trabajar en diferentes puntos estacionarios.
- Permutabilidad: habilidad del sistema de moverse de un estado estacionario a otro y viceversa.
- Controlabilidad: la capacidad del proceso de alcanzar y mantener un valor de equilibrio deseado.
- Controlabilidad en lazo cerrado: se define como el “mejor” comportamiento dinámico del sistema (seguimiento de referencia y rechazo de perturbaciones), que puede alcanzar el sistema en lazo cerrado. Por definición esta característica no depende solo del controlador, sino de la propia planta, ya que la controlabilidad depende en gran medida de la

flexibilidad y permutabilidad que son proporcionadas por el propio diseño del proceso.

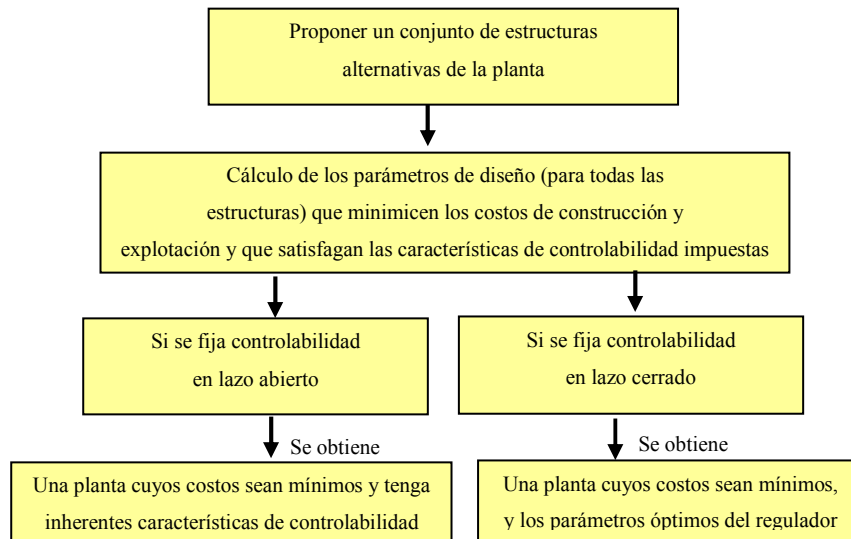
El problema de síntesis de procesos y de control conduce a la solución de tres problemas fundamentales:

- ¿Cuáles son las unidades del proceso que se deben incluir en el proceso?
- ¿Cómo deben interconectarse las unidades de proceso?
- ¿Qué condiciones de operación y dimensiones deben tener las unidades de proceso?

Estos problemas se resuelven utilizando algoritmos de Optimización No Lineal Mixta Entera con restricciones algebraicas y diferenciales. Debido a la complejidad del problema, la secuencia de la etapa de síntesis seguida por una de análisis es iterativa hasta que se encuentra un diseño satisfactorio.

Ambos pasos pueden involucrar desde el uso de modelos muy simples a bastante complejos; por ejemplo, técnicas auxiliares de computación que ayudan al ingeniero a manejar variables estructurales (generación de estructuras expandiéndolas en un árbol o método equivalente) y la evaluación de los costos (que involucra el diseño preliminar de los equipos y a su vez implica la resolución de los balances de materia y energía y a veces de cantidad de movimiento) para analizar su viabilidad técnica y económica. Para ello serán necesarias herramientas de simulación estacionaria y/o dinámica, programas de costo, optimización, programas de diseño de equipos, entre otros. El objetivo ideal, difícil de lograr, es incorporar y crear metodologías eficientes y métodos de diseño automatizados o asistidos por computadora, que contemplen todos estos aspectos en un modelo,

implementado en un procedimiento, metodología o herramienta única. El enfoque simultáneo o el enfoque de Diseño Integrado se ilustra en la Figura 2.2.



**Figura 2.2. Metodología del enfoque simultáneo.** Fuente: Gutiérrez y de Prada (2003)

## 2.2. DISEÑO DE REACTORES MEZCLA COMPLETA

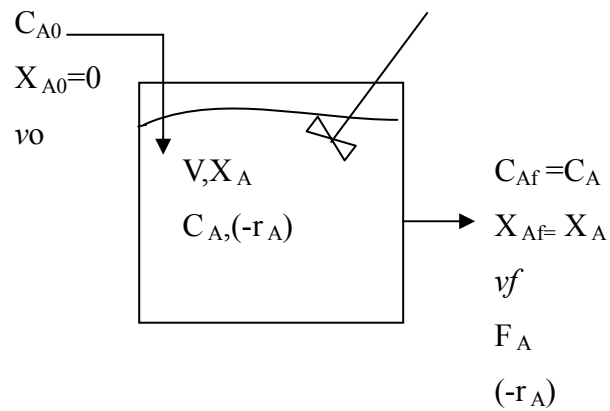
### 2.2.1. El reactor de mezcla completa

En un reactor de mezcla completa las propiedades no se modifican ni con el tiempo ni con la posición, cuando se supone que se está trabajando en estado de flujo estacionario y la mezcla de reacción es completamente uniforme. El tiempo de mezcla tiene que ser muy pequeño en comparación con el tiempo de permanencia en el reactor. En la práctica se puede llevar a cabo siempre que la mezcla fluida sea poco viscosa y esté bien agitada.

La ecuación de diseño para el reactor mezcla completa se obtiene efectuando el balance de un componente determinado en todo el reactor. Considerando el reactante A en la reacción del tipo irreversible de primer orden:  $A \rightarrow B$ , se puede escribir el balance de masa en estacionario y transitorio según las ecuaciones 2 y 3:

$$\text{ENTRADA} - \text{SALIDA} + \text{GENERACIÓN} - \text{REACCIONA} = \text{ACUMULACIÓN} \quad (2)$$

$$\text{ENTRADA} - \text{SALIDA} - \text{REACCIONA} = 0 \quad (3)$$



**Figura 2.3. Nomenclatura utilizada en un reactor mezcla completa.** Fuente: Levenspiel, O. (1998).

Donde:

$C_{A0}$  es la concentración de A a la entrada del reactor,

$X_{A0}$  es la conversión de A a la entrada del reactor que se toma como nula,

$v_0$  es el caudal volumétrico de la corriente de entrada al reactor ( $m^3/s$ )

$V$  es el volumen del reactor ( $m^3$ ),

$X_A$  es la conversión de A en el reactor,

$C_A$  es la concentración de A en el reactor [ $(mol\ A)/(m^3)$ ]

$(-r_A)$  representa la velocidad de reacción [ $(mol\ A)/(m^3.s)$ ]

$C_{Af}$  es la concentración de A a la salida del reactor [ $(mol\ A)/(m^3)$ ]

$X_{Af}$  es la conversión de A a la salida del reactor

$v_f$  es el caudal volumétrico de la corriente de salida del reactor ( $m^3/s$ )

$F_A$  flujo molar del componente A en la salida del reactor (mol A/s)

Como se muestra en la Figura 2.3, según Levenspiel, (1998) si se considera al reactor como un todo se tiene:

$$F_{A0} = v_0 C_{A0} \text{ (flujo molar del componente A en la alimentación al reactor)} \quad (4)$$

$$\text{Entrada de A, moles/tiempo} = F_{A0}(1 - X_{A0}) = F_{A0} \quad (5)$$

$$\text{Salida de A, moles/tiempo} = F_A = F_{A0}(1 - X_A) \quad (6)$$

$$\begin{array}{l} \text{Moles que reaccionan} \\ \text{de A por unidad de} \\ \text{tiempo (mol/s)} \end{array} = (-r_A)V = \left( \frac{\text{moles de A que reaccionan}}{(\text{tiempo})(\text{volumen de fluido})} \right) * \left( \begin{array}{l} \text{volumen} \\ \text{de reactor} \end{array} \right) \quad (7)$$

$F_{A0}$  son los moles de la especie A por unidad de tiempo referidos a conversión cero, (diferentes de los moles por unidad de tiempo en la entrada del reactor, que pueden introducirse en el reactor con o sin conversión).  $F_A$  son los moles de la especie A por unidad tiempo, referidos a conversión  $X_A$  y tiempo de residencia de las especies dentro del reactor.

De estas ecuaciones se obtiene:

$$F_{A0}X_A = (-r_A)V \quad (8)$$

$$\frac{V}{F_{A0}} = \frac{\tau}{C_{A0}} = \frac{\Delta X_A}{-r_A} = \frac{X_A}{-r_A} \quad (9)$$

$$\tau = \frac{1}{s} = \frac{V}{v_0} = \frac{VC_{AO}}{F_{AO}} = \frac{C_{AO}X_A}{-r_A} \quad (10)$$

Donde  $X_A$  y  $r_A$  se evalúan para las condiciones de la corriente de salida, que son iguales a las existentes dentro del reactor. De modo más general, si la alimentación sobre la que se basa la conversión (que se expresa con el subíndice 0) entra en el reactor parcialmente convertida (que expresamos con el subíndice i) y sale en las condiciones expresado con el subíndice f, para sistemas de densidad constante, se tiene que:

$$\frac{V}{F_{AO}} \frac{X_A}{-r_A} = \frac{C_{AO} - C_A}{C_{AO}(-r_A)} \quad (11)$$

$$\tau = \frac{V}{v_0} = \frac{C_{AO}X_A}{-r_A} = \frac{C_{AO} - C_A}{-r_A} \quad (12)$$

Estas expresiones relacionan de manera sencilla cuatro términos  $X_A$ ,  $-r_A$ ,  $V$  y  $F_{AO}$ ; por consiguiente, conociendo tres cualquiera de ellos se obtiene directamente el cuarto. Entonces, en el diseño, el tamaño del reactor necesario para una función determinada, o el grado de conversión en un reactor de tamaño conocido, se calculan directamente.

En el proceso en estado dinámico las variables tienen historia, son dependientes del tiempo, y las variables de salida dependen de los valores actuales y pasados de las variables de entrada. Si se considera nuevamente el reactante A en la reacción del tipo irreversible de primer orden:  $A \longrightarrow B$ , se aplica el balance de masa de la ecuación 2 y como se está en estado dinámico el término de acumulación no se anula, con lo que queda:

$$\text{Entrada} - \text{Salida} - \text{Reacciona} = \text{Acumulación} \quad (13)$$

$$F_{A0}(t) - F_A(t) - (-r_A(t))V = \frac{dNA}{dt} \quad (14)$$

Con respecto al balance de energía en estado dinámico se puede expresar:

$$\text{ENTRADA} - \text{SALIDA} - \text{REACCIONA} - \text{TRANSMITE} = \text{ACUMULACIÓN} \quad (15)$$

$$F \cdot \rho \cdot h_o - F \cdot \rho \cdot h - \Delta H \cdot r_A \cdot V - Q = \frac{dU}{dt} \quad (16)$$

Donde:

F = Flujo volumétrico (mol A/s)

p = Densidad del líquido en (kg/m<sup>3</sup>)

h<sub>o</sub> = Entalpía de la corriente de entrada, (J/kg)

h = Entalpía de la corriente de salida (J/kg)

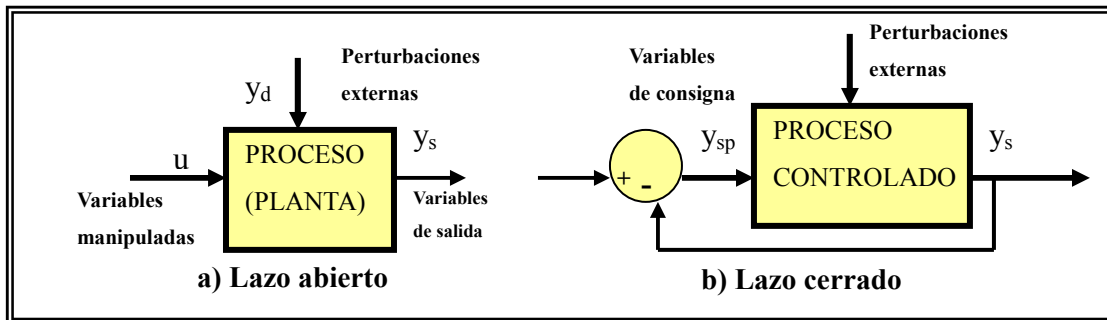
$\Delta H$  = Calor de reacción

U = Energía interna del líquido en el tanque, (J/kg).

Q = Flujo de calor por transferencia (J/s).

### 2.3. CONTROL DE PROCESOS

La Figura 2.4 (a) muestra un esquema global de un proceso no controlado, usualmente denominado la “planta” y un esquema en bloques de un proceso controlado se indica en la Figura 2.4 (b).



**Figura 2.4. Esquemas globales de procesos controlados y no controlados.**

Fuente: Scenna (1999).

Un proceso es controlado automáticamente cuando existen componentes operativos (los controladores) que permiten recibir el valor deseado o consigna, de manera tal que las variables de salida evolucionen automáticamente hacia esos valores especificados, aun en presencia de perturbaciones externas.

El Control de Procesos es definido por Smith y Corripio (1991) como una manera de mantener la variable controlada en el punto de control a pesar de las perturbaciones. Así, el objetivo del control de procesos es mantener en determinado valor de operación las variables del proceso tales como: temperaturas, presiones, flujos y compuestos. Ya que los procesos son de naturaleza dinámica, en



ellos siempre ocurrirán cambios y si no se emprenden las acciones pertinentes, las variables importantes del proceso, es decir, aquellas que se relacionan con la seguridad, la calidad del producto y los índices de producción, no cumplirán con las condiciones de diseño.

El control automático de procesos fue posible gracias al desarrollo de la tecnología digital, los sensores “en línea”, las comunicaciones digitales, entre otros. Con el advenimiento de sistemas de cómputo cada vez más potentes y con mayor capacidad de soporte de datos, se posibilitó la operación del proceso (o de la planta) en forma interrelacionada con información proveniente de sectores diversos de la empresa, permitiendo integrar datos provenientes de los sectores de producción, administrativo, de mantenimiento, de ventas, entre otros. Así, la decisión sobre la operación de toda la empresa (incluido el proceso específico) puede ser comandada directamente desde los niveles gerenciales. Más aun, las decisiones pueden ser tomadas no sólo ya por una empresa aislada, sino por un conjunto de empresas con un interés o una política productiva común.

Hoy en día es deseable disponer de sistemas de control cada vez más elaborados que permitan un alto grado de automatización del proceso, y que aseguren la obtención de productos finales con características de calidad ubicadas en un rango de especificación predeterminado en un entorno más seguro.

### **2.3.1 Los controladores clásicos**

En los sistemas de control, el controlador es el componente activo que recibe las señales de “set point” y las mediciones, y efectúa acciones de control para ajustar los valores de las variables manipuladas.

Internamente, en un controlador puede implementarse una ley de control muy sencilla (por ejemplo, sólo un efecto proporcional), o relativamente más compleja, como en el caso de los controladores no lineales con realimentación de estados.

Aunque actualmente es posible implementar algoritmos de control sofisticados, todavía un gran número de controladores utilizados en la industria son lineales con acciones proporcional, integral y derivativa (PID). En un controlador PID, la variable manipulada se relaciona con la señal de error a través de:

Controlador lineal Con acciones P+I+D

$$u(T) = K_p \varepsilon + \frac{K_p}{T_f} \int_0^t \varepsilon dt + K_p T_D \frac{d\varepsilon}{dt} + c(t) \quad (17)$$

Donde  $K_p$ ,  $TI$ , y  $TD$  son las constantes proporcional, integral y derivativa, respectivamente; y  $c(t)$  es la señal del controlador cuando . Las constantes  $K_p$ ,  $TI$ , y  $TD$  son los parámetros de ajuste del controlador.

El ajuste de los parámetros de un controlador se efectúa teniendo en cuenta criterios de estabilidad del sistema operando en lazo cerrado, y características preestablecidas para la evolución temporal de la respuesta, tales como el grado de amortiguamiento, el máximo sobrepaso, a este se le denomina “Criterio basado en algunos puntos de la respuesta”, también existen los basados en la respuesta completa, o “Criterio Integral”.

Si bien los controladores PID son estructuralmente adecuados para sistemas lineales, también es aceptable su uso en muchos sistemas no lineales (como lo son

la mayor parte de los procesos químicos), al menos dentro de ciertos rangos operativos.

En el caso de los criterios basados en la respuesta completa del sistema, el controlador se diseñará eligiendo los parámetros que minimizan la integral de alguna medida de error, ya sea con el uso de la integral del error cuadrático ISE, la integral del valor absoluto IAE o la integral del valor absoluto ponderado en el tiempo ITAE.

Obviamente, los parámetros resultantes de la aplicación de este o de otros métodos de ajuste, conforman solo una primera aproximación de los ajustes definitivos, que deberán ser posteriormente efectuados para cumplir con las especificaciones de control deseadas. Existen varias aproximaciones que pueden emplearse para ajustar el controlador, pero los objetivos que conllevan todas ellas pueden resumirse en base a las siguientes pautas:

- Rápida Respuesta
- Suprimir el efecto de las perturbaciones
- Insensible a los errores de modelado y mediciones
- Evitar una excesiva acción del controlador
- Sea adecuado sobre un amplio rango de condiciones operativas

#### **2.4. CONCEPTOS ASOCIADOS A LOS MODELOS MATEMÁTICOS DE OPTIMIZACIÓN**

**Función Objetivo.** Es la medida cuantitativa del funcionamiento del sistema que se desea optimizar (maximizar o minimizar). Como ejemplo de funciones objetivo se pueden mencionar: la minimización de los costos variables de operación de un

sistema eléctrico, la maximización de los beneficios netos de venta de ciertos productos, la minimización del cuadrado de las desviaciones con respecto a unos valores observados, la minimización del material utilizado en la fabricación de un producto, entre otros.

**Variabes.** 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 dependientes o auxiliares o de estado y variables independientes o principales o de control, aunque matemáticamente todas son iguales. En el caso de un sistema eléctrico serán los valores de producción de los grupos de generación o los flujos por las líneas. En el caso de las ventas, la cantidad de cada producto fabricado y vendido. En el caso de la fabricación de un producto, sus dimensiones físicas.

**Restricciones.** Representan el conjunto de relaciones (expresadas mediante ecuaciones e inecuaciones) que ciertas variables están obligadas a satisfacer. Por ejemplo las potencias máximas y mínimas de operación de un grupo de generación, la capacidad de producción de la fábrica para los diferentes productos, las dimensiones del material bruto del producto.

Resolver un problema de optimización consiste en encontrar el valor que deben tomar las variables para hacer óptima la función objetivo satisfaciendo el conjunto de restricciones.

Este problema puede ser representado en la siguiente forma:

*Dada:* una función  $f: A \rightarrow \mathbb{R}$  de algún conjunto  $A$  a los números reales

**(18)**

*Se busca:* un elemento  $x_0$  en  $A$  tal que  $f(x_0) \leq f(x)$  para todo  $x$  en  $A$  ("minimización") o tal que  $f(x_0) \geq f(x)$  para todo  $x$  en  $A$  ("maximización").

Tal formulación se llama "problema de optimización" o "problema de programación matemática" (un término que no está directamente relacionado con la programación de computadoras) pero aún en uso, por ejemplo "programación lineal". Muchos problemas teóricos y de la vida real pueden ser modelados en este marco.

**Región Factible.** Es el conjunto  $A$  de puntos  $x \in R^n$  que satisfacen simultáneamente todas las restricciones del programa matemático.

**Punto Factible:** es todo punto  $x \in A$ .

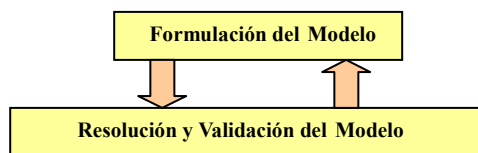
**Punto No Factible:** es todo punto  $x \notin A$ .

**Solución Óptima del Modelo.** Es todo punto factible  $x^*$  que brinda el mejor valor de la función objetivo.  $x^*$  es un punto óptimo y  $f(x^*)$  el valor óptimo de la función objetivo. Es necesario destacar aquí que el concepto de óptimo está asociado al modelo y no al problema.

Es decir, el óptimo corresponde al modelo y podrá aproximar más o menos bien al óptimo del problema dependiendo de cuan bien el modelo representa al problema de optimización en cuestión.

### 2.4.1. Etapas para resolver el problema de optimización

Para resolver un problema de optimización se requieren dos etapas principales que se muestran en la Figura 2.5.



**Figura 2.5. Etapas para resolver el problema de optimización.** Fuente: Scenna (1999)

La formulación del modelo de optimización no es un procedimiento formal estructurado, sino más bien es un proceso que requiere de experiencia y creatividad. Una vez generado el modelo, la etapa siguiente es resolver y validar dicho modelo. Esta etapa puede considerarse suficientemente formalizada puesto que los modelos de problemas de optimización han sido muy estudiados y se han desarrollado innumerables métodos y estrategias para resolverlos.

### 2.4.2. Tipos de optimización. Optimización global y local

La optimización numérica ha adquirido mucha atención entre la comunidad científica durante las últimas décadas, y quizás lo más confuso para el diseñador reside en decidir qué algoritmo de optimización se ajusta mejor a las características del problema físico bajo análisis aunado a tratar de encontrar una solución óptima con un coste computacional razonable.

Sobre la base de estas dos premisas, Pérez y Basterrechea (2005) establecen una clasificación preliminar de los métodos de optimización en dos grandes bloques, distinguiendo por un lado los métodos de búsqueda local y por otro, las técnicas de optimización global.

Los métodos de optimización local son también conocidos como métodos clásicos y los métodos heurísticos o metaheurísticos son llamados métodos de optimización global. Dentro de los primeros se encuentra la optimización lineal, lineal entera mixta, no lineal, dinámica, entre otros. En el segundo grupo se incluyen los algoritmos evolutivos (por ejemplo, genéticos), el método del recocido simulado (simulated annealing) o las búsquedas heurísticas (método tabú, búsqueda aleatoria).

#### **2.4.2.1 Optimización local**

Los métodos de optimización local obtienen la mejor solución posible en las inmediaciones del punto inicial, atribuyéndoseles una fuerte dependencia del punto de arranque del algoritmo. La mayor parte de los métodos locales utilizan la información del gradiente, requieren el cálculo de derivadas y, en definitiva, imponen sobre el espacio de búsqueda unas condiciones de diferenciabilidad y continuidad difíciles de garantizar y controlar en la práctica.

### 2.4.2.2 Optimización Global

En el extremo opuesto están las técnicas de optimización global que exhiben una gran independencia de la naturaleza del espacio de soluciones y, a diferencia de las técnicas de búsqueda local, son capaces de atravesar un espacio de búsqueda con múltiples mínimos o máximos locales y alcanzar una solución global al problema, entendiendo como tal la mejor solución posible o una solución en las inmediaciones de la región que contiene a la solución óptima. Como contrapartida, los métodos de optimización global exploran de forma exhaustiva el espacio de soluciones, sin aprovechar las características locales del espacio de búsqueda como ocurre al operar por ejemplo con gradientes, ralentizando así la convergencia.

### 2.4.3 Clasificación histórica de los métodos de optimización

Todos los algoritmos confiables y universalmente aplicables a problemas de optimización global tienen tiempos que incrementan exponencialmente con el número de variables. Pérez y Basterrechea (2005), presentan en su trabajo una clasificación histórica de los métodos de optimización:

- **Estocásticos o heurísticos:** evalúan la función  $f$  sobre una muestra aleatoria sobre la región de interés, denominados también no confiables. Entre ellos está el método de algoritmo genético (AG), recocido simulado (simulated annealing) y el de búsqueda aleatoria (random search). La



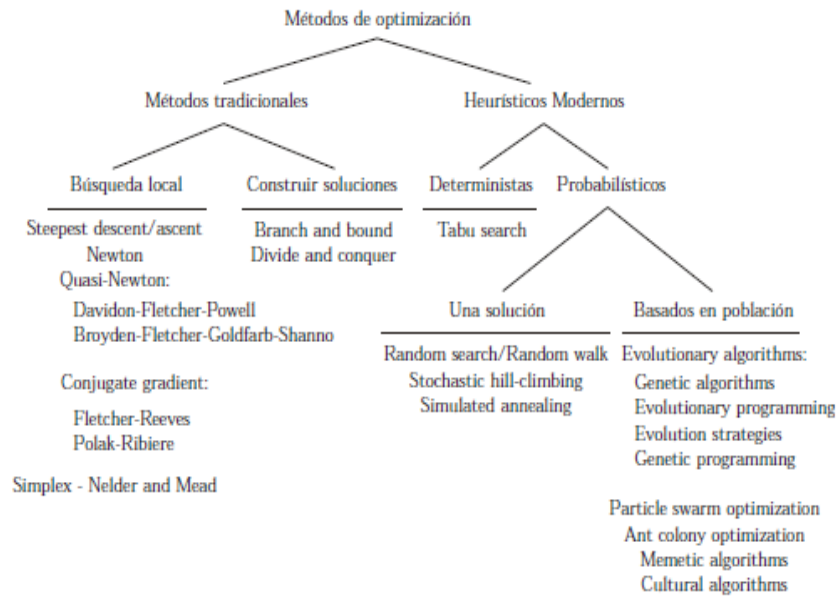
eficiencia es una característica de estos métodos y se emplean para resolver problemas de gran escala (+100 variables).

- **Determinísticos o clásicos:** no envuelve ningún elemento aleatorio, son los denominados métodos confiables. Entre ellos el método de ramas y bordes (branch and bound) y métodos de intervalos (interval methods).
- **Métodos de puntos** (point methods). Estos métodos calculan valores de la función sobre puntos de muestras pero son incapaces para resolver confiablemente un problema de optimización global.
- **Métodos de bordes** (boundary methods): calculan cotas sobre conjuntos compactos. Si son implementados apropiadamente y considerando errores de redondeo, pueden producir soluciones globales rigurosas.

Al exponer aquí los principales métodos de optimización, no se pretende hacer ver que los métodos que se esbozaron son los mejores para cualquier proceso. Todo lo contrario, depende del proceso y cómo se aborde el problema de optimización. Lo que se quiere es dar una visión general sobre los métodos más explorados actualmente y que para el caso específico del problema de diseño integrado, se han aplicado tres de ellos para determinar cuales se adaptan mejor y arrojan la solución óptima. Y es así que se determina que para un proceso muy particular, algún método o combinación de ellos, será la herramienta adecuada.

Al margen de la mera distinción entre técnicas locales y globales o de la clasificación histórica de los métodos de optimización, en la Figura 2.6 se muestra

otra clasificación detallada que recoge los métodos de optimización más representativos presentados por Pérez y Basterrechea (2005).



**Figura 2.6. Clasificación de los métodos de optimización más relevantes.** Por uniformidad con la literatura científica se conserva la denominación original en inglés de cada método. Fuente: Pérez y Basterrechea (2005).

Junto a los métodos de búsqueda local basados en derivadas, se incluye un método directo basado en simplex, cuyo atractivo reside en que únicamente requiere los valores de la función objetivo a optimizar.

Los métodos tradicionales se completan con los así denominados constructivos, que básicamente trabajan sobre soluciones parciales del problema y que de alguna forma aíslan partes del espacio de búsqueda sin examinarlas. Como alternativa a los métodos tradicionales, una parte importante de los así denominados métodos heurísticos modernos han adquirido durante la última década una notable

aceptación en diferentes campos de la ingeniería, y entorno a algunos de ellos se centra esta investigación.

Dentro de los métodos heurísticos de naturaleza probabilística, denominación asociada con el hecho de que la optimización depende de eventos aleatorios, existen dos familias diferenciadas: aquellos que utilizan un único punto de partida y aquellos que utilizan una población.

Todas las variantes de los métodos heurísticos probabilísticos que utilizan un único punto de partida, a excepción del recocido simulado, tienen unos fundamentos muy sencillos, que se limitan a hacer evolucionar una solución inicial perturbando aleatoriamente los parámetros a optimizar.

En lo que respecta al recocido simulado, este método imita a nivel computacional el proceso físico a seguir para obtener sólidos con configuraciones de energía mínima. Su sencillez y su potencial frente a otros métodos es un hecho contrastado, y su uso está muy extendido en aplicaciones muy diversas.

En línea con el recocido simulado y en un intento por imitar procesos naturales tan dispares como la evolución de las especies o los propios comportamientos sociales y culturales de diferentes colectivos, entre los cuales puede incluirse a los propios seres humanos, surgen nuevos métodos que establecen una nueva concepción de la optimización.

Los algoritmos genéticos merecen una mención especial al ser el método que más atención ha recibido en los últimos años. Basado en los principios de la selección y evolución natural, el esquema clásico de los algoritmos genéticos hace evolucionar la población en base a la presión que ejercen los operadores de

selección, cruce y mutación. Sin embargo, la amplia difusión del método de optimización ha hecho que los propios investigadores propongan continuas variantes y mejoras al esquema clásico, en un intento por incrementar el rendimiento del método.

#### **2.4.4 Métodos Híbridos de Optimización**

Como ya se mencionaba en el estado del arte, cuando se trabaja con algoritmos de optimización no existe uno mejor que otro y que la mayoría tiene sus ventajas y desventajas. Es por ello los algoritmos híbridos o algoritmos derivados de otros han tenido mucho éxito.

Un ejemplo del desarrollo de algoritmos híbridos exitosos está en el trabajo de Petcu y Faltings (2007) sobre optimización combinatoria basado en métodos de búsqueda locales (que solo toman decisiones basados en información local) y métodos completamente inferenciales.

En el estado del arte se presentaron los hallazgos de algunos trabajos que emplearon con éxito la hibridación de métodos de optimización entre ellos están los de Elhossini y otros (2010), Martinez y Behar (2011), Liu y otros (2009), Zhou y otros (2009) y Pirona en el 2009, presentados en el estado del arte.

Por lo tanto, al escudriñar los conceptos asociados a la optimización, carece de sentido enjuiciar la capacidad de un método de optimización para una aplicación concreta sin abrirse a nuevas configuraciones y combinaciones. Si bien es cierto que la configuración del algoritmo está íntimamente ligada con el tipo de problema

a tratar, resulta interesante expandir la investigación modificando las estrategias, los valores de los parámetros e incluso el propio esquema del algoritmo, pues en realidad, esta información convenientemente presentada marca una tendencia que, por lo general, puede ser extrapolada a otros problemas de interés.

Por todo ello, el objetivo de esta tesis se centra en contribuir metodológicamente, mediante un estudio sistemático de la aplicación de diferentes técnicas de optimización al diseño integrado de reactores mezcla completa.

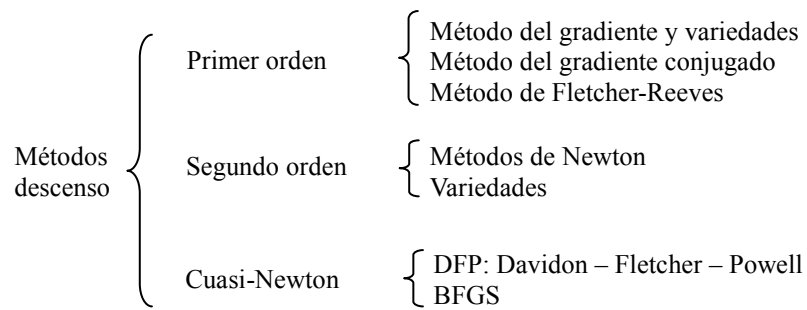
A continuación se explica a detalle las características de los métodos de optimización explorados en el presente trabajo para resolver el problema de diseño integrado de reactores mezcla completa, los cuales corresponden a:

- (1) Método del gradiente,
- (2) Algoritmo DIRECT,
- (3) Recocido simulado y
- (4) Algoritmos genéticos.

#### **2.4.5 Método del gradiente**

Para obtener un mínimo local en la resolución de un problema de optimización, se debe resolver un sistema de ecuaciones no lineales, en general complejo. Surge de esta forma la necesidad de aplicar métodos numéricos.

Considerando inicialmente dentro de la optimización local, los métodos numéricos de descenso, se puede observar la clasificación dada por Hernández G. (2006):



**Figura 2.7. Clasificación métodos de búsqueda local. Métodos de Descenso.**

Fuente: Hernández G. (2006).

Todos los métodos numéricos que se utilizan en la actualidad reducen este problema a una secuencia de problemas unidimensionales.

El método del gradiente (conocido también como método de Cauchy o del descenso más pronunciado) consiste en un algoritmo específico para la resolución de modelos de programación no lineal sin restricciones, perteneciente a la categoría de algoritmos generales de descenso (ver figura 2.7), donde la búsqueda de un mínimo está asociado a la resolución secuencial de una serie de problemas unidimensionales.

Matemáticamente se define de la siguiente forma: Sea  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  diferenciable. La derivada direccional de  $f$  en la dirección  $d \in \mathbb{R}^n$  está dada por:

$$Df(x; d) = \nabla f(x)^t d \quad (19)$$

Para obtener la dirección de máximo descenso de la función  $f$  en un punto  $x \in \mathbb{R}^n$  tal que  $f(x)^t = 0$ , se debe resolver el problema:

$$\boxed{\begin{array}{l} \min_{d \in \mathbb{R}^n} \nabla f(x)^t d \\ \|d\|^2 = 1 \end{array}} \quad (20)$$

La solución de este problema es:

$$d = -\frac{\nabla f(x)}{\|\nabla f(x)\|} \quad (21)$$

Y por lo tanto la dirección de máximo descenso de la función  $f$  es:

$$d = -\nabla f(x) \quad (22)$$

**Definición:** El vector  $d \in \mathbb{R}^n$  es una dirección de descenso de  $f$ , función diferenciable, en el punto  $x \in \mathbb{R}^n$  si:

$$\nabla f(x)^t \cdot d < 0 \iff (-\nabla f(x))^t \cdot d > 0 \quad (23)$$

A través del método del gradiente, se busca definir una sucesión de puntos que tenga la dirección de máximo descenso de la función  $f$ .

**Definición:** Las curvas de nivel  $\theta$  de  $f$  están definidas por:

$$\varphi(f, \theta) = \{x \in \mathbb{R}^n \text{ tal que: } f(x) = \theta\} \quad (24)$$

**Lema:** Sea  $d \in \mathbb{R}^n$  una dirección de descenso de la función  $f$  en el punto  $x$ . Se define la función unidimensional:

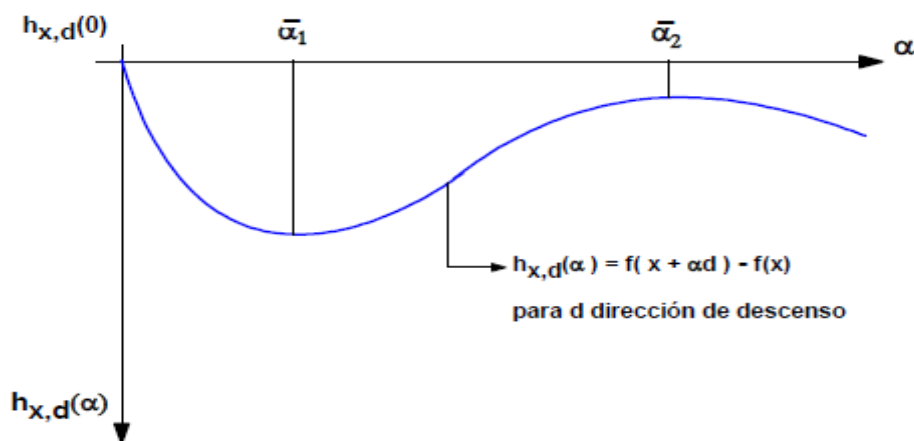
$$h_{x,d} : \mathbb{R}_+ \rightarrow \mathbb{R} \\ \alpha \rightarrow h_{x,d}(\alpha) = f(x + \alpha d) - f(x) \quad \text{para } x, d \in \mathbb{R}^n \text{ fijos} \quad (25)$$

$$h_{x,d}(\bar{\alpha}) < h_{x,d}(\alpha = 0) = 0 \Leftrightarrow f(x + \bar{\alpha}d) < f(x) \quad (26)$$

donde  $\alpha$  verifica:

$$h'_{x,d}(\bar{\alpha}) = 0 \Leftrightarrow \nabla f(x + \bar{\alpha}d)^t d = 0 \quad (27)$$

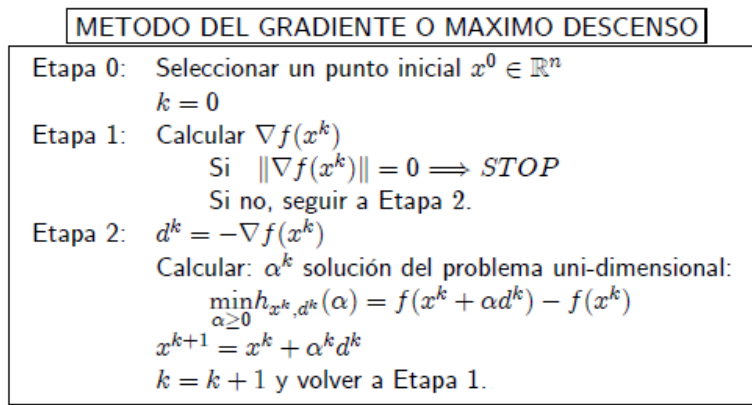
De las definiciones hechas para  $h_{x,d}$  se observa lo siguiente:



**Figura 2.8.** Función unidimensional  $h_{x,d}$ . Fuente. Hernández, G. (2006).

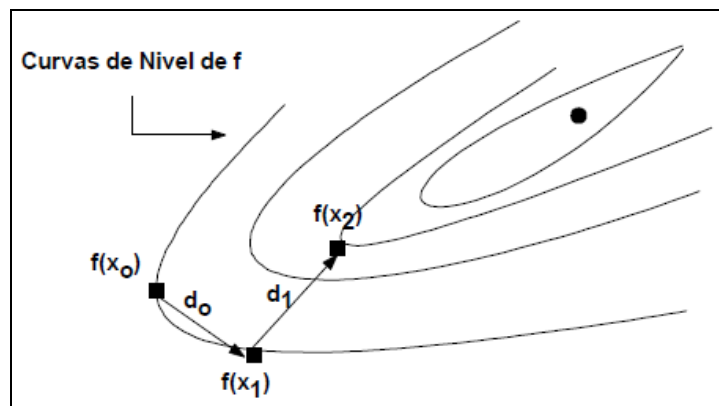


De acuerdo a los resultados anteriores, se define la iteración canónica del Método del Gradiente o Máximo Descenso, según:



**Figura 2.9. Método del gradiente.** Fuente. Hernández, G. (2006).

El método del gradiente efectúa las iteraciones a través de las llamadas curvas de nivel, en la siguiente figura se ilustra gráficamente este concepto.



**Figura 2.10. Esquema de iteración del método del gradiente.** Fuente. Hernández, G. (2006).

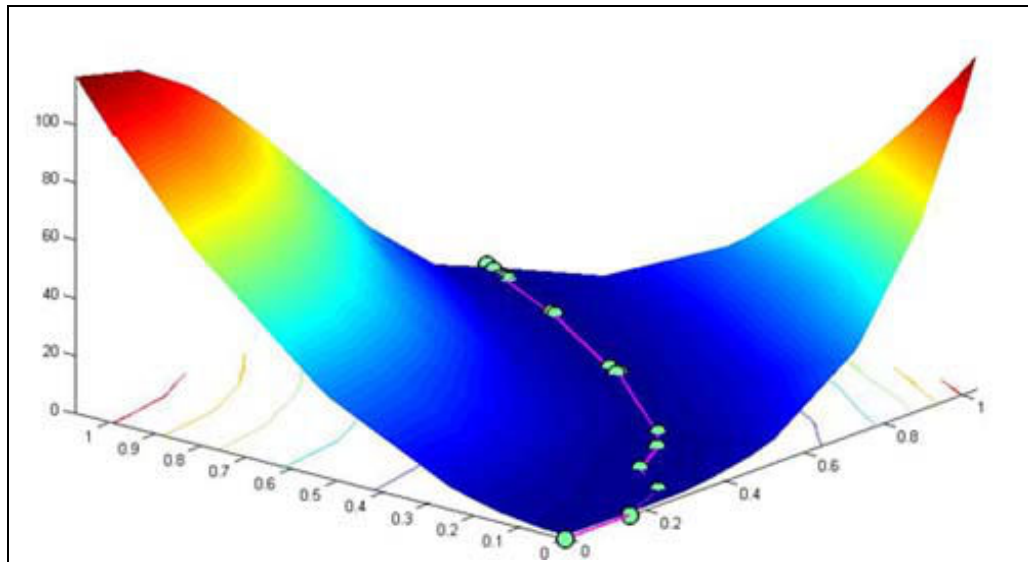
Hernández G. (2006) indica que algunos métodos que evalúan gradientes o aproximan gradientes usan diferencias finitas (o incluso subgradientes), ellos son:

- **Métodos Quasi-Newton:** métodos iterativos para problemas medianos-grandes (ejemplo  $N < 1000$ ).
- **Métodos de gradiente conjugado:** métodos iterativos para problemas grandes. (En teoría, estos métodos terminan en un número finito de pasos con funciones objetivo cuadráticas, pero esta terminación finita no se observa en la práctica en computadoras de precisión finita.)
- **Métodos de punto interior:** esta es una gran clase de métodos para la optimización restringida. Algunos métodos de punto interior usan solamente información del subgradiente, y otros requieren la evaluación de las Hessianas.
- **Descenso del gradiente** (alternativamente, descenso pronunciado o ascenso pronunciado): un método lento de interés teórico e histórico, el cual ha sido renovado para encontrar soluciones aproximadas de problemas enormes.
- **Método del subgradiente:** un método iterativo para grandes funciones de Lipschitz localmente usando gradientes generalizados.

Dentro de los métodos que evalúan gradientes, el método QUASI Newton, es empleado en la resolución del problema de diseño integrado de reactores mezcla completa.

### 2.4.5.1 Método QUASI Newton

Zapata J. (2012) indica que los métodos Quasi-Newton se utilizan si la derivada de la función objetivo es difícil de calcular, o esta viene dada de forma numérica. Se basan en sustituir las derivadas por aproximaciones en diferencias finitas.



**Figura 2.11. Espacio de soluciones y el Método Quasi-Newton.** Fuente: Zapata J. (2012).

La idea fundamental de los métodos Quasi-Newton es intentar construir una aproximación de la inversa del Hessiano (discriminante), usando información obtenida durante el proceso de descenso.

Estos métodos son similares a los métodos de gradiente conjugado en el sentido de que se basan principalmente en propiedades de las funciones cuadráticas. Sin embargo, en el método del gradiente conjugado, la principal fortaleza de la

búsqueda se deriva del uso de las direcciones conjugadas de búsqueda, mientras que los métodos de Quasi-Newton están diseñados para imitar más directamente las características positivas del método de Newton pero usando solo información de primer orden.

#### **2.4.6 Algoritmo DIRECT**

Según Bjorkman y Holmström (2012), los métodos de descenso basados en la existencia del gradiente de la función objetivo, se caracterizan por converger hacia extremos locales de la función sin garantía de que estos extremos correspondan a extremos globales. Más aún, localizan solo un extremo.

Es por ello que a pesar de la popularidad de los métodos de búsqueda local, se han desarrollado paralelamente métodos de optimización global cuyo propósito es obtener el conjunto de puntos donde la función se optimiza. Es frecuente, además, que debiliten las hipótesis de suavidad de la función objetivo y no exijan, por ejemplo, la existencia del gradiente.

Entre estos métodos de optimización global existe un conjunto que incorporan mecanismos estocásticos, entre los que pueden citarse los métodos de: Monte Carlo, recocido simulado (simulated annealing), algoritmos genéticos, y optimización bayesiana; y otros determinísticos, que no presuponen derivación.

Entre los determinísticos, los métodos Lipschitzianos han probado su eficiencia. El DIRECT (dividing rectangles), es una propuesta eficiente que por su importancia se emplea en el problema de diseño integrado.

DIRECT es un algoritmo desarrollado por Jones D. y otros (1993), para encontrar el mínimo global de una función multivariable sujeta a límites simples, sin usar la información de la derivada.

El algoritmo DIRECT es una modificación del enfoque de la aproximación estándar Lipschitziana, que elimina la necesidad de especificar una constante de Lipschitz. Jones D. y otros (1993), introducen una forma distinta de ver la constante de Lipschitz. Su idea es realizar búsquedas simultáneas utilizando todas las constantes posibles de cero hasta el infinito, por ello opera tanto local como globalmente. Adicionalmente, DIRECT reduce la complejidad computacional al evaluar la función objetivo en el centro de cada hiperrectángulo en lugar de sus vértices.

El método estándar de la constante de Lipschitz es visto como un parámetro de ponderación que indica cuánto énfasis se debe colocar en la optimización global frente a la búsqueda local.

En el método Lipschitziano estándar, esta constante suele ser grande, porque debe ser igual o superior a la tasa máxima de cambio de la función objetivo. Como resultado, estos métodos ponen un gran énfasis en la búsqueda global, lo que lleva a reducir la velocidad de convergencia.

Como se mencionó anteriormente los métodos Lipschitzianos se caracterizan por ser determinísticos, por lo tanto presentan como ventaja la ausencia de la necesidad de realizar múltiples corridas.

Sin embargo, según Jones D. y otros (1993), la optimización Lipschitziana presenta inconvenientes importantes relacionados con la especificación de la

constante de Lipschitz, la velocidad de convergencia y la complejidad computacional en problemas de varias dimensiones.

La determinación de la constante de Lipschitz es un problema en la práctica ya que esta puede no existir o su cálculo puede resultar difícil. Por ejemplo, en la optimización de un proceso en ingeniería, la función objetivo puede estar basada en una simulación de alto costo computacional, o en un experimento sobre el sistema real, donde no se dispone de la función objetivo en forma cerrada, por lo que el cálculo de la constante de Lipschitz es usualmente difícil o imposible.

El problema de velocidad de convergencia está relacionado con la especificación de la constante de Lipschitz, pues ésta puede ser vista como una medida del énfasis entre la búsqueda global y local.

Como resultado, estos métodos le dan mayor énfasis a la búsqueda global, razón por la cual tienen una lenta convergencia.

Con relación a la complejidad computacional, cuando se optimiza una función de  $n$  variables sujeta a restricciones de borde, el espacio de búsqueda es un hiperrectángulo de  $n$  dimensiones. La mayoría de los algoritmos Lipschitzianos dividen el espacio de búsqueda en pequeños hiperrectángulos y la función objetivo es evaluada en sus vértices. Para iniciar la búsqueda, el algoritmo debe evaluar la función objetivo en los  $2^n$  vértices de espacio de búsqueda.

En contraste, las variaciones introducidas por Jones D y otros (1993), con el algoritmo DIRECT, se realizan búsquedas simultáneas utilizando todas las constantes posibles, y por lo tanto opera tanto a nivel global como local.

La clave para esto es modificar como se divide el espacio. En vez de evaluar en los extremos del intervalo, se evalúa en el punto central. Esto significa que, en múltiples dimensiones, el algoritmo será inicializado solo muestreando en un punto (a diferencia de  $2n$  puntos).

A continuación se presenta una descripción general del algoritmo DIRECT:

1. Inicialmente, se transforma el espacio de búsqueda en un hipercubo unitario. La función es evaluada en el centro de este hipercubo.
2. Empleando un procedimiento establecido, el hipercubo inicial es dividido en hiperrectángulos.
3. Mediante la determinación de la cota mínima correspondiente a cada hiperrectángulo, se identifican los hiperrectángulos potencialmente óptimos.
4. Todos los hiperrectángulos potencialmente óptimos son divididos en hiperrectángulos aún más pequeños utilizando el mismo procedimiento mencionado anteriormente y la función objetivo es evaluada en sus centros.
5. El algoritmo se detiene si se ha alcanzado el número máximo de iteraciones, en caso contrario se va al paso 3.

La selección de los rectángulos potencialmente óptimos se realiza mediante la determinación de los puntos extremos de una cápsula convexa inferior monótona creciente a partir de un conjunto de puntos en el plano empleando el llamado “algoritmo de Graham”.

Una de las limitaciones de DIRECT es que requiere, relativamente, gran cantidad de evaluaciones de la función objetivo, en particular cuando la solución óptima se encuentra cerca de la frontera del espacio de búsqueda.

En ésta sección se presenta más adelante, de manera equivalente el término algoritmo DIRECT y “gblSolve”, que es el nombre de la rutina de MATLAB R2012a correspondiente.

#### **2.4.7 Algoritmos genéticos**

El algoritmo genético es una técnica de búsqueda basada en la teoría de la evolución de Darwin (1859), que ha cobrado tremenda popularidad en todo el mundo durante los últimos años. Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Hoy en día se sabe que estos cambios se efectúan en los genes de un individuo (unidad básica de codificación de cada uno de los atributos de un ser vivo), y que sus atributos más deseables (es decir, los que le permiten adaptarse mejor a su entorno) se transmiten a sus descendientes cuando este se reproduce sexualmente.

Un investigador de la Universidad de Michigan llamado John Holland era consciente de la importancia de la selección natural, y a fines de los años 60 desarrolló una técnica que permitió incorporarla a un programa. Su objetivo era lograr que las computadoras aprendieran por sí mismas. A la técnica que inventó Holland se le llamó originalmente "planes reproductivos", pero se hizo popular bajo el nombre "algoritmo genético" tras la publicación de su libro en 1975.

Alander J. (1992) exponen versiones más complejas de algoritmos genéticos, las cuales generan un ciclo iterativo que directamente toma a la especie (el total de los ejemplares) y crea una nueva generación que reemplaza a la antigua una cantidad



de veces determinada por su propio diseño. Una de sus características principales es la de ir perfeccionando su propia heurística en el proceso de ejecución, por lo que no requiere largos períodos de entrenamiento especializado por parte del ser humano.

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un gran número de descendientes.

Por el contrario individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagarán en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes "superindividuos", cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

Los algoritmos usan una analogía directa con el comportamiento natural, ellos trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma.

Este cruce producirá nuevos individuos descendientes de los anteriores, los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones. De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población.

Al favorecer el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda.

#### **2.4.7.1 El algoritmo genético simple**

El algoritmo genético simple, también denominado canónico, se representa en la Figura 2.12. Se necesita una codificación o representación del problema, que resulte adecuada al mismo. Además se requiere una función de ajuste o adaptación al problema, la cual asigna un número real a cada posible solución codificada. Durante la ejecución del algoritmo, los padres deben ser seleccionados para la reproducción, a continuación dichos padres seleccionados se cruzarán generando dos hijos, sobre cada uno de los cuales actuará un operador de mutación. El resultado de la combinación de las anteriores funciones será un conjunto de individuos (posibles soluciones al problema), los cuales en la evolución del algoritmo genético formarán parte de la siguiente población.

```

BEGIN /* Algoritmo Genético Simple */
  Generar una población inicial.
  Computar la función de evaluación de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generación */
      FOR Tamaño población/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generación,
          para el cruce (probabilidad de selección proporcional
          a la función de evaluación del individuo).
          Cruzar con cierta probabilidad los dos
          individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Computar la función de evaluación de los dos
          descendientes mutados.
          Insertar los dos descendientes mutados en la nueva generación.
        END
      END
    IF la población ha convergido THEN
      Terminado := TRUE
    END
  END

```

**Figura 2.12. Algoritmo genético simple.** Fuente: Larrañaga y otros (1999).

### **Codificación, fenotipo y genotipo**

Se supone que los individuos (posibles soluciones del problema), pueden representarse como un conjunto de parámetros (denominados genes), los cuales agrupados forman una cadena de valores (a menudo referida como cromosoma). Si bien el alfabeto utilizado para representar los individuos no debe necesariamente estar constituido por el (0, 1), buena parte de la teoría en la que se fundamentan los algoritmos genéticos utiliza dicho alfabeto. En términos biológicos, el conjunto de parámetros representando un cromosoma particular se denomina fenotipo.

**El fenotipo** contiene la información requerida para construir un organismo, el cual se refiere como genotipo. Los mismos términos se utilizan en el campo de los Algoritmos Genéticos.

La adaptación al problema de un individuo depende de la evaluación del genotipo. Esta última puede inferirse a partir del fenotipo, es decir puede ser computada a partir del cromosoma, usando la función de evaluación.

**La función de adaptación** debe ser diseñada para cada problema de manera específica. Dado un cromosoma particular, la función de adaptación le asigna un número real, que se supone refleja el nivel de adaptación al problema del individuo representado por el cromosoma.

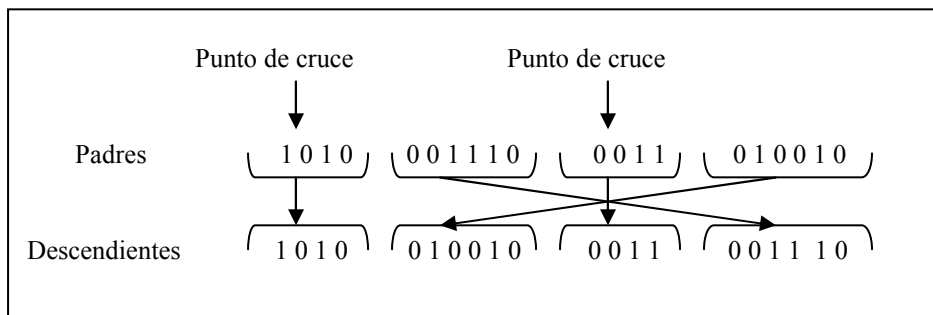
Durante **la fase reproductiva** se seleccionan los individuos de la población para cruzarse y producir descendientes, que constituirán, una vez mutados, la siguiente generación de individuos.

**La selección de padres** se efectúa al azar usando un procedimiento que favorezca a los individuos mejor adaptados, ya que a cada individuo se le asigna una probabilidad de ser seleccionado que es proporcional a su función de adaptación. Este procedimiento se dice que está basado en la ruleta sesgada. Según dicho esquema, los individuos bien adaptados se escogerán probablemente varias veces por generación, mientras que, los pobremente adaptados al problema, no se escogerán más que de vez en cuando.

Una vez seleccionados dos padres, sus cromosomas se combinan, utilizando habitualmente los operadores de cruce y mutación. Las formas básicas de dichos operadores se describen a continuación.

### Operadores de cruce y mutación

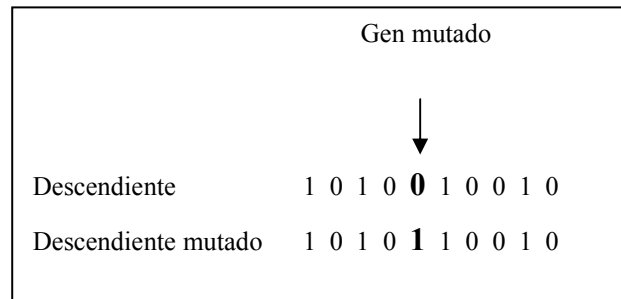
**El operador de cruce**, toma dos padres seleccionados y corta su cadena de cromosomas en una posición escogida al azar, para producir dos subcadenas iniciales y dos subcadenas finales. Después se intercambian las subcadenas finales, produciéndose dos nuevas cromosomas completos (véase la Figura 2.13). Ambos descendientes heredan genes de cada uno de los padres. Este operador se conoce como operador de cruce basado en un punto.



**Figura 2.13. Operador cruce basado en un punto.** Fuente: Larrañaga y otros (1999).

**El operador de mutación** se aplica a cada hijo de manera individual, y consiste en la alteración aleatoria (normalmente con probabilidad pequeña) de cada gen componente del cromosoma. La Figura 2.14 muestra la mutación del quinto gen

del cromosoma. Si bien puede en principio pensarse que el operador de cruce es más importante que el operador de mutación, ya que proporciona una exploración rápida del espacio de búsqueda, éste último asegura que ningún punto del espacio de búsqueda tenga probabilidad cero de ser examinado, y es de capital importancia para asegurar la convergencia de los algoritmos genéticos.



**Figura 2.14. Operador de mutación.** Fuente: Larrañaga y otros (1999)

## Población

Las poblaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que el trabajar con poblaciones de gran tamaño puede acarrear problemas relacionados con el excesivo costo computacional.

Goldberg (1989) efectuó un estudio teórico, obteniendo como conclusión que el tamaño óptimo de la población para cadenas de longitud, con codificación binaria, crece exponencialmente con el tamaño de la cadena. Este resultado traería como consecuencia que la aplicabilidad de los algoritmos genéticos en problemas reales sería muy limitada, ya que resultarían no competitivos con otros métodos de optimización combinatoria. Alander (1992), basándose en evidencia empírica sugiere que un tamaño de población comprendida entre  $\sqrt{n}$  y  $2\sqrt{n}$  es suficiente para

atacar con éxito los problemas por el considerados. Habitualmente la población inicial se escoge generando cadenas al azar, pudiendo contener cada gen uno de los posibles valores del alfabeto con probabilidad uniforme.

### **Función objetivo**

Dos aspectos que resultan cruciales en el comportamiento de los algoritmos genéticos son la determinación de una adecuada función de adaptación o función objetivo, así como la codificación utilizada.

Idealmente interesa construir funciones objetivo con "ciertas regularidades", es decir funciones objetivo que verifiquen que para dos individuos que se encuentren cercanos en el espacio de búsqueda, sus respectivos valores en las funciones objetivo sean similares. Por otra parte una dificultad en el comportamiento del algoritmo genético puede ser la existencia de gran cantidad de óptimos locales, así como el hecho de que el óptimo global se encuentre muy aislado.

### **Selección**

La función de selección de padres más utilizada, es la denominada "función de selección proporcional a la función objetivo, en la cual cada individuo tiene una, probabilidad de ser seleccionado como padre que es proporcional al valor de su función objetivo.

#### **2.4.8 Método recocido simulado o simulated annealing (SA)**

El recocido simulado es un algoritmo de aproximación a la solución óptima, fundamentado en una analogía del comportamiento de sistemas termodinámicos simples que viene siendo utilizado en ciertos problemas de ingeniería.

El recocido simulado es un algoritmo en el mismo sentido en que lo son las reglas para multiplicar o dividir dos números, pero se trata de un conjunto de operaciones mucho más complicado, de manera que en la práctica solo puede ejecutarse mediante máquinas de cálculo. El algoritmo simula el comportamiento de sistemas físicos simples sometidos a un descenso paulatino y lento de su temperatura, tal es el caso de la producción de acero laminado o la cristalización de azúcar en soluciones sobresaturadas. Tras un lento proceso, tales sistemas alcanzan un estado caracterizado por ser mínima su energía, a pesar del comportamiento intrínsecamente aleatorio de sus partículas microscópicas.

En ocasiones, tales estados se reconocen por una particular perfección visual, por ejemplo, en un macrocristal de azúcar, cúbico y regular (en teoría). El recocido simulado permite explotar el conocimiento físico de tales procesos con vistas a buscar el mínimo absoluto de casi cualquier función matemática, que también en ocasiones se corresponderá con formas que muestran cualidades apreciables en tanto ordenadas, razonables o sensatas.

El algoritmo no es nuevo y viene siendo empleado con éxito en diversas disciplinas desde hace más de una década. El recocido simulado de un objeto para mejorar alguna de sus propiedades consiste en producir al azar cambios en su descripción.



### 2.4.8.1 El Proceso de Recocido de un Sólido

De acuerdo con Gutierrez, A. y otros (1998), el algoritmo de recocido simulado está basado en una analogía entre la simulación de recocido de sólidos y la problemática de resolver problemas de optimización combinatoria de gran escala. Por esta razón el algoritmo se conoce como recocido simulado. Recocido denota un proceso de calentamiento de un sólido a una temperatura en la que sus granos deformados recrystalizan para producir nuevos granos. La temperatura de recocido o de recrystalización, depende del tipo de material, del grado de deformación del mismo, además de su uso futuro.

Seguida a la fase de calentamiento, viene un proceso de enfriamiento en donde la temperatura se baja poco a poco. De esta manera, cada vez que se baja la temperatura, las partículas se acomodan en estados de más baja energía hasta que se obtiene un sólido con sus partículas acomodadas conforme a una estructura de cristal. Si se comienza con un valor máximo de la temperatura, en la fase de enfriamiento del proceso de recocido, para cada valor de la temperatura  $T$  debe permitirse que se alcance su equilibrio térmico.

Sin embargo, si el proceso de enfriamiento es demasiado rápido y no se alcanza en cada etapa el equilibrio térmico, el sólido congelará en un estado cuya estructura será amorfa en lugar de la estructura cristalina de más baja energía. La estructura amorfa está caracterizada por una imperfecta cristalización del sólido.

El equilibrio térmico está caracterizado por la distribución de Boltzmann. De acuerdo a esta distribución, la probabilidad de que el sólido esté en un estado  $i$

con energía  $E_i$  a la temperatura  $T$ , viene dada por:

$$P_T\{X = i\} = \frac{1}{Z(T)} \exp\left(\frac{-E_i}{k_B T}\right), \quad (28)$$

donde  $X$  es una variable aleatoria que denota el estado actual del sólido.  $Z(T)$  es una constante de normalización llamada la función partición, que está definida como:

$$Z(T) = \sum_j \exp\left(\frac{-E_j}{k_B T}\right), \quad (29)$$

donde la sumatoria se extiende sobre todos los posibles estados y  $k_B$  es una constante física conocida como la constante de Boltzmann. El factor  $\exp\left(\frac{-E_i}{k_B T}\right)$  se conoce como el factor de Boltzmann.

Obviamente  $P_T$  es una función de densidad de probabilidad ya que siempre es mayor o igual a cero y la suma sobre todos los valores es igual a la unidad.

Se puede observar la expresión para  $Z(T)$  que cuando el valor de  $T$  disminuye, la distribución de Boltzmann se concentra en los estados de menor energía mientras que si la temperatura se aproxima a cero, únicamente los estados con mínima energía tienen una probabilidad de ocurrencia diferente de cero.

Por lo dicho anteriormente, el proceso de recocido consta de dos pasos fundamentales que son:

- Incrementar la temperatura del “baño térmico” a un valor máximo.
- Disminuir cuidadosamente la temperatura del “baño térmico” hasta que las

partículas se reacomoden por sí mismas en un estado de mínima energía.

El proceso físico de recocido puede moderarse exitosamente usando métodos de simulación, el algoritmo introducido para tal propósito se basa en técnicas “Monte Carlo” y genera una sucesión de estados del sólido de la siguiente manera:

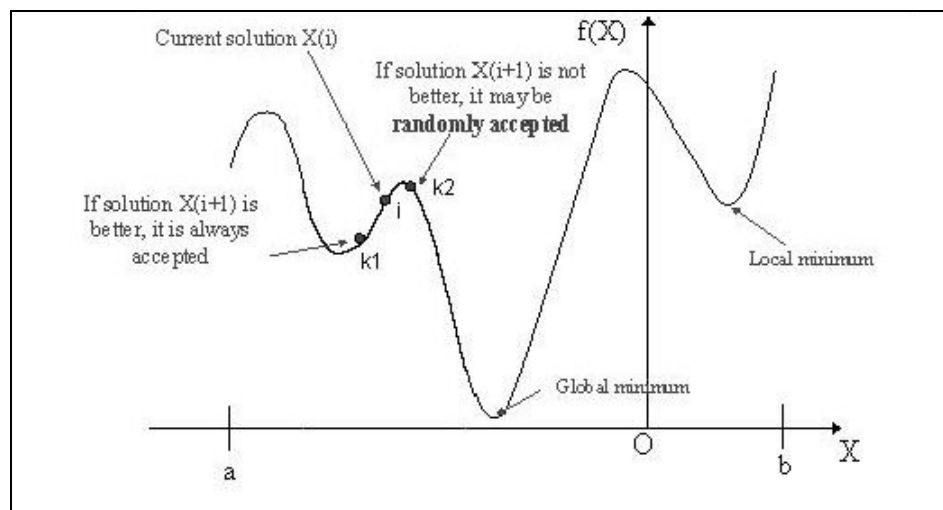
Dado un estado  $i$  del sólido con energía  $E_i$ , se genera un estado subsecuente  $j$  aplicando un mecanismo de perturbación que transforma el estado actual en el siguiente estado por medio de una pequeña distorsión, por ejemplo, por el desplazamiento de una partícula. La energía del siguiente estado es  $E_j$ . Si la diferencia de energía,  $E_j - E_i$ , es menor o igual a cero, el estado  $j$  se acepta como el estado actual. Si la diferencia de energía, es mayor que cero, el estado  $j$  se acepta con una probabilidad que está dada por:  $\exp\left(\frac{-E_j}{k_B T}\right)$ , donde  $T$  denota la temperatura del baño térmico y  $k_B$  es la constante de Boltzman.

La regla de decisión descrita arriba se conoce como el “criterio de Metropolis” y al algoritmo se le conoce como “algoritmo de Metropolis”.

Si la temperatura se baja poco a poco, el sólido puede alcanzar su equilibrio térmico en cada temperatura. En el algoritmo de Metropolis esto se lleva a cabo generando un número grande de transiciones para un valor dado de la temperatura.

Por ejemplo, comenzando en el estado  $i$ , como se ilustra en la Figura 2.15, el nuevo estado  $k_1$  es aceptado, pero el nuevo estado  $k_2$  sólo se acepta con una cierta probabilidad. La probabilidad de aceptar un estado peor es alta en el principio y decrece a medida que la temperatura disminuye. Para cada

temperatura, el sistema debe llegar a un equilibrio, es decir, una serie de nuevos estados deben ser juzgados antes que la temperatura se reduzca por lo general en un 10%. En la bibliografía se encuentran demostraciones indicando que el algoritmo va a encontrar, bajo ciertas condiciones, un mínimo global y no se atasca en un mínimo local.



**Figura 2.15. Selección de nuevos estados en Recocido Simulado o Simulated Annealing (SA).** Fuente: Gutiérrez y otros (1998)

#### 2.4.8.2 El Algoritmo de recocido simulado

La simulación del proceso de recocido puede usarse para describir un proceso de generación de una sucesión de soluciones de un problema de optimización combinatoria, en donde se vayan obteniendo, conforme el proceso avanza, mejores soluciones al mismo.

Para este propósito, se puede observar una analogía entre el sistema físico y un problema de optimización combinatoria en donde cada solución del problema puede verse como un estado del sólido y el valor de la función objetivo para la el nivel de energía del sólido. En resumen, se puede pensar en que las soluciones de un problema de optimización combinatoria son equivalentes a los estados de un sistema físico. El costo de una solución es equivalente a la energía de un estado.

Kirkpatrick S. y otros (1983) introducen las siguientes definiciones:

**Definición:** Sea  $(S, f)$  una instancia de un problema de optimización combinatoria, y denote por  $i$  y  $j$  dos soluciones con costo  $f(i)$  y  $f(j)$ , respectivamente. Entonces el criterio de aceptación determina si  $j$  se acepta de  $i$  a partir de aplicar la siguiente probabilidad de aceptación:

$$P_c\{\text{aceptar } j\} = \begin{cases} 1 & \text{si } f(j) \leq f(i) \\ \exp\left(\frac{f(i)-f(j)}{c}\right) & \text{si } f(j) > f(i), \end{cases} \quad (30)$$

donde  $c \in \mathbb{R}^+$  denota el parámetro de control.

Claramente, el mecanismo de generación corresponde al mecanismo de perturbación en el algoritmo de Metropolis, mientras que el criterio de aceptación corresponde al criterio de Metropolis.

**Definición:** Una transición es una acción combinada que transforma la solución actual en una subsecuente, consiste de los siguientes dos pasos: (i) aplicación del mecanismo de generación, (ii) aplicación del criterio de aceptación.

Sea  $C_k$  el valor del parámetro de control y  $L_k$  el número de transiciones generadas en la  $k$ -ésima iteración del algoritmo de Metrópolis. Entonces el algoritmo de recocido simulado puede describirse en pseudo-código como se muestra en la Figura 2.15.

El algoritmo de la Figura 2.15 comienza llamando a un procedimiento de inicialización donde se definen la solución inicial, la temperatura inicial y el número inicial de generaciones necesarias para alcanzar el equilibrio térmico para la temperatura inicial. La parte modular del algoritmo consta de dos ciclos. El externo repite ... hasta y el interno para ...fin...para.

El ciclo interno mantiene fijo el parámetro de control hasta que se generan  $L_k$  soluciones y se acepta o se rechaza la solución generada conforme los criterios de aceptación ya discutidos. El ciclo externo disminuye el valor de la temperatura mediante el procedimiento CALCULA-CONTROL y calcula el número de soluciones a generar para alcanzar equilibrio térmico mediante el procedimiento CALCULA-LONGITUD. Este ciclo finaliza cuando la condición de paro se cumple.

Un rasgo característico del algoritmo de recocido simulado es que, además de aceptar mejoras en el costo, también acepta soluciones peores en costo.

Inicialmente para valores grandes, donde  $c$  puede aceptar soluciones con un valor objetivo mucho mayor a la solución actual, si  $c$  decrece se aceptarán únicamente pequeñas desviaciones y finalmente cuando el valor de  $c$  se aproxima a cero, no se aceptarán desviaciones.

Este hecho significa que el algoritmo de recocido simulado, en contraste con el

algoritmo de búsqueda local, puede escapar de mínimos locales además de exhibir los rasgos favorables de los algoritmos de búsqueda local, es decir, simplicidad y aplicabilidad general.

### **Aspectos generales del algoritmo SA**

En las aplicaciones del algoritmo de recocido simulado, comúnmente se desea implementar este, de manera que la sucesión de soluciones estén generadas a partir de valores decrecientes del parámetro de control.

Las soluciones se generan continuamente tratando de transformar la solución actual en una subsecuente por medio de la aplicación del mecanismo de generación y el criterio de aceptación.

Las aplicaciones del algoritmo de recocido simulado requieren de la especificación de los siguientes puntos: (i) Una representación concisa del problema, (ii) un mecanismo de transición y (iii) un programa de enfriamiento. Cada uno de estos puntos se pueden enumerar con más detalle.

- i. Una descripción concisa de la representación del problema consiste de una representación del espacio de soluciones y una expresión de la función de costo. La función de costo debe escogerse de manera que represente la efectividad de las soluciones con respecto al objetivo de optimización. La representación del problema y la función de costo deben darse por medio de expresiones simples que permitan una fácil manipulación.

- ii. La generación de ensayos para transformar la solución actual en una subsecuente consiste de tres pasos. Primero, se debe generar una nueva solución aplicando un mecanismo de generación. Enseguida se debe calcular la diferencia de costo de las dos soluciones, por último se toma la decisión de aceptar o descartar la nueva solución. La evaluación de la nueva solución es lo que más tiempo consume en el algoritmo de recocido simulado y por lo tanto debe hacerse lo más eficientemente posible. El mecanismo de generación usualmente se escoge de tal manera que se obtengan modificaciones simples para que puedan ser ejecutadas rápidamente, por ejemplo permutaciones, cambios o inversiones. El cálculo de la diferencia de costo se hace tomando en cuenta las diferencias entre ambas soluciones. Para muchas aplicaciones este es el camino más rápido para calcular las diferencias en el costo. La decisión de aceptar la nueva solución se basa en un criterio de aceptación.
  
- iii. Ejecutar el proceso de recocido requiere de la especificación de los parámetros que determinan el programa de enfriamiento. Estos parámetros son el valor inicial del parámetro de control, una función que especifique el decremento del parámetro de control, la longitud de cada bloque donde permanece constante el parámetro de control y el criterio de paro.



## **2.5 MATLAB R2012A EN EL PROBLEMA DE DISEÑO INTEGRADO.**

MATLAB R2012a (abreviatura de MATrix LABoratory, "laboratorio de matrices", versión del año 2012) es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux .

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

El paquete MATLAB R2012a dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB R2012a con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets).

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL.

La caja de herramientas o “toolbox” de optimización de MATLAB R2012a es una colección de funciones que amplía la capacidad de computación numérica.

Dentro de este “toolbox”, las funciones de optimización no son más que rutinas con extensión .m que se llevan a cabo mediante un conjunto de sentencias y algoritmos de optimización específicos.

Este conjunto de funciones logran minimizar o maximizar funciones no lineales generales. Es decir, al usar la palabra “optimización” en MATLAB R2012a se refiere al proceso de búsqueda del mínimo o máximo de una función, denominada comúnmente “función objetivo” en las funciones de optimización que incluye el “toolbox” de optimización de MATLAB R2012a. Además, este “toolbox” también incluye funciones para la resolución de ecuaciones no lineales y problemas de mínimos cuadrados.

Todas las funciones que incluye el “toolbox” son ficheros con extensión .m, contruidos con declaraciones de MATLAB R2012a e implementan algoritmos de optimización específicos.

Cada una de las funciones resuelve un tipo de optimización, para el caso de estudio, se sabe de acuerdo con el modelado matemático presentado más adelante en la “descripción del proceso o sistema”, que el problema es uno de programación no lineal entera mixta con ecuaciones algebraico-diferenciales (MINLP-DAE), requiriéndose la minimización de la función costo (no lineal), sujeta a restricciones.

En el “toolbox” de optimización de MATLAB R2012a, se hace una clasificación de los algoritmos de optimización internos que usan las rutinas del “toolbox” para

operar. Así pues, los algoritmos que desarrollan el conjunto de funciones de optimización pueden ser algoritmos de media escala o bien algoritmos de amplia escala.

Esta distinción entre algoritmos se hace para diferenciar los algoritmos clásicos de los algoritmos de amplia escala, los cuales han sido diseñados para abordar problemas de amplia escala con más eficiencia.

Algunos ejemplos de los algoritmos que pertenecen a uno u otro de estos dos conjuntos diferenciados, son:

- **Algoritmos de media escala:** Entre los algoritmos clasificados como de media escala se encuentran con los algoritmos de programación cuadrática secuencial (SQP) usados en las funciones de minimización con restricciones, problemas de mínimo-máximo, alcance de objetivos y optimización semi-infinita. Y por último, se encuentran los algoritmos que usan los problemas de mínimos cuadrados no lineales, tales como los métodos de Gauss-Newton y Levenberg-Marquardt.

- **Algoritmos de amplia escala:** Los principales algoritmos de amplia escala son iterativos, es decir, generan una secuencia de soluciones aproximadas. En cada iteración se resuelve un sistema lineal de forma aproximada. Entre este tipo de algoritmos se encuentran los problemas de límites con restricciones que son resueltos usando métodos de Newton reflejados. También problemas de igualdades con restricciones para los cuales se usa una iteración de gradiente conjugado preconditionado. Para resolver los problemas de amplia escala de forma eficiente, se requiere información adicional (por ejemplo, si se usa uno de estos algoritmos para la minimización no lineal se requiere calcular computacionalmente el gradiente en la función objetivo que realiza el usuario).

Además, es importante tener en cuenta que los algoritmos de amplia escala no son aplicables a todo tipo de problemas que se puedan formular; por ejemplo, para una de las funciones empleadas en este trabajo “fmincon”, el algoritmo de amplia escala que usa MATLAB R2012a cubre los casos en los que solamente hay límites con restricciones. Para cada caso en el que se aplica un algoritmo de amplia escala se necesita una información adicional específica.

La función antes mencionada “fmincon” emplea uno de los siguientes algoritmos: active-set, interior-point, sqp, or trust-region-reflective .

### **2.5.1 Método del gradiente en MATLAB R2012a. La función “fmincon”**

De acuerdo con Vizcaíno G. (2006), “fmincon” se define como una función de optimización empleada para minimización con restricciones, la cual encuentra el mínimo de una función no lineal de varias variables con restricciones.

El propósito de esta función es encontrar el mínimo del valor escalar que devuelve una función de varias variables no lineal y sujeta a una serie de restricciones. Normalmente se conoce este tipo de problemas como optimización no lineal con restricciones o programación no lineal. La función escalar de varias variables se llamará función objetivo. El usuario debe implementar esta función objetivo para que se realicen los cálculos oportunos para obtener el valor escalar que devuelve la función. Se denota de la siguiente manera:

$$\begin{aligned}
 & \min_x f(x) \\
 & \text{sujeto a } c(x) = 0, \text{ ceq}(x) = 0, \\
 & A \cdot x = b, \text{ Aeq} \cdot x = \text{beq}, \\
 & \text{lb} = x = \text{ub}
 \end{aligned}
 \tag{31}$$

A, Aeq son matrices. f, x, b, beq, lb, ub son vectores. c(x) y ceq(x) son funciones que devuelven un vector. c(x), ceq(x) y f(x) pueden ser funciones no lineales.

De igual manera Vizcaíno G. (2006) expresa la necesidad del uso de funciones de resolución de ecuaciones no lineales. Para el caso que ocupa, una de las funciones que se emplea es la función “lsqnonlin” (resolvedor de problemas no lineales de mínimos cuadrados, incluyendo ajuste de datos) y “ode23” (resolvedor de ecuaciones diferenciales).

Los algoritmos de optimización internos que usan las rutinas del “toolbox” para operar son seleccionados como opciones de la función.

Uno de los argumentos de entrada más interesantes de ésta y otras funciones de minimización es la ya comentada función objetivo, *fun*, o función a ser minimizada. *fun* toma el vector x y devuelve un valor escalar, por ejemplo f, que es el valor que toma la función evaluada en x.

El argumento que define la función objetivo se puede especificar de dos formas. *fun* puede ser el nombre de una función con extensión .m, realizada por el usuario, y que realiza los cálculos necesarios para evaluar f(x) en x. La segunda forma de pasar el argumento *fun* a la función ‘fmincon’ es como un objeto en línea, es decir,

en el lugar del parámetro *fun* se escribe directamente la expresión de la función  $f(x)$ .

Se puede además obtener el gradiente de la función objetivo, para ello habrá que activar un elemento de la estructura de opciones de optimización “options” que permite que la función objetivo compute el cálculo del gradiente. Para activar dicho elemento en la estructura de optimización se usa ‘optimset’, que es una instrucción que permite crear o modificar los parámetros de optimización de la estructura. El uso de esta instrucción es como sigue en este caso:

```
options = optimset('GradObj', 'on')
```

El gradiente es la derivada parcial de  $f$  en el punto  $x$ . Es decir, si se llama  $g$  al gradiente, el  $i$ -ésimo componente de  $g$  sería la derivada parcial de la función  $f(x)$  respecto del  $i$ -ésimo componente del vector  $x$  y evaluada en la solución  $x$ . En el caso de que se active la opción de cálculo del gradiente, la función *fun* devuelve dos parámetros de salida, es decir, se tiene en este caso algo tal que así:

```
function [f,g] = myfun(x)
f = ...;
if nargin > 1 % fun se usa con dos argumentos de salida
g = ...;
end
```

También se puede calcular la matriz Hessiana o discriminante, activando esta opción en la estructura *options*. Es decir, se debería escribir antes de usar “fmincon”:

```
options = optimset('Hessian', 'on')
```

En este caso, también se puede chequear el valor de la variable “nargout” para evitar el computo de H si la función fun se llama sólo con uno o dos argumentos de salida (es decir, en el caso en el que la optimización solo necesita los valores de f y g, pero no H).

La matriz Hessiana es una matriz simétrica, donde cada elemento es una derivada parcial segunda. Esto es, el elemento (i,j) de la matriz H sería la derivada parcial segunda de f con respecto al elemento i-ésimo y j-ésimo de x.

La estructura de la función objetivo en el caso de que se compute la matriz Hessiana sería algo tal que así:

```
function [f,g,H] = myfun(x)
f = ...;
if nargout > 1 % fun se usa con dos argumentos de salida
g = ...;
if nargout > 2
H = ...; % matriz Hessiana evaluada en x
end
end
```

Otro argumento de entrada de gran importancia a la hora de usar tanto la función de minimización no lineal “fmincon”, como cualquiera de las demás funciones del “toolbox” de Optimización de MATLAB R2012a, es la estructura de opciones de optimización, llamada “options”. Esta estructura contiene varios campos que permiten influir de manera notable en el proceso que lleva a cabo la rutina de optimización.

Hay dos argumentos de salida que merecen ser tenidos en cuenta. Estos dos parámetros son *exitflag* y *output*. El argumento de salida *exitflag* es una bandera que indica al usuario la razón por la que ha finalizado la función de optimización y puede tomar los siguientes valores:

- $\text{Exitflag} > 0$ , indica que la función  $f(x)$  converge en la solución  $x$ .
- $\text{Exitflag} = 0$ , indica que se ha alcanzado el número máximo de evaluaciones permitidas a la función  $f(x)$  o de iteraciones de la función `fmincon`.
- $\text{Exitflag} < 0$ , indica que la función  $f(x)$  no converge a una solución.

El argumento de salida *output* es una estructura, cuyos campos contienen información sobre la optimización que se ha llevado a cabo en la rutina. Los parámetros de esta estructura y el significado de ellos son:

- `output.Iterations`: iteraciones llevadas a cabo por `fmincon`.
- `output.funcCount`: número de evaluaciones de la función objetivo.
- `output.algorithm`: el algoritmo usado por la función `fmincon`.
- `output.cgiterations`: número de iteraciones de PCG (solo para el caso de algoritmos de escala amplia).
- `output.stepsize`: el tamaño del paso final tomado (solo para algoritmos de media escala).
- `output.firstorderopt`: Es una medida de optimicidad de primer orden (solo para algoritmos de escala amplia).



### 2.5.1.1 Estructura de parámetros de optimización “options” para “fmincon”

Algunos de los parámetros son aplicables a todos los algoritmos, otros solo son relevantes si la rutina de optimización usa un algoritmo de amplia escala y de forma similar, hay parámetros que solo tienen sentido para la función de optimización si esta usa un algoritmo de escala media. Para la función *fmincon*, solo tiene sentido usar algunos de los parámetros de la estructura *options*, pero antes de hablar de cómo afectan estos a la función, es necesario ver todos los parámetros que componen la estructura *options* y una breve descripción de cada uno en la Tabla 2.1.

### 2.5.1.2 Parámetros de la estructura “options” para “fmincon”

Una vez vistos todos los parámetros que integran los campos de la estructura de opciones de optimización, se presentan los parámetros de mayor influencia en el caso de estudio. Hay varios parámetros que influyen en el proceso de optimización que lleva a cabo la rutina, entre los más influyentes están: *TolFun*, *TolX*, *MaxFunEvals*, *MaxIter*. Hay que decir que el parámetro *TolCon* tiene tanta o más importancia que los parámetros *TolFun* y *TolX*.

- Los parámetros *TolFun* y *TolX*: Los campos *TolFun* y *TolX* son parámetros de tolerancias relativas que pueden influenciar en cómo y cuando se converge a una solución, y por tanto afectan a la solución que encuentra la función de optimización, *x* y *fval*. *TolFun* es la tolerancia de finalización que se permite a la función objetivo *fun*.

**Tabla 2.1. Descripción de los parámetros para “fmincon”**

<b>PARÁMETRO</b>	<b>DESCRIPCIÓN</b>
DerivativaCheck	Compara derivadas analíticas proporcionadas por el usuario (gradientes y jacobianos) y derivadas diferenciales finitas.
Diagnostic	Imprime información de diagnóstico sobre la función.
DiffMinChange	Mínimo incremento diferencial en los elementos del vector $x$ .
DiffMaxChange	Máximo incremento diferencial en los elementos del vector $x$ .
Display	Nivel de Display. Si se establece éste parámetro como “off” no muestra nada a la salida de la función. Si se establece como “iter” muestra la salida en cada iteración de la función; si se establece como “final” solo muestra la salida al finalizar la función.
GoalsExatAchive	Números de logros que se desean conseguir.
GradConstr	Gradiente de las restricciones no lineales definidas por el usuario
GradObj	Gradiente de la función objetivo definida por el usuario. El gradiente de la función debe ser proporcionado cuando se usa un algoritmo de amplia escala. En caso contrario, se usa uno de media escala, es opcional.
Hessian	Matriz Hessiana para la función objetivo
HessPattern	Grado de libertad de la matriz hessiana en diferenciales finitos.
HessUpdate	Esquema de actualización Quasi-Newton.
Jacobian	Jacobiano de la función objetivo.
JacobPattern	Usar un algoritmo de amplia escala si es posible. Cuando el parámetro “largeScale” se establece a “on”. Cuando está establecido como “off” la función usará uno de media escala.
LeverbergMarquardt	Elegir un algoritmo de Leverberg-Marquardt en lugar de un algoritmo de Gauss-Newton.
LineSearchType	Línea de búsqueda del algoritmo elegido.
MaxFunEvals	Máximo número de evaluaciones de “fun” permitidos.
MaxIter	Máximo número de iteraciones de la función de optimización permitidos.
MaxPCGIter	Máximo número de iteraciones PCG (precondiciones conjugate gradient) permitido. Sólo será necesario cuando se use un algoritmo de amplia escala en la rutina de optimización.
MeritFunction	Usar una función de mérito.
MinAbsMax	Número de $F(x)$ a minimizar en el peor de los casos.
PrecondBandwidth	Ancho de banda superior del precondicional para el PCG.
TolCon	Tolerancia de finalización en la violación de las restricciones.
TolFun	Tolerancia de finalización en el valor de “fun”
TolPCG	Tolerancia de finalización en la iteración de “PCG”
TolX	Tolerancia de finalización en el valor de “x”.
TypicalX	Valores típicos de los elementos del vector “x”. Sólo se usará en los algoritmos de amplia escala.

Fuente: VIZCAÍNO, G., (2006).

Cuando la función de optimización use como criterio de optimización el valor del parámetro TolFun, entonces buscará que la derivada direccional en la dirección de búsqueda del valor que devuelve fun sea al menos  $2 \cdot \text{options.TolFun}$ . Es decir, se

converge a una solución cuando la pendiente entre dos valores consecutivos del valor que devuelve la función objetivo sea al menos del orden de  $2 * TolFun$ .

La función de optimización se parará si la pendiente entre dos valores consecutivos del valor devuelto es menor de lo especificado. Si se sigue este criterio de optimización, al finalizar satisfactoriamente la rutina de optimización, se mostrará en la pantalla del computador un mensaje como el siguiente:

*“Optimization terminated successfully: Magnitude of directional derivative in search direction less than  $2 * options.TolFun$  and maximum constraint violation is less than  $options.TolCon$ .”*

Del mismo modo, el parámetro *TolX* es una tolerancia de terminación para los valores estimados del vector solución *x*. Si se usa como criterio de optimización el valor del parámetro *TolX*, se buscará que el cambio del valor de los parámetros del vector *x* sea menor que *TolX*. El mensaje que se mostrará por pantalla cuando se haya encontrado una solución y finalice la rutina sería como el siguiente:

*“Optimization terminated successfully: Search direction less than  $2 * options.TolX$  and maximum constraint violation is less than  $options.TolCon$ .”*

Es muy importante tener en cuenta que las rutinas de optimización hacen un OR entre los parámetros de optimización *TolFun* y *TolX*, así que el hecho de seguir uno u otro criterio de terminación dependerá del valor que tengan tales parámetros. Por defecto, *TolFun* y *TolX* tienen el mismo valor,  $1e-006$ . En caso de que no se cambien estos dos valores, y al ser los dos parámetros iguales, la función sigue cualquiera de los dos criterios de optimización vistos. Para cambiar el valor de cualquier parámetro de la estructura *options*, como ya se ha visto anteriormente, se

usa la función *optimset*. Alterar el valor de los parámetros *TolFun* y *TolX* supone alterar la solución a la que converge la rutina de optimización.

De forma similar, si se disminuye el valor de *TolFun* y/o *TolX*, por ejemplo de  $1e-006$  a  $1e-008$  se limita uno o los dos criterios de optimización, resultando en una solución más exacta, aunque muy posiblemente se requieran un mayor número de iteraciones de la función de optimización cuando se observa que la solución encontrada es subóptima. Normalmente el usuario disminuye los valores de estos parámetros cuando observa que la solución encontrada es subóptima.

***Los parámetros MaxFunEvals y MaxIter:*** El parámetro *MaxIter* es el número de iteraciones permitidas en el proceso de estimación que lleva a cabo la función *fmincon*. Cada iteración involucra una optimización, y por lo tanto distintas evaluaciones de la función objetivo *fun*. El valor por defecto que tiene *MaxIter* es 400. Aunque el proceso de estimación raramente excede este número de iteraciones de la función de optimización, se puede incrementar este valor usando *optimset* si se sospechase que el proceso ha terminado prematuramente.

*MaxFunEvals*, un campo estrechamente relacionado con el anterior, *MaxIter*, especifica el máximo número de evaluaciones que se realizarán, a lo largo de todo el proceso de estimación de una solución óptima, de la función objetivo. El valor por defecto de *MaxFunEvals* es 100 veces el número de parámetros que se están estimando. Es decir, si la función objetivo recibe un vector con 3 elementos, el valor de *MaxFunEvals* es 300. Cuando la rutina de optimización acaba prematuramente, sin encontrar una solución, será normalmente porque se ha excedido este parámetro. Al igual que en el caso anterior, se puede incrementar este valor usando *optimset* si el proceso ha finalizado y se sospecha que se puede conseguir una mejor solución.

Los campos *MaxFunEvals* y *MaxIter* son por naturaleza puramente mecánicos. Aunque el usuario puede encontrar situaciones en las que se alcance el valor de *MaxFunEvals* y/o *MaxIter*, no suelen ser situaciones comunes. Si se aumenta el valor de estos campos, se puede converger a una solución mas satisfactoria; ahora bien, si la rutina de optimización finaliza por haber alcanzado alguno de estos valores, se está indicando que el modelo descrito es demasiado complicado. Finalmente, aunque los parámetros *MaxFunEvals* y *MaxIter* pueden ser causa de finalización de la función de optimización, hay que decir que su valor no afecta a la solución una vez ésta ha sido encontrada.

### **2.5.2 Algoritmo DIRECT en MATLAB R2012a: “gblSolve”.**

Como se mencionó en el apartado 2.4 de este capítulo, el algoritmo DIRECT, que se implementa en el trabajo, se hace mediante el solucionador “gblSolve.m” de MATLAB R2012a, que resuelve los problemas de optimización global de una función multivariable sujeta a restricciones de borde, utilizando únicamente los valores de la función.

Es importante destacar que MATLAB R2012a contiene dos (2) rutinas de implementación de DIRECT. La primera, incluida en el llamado “NLPLIB TB”, o caja de herramientas de MATLAB para la programación no lineal y estimación de parámetros de TOMLAB. Se denota por “gblsolve”. TOMLAB desarrollado por la Optimización Aplicada y el grupo de modelado (TOM) en la Universidad de Mälardalen Universidad, es un entorno MATLAB abierto para la investigación y la enseñanza en la optimización. TOMLAB es basado en NLPLIB TB y OPERA TB. Aunque TOMLAB incluye más de 65 algoritmos diferentes de optimización, hasta hace poco tiempo no se le había incluido una rutina que manejara los problemas de optimización global, como lo es el caso de algoritmo DIRECT.

La otra rutina de implementación del algoritmo DIRECT en MATLAB es la versión independiente usada en este trabajo y denotada por “gblsolve” o “GBL”, que se diferencia de la primera solo en el formato de entrada y salida. La siguiente tabla muestra los parámetros de entrada “gblsolve” en MATLAB R2012a.

**Tabla 2.2. Parámetros de entrada “gblsolve” en MATLAB R2012a**

<b>Notación</b>	<b>Descripción</b>
x_L	límites inferiores para x
x_U	límites superiores para x
GLOBAL.iterations	número de iteraciones para las corridas. El Número por defecto que asigna MATLAB es de 50.
GLOBAL.epsilon	Parámetro de asignación del peso a la búsqueda Global/local. El valor por defecto es de $1 \times 10^{-4}$ .
Si se requiere reiniciar la corrida del algoritmo, se deben definir los siguientes campos en GLOBAL y la igualdad de los campos correspondientes en la estructura de resultados de la ejecución anterior:	
% GLOBAL.C	Matriz con todos los puntos centrales del rectángulo.
% GLOBAL.D	Vector con distancias de punto central a los vértices.
GLOBAL.L	Matriz con todas las longitudes de los lados del rectángulo en cada dimensión.
GLOBAL.F	Vector con los valores de la función.
GLOBAL.d	Vector fila de todas las diferentes distancias, ordenados.
GLOBAL.d_min	Vector fila del valor mínimo de función para cada distancia.
PriLev	Nivel de impresión
PriLev >= 0	Advertencias
PriLev > 0	información pequeña
PriLev > 1	información de cada iteración.
<b>Parámetros de salida</b>	
x_k	Matriz con el cumplimiento de todos los puntos $f(x)=\min(f)$ .
f_k	Valor más pequeño de la función hallado.
Iter	Número de iteraciones
FuncEv	Número de evaluaciones de la función.
GLOBAL.C	Matriz con todos los puntos centrales del rectángulo
GLOBAL.D	Vector con las distancias de punto central a los vértices.
GLOBAL.L	Matriz con todas las longitudes de los lados del rectángulo en cada dimensión.
GLOBAL.F	Vector con los valores de la función
GLOBAL.d	Vector fila de todas las diferentes distancias, ordenada.

**Fuente:** BJORKMAN, M. y Holmström, K. (2012).

### 2.5.3 Algoritmos genéticos en MATLAB R2012a

Existen dos formas para especificar las opciones para el caso de algoritmos genéticos, dependiendo si se está usando la herramienta de algoritmos genéticos (AG) o usando la opción “ga” en la línea de comando. Para este trabajo se llama el algoritmo genético por la línea de comando y se especifican las opciones por la creación de una estructura de opciones usando la función “gaoptimset”, como sigue:

```
options = gaoptimset;
```

Ya que no se colocan argumentos de entrada, se crea una estructura llamada opciones que contiene los parámetros del algoritmo genético y el conjunto de valores por defecto aparecen en la Tabla 2.3.

Las opciones están divididas en las siguientes categorías:

#### 2.5.3.1 Opciones de población

Las opciones de población permiten especificar los parámetros de población que usa el algoritmo genético.

**El tipo de población** (PopulationType) especifica el tipo de datos de la entrada de la función objetivo. El tipo de población vector doble (‘doubleVector’) se usa si los individuos en la población están en el conjunto de los números reales.

**Tabla 2.3. Opciones del algoritmo genético empleadas para el problema de optimización del sistema en estudio**

Nombre de la función	Nombre de la función según MATLAB R2012a	Especificación
Tipo de población	PopulationType	'doubleVector'
Rango inicial de población	PopInitRange	[2x7 double]
Tamaño de población	PopulationSize	20
Elite	EliteCount	2
Fracción de cruce	CrossoverFraction	0.8000
Dirección de la migración	MigrationDirection	'forward'
Intervalo de migración	MigrationInterval	20
Fracción de migración	MigrationFraction	0.2000
Generaciones	Generations	100
Límite de tiempo	TimeLimit	Inf
Límite de ajuste	FitnessLimit	-Inf
Generaciones de puesto	StallGenLimit	Inf
Tiempo Limite	StallTimeLimit	: Inf
Tolerancia de la función	TolFun	1.0000e-006
Tolerancia de restricción no lineal	TolCon	1.0000e-006
Población inicial	InitialPopulation	[]
Puntuaciones iniciales	InitialScores	[]
Penalización inicial	InitialPenalty	10
Factor de Penalización	PenaltyFactor	100
Intervalo de gráfico	PlotInterval	: 1
Función creación	CreationFcn	@gacreationuniform
Función de escalamiento de ajuste	FitnessScalingFcn	@fitscalingprop
Función selección	SelectionFcn	@selectionuniform
Función de cruce	CrossoverFcn	@crossoverscattered
Función Mutación	MutationFcn	{[1x1 function_handle] [0.3000]; [1x1 function_handle] [0.1000]}
Función Híbrido	HybridFcn	[]
Despliegue	Display	'iter'
Función de gráfico	PlotFens	[]
Función de salida	OutputFens	[]
Vectorización	Vectorized	'off'

**El rango inicial** (PopInitRange) especifica el rango de los vectores en la población inicial que se genera por la función creación. El rango inicial se establece como una matriz con dos (2) filas y un número de columnas variable, cada una de las cuales tiene la forma [lb; ub], donde lb es el límite inferior y ub es el límite



superior para las entradas. Se establece el valor por defecto que es un vector 2 x 1, así cada entrada se expande a una fila constante de longitud de número de variables.

El tamaño de la población (PopulationSize) especifica cuantos individuos hay en cada generación. Con un tamaño de población grande, el algoritmo genético busca la solución en un espacio más exhaustivamente, reduciendo así las oportunidades de que el ga retorne a un mínimo local que no es un mínimo global. Sin embargo, una población grande también hace que el ga corra lentamente.

Si se selecciona un vector como tamaño de población crea múltiples subpoblaciones de número correspondiente a la longitud del vector. El valor por defecto es 20.

**Población Inicial:** (InitialPopulation) especifica una población inicial para el algoritmo genético. El valor por defecto es [], en tal caso el algoritmo usa la función creación para crear una población inicial. Si no se introduce un vector vacío en éste campo el vector debe tener el número de filas del tamaño de la población y el número de variables de las columnas, para tal caso no se emplea la función creación. Por su parte las puntuaciones iniciales (InitialScores) se especifican para la población inicial.

**La función creación** (CreationFcn) especifica la función que crea la población inicial para el ga. Se puede escoger entre las siguientes funciones:

**Uniforme** (@gacreationuniform): crea una población inicial al azar con una distribución uniforme. La función Custom permite escribir una propia función, la cual debe generar datos del tipo especificados en el tipo de población.

### 2.5.3.2 Opciones de reproducción

Estas opciones especifican cómo el algoritmo genético crea los individuos o descendientes para la próxima generación.

**La cuenta de elite** (EliteCount) especifica el número de individuos que sobrevivirán hasta la próxima generación. EliteCount se debe seleccionar como un entero positivo menor o igual que el tamaño de la población. El valor por defecto es dos (2).

**La fracción de cruce** (CrossoverFraction) especifica la fracción de la próxima generación aparte de los descendientes de la élite que son producidos por cruce. El establecimiento de la fracción de cruce debe ser entre 0 y 1. El valor por defecto es 0.8.

Una fracción de 1 significa que todos los descendientes aparte de los individuos élite provienen del cruce, mientras que un valor de cero (0) significa que todos los descendientes provienen de la mutación.

### 2.5.3.3 Opciones de migración

Las opciones de migración especifican cómo los individuos se mueven entre subpoblaciones. La migración ocurre si el tamaño de la población es un vector de longitud mayor que 1. Cuando la migración ocurre, los mejores individuos desde una subpoblación reemplazan los peores individuos en otra subpoblación. Los

individuos que migran de una subpoblación a otra son copiados, por lo que no son removidos de la subpoblación fuente. Es posible controlar cómo ocurre la migración especificando los siguientes tres campos:

**Dirección** (MigrationDirection): la migración toma lugar en una o dos direcciones. Si se selecciona la dirección hacia adelante ('forward'), la migración toma lugar hacia la última subpoblación. Es decir, la subpoblación  $n$  migra hacia la subpoblación  $(n+1)$ . Si se selecciona la dirección hacia ambos ('both'), la subpoblación  $n$  migra hacia la subpoblación  $(n+1)$  y hacia subpoblación  $(n-1)$ .

**Intervalo** (MigrationInterval): especifica cuántas generaciones pasan entre migraciones. Por ejemplo si se selecciona el intervalo a 20, la migración toma lugar cada 20 generaciones.

**Fracción** (MigrationFraction): especifica cuántos individuos se mueven entre subpoblaciones. La fracción especifica la fracción de la más pequeña de las 2 subpoblaciones que se mueve. Por ejemplo si los individuos migran desde una subpoblación de 50 individuos hacia una subpoblación de 100 individuos y si se coloca una fracción de 0.2, el número de individuos que migrará será de 10.

#### 2.5.3.4 Opciones de criterios de parada

Estos criterios determinan las causas de culminación del algoritmo. Especificando las generaciones (Generations) se establece el número máximo de iteraciones que puede llevar a cabo el algoritmo. El valor por defecto es 100.

**Tiempo Límite (TimeLimit):** especifica el tiempo máximo en segundos que corre el algoritmo genético antes de parar.

**Límite de ajuste:** el algoritmo se detiene si el mejor valor de ajuste es menor que o igual al límite de ajuste.

**Generaciones de puesto (StallGenLimit):** el algoritmo se detiene si el cambio en el promedio ponderado del valor de la función objetivo sobre las generaciones de puesto es menor que la tolerancia de la función.

**Tolerancia de la función (TolFun):** el algoritmo corre hasta que el cambio acumulado en el valor la función objetivo sobre las Generaciones de puesto es menor que la tolerancia de la función.

**Tolerancia de restricción no lineal (TolCon):** no es usado como criterio de parada, sino para determinar la factibilidad con respecto a restricciones no lineales.

**Escenarios del algoritmo:** define los parámetros específicos del algoritmo, los que se pueden especificar incluyen la **penalización inicial (InitialPenalty)** que especifica el valor del parámetro de la penalización, debe ser mayor o igual a 1. Por su parte el **factor de penalización (PenaltyFactor)** incrementa el parámetro de penalización cuando el problema no se resuelve con la precisión requerida y las restricciones no se satisfacen. Debe especificarse mayor que 1.

#### 2.5.3.5 Opciones de gráfico.

Permiten graficar los datos desde el algoritmo genético mientras está corriendo. Cuando se seleccionan las funciones de gráfico y se corre el algoritmo genético, se

despliega una ventana de gráfico que grafica en ejes separados. Se puede parar la corrida del algoritmo en cualquier momento haciendo clic en el botón de parada en la ventana de gráficos.

**Intervalo de Gráfico** (PlotInterval): especifica el número de generaciones entre llamadas consecutivas de la función objetivo.

#### 2.5.3.6 Opciones de escalamiento de ajuste

El escalamiento de ajuste convierte las puntuaciones de ajuste que se retornan por la función ajuste a valores en un rango adecuado para la función de selección.

**La función escalamiento** (FitnessScalingFcn) especifica la función que representa el escalamiento. Una de las opciones es el **rango** (@fitscalingrank), que es la función por defecto la cual escala las puntuaciones basadas en el rango de cada individuo en lugar de su puntuación. El rango de un individuo es su posición en la clasificación de las puntuaciones. El rango del mejor individuo es 1, el rango del siguiente mejor individuo es 2 y así sucesivamente.

#### 2.5.3.7 Opciones de selección

Las opciones de selección especifican cómo el algoritmo genético escoge los padres para la próxima generación. Se puede especificar la función, en el campo de **selección** (SelectionFcn) Entre las opciones está la función **uniforme**

**estocástica** (@selectionstochunif), que es la función por defecto que distribuye una línea en la cual cada padre corresponde a una sección de la línea de longitud proporcional a su valor escalado. El algoritmo se mueve a lo largo de la línea de pasos en pasos de igual tamaño. A cada paso el algoritmo localiza un padre de una sección donde cae. El primer paso es un número uniforme aleatorio menor que el tamaño del paso.

#### 2.5.3.8 Opciones de cruce

Las funciones de cruce especifican cómo el algoritmo genético combina dos individuos o padres para formar un descendiente de cruce para la próxima generación. **La función de cruce** (CrossoverFcn) especifica la función que representa el cruce. La función por defecto es **dispersión** (@crossoverscattered), la cual crea un vector binario aleatorio y selecciona los genes donde el vector es un 1 desde el primer padre y los genes donde el vector es cero (0) desde el segundo padre y combina los genes para formar el descendiente.

#### 2.5.3.9 Opciones de mutación

Las opciones de mutación especifican cómo el algoritmo genético efectúa los pequeños cambios aleatorios en los individuos de una población para crear los descendientes. La mutación provee diversidad genética y permite al algoritmo genético buscar en un espacio más amplio.

Se especifica la mutación en el campo de la **función mutación** (MutationFcn). La función por defecto es la **Gaussiana** (mutationgaussian), la cual añade un número aleatorio tomado de una distribución gaussiana la cual es cero (0) para cada entrada del vector progenitor. La varianza de esta distribución se determina por la escala y reducción de parámetros las cuales se despliegan cuando se selecciona la función Gaussiana y por el establecimiento del rango inicial en las opciones de población.

#### **2.5.4 Recocido simulado (SA) en MATLAB R2012a**

De acuerdo con la librería de MATLAB R2012a, el resolvidor de “Simulated Annealing” o recocido simulado (SA) asume que la función objetivo tomará una entrada  $x$ , donde  $x$  tiene tantos elementos como el número de variables en el problema. La función objetivo calcula el valor escalar del objetivo y lo devuelve en un único argumento “ $y$ ”.

##### **2.5.4.1 Minimización mediante el uso de SIMULANNEALBND**

Para reducir al mínimo la función objetivo se emplea la función SIMULANNEALBND, se tiene que pasar a una función manejable de la función objetivo, así como la especificación de un punto de partida como el segundo argumento.

```
ObjectiveFunction = @simple_objective;
X0 = [0.5 0.5]; % Starting point
[x,fval,exitFlag,output] = simulannealbnd(ObjectiveFunction,X0)
```

Los dos primeros argumentos de salida devueltos por SIMULANNEALBND son  $x$ , el mejor punto de encontrado, y  $FVal$ , el valor de la función en el mejor punto. Un tercer argumento de salida,  $exitFlag$  devuelve un indicador que corresponde a la razón por la que SIMULANNEALBND se detuvo. SIMULANNEALBND también puede devolver un cuarto argumento, la producción, la cual contiene información sobre el desempeño del programa de solución.

#### **2.5.4.2 Opciones de recocido simulado. Configuración problema de optimización**

El SIMULANNEALBND realiza la búsqueda para un mínimo de una función mediante el recocido simulado. Para el presente trabajo se emplea SIMULANNEALBND para minimizar la función objetivo.

#### **Especificación de las opciones de temperatura**

El parámetro de temperatura utilizado en el recocido simulado controla los resultados de búsqueda generales. La temperatura para cada dimensión se utiliza para limitar el alcance de la búsqueda en esa dimensión. La caja de herramientas o



el “toolbox” de MATLAB R2012a, le permite especificar la temperatura inicial, así como formas de actualizar la temperatura durante el proceso de solución. Las dos opciones relacionadas con la temperatura son los “InitialTemperature” y “TemperatureFcn”.

### **Especificación de la temperatura inicial**

La temperatura inicial por defecto es 100 para cada dimensión. Si desea que la temperatura inicial sea diferente en diferentes dimensiones se debe especificar un vector de temperaturas. Esto puede ser necesario en los casos en los que problema se escala de manera diferente para cada dimensión.

```
options = saoptimset('InitialTemperature',[300 50]);
```

### **Especificar una función de la temperatura**

La función de la temperatura por defecto utilizado por SIMULANNEALBND se llama TEMPERATUREEXP. En el programa de “temperatureexp”, la temperatura en un paso determinado es 0,95 veces la temperatura en el paso anterior. Esto hace que la temperatura baje lentamente al principio, pero en último momento se hacen más frías más rápido que otros esquemas. Si se desea otro esquema, por ejemplo, el Boltzmann o el calendario "fast" recocido, entonces TEMPERATUREBOLTZ o TEMPERATUREFAST se pueden utilizar respectivamente.

```
options = saoptimset(options,'TemperatureFcn',@temperaturefast);
```

Para el sistema que se estudia se mantiene la opción por defecto “temperatureexp”.

### **Especificación de reasociación**

La reasociación o reannealing (en inglés) es una parte del proceso de recocido. Después que se acepta un cierto número de nuevos puntos, se eleva la temperatura a un valor más alto con la esperanza de reiniciar la búsqueda y salir de un mínimo local. Si se efectúa la reasociación demasiado pronto, no ayuda al solucionador a identificar un mínimo, por lo que un intervalo relativamente alto es una buena opción. La frecuencia con la que ocurre reasociación se puede ajustar mediante la opción “ReannealInterval”.

```
options = saoptimset(options,'ReannealInterval',50);
```

Para el sistema estudiado se escoge “ReannealInterval” de 500, con la intención de reiniciar la búsqueda de forma más frecuente y escapar de mínimo locales.

### **La modificación de los criterios de parada**

SIMULANNEALBND utiliza seis criterios diferentes para determinar cuándo detener el solucionador. SIMULANNEALBND se detiene cuando se sobrepasa el

número máximo de iteraciones de evaluación de la función (MaxIter). Por defecto el número máximo de evaluaciones de la función es:

$$3000 * (\text{numberOfVariables})$$

SIMULANNEALBND realiza un seguimiento de la evolución media del valor de la función para las iteraciones “StallIterLimit”. Si el cambio promedio es menor que la tolerancia de la función, “TolFun”, entonces el algoritmo se detiene. El solucionador también se detendrá cuando el valor de la función objetivo alcanza “ObjectiveLimit”.

Por último, el resolvidor se parará después de correr para los segundos del “TimeLimit”. Por ejemplo; se establece la TolFun a 1e-5.

```
options = saoptimset(options,'TolFun',1e-5);
```

Se establece para el sistema estudiado, el número máximo de iteraciones (MaxIter) en 20000.

### **Adición de visualización**

SIMULANNEALBND puede aceptar una o más funciones de gráficas a través de un argumento "opciones". Esta característica es útil para visualizar el rendimiento

del solucionador en el tiempo de ejecución o corrida del programa. Las gráficas de las funciones se seleccionan mediante SAOPTIMSET. La ayuda para SAOPTIMSET contiene una lista de funciones de gráficos para seleccionar, o también se puede proporcionar funciones personalizadas de trazado de gráficos.

Para seleccionar varias funciones de gráficos, se utiliza SAOPTIMSET para crear una estructura de opciones. Para el caso de estudio se emplea: options.PlotFcns: @saplotbestf, que traza el mejor valor de la función de cada iteración.

### **Límites de minimización con restricciones**

SIMULANNEALBND se puede utilizar para resolver problemas con restricciones limitadas. Los límites inferior y superior se pasan al solucionador como vectores. Para cada dimensión  $i$ , el solucionador asegura  $LB(i) \leq x(i) \leq UB(i)$ , donde  $x$  es un punto seleccionado por el solucionador durante la simulación.

# **CAPÍTULO III**

## **MARCO METODOLÓGICO**

## CAPÍTULO III. MARCO METODOLÓGICO

### 3.1. TIPO DE INVESTIGACIÓN

**Nivel de investigación:** El tipo de investigación según el nivel o el grado de profundidad con el cual se aborda el fenómeno constituido por el diseño integrado se refiere a una investigación del tipo descriptiva en el área de control de procesos, ya que según Arias (2000) ésta consiste en la caracterización de un hecho, fenómeno o grupo con el fin de establecer su estructura o comportamiento.

**Diseño de investigación:** El tipo de investigación según el diseño, se enmarca dentro de una investigación documental, porque para realizar el presente estudio se tuvo que realizar una exhaustiva revisión bibliográfica de todos los estudios previos y actualizados sobre diseño integrado. El diseño integrado ha sido la consecuencia de todo el desarrollo industrial y del control automático de procesos. La tecnología de las plantas, requerimientos de calidad, operabilidad, rentabilidad y controlabilidad son cada día más exigentes y por ende los avances en el desarrollo de diseños que cumplan dichos requerimientos se actualizan constantemente.

Las técnicas de recolección de datos consisten en el análisis documental y el análisis de contenido y la información obtenida se tabula y procesa empleando

procedimientos propios de la investigación científica como balances de materia y energía en estado estacionario y transitorio. Se emplean además las distintas ecuaciones expuestas en el diseño integrado de reactores mezcla completa, se codifican en un programa computacional y se registran los resultados obtenidos bien sea mediante tablas o gráficas los resultados obtenidos. Entre las técnicas lógicas se emplean tanto la síntesis como el análisis siendo los principales instrumentos el computador junto con una herramienta computacional adecuada.

El cumplimiento de los objetivos propuestos se desarrollará a lo largo de tres etapas: diagnóstico, estudio de factibilidad y diseño propiamente dicho en el cual se plantea un procedimiento dividido en 6 pasos.

**La etapa de diagnóstico** corresponde a la recopilación de la información necesaria a través de revisiones bibliográficas y documentales de libros, publicaciones y trabajos asociados al diseño integrado de una red de reactores mezcla completa. También se contempla en esta etapa la organización, clasificación y selección de la información recopilada.

**El estudio de factibilidad** de la investigación está íntimamente relacionado con la disposición de recursos materiales, técnicos, económicos, humanos, tiempo y de información que permitan establecer que el proyecto es viable y que se puede realizar. Entre los recursos materiales y técnicos comprenden básicamente el computador y la herramienta computacional, mientras que los recursos económicos representan montos despreciables considerando que ya se dispone previamente de dichos recursos técnicos. El recurso más importante en el proyecto se refiere al tiempo que se necesita invertir en la investigación documental pasando por la programación o implementación del algoritmo hasta la obtención de resultados producto de la corrida del programa. El proyecto se hace

completamente viable y accesible tomando en consideración la disponibilidad del tiempo.

**En la etapa de diseño** el procedimiento que se utiliza para efectuar el diseño integrado del proceso reactivo consiste de 6 pasos básicos. El primer paso puede responderse en la fase preliminar del trabajo de grado, es decir, en el proyecto, permitiendo obtener las premisas para el desarrollo del diseño integrado para el proceso seleccionado.

### **3.2. PROCEDIMIENTO DE TRABAJO**

A continuación se señalan tres pasos que se emplean para la consecución de los objetivos planteados.

**Paso 1: Plantear todos los aspectos técnicos necesarios que describan el proceso, definiendo el conjunto de variables y parámetros, así como los niveles deseados de las salidas y restricciones del caso específico a diseñar.**

Establecer las consideraciones que permitan conocer en detalle el proceso que representa el objeto de estudio. Se debe especificar todo lo relacionado a:

- Tipo de unidades que presenta el proceso.
- La operación unitaria que en ellos ocurre.
- El tipo de reactores que se empleará para el estudio.
- La reacción que ocurre en ellos.



En el caso que ocupa al presente trabajo, el sistema lo conforma un proceso reactivo que constituye una operación unitaria química. El tipo de unidades que presenta el proceso son dos reactores mezcla completa con chaquetas de enfriamiento, en donde ocurre una reacción de primer orden elemental irreversible y exotérmica:  $A \longrightarrow B$ . Se consideran 2 reactores y la configuración será determinada.

Establecer los resultados a obtener:

- Los objetivos de efectuar el diseño del proceso
- Finalidad primordial del sistema de dos reactores mezcla completa
- Tipo de sistema de control.
- Objetivo de implementar el sistema de control

De esta manera se establece que el objetivo es determinar las dimensiones físicas de las unidades de procesos, condiciones de operación y parámetros de entonación del control que optimicen tanto los criterios económicos como la controlabilidad del proceso y a la vez garantice una operación factible.

**Definir el conjunto de variables su clasificación y las técnicas de optimización objeto de estudio.**

Especificar:

- Variable manipulada
- Variable controlada
- Los diferentes tipos de variables que caracterizan la formulación matemática para éste problema.

En el caso de estudio, la variable manipulada la representa el flujo de refrigerante y la variable controlada la representa la temperatura de los reactores. Las variables dinámicas del modelo matemático se representan por  $Z_i$  y las variables del problema de optimización por  $X_i$ .

Para resolver el problema de diseño integrado es necesario especificar bajo que técnica de optimización se pretende dar solución al planteamiento matemático, el objetivo del trabajo es explorar las dos áreas definidas en el campo de la optimización, como lo son:

- Métodos clásicos: Se estudian los métodos de búsqueda local, específicamente la técnica de optimización basada en búsqueda de gradientes y DIRECT.
- Métodos heurísticos: Se utilizan los métodos probabilísticos como lo es el recocido simulado y basados en población como lo son los algoritmos genéticos. Ambos son métodos globales de optimización.

### **Establecer cuáles son los objetivos de optimización.**

Los objetivos están representados por:

- “El costo total” para la parte de diseño económico y
- “El error cuadrático integral (ISE)” para el diseño del control.

El costo total es una función del costo de capital y el costo de utilidad. El costo de capital está determinado por el diámetro y la altura de los reactores, por lo tanto se debe plantear la correlación matemática, así como también para el costo de

utilidad, que está relacionado con el flujo de agua de enfriamiento. Finalmente se debe obtener una expresión para el costo total.

Por su parte “el error cuadrático integral (ISE)” es uno de los criterios de desempeño que permite evaluar el comportamiento de los sistemas y representa la base para optimizarlos respecto del comportamiento de su error en el estado transitorio.

El ISE se representa a través de una ecuación diferencial que involucra las temperaturas del reactor y sus puntos de consigna. El objetivo de la controlabilidad dinámica es la minimización del ISE y se denota con la letra “ $\mu$ ”. Las ventajas de su empleo radican en que proporciona importancia a los errores grandes y presenta una rápida respuesta.

**Paso 2: Establecer la estructura del proceso con su respectiva modelación matemática para la resolución del problema de diseño integrado del sistema, empleando métodos de optimización clásica y heurística.**

Se establece el modelo dinámico para la optimización del sistema que representa la formulación del diseño integrado y se resuelve el problema de diseño integrado con el uso de los métodos de optimización seleccionados, implementándolo en una plataforma computacional adecuada.

El problema de diseño integrado para el sistema se resuelve de la siguiente forma:

**Establecer una estructura algorítmica** que determine los pasos necesarios para la obtención de los valores óptimos de diseño, configuración de los equipos y parámetros del controlador.

Esta estructura algorítmica se descompone en tres partes:

- **La naturaleza multiobjetivo:** el planteamiento matemático es multiobjetivo, un objetivo es la minimización los costos totales y el otro es la minimización del error cuadrático integral para resolver el problema multiobjetivo se convierte uno de los objetivos en restricción  $(obj_j \leq \varepsilon)$ , en este caso el error cuadrático integral pasa a formar parte de las restricciones no lineales del problema, lo que implica resolver el problema multiobjetivo de control óptimo entero mixto (MIOCP) a diferentes valores de  $\varepsilon$  que van desde el óptimo controlable hasta el óptimo económico, en este punto el problema multiobjetivo se convierte en un problema de programación no lineal entero mixto con ecuaciones algebraico diferenciales MINLP-DAE.
- **El problema entero mixto** conocido por sus siglas en inglés MINLP-DAE (Mixed Integer No Linear Programming- Differential Algebraic Equations) o programación no lineal entera mixta con ecuaciones algebraico diferenciales. La característica entero-mixta se resuelve obteniendo separadamente el diseño integrado para cada topología y seleccionando la que minimice los costos. De esta forma se reduce el problema a uno NLP-DAE (No Linear Programming.- Differential Algebraic Equations), o problema de

programación no lineal con ecuaciones algebraico diferenciales. Nótese que la característica entera-mixta se resuelve en esta parte.

- **El problema de control óptimo:** involucra la aplicación de parametrización de control al sistema para reducirlo de programación dimensional infinito a uno de programación dimensional finito.

**Implementación de la estructura algorítmica en la herramienta computacional seleccionada, empleando MATLAB R2012a:**

- Definir la perturbación y las condiciones iniciales de las ecuaciones algebraico-diferenciales.
- Formular las restricciones lineales de igualdad y desigualdad.
- Resolver las ecuaciones diferenciales ordinarias con el uso de MATLAB R2012a.
- Formar las restricciones no lineales.
- Emplear como herramienta matemática MATLAB R2012a para resolver el problema de diseño integrado, para los diferentes métodos de optimización seleccionados.

De las corridas del programa se deberán obtener: topologías del sistema (tipo de interconexión), las dimensiones físicas de las unidades de procesos, condiciones de operación y los parámetros de entonación así como también la relación cuantitativa entre los objetivos económicos del estado estacionario y el objetivo de controlabilidad dinámica.

### **Paso 3: Determinar la técnica de optimización de mejor desempeño en la aplicación del diseño integrado de reactores mezcla completa**

Los resultados establecen cuantitativamente el mejor desempeño de los métodos de optimización para el diseño integrado de reactores mezcla completa.

El procedimiento metodológico de los pasos 1 y 2 se desglosa a continuación, mientras que los resultados de la implementación de la estructura algorítmica para cada método de optimización se presentan y discuten en el Capítulo IV.

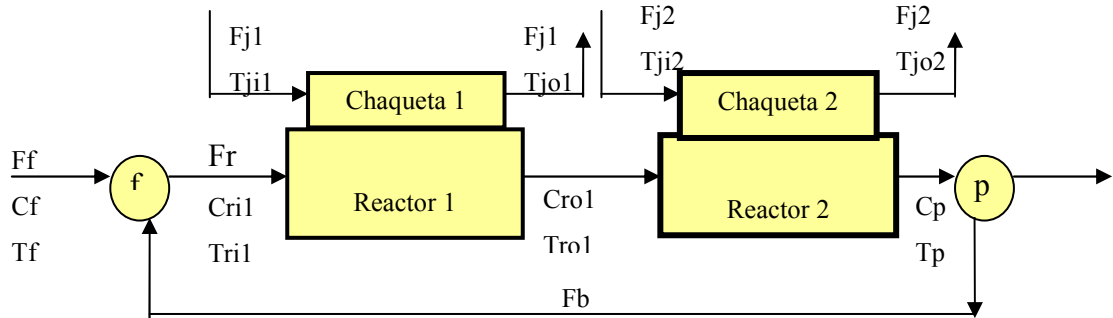
#### **Planteamiento de los aspectos técnicos que describen el proceso**

##### **El sistema**

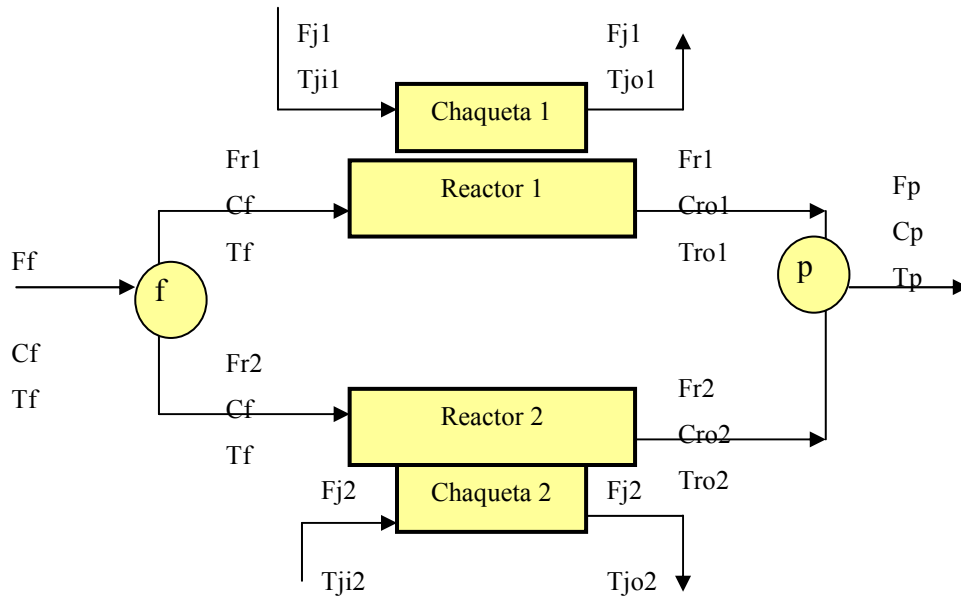
##### **Tipo de unidades. Operación unitaria. Tipo de reacción**

El sistema a considerar y que se describe a continuación está constituido por dos reactores mezcla completa con chaquetas de enfriamiento, donde ocurre una reacción de primer orden irreversible y exotérmica:  $A \longrightarrow B$ . Las velocidades de reacción en cada reactor son  $k_1$  y  $k_2$ , respectivamente. Para el sistema se supone que tanto el refrigerante en la chaqueta de enfriamiento como el fluido del proceso en el reactor, están bien mezclados y que la densidad y la capacidad calorífica de ambos no cambia significativamente con la temperatura, el nivel y el área de transferencia de calor en el tanque son constantes, el sistema se encuentra bien aislado y se desprecia la dinámica de transferencia de calor entre las paredes del tanque. Antes de perturbar el sistema se supone que el mismo se encuentra

operando en estado estacionario. El esquema del proceso se muestra en la figura 3.1 y 3.2, respectivamente.



**Figura 3.1 Diagrama de bloques del proceso con los reactores en serie.** Fuente: Schweiger y Floudas (1997)



**Figura 3.2 Diagrama de bloques del proceso con los reactores en paralelo.** Fuente: Schweiger y Floudas (1997)

A fin de mantener una operación estable es necesario controlar la temperatura de reacción, para lo cual se disponen controladores del tipo proporcional-integral

(PI), uno para cada reactor que controlarán el flujo de refrigerante a través de las chaquetas de enfriamiento.

### **Resultados que se deben obtener**

Ellos son: las dimensiones físicas óptimas (diámetro y altura) de los reactores mezcla completa para la reacción dada y los parámetros de entonación del controlador asociados al control de la temperatura del sistema para que se lleve a cabo efectivamente la reacción. El diseño es óptimo tanto a los criterios económicos como a la controlabilidad del proceso garantizando una operación factible.

Definiendo los términos empleados en la resolución del problema se tiene que los objetivos están representados por “el costo total” que representa explícitamente el diseño del proceso y “el error cuadrático integral” (ISE), el cual se denota con la letra “ $\mu$ ” y esta en función de la suma de los errores entre la temperatura de los reactores y su respectivos set-points.

El costo total es una función del costo de capital y el costo de utilidad. El costo de capital está determinado por el diámetro y la altura de los reactores, así como el costo de utilidad se relaciona con el flujo de agua de enfriamiento. El ISE se representa a través de una ecuación diferencial que involucra las temperaturas del reactor y sus set-points. El objetivo de la controlabilidad dinámica es la minimización del ISE.



Como se puede apreciar, el planteamiento matemático es multiobjetivo, un objetivo es la minimización los costos totales y el otro es la minimización del error cuadrático integral.

Para resolver el problema multiobjetivo se convierte uno de los objetivos en restricción ( $obj_j \leq \varepsilon$ ), en este caso el error cuadrático integral pasa a formar parte de las restricciones no lineales del problema, lo que implica resolver el problema multiobjetivo de control óptimo entero mixto (MIOCP) a diferentes valores de  $\varepsilon$  que van desde el óptimo controlable hasta el óptimo económico, en este punto el problema multiobjetivo se convierte en un problema de programación no lineal entero mixto con ecuaciones algebraico diferenciales MINLP-DAE.

Lo que convierte el planteamiento matemático en una programación entera mixta es la topología del mismo, la cual puede ser en paralelo o en serie, para tratar este problema se decidió estudiar ambas topología para luego establecer las comparaciones de una con respecto a la otra, tanto desde el punto de vista económico como desde el punto de vista de su comportamiento dinámico ante perturbaciones, por esta razón el problema pasa a ser una programación no lineal con ecuaciones algebraico diferenciales NLP-DAE

A continuación se enlistan los parámetros constantes del proceso y sus variables. La idea de emplear estos datos corresponde a la validación de la metodología presentada con el mismo caso de reacción química resuelto por Schweiger y Floudas (1997).

## Parámetros constantes del proceso

**Tabla 3.1 Parámetros que se asumen constantes en el problema**

Descripción	Parámetro	Valor
Calor de reacción	$\Delta H$	-3000 Btu/lbmol
Coefficiente de transferencia de calor	U	300 Btu
Energía de avivación	E/R	15075°R
Constante cinética de velocidad de reacción	Ko	40.8 hr <sup>-1</sup>
Densidad del líquido	$\rho$	50 lb/ft <sup>3</sup>
Capacidad calorífica del líquido	Cp	0.75 Btu/(lb°R)
Densidad del refrigerante		62.3 Btu/(ft <sup>3</sup> °R)
Capacidad calorífica del refrigerante	Cj	1.0 Btu/(lb°R)
Flujo de alimentación	Ff	100 ft <sup>3</sup> /hr
Composición de la alimentación	Cf	1 lbmol A/ ft <sup>3</sup>
Temperatura de la alimentación	Tf	600°R
Temperatura de entrada del refrigerante	Tji	530° R
Costo del refrigerante	Cost	3.74x10 <sup>-6</sup> \$/ ft <sup>3</sup>

Fuente: Schweiger C., Floudas C.,(1997).

## Variables y su clasificación.

Se debe especificar cuáles son las variables manipuladas, controlada y todos los diferentes tipos de variables que caracterizan la formulación matemática para éste problema.

En el caso de estudio la variable manipulada la representa el flujo de refrigerante, la variable controlada la representa la temperatura de los reactores y la

perturbación esta representada por incrementos tipo escalón en la temperatura de entrada de los reactantes.

Según Schweiger y Floudas, (1997), para el problema de diseño integrado de un sistema como el mostrado en las figuras anteriores se presentan diferentes tipos de variables y restricciones. Las variables enlistadas en la tabla 3.1 están divididas en dos categorías: continuas y enteras.

Las variables continuas las representan las velocidades de flujo, composiciones, temperaturas, dimensiones de equipos, entre otras. Las variables continuas son adicionalmente clasificadas como variables de diseño o variables de decisión invariantes en el tiempo “ $v$ ”, variables de estado dinámico “ $z(t)$ ” y variables de control o variables de decisión dependientes “ $u(t)$ ”. Las variables enteras “ $y$ ” son usadas para representar la existencia de unidades de proceso.

**Tabla 3.2. Tipos de variables para la formulación matemática**

Notación	Descripción
$V$	Variables de decisión invariantes en el tiempo
$Y$	Variables enteras
$z(t)$	Variables de estado dinámico
$u(t)$	Variables de control o variables de decisión dependientes
$T$	Tiempo
$T_i$	Instante de tiempo

**Fuente:** Schweiger C., Floudas C.(1997)

Las variables dinámicas son aquellas tales como composición y temperatura las cuales varían con el tiempo.

Las variables de control representan cantidades que pueden ser manipuladas en el tiempo para mantener controladas las variables a niveles preespecificados, la representa el flujo de refrigerante. Las variables controladas son un subconjunto de las variables dinámicas las cuales tienen un valor deseado o set-point, la representa la temperatura de los reactores.

La perturbación está representada por incrementos tipo escalón en la temperatura de entrada de los reactantes.

La definición de las variables dinámicas y de diseño con las unidades en el sistema inglés se muestra en las Tablas 3.3 y 3.4. Es importante señalar que para efectos del control de los reactores se usará un controlador para cada uno de ellos del tipo proporcional-integral (PI).

### **Objetivos en el problema de diseño integrado.**

El objetivo primordial que se persigue con el diseño del proceso objeto de estudio es determinar las dimensiones físicas óptimas (diámetro y altura) de los reactores mezcla completa para la reacción y los parámetros de entonación del controlador asociados al control de la temperatura del sistema para que se lleve a cabo efectivamente la reacción.

**Tabla 3.3. Variables del estado dinámico**

<b>Variables</b>	<b>Descripción</b>	<b>Variables</b>	<b>Descripción</b>
$C_{ri1}$ ( $ft^3 / h$ )	Concentración a la entrada del primer reactor	$k_1$ ( $h^{-1}$ )	Velocidad de reacción en el primer reactor
$T_{ri1}$ (R)	Temperatura a la entrada del primer reactor	$k_2$ ( $h^{-1}$ )	Velocidad de reacción en el segundo reactor
$C_{ro1}$ ( $ft^3 / h$ )	Concentración a la salida del primer reactor	$I_{Tr1}$ (R.h)	Termino integral del PI del primer reactor
$T_{ro1}$ (R)	Temperatura a la salida del primer reactor	$I_{Tp}$ (R.h)	Termino integral del PI del segundo reactor
$C_p$ ( $ft^3 / h$ )	Composición del producto final	$\mu$ ( $R^2 .h$ )	Objetivo de la controlabilidad (Error Cuadratico Integral)
$T_p$ (R)	temperatura del producto final		
$T_{ro1}^*$ (R)	Set-Point temperatura salida primer reactor		
$T_p^*$ (R)	Set-Point temperatura del producto final		

**Fuente:** Schweiger C., Floudas C.(1997)

**Tabla 3.4. Variables de diseño**

Notación	Descripción
$T_{ro1}^*$ (R)	Set-Point temperatura a la salida del primer reactor
$T_p^*$ (R)	Set-Point temperatura del producto final
$V_{r1}$ (ft) <sup>3</sup>	Volumen del primer reactor
$V_{r2}$ (ft) <sup>3</sup>	Volumen del segundo reactor
$A_1$ (ft) <sup>2</sup>	Área de intercambio de calor del primer reactor
$A_2$ (ft) <sup>2</sup>	Área de intercambio de calor del segundo reactor
$D_{r1}$ (ft)	Diámetro del primer reactor
$D_{r2}$ (ft)	Diámetro del segundo reactor
$H_{r1}$ (ft)	Altura del primer reactor
$H_{r2}$ (ft)	Altura del segundo reactor
$V_{j1}$ (ft) <sup>3</sup>	Volumen de la chaqueta de enfriamiento del primer reactor
$V_{j2}$ (ft) <sup>3</sup>	Volumen de la chaqueta de enfriamiento del segundo reactor
$F_r$ (ft <sup>3</sup> / h)	Flujo de reacción
$F_b$ (ft <sup>3</sup> / h)	Flujo de recirculación
$F_p$ (ft <sup>3</sup> / h)	Flujo del producto final
$F_{jn1}$ (ft <sup>3</sup> / h)	Flujo nominal de la chaqueta de enfriamiento del reactor 1
$F_{jn2}$ (ft <sup>3</sup> / h)	Flujo nominal de la chaqueta de enfriamiento del reactor 2
$k_{j1}$ (ft <sup>3</sup> / (h.R))	Ganancia del PI del primer reactor
$k_{j2}$ (ft <sup>3</sup> / (h.R))	Ganancia del PI del segundo reactor
$\tau_{j1}$ (h)	Tiempo de integración del PI del primer reactor
$\tau_{j2}$ (h)	Tiempo de integración del PI del segundo reactor

Fuente: Schweiger C., Floudas C.(1997)

Al ser el diseño integrado, un problema de optimización, los objetivos se traducen en:

- Minimizar el costo total, que es el objetivo de diseño de procesos y simultáneamente,
- Minimizar el ISE, que es el objetivo de la controlabilidad dinámica.

De acuerdo con Schweiger y Floudas (1997), se asumen las siguientes expresiones y las restricciones para el proceso establecido, primero la función objetivo se relaciona con los costos de diseño y utilidades a través de la siguiente expresión matemática, en donde los costos de diseño están relacionados con las dimensiones de los equipos y el de utilidades con los flujos nominales del refrigerante, esto es válido para un sistema conformado por dos reactores mezcla completa, en donde ocurra una reacción exotérmica irreversible de primer orden y exista un control de esta por medio de chaquetas de enfriamiento, siendo este el caso estudiado en la presente investigación. Las siguientes expresiones son de uso estándar para el diseño de reactores mezcla completa (Schweiger y Floudas (1997)).

$$\text{cost}_{cap} = 1916,9.D_{r1}^{1,066}.(H_{r1})^{0,802} + 1916,9.D_{r2}^{1,066}.(H_{r2})^{0,802} \quad (32)$$

$$\text{cost}_{util} = 32,77.(F_{jn1} + F_{jn2}) \quad (33)$$

$$\text{cost}_{total} = \text{cost}_{cap} + 4_{yr}.\text{cost}_{util} \quad (34)$$

$$\text{cost}_{total} = 1916,9.D_{r1}^{1,066}.(H_{r1})^{0,802} + 1916,9.D_{r2}^{1,066}.(H_{r2})^{0,802} + 131,8.(F_{jn1} + F_{jn2}) \quad (35)$$

La expresión matemática de la función objetivo que relaciona la controlabilidad del sistema es:

$$\frac{d\mu}{dt} = (T_{ro1} - T_{ro1}^*)^2 + (T_p - T_p^*)^2 \quad (36)$$

Donde  $\mu$  es el error cuadrático integral, que no es más que la integral de la diferencia entre la temperatura de salida en cada reactor y sus respectivos set point elevada al cuadrado.

De acuerdo con Schweiger y Floudas (1997), entre las restricciones del sistema se tiene la composición de A en la corriente final :

$$C_p \leq 0,025 \quad (37)$$

La restricción de la controlabilidad del sistema:

$$\mu \leq \varepsilon \quad (38)$$

### **Estructura del proceso con su respectiva modelación matemática**

#### **Modelación matemática**

A continuación se muestra la modelación matemática tanto para la configuración de los reactores en serie como en paralelo (el cálculo típico del balance de materia y energía se presenta en los anexos M y N).



Usando la Figura 3.1 se obtiene el modelo dinámico para la optimización del sistema de los reactores en serie y se representa de la siguiente manera:

Balance de masa en F:

$$F_f \cdot \rho_f + F_b \cdot \rho_b = F_r \cdot \rho_r \quad (39)$$

Balance de masa en P:

$$F_r \cdot \rho_r + F_p \cdot \rho_p = F_b \cdot \rho_b \quad (40)$$

Balance de molar de A en F:

$$F_f \cdot C_f + F_b \cdot C_p = F_r \cdot C_{r1} \quad (41)$$

Balance energético en F:

$$F_f \cdot h_f \cdot \rho_f + F_b \cdot h_b \cdot \rho_b = F_r \cdot h_{r1} \cdot \rho_r \Rightarrow F_f \cdot h_f + F_b \cdot h_b = F_r \cdot h_{r1} \quad (42)$$

Balance molar de A en el reactor 1

$$\frac{d(C_{r01} \cdot V_{r1})}{dt} = F_r \cdot C_{r1} - F_r \cdot C_{r01} - r_{r1} \cdot V_{r1} \quad (43)$$

Balance molar de A en el reactor 2

$$\frac{d(C_{r2} \cdot V_{r2})}{dt} = F_r \cdot C_{r01} - F_r \cdot C_{r2} - r_{A2} \cdot V_{r2} \quad (44)$$

Balance de energía en el reactor 1

$$\frac{dU}{dt} = F_r \cdot \rho_r \cdot h_{r1} - F_r \cdot \rho_r \cdot h_{r01} - \Delta H \cdot r_A \cdot V_{r1} - Q \quad (45)$$

Balance de energía en el reactor 2

$$\frac{dU}{dt} = F_r \cdot \rho_r \cdot h_{r01} - F_r \cdot \rho_r \cdot h_{r02} - \Delta H \cdot r_A \cdot V_{r2} - Q \quad (46)$$

Balance de energía en la chaqueta 1

$$\frac{dU}{dt} = F_{j1} \cdot \rho_{j1} \cdot h_{j1} - F_{j1} \cdot \rho_{j1} \cdot h_{j01} + Q \quad (47)$$

Balance de energía en la chaqueta 2

$$\frac{dU}{dt} = F_{j2} \cdot \rho_j \cdot h_{ji2} - F_{j2} \cdot \rho_j \cdot h_{jo2} + Q \quad (48)$$

En donde F representa flujo en ft<sup>3</sup>/hr, las C representan concentraciones molares en lbmol A/ft<sup>3</sup>, las T representan temperaturas en R, V representa volumen en ft<sup>3</sup>, las  $k$  son constantes cinéticas de velocidad de reacción en hr<sup>-1</sup>, A es el área de transferencia de calor efectiva en ft<sup>2</sup>.

Similarmente, usando la figura 4.1.2 se obtiene el modelo dinámico para la optimización del sistema de los reactores en paralelo y se representa de la siguiente manera:

Balance de masa en p:

$$F_{r1} \cdot \rho_{r1} + F_{r2} \cdot \rho_{r2} = F_p \cdot \rho_p \quad (49)$$

Balance de masa en f:

$$F_{r1} \cdot \rho_{r1} + F_{r2} \cdot \rho_{r2} = F_f \cdot \rho_f \quad (50)$$

Balance de molar de A en p:

$$F_{r1} \cdot C_{ro1} + F_{r2} \cdot C_{ro1} = F_p \cdot C_p \quad (51)$$

Balance energético en p:

$$F_{r1} \cdot h_{ro1} \cdot \rho_{ro1} + F_{r2} \cdot h_{ro2} \cdot \rho_{ro2} = F_p \cdot h_p \cdot \rho_p \Rightarrow F_{r1} \cdot h_{ro1} + F_{r2} \cdot h_{ro2} = F_p \cdot h_p \quad (52)$$

Balance molar de A en el reactor 1

$$\frac{d(C_{ro1} \cdot V_{r1})}{dt} = F_{r1} \cdot C_f - F_{r1} \cdot C_{ro1} - r_{r1} \cdot V_{r1} \quad (53)$$

Balance molar de A en el reactor 2

$$\frac{d(C_p \cdot V_{r2})}{dt} = F_{r2} \cdot C_f - F_{r2} \cdot C_{ro2} - r_{A2} \cdot V_{r2} \quad (54)$$

Balance de energía en el reactor 1

$$\frac{dU}{dt} = F_{r1} \cdot \rho_f \cdot h_f - F_{r1} \cdot \rho_{ro1} \cdot h_{ro1} - \Delta H \cdot r_A \cdot V_{r1} - Q \quad (55)$$

Balance de energía en el reactor 2

$$\frac{dU}{dt} = F_{r2} \cdot \rho_f \cdot h_f - F_{r2} \cdot \rho_{ro2} \cdot h_{ro2} - \Delta H \cdot r_A \cdot V_{r2} - Q \quad (56)$$

Balance de energía en la chaqueta 1

$$\frac{dU}{dt} = F_{j1} \cdot \rho_{j1} \cdot h_{ji1} - F_{j1} \cdot \rho_{j1} \cdot h_{jo1} + Q \quad (57)$$

Balance de energía en la chaqueta 2

$$\frac{dU}{dt} = F_{j2} \cdot \rho_j \cdot h_{ji2} - F_{j2} \cdot \rho_j \cdot h_{jo2} + Q \quad (58)$$

De acuerdo con Schweiger y Floudas (1997), se asumen las siguientes expresiones y las restricciones para el proceso establecido:

La función objetivo se relaciona con los costos de diseño a través de las ecuaciones numero (32), (33), (34) y (35) antes descritas:

La expresión matemática de la función objetivo que relaciona la controlabilidad del sistema para la disposición en serie es:

$$\frac{d\mu}{dt} = (T_{ro1} - T_{ro1}^*)^2 + (T_p - T_p^*)^2 \quad (59)$$

La expresión matemática de la función objetivo que relaciona la controlabilidad del sistema para la disposición en paralelo es:

$$\frac{d\mu}{dt} = (T_{ro1} - T_{ro1}^*)^2 + (T_{ro2} - T_{ro2}^*)^2 \quad (60)$$

La expresión matemática para el volumen del reactor y el área de transferencia de calor es la siguiente:

$$V_{r1} = \frac{\pi}{4} \cdot D_{r1}^2 \cdot H_{r1} \quad (61)$$

$$V_{r2} = \frac{\pi}{4} \cdot D_{r2}^2 \cdot H_{r2} \quad (62)$$

$$A_1 = \pi \cdot D_{r1} \cdot H_{r1} \quad (63)$$

$$A_2 = \pi \cdot D_{r2} \cdot H_{r2} \quad (64)$$

La expresión matemática para el volumen de la chaqueta de enfriamiento se representa a través de:

$$V_{j1} = A_1 \cdot \frac{1}{3} \cdot ft \quad (65)$$

$$V_{j2} = A_2 \cdot \frac{1}{3} \cdot ft \quad (66)$$

Se establece la ley de control correspondiente al controlador y el conjunto de perturbaciones, al igual que el horizonte de tiempo y la restricción sobre la composición del producto. El modelo matemático que representa la ley de control (Proporcional-Integral) del sistema en el arreglo en serie es:

$$\frac{dI_{Tro1}}{dt} = T_{ro1} - T_{ro1}^* = 0 \quad (67)$$

$$\frac{dI_{Tp}}{dt} = T_p - T_p^* = 0 \quad (68)$$

$$F_{j1} = F_{jn1} + k_{j1} \cdot (T_{ro1} - T_{ro1}^*) + \frac{k_{j1}}{\tau_{j1}} \cdot I_{Tro1} \quad (69)$$

$$F_{j2} = F_{jn2} + k_{j2} \cdot (T_p - T_p^*) + \frac{k_{j2}}{\tau_{j1}} \cdot I_{Tp} \quad (70)$$

Para el arreglo en paralelo el modelo matemático que representa la ley de control (Proporcional-Integral) del sistema es:

$$\frac{dI_{Tro1}}{dt} = T_{ro1} - T_{ro1}^* = 0 \quad (71)$$

$$\frac{dI_{Tp}}{dt} = T_{ro2} - T_{ro2}^* = 0 \quad (72)$$

$$F_{j1} = F_{jn1} + k_{j1} \cdot (T_{ro1} - T_{ro1}^*) + \frac{k_{j1}}{\tau_{j1}} \cdot I_{Tro1} \quad (73)$$

$$F_{j2} = F_{jn2} + k_{j2} \cdot (T_{ro2} - T_{ro2}^*) + \frac{k_{j2}}{\tau_{j1}} \cdot I_{Tro2} \quad (74)$$

Entre las restricciones del sistema se tiene la restricción del compuesto A en la corriente final:

$$C_p(t_f) \leq 0.025 \quad (75)$$

y la restricción de la controlabilidad del sistema:

$$\mu \leq \varepsilon \quad (76)$$

### **Estructura algorítmica para la obtención de los valores óptimos de diseño, configuración de los equipos y parámetros del controlador**

Se requiere encontrar los puntos de partida del algoritmo, que no son más que aquellos valores que delimitan la solución de un problema. Los métodos globales exhiben independencia de la naturaleza del espacio de soluciones y poseen la capacidad de atravesar un espacio de búsqueda con múltiples mínimos o máximos locales para alcanzar una solución en las inmediaciones de la región que contiene a la solución óptima. Esto representa una ventaja al prescindir de puntos de partida. Pero los mismos presentan limitaciones en cuanto a su estancamiento en óptimos locales y la imposibilidad de escapar de ellos.

Lo importante es garantizar que dicho algoritmo converja, de lo contrario el algoritmo seguiría iterando de forma infinita sin arrojar ningún resultado.

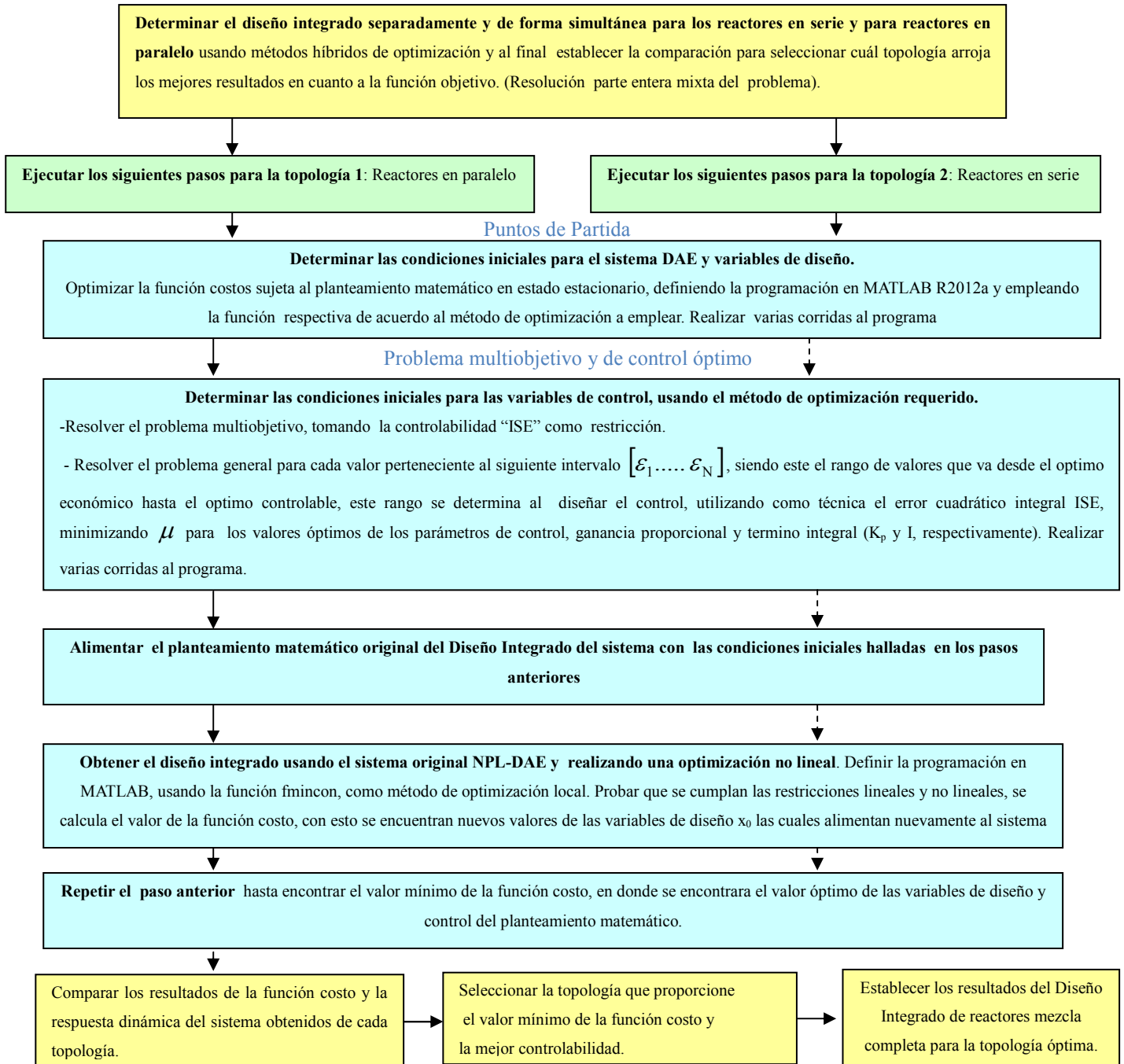
Por su parte, los métodos de optimización local requieren puntos de partida debido a su incapacidad de pasearse por todo el espacio de búsqueda. Por ello ocurre que si la optimización local parte de puntos muy cercanos al óptimo global, el algoritmo de solución convergerá rápidamente y los resultados serían óptimos. Y de manera recíproca el método global, al reducir el espacio de búsqueda, acercándose al óptimo global, puede servirse posteriormente de los métodos locales para alcanzar convergencia.

Este hallazgo ya fue presentado en 2009 por Pirona, al establecer en su trabajo con reactores mezcla completa, que al hibridar técnicas de optimización local del tipo determinístico con métodos globales estocásticos, se compensan las debilidades de una técnica con las habilidades de la otra. (Ver métodos híbridos de optimización, Capítulo II).

En función a esto, se asume en el presente trabajo que para obtener la mejor aproximación a la solución óptima al problema de diseño integrado de reactores mezcla completa, según el sistema descrito, es necesario que para inicializar el algoritmo, se empleen los puntos de partida obtenidos de métodos de optimización global. La estructura algorítmica en forma de diagrama de bloques, se observa en la Figura 3.3.

Para formar la estructura algorítmica propiamente dicha, a partir de los pasos descritos en el diagrama de bloques, se define el conjunto de expresiones matemáticas que se debe usar en los diferentes planteamientos matemáticos para ambas topologías.

Problema entero-mixto





**Figura 3.3. Diagrama de bloques de la estructura algorítmica para la resolución del problema de diseño integrado de dos reactores mezcla completa.**

**Programacion no lineal (NLP) del planteamiento original. Topología de reactores en serie**

Minimizar

$$\text{cost}_{total} = 19169.D_{r1}^{1,066} \cdot (H_{r1})^{0,802} + 19169.D_{r2}^{1,066} \cdot (H_{r2})^{0,802} + 1318.(F_{jn1} + F_{jn2}) = \text{ObjM} \quad (77)$$

Sujeto a:

$$F_r = F_f + F_b \quad (78)$$

$$F_r = F_p + F_b \quad (79)$$

$$F_f \cdot C_f + F_b \cdot C_p = F_r \cdot C_{ri1} \quad (80)$$

$$T_f \cdot F_f + T_p \cdot F_b = T_{ri1} \cdot F_r \quad (81)$$

$$F_r \cdot C_{ri1} - F_r \cdot C_{ro1} - k_1 \cdot C_{ro1} \cdot V_{r1} = 0 \quad (82)$$

$$F_r \cdot C_{ro1} - F_r \cdot C_p - k_2 \cdot C_p \cdot V_{r2} = 0 \quad (83)$$

$$F_r \cdot T_{ri1} - F_r \cdot T_{ro1} - \frac{U \cdot A_1}{\rho \cdot Cp} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_1 \cdot C_{ro1} \cdot V_{r1} = 0 \quad (84)$$

$$F_r \cdot T_{ro1} - F_r \cdot T_p - \frac{U \cdot A_2}{\rho \cdot Cp} \cdot (T_p - T_{jo2}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_2 \cdot C_p \cdot V_{r2} = 0 \quad (85)$$

$$F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1}) = 0 \quad (86)$$

$$F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_p - T_{jo2}) = 0 \quad (87)$$

$$k_1 = k_0 \cdot e^{\frac{-E}{R \cdot T_{ro1}}} \quad (88)$$

$$k_2 = k_0 \cdot e^{\frac{-E}{R \cdot T_p}} \quad (89)$$

$$V_{r1} = \frac{\pi}{4} \cdot D_{r1}^2 \cdot H_{r1} \quad (90)$$

$$V_{r2} = \frac{\pi}{4} \cdot D_{r2}^2 \cdot H_{r2} \quad (91)$$

$$A_1 = \pi \cdot D_{r1} \cdot H_{r1} \quad (92)$$

$$A_2 = \pi \cdot D_{r2} \cdot H_{r2} \quad (93)$$

$$V_{j1} = A_1 \cdot \frac{1}{3} f \quad (94)$$

$$V_{j2} = A_2 \cdot \frac{1}{3} f \quad (95)$$

Restricción (96)

$$C_p \leq 0,025$$

### Programacion no lineal (NLP) del planteamiento original. Topología de reactores en paralelo

Minimizar

*ObjM*

Sujeto a:

$$F_{r1} = F_p - F_{r2} \quad (97)$$

$$F_{r1} = F_f - F_{r2} \quad (98)$$

$$F_{r1} \cdot C_{ro1} + F_{r2} \cdot C_{ro2} = F_p \cdot C_p \quad (99)$$

$$T_{ro1} \cdot F_{r1} + T_{ro2} \cdot F_{r2} = T_p \cdot F_p \quad (100)$$

$$F_{r1} \cdot C_f - F_{r1} \cdot C_{ro1} - k_1 \cdot C_{ro1} \cdot V_{r1} = 0 \quad (101)$$

$$F_{r2} \cdot C_f - F_{r2} \cdot C_{ro2} - k_2 \cdot C_{ro2} \cdot V_{r2} = 0 \quad (102)$$

$$F_{r1} \cdot T_f - F_{r1} \cdot T_{ro1} - \frac{U \cdot A_1}{\rho \cdot Cp} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_1 \cdot C_{ro1} \cdot V_{r1} = 0 \quad (103)$$

$$F_{r2} \cdot T_f - F_{r2} \cdot T_{ro2} - \frac{U \cdot A_2}{\rho \cdot Cp} \cdot (T_{ro2} - T_{jo2}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_2 \cdot C_{ro2} \cdot V_{r2} = 0 \quad (104)$$

$$F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1}) = 0 \quad (105)$$

$$F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_{ro2} - T_{jo2}) = 0 \quad (106)$$

**Programacion no lineal con ecuaciones algebraico diferenciales NLP-DAE.  
Modelado matematico de la topología de reactores en serie**

Minimizar

*ObjM*

$$F_{j1} = F_{jn1} + k_{j1} \cdot (T_{ro1} - T_{ro1}^*) + \frac{k_{j1}}{\tau_{j1}} \cdot I_{tro1} \quad (107)$$

$$F_{j2} = F_{jn2} + k_{j2} \cdot (T_p - T_p^*) + \frac{k_{j2}}{\tau_{j1}} \cdot I_{tp} \quad (108)$$

$$V_{r1} \cdot \frac{dC_{ro1}}{dt} = F_r \cdot C_{ri1} - F_r \cdot C_{ro1} - k_1 \cdot C_{ro1} \cdot V_{r1} \quad (109)$$

$$(110)$$

$$V_{r2} \cdot \frac{dC_p}{dt} = F_r \cdot C_{ro1} - F_r \cdot C_p - k_2 \cdot C_p \cdot V_{r2}$$

$$V_{r1} \cdot \frac{dT_{ro1}}{dt} = F_r \cdot T_{ri1} - F_r \cdot T_{ro1} - \frac{U \cdot A_1}{\rho \cdot C_p} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot C_p} k_1 \cdot C_{ro1} \cdot V_{r1} \quad (111)$$

$$V_{r2} \cdot \frac{dT_p}{dt} = F_r \cdot T_{ro1} - F_r \cdot T_p - \frac{U \cdot A_2}{\rho \cdot C_p} \cdot (T_p - T_{jo2}) - \frac{\Delta H}{\rho \cdot C_p} k_2 \cdot C_p \cdot V_{r2} \quad (112)$$

$$V_{j1} \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1}) \quad (113)$$

$$V_{j2} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_p - T_{jo2}) \quad (114)$$

$$\frac{d\mu}{dt} = (T_{ro1} - T_{ro1}^*)^2 + (T_p - T_p^*)^2 \quad (115)$$

$$\frac{dI_{T_{ro1}}}{dt} = T_{ro1} - T_{ro1}^* \quad (116)$$

$$\frac{dI_{T_p}}{dt} = T_p - T_p^* \quad (117)$$

Tomando como restricciones:

$$\mu \leq \varepsilon \quad (118)$$

**Programacion no lineal con ecuaciones algebraico diferenciales NLP-DAE.**

**Modelado matemático de la topología de reactores en paralelo**

Minimizar

*ObjM*

Sujeto a:

$$V_{r1} \cdot \frac{dC_{ro1}}{dt} = F_{r1} \cdot C_f - F_{r1} \cdot C_{ro1} - k_1 \cdot C_{ro1} \cdot V_{r1} \quad (119)$$

$$V_{r2} \cdot \frac{dC_{ro2}}{dt} = F_{r2} \cdot C_f - F_{r2} \cdot C_{ro2} - k_2 \cdot C_{ro2} \cdot V_{r2} \quad (120)$$

$$V_{r1} \cdot \frac{dT_{ro1}}{dt} = F_{r1} \cdot T_f - F_{r1} \cdot T_{ro1} - \frac{U \cdot A_1}{\rho \cdot Cp} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_1 \cdot C_{ro1} \cdot V_{r1} \quad (121)$$

$$V_{r2} \cdot \frac{dT_{ro2}}{dt} = F_{r2} \cdot T_f - F_{r2} \cdot T_{ro2} - \frac{U \cdot A_2}{\rho \cdot Cp} \cdot (T_{ro2} - T_{jo2}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_2 \cdot C_{ro2} \cdot V_{r2} \quad (122)$$

$$V_{j1} \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1}) \quad (123)$$

$$V_{j2} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_{ro2} - T_{jo2}) \quad (124)$$

En el trabajo de Segura (2007), se presentan las restricciones adicionales al problema. En su investigación solamente se resuelve el problema del diseño integrado mediante métodos determinísticos (función fmincon). Las restricciones son del tipo:

Lógicas: no existencia de flujos o dimensiones negativas, relación entre la altura y el diámetro de los equipos.

Condiciones de proceso: relación entre el flujo nominal y el flujo dinámico máximo del refrigerante (variable manipulada), norma “PDVSA N° K-332 Control Valve”, la cual establece que el diseño final de una válvula de control debe estar sujeto a la apertura a la cual tiene que trabajar para un determinado flujo nominal en la variable manipulada [17 – 90] %.

$$F_{jn1} \geq 0,3F_{j1 \max} \quad (125)$$

$$F_{jn2} \geq 0,3F_{j2 \max} \quad (126)$$

Condiciones de control: la perturbación es una entrada tipo escalón con aumento en el 10 % de temperatura de alimentación al sistema. Se selecciona este tipo de perturbación porque es la que más se asemeja al comportamiento de la temperatura en el sistema reactivo. Se emplea el índice de disminución de un cuarto, expresado matemáticamente como:

$$\alpha.(T_{ro1 \max} - T_{ro1}^*) \leq (T_{ro1}^* - T_{ro1 \min}) \quad (127)$$

$$\alpha.(T_{ro2 \max} - T_{ro2}^*) \leq (T_{ro2}^* - T_{ro2 \min}) \quad (128)$$

Donde  $\alpha$  representa el índice o razón de disminución de la cresta inicial (máxima) con respecto a la cresta mínima de cada respuesta. Se toma  $\alpha = 0,3$  (Segura (2007)).

Como se nombró anteriormente, el problema de diseño integrado de un sistema como los planteados presentan diferentes tipos de variables y restricciones las cuales se encuentran enlistadas en la Tabla 3.1, siguiendo esta clasificación se puede representar simplificadaamente el problema de diseño integrado para el sistema.

**Programación no lineal con ecuaciones algebraico diferenciales NLP-DAE modelado matemático para la estructura de control reactores en serie**

Minimizar

$\mu$

Sujeto a:

Ecuación (78) a la ecuación (81)

Ecuación (107) a la ecuación (117)

Ecuación (88) a la ecuación (95)

**Programación no lineal con ecuaciones algebraico diferenciales NLP-DAE.  
Modelado matemático para la estructura de control reactores en paralelo**

Minimizar

$\mu$

Sujeto a:

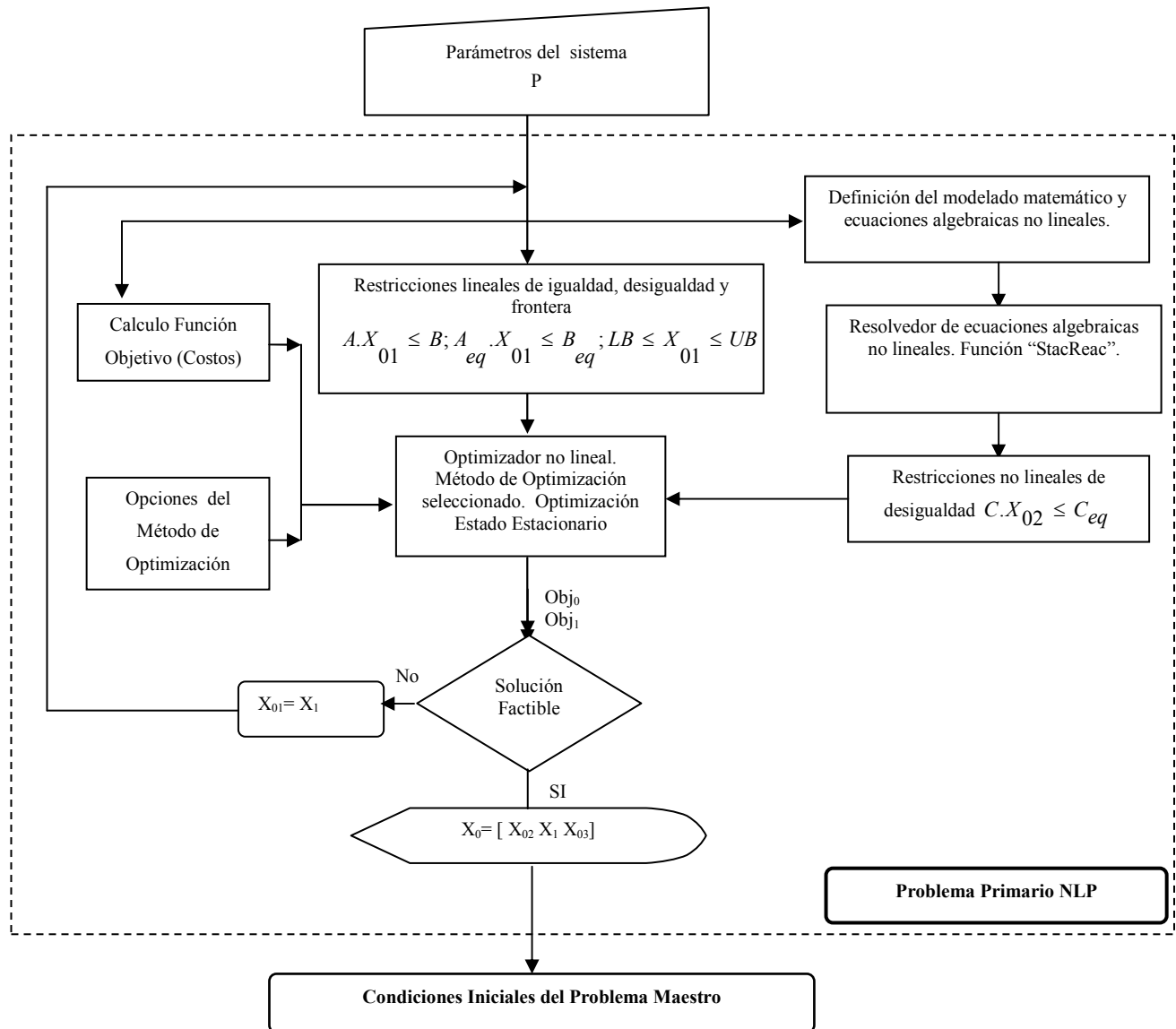
Ecuacion (97) a la ecuación (100)

Ecuacion (119) a la ecuación (124)

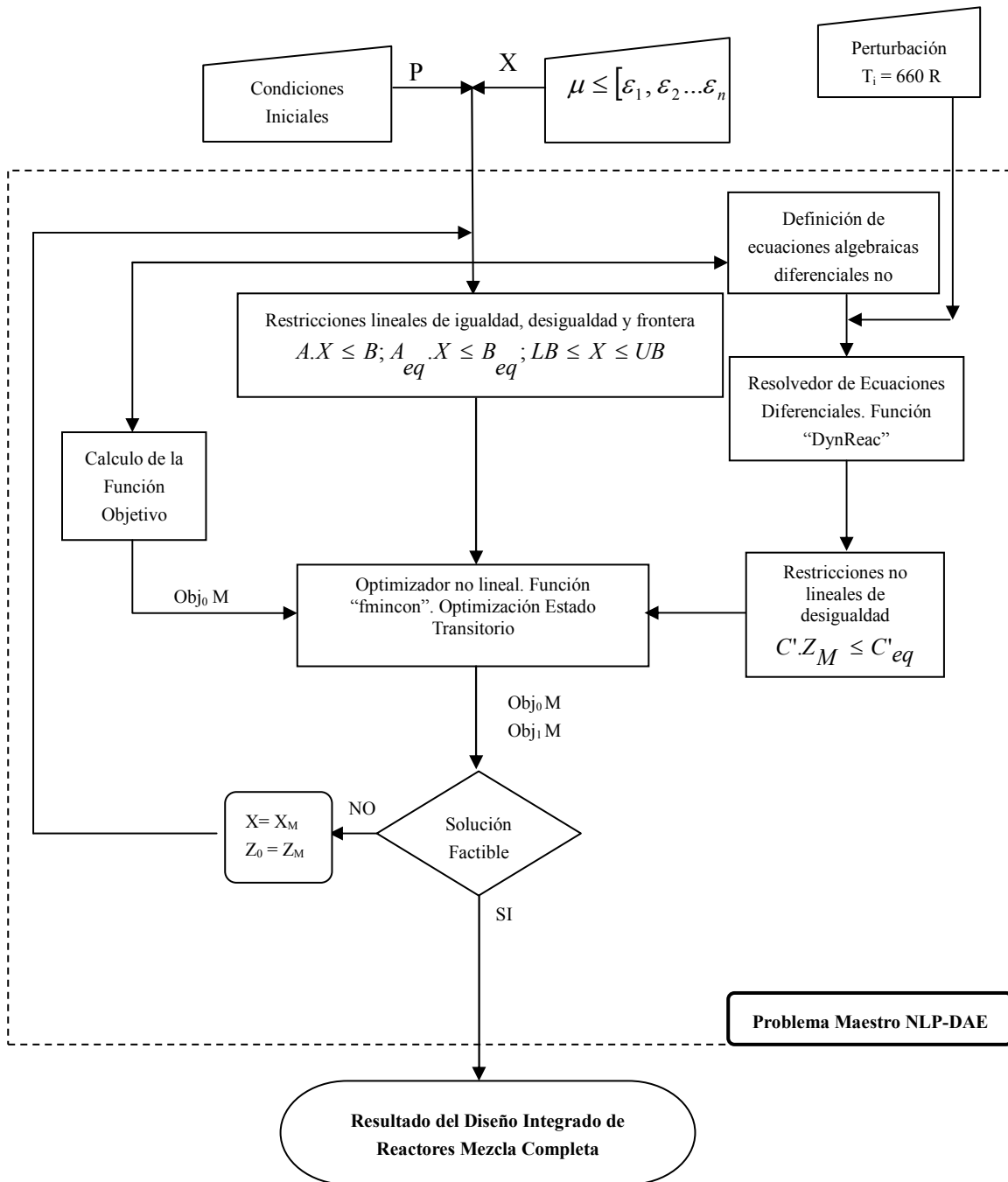
Ecuacion (88) a la ecuación (95)

En la siguiente figura se presenta la estructura algorítmica para la resolución del problema NLP-DAE del diseño integrado de reactores mezcla completa.





**Figura 3.4. Algoritmo para resolver el problema NLP-DAE del diseño integrado de reactores mezcla completa. Módulo I: optimización en estado estacionario.**



**Figura 3.5.** Algoritmo para resolver el problema NLP-DAE del diseño integrado de reactores mezcla completa. Módulo II: optimización en estado transitorio.

La definición de las variables utilizadas en el algoritmo se muestra a continuación:

### **Parámetros constantes para el problema primario y maestro**

$$P = [k_0, E/R, C_f, F_f, \Delta H, \rho, C_p, U, T_{j1}, T_{j2}, \rho_j, C_j]$$

### **Optimización no lineal estado estacionario**

VARIABLES DE DISEÑO PARA LAS RESTRICCIONES DE DESIGUALDAD NO LINEALES (PROBLEMA PRIMARIO)

$$X_{02} = [T_{r01}, C_p, T_p]$$

### **Resultados optimización no lineal (estado estacionario)**

$$X_0 = [H_{r1}, H_{r2}, F_b, D_1, D_2, F_{jn1}, F_{jn2}, F_r, C_{r1}, T_{r1}, C_{r01}, T_{r01}, T_{j01}, C_p, T_p, T_{j02}]$$

### **Optimización no lineal estado transitorio**

VARIABLES DE DISEÑO (CONDICIONES INICIALES PROBLEMA MAESTRO)

$$X = [H_{r1}, H_{r2}, F_b, D_1, D_2, F_{jn1}, F_{jn2}, k_{j1}, k_{j2}, \tau_{j1}, \tau_{j2}]$$

### **VARIABLES DE DISEÑO (CONDICIONES INICIALES DEL SISTEMA DAE)**

$$Z_0 = [C_{r01}, T_{r01}, T_{j01}, C_p, T_p, T_{j02}, I_{tro1}, I_{tro2}, \mu]$$

### **Resultados del diseño integrado de reactores mezcla completa**

[ $H_{r1}$ ,  $H_{r2}$ ,  $F_b$ ,  $D_1$ ,  $D_2$ ,  $F_{jn1}$ ,  $F_{jn2}$ ,  $k_{j1}$ ,  $k_{j2}$ ,  $\tau_{j1}$ ,  $\tau_{j2}$ ,  $C_{ro1}$ ,  $T_{ro1}$ ,  $T_{jo1}$ ,  $C_p$ ,  $T_p$ ,  $T_{jo2}$ ,  $I_{tro1}$ ,  
 $I_{tro2}$ ,  $\mu$  ]

# **CAPÍTULO IV**

## **DISCUSIÓN DE RESULTADOS**

## **CAPÍTULO IV. PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS**

A lo largo de este capítulo se presentan los detalles de resolución del problema, ya que por ser de naturaleza matemática altamente no lineal, se requiere la revisión exhaustiva y cuantitativa, que al final determinan cual método desarrollado se ajusta más al sistema empleado, cumpliendo con el último paso de la metodología experimental y obteniendo el objetivo deseado.

### **4.1 ALGORITMOS GENÉTICOS**

#### **Implementación de la estructura algorítmica en MATLAB R2012a**

Para obtener las condiciones iniciales del problema matemático del diseño integrado se optimiza en estado estacionario por separado y en forma secuencial primeramente las variables del diseño de los reactores, es decir, aquellas que influyen en el costo capital por estar directamente relacionadas a las dimensiones del reactor, y segundo se optimizan también en el estado estacionario las variables de control que se relacionan con los parámetros del controlador.

Aplicando los tres primeros pasos del diagrama de bloques, se realiza una optimización de la función costos sujeta al planteamiento matemático en estado estacionario para cada topología por separado definiendo la programación en MATLAB R2012a , usando algoritmos genéticos.

La definición de la programación en MATLAB R2012a se presenta en el Anexo K: Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en paralelo y en el Anexo L: Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie). Estos programas manejan a su vez varias funciones que se definen en MATLAB R2012a en “*scripts*” separados para facilitar la tarea de programación, longitud de los programas y tiempo de corrida de los mismos. Estas funciones que son llamadas por el programa OptEstatPar.m (Anexo K) son: la función “Costo” (Anexo A), “DynReacPar” (Anexo B) y “StatReacPar” (Anexo D), que calcula la función costo, realiza la simulación del sistema de reactores en estado dinámico, y simula el sistema de reactores en estado estacionario para la topología paralelo, respectivamente. Similarmente se definen las funciones que son llamadas por el programa OptEstatSer.m (Anexo L): la función “Costo” (Anexo A), “DynReacSer” (Anexo C) y “StatReacSer” (Anexo E), que calcula la función costo, hace la simulación del sistema de reactores en estado dinámico, y simula el sistema de reactores en estado estacionario para la topología serie, respectivamente.

El establecimiento de las opciones del algoritmo genético se realiza ensayando distintas opciones para los parámetros que se muestran en la Tabla 2.3 del Capítulo II. Las opciones y los tipos de operadores genéticos afectan significativamente el desempeño de un algoritmo genético. En general, el algoritmo genético es extremadamente sensible a la introducción o combinación de las técnicas empleadas. La selección de las opciones y los tipos de operadores es empírica, sin embargo está relacionada con el tipo de problema.

Para el sistema estudiado se observó en diferentes corridas que al usar las funciones por defecto que establece MATLAB R2012a para cada una de las

opciones (Tabla 2.3), se producían los resultados esperados en lapsos de tiempo muy cortos, solamente se optimizó el parámetro de probabilidad de mutación para mejorar los resultados en cuanto a los costos totales.

Las opciones de mutación especifican cómo el algoritmo genético efectúa los pequeños cambios aleatorios en los individuos de una población para crear los descendientes. Un algoritmo genético trata de explorar las regiones más prometedoras de un enorme espacio de posibilidades. Al encontrar una zona con pesos altos, esta se explora más a fondo. Pero hay que evitar que el algoritmo se estanque en una determinada zona, produciendo multitud de cadenas muy parecidas. El aumento de variedad se consigue con las mutaciones. La mutación provee diversidad genética y permite al algoritmo genético buscar en un espacio más amplio.

Se varió la probabilidad de mutación desde el valor más bajo aceptado (0,1) hasta hallar valor (0,3) que produjo mejores resultados en cuanto a costos totales, como se verá más adelante.

Debido a la naturaleza estocástica de los algoritmo genético, es necesario realizar varias corridas.

En las Tablas 4.1.1 y 4.1.2 se muestran los resultados obtenidos por diferentes corridas del programa del Anexo K y L, respectivamente y para los dos mejores valores para la probabilidad de mutación del sistema.



**TABLA 4.1.1 Costo total para varias corridas del programa OptEStatPar.m y probabilidad de mutación 0,1 y 0,3, topología de reactores en paralelo**

<b>Mutación 0.1</b>		<b>Mutación 0.3</b>	
<b>Corrida número</b>	<b>Costo Total (\$)</b>	<b>Corrida número</b>	<b>Costo Total (\$)</b>
1	303.621	1	301.495
2	305.882	2	311.428
3	305.974	3	313.193
4	308.598	4	314.803
5	331.895	5	316.325
6	336.320	6	321.505
7	349.101	7	321.675
8	384.720	8	347.703
9	396.302	9	380.208
10	399.993	10	385.929
11	430.696	11	405.960
12	434.937	12	446.718
13	443.472	13	467.361
14	458.518	14	499.362
15	483.524	15	547.658
16	485.496	16	587.630
17	494.794	17	874.078
18	551.453	18	903.683
19	1.222.873	19	1.342.664
20	1.583.158	20	2.138.429

**TABLA 4.1.2. Reporte de resultados del costo total para varias corridas del programa OptEStatSer.m (Anexo L) en el estado estacionario y probabilidad de mutación 0,1 y 0,3, topología de reactores en serie**

<b>Mutación 0.1</b>		<b>Mutación 0.3</b>	
<b>Corrida número</b>	<b>Costo Total (\$)</b>	<b>Corrida número</b>	<b>Costo Total (\$)</b>
1	176.815	1	183.404
2	179.015	2	195.291
3	179.287	3	196.172
4	185.014	4	199.134
5	193.083	5	202.924
6	193.213	6	214.167
7	211.209	7	220.356
8	223.032	8	222.424
9	233.666	9	239.826
10	235.256	10	244.963
11	237.707	11	261.226
12	248.850	12	281.675
13	252.279	13	299.900
14	271.313	14	313.440
15	283.335	15	313.913
16	300.880	16	326.519
17	309.040	17	353.084
18	323.060	18	396.487
19	509.071	19	420.941
20	1.227.675	20	437.094

La razón por la cual es necesario realizar varias corridas a los programas que emplean algoritmos genéticos es porque son sistemas no lineales con comportamiento fuertemente ecológico, ellos combinan cambios aleatorios con procesos probabilísticos, son por lo tanto, estocásticos: difícilmente repiten un

resultado de un experimento para otro, por ello se producen resultados distintos cada vez que se corre el programa.

La población inicial suele ser generada mediante un proceso aleatorio que cubre todo el espacio del problema. Esto garantiza una amplia variedad en la reserva genética. Debido a que se genera aleatoriamente la población inicial al especificar el vector vacío en el campo InitialPopulation, el algoritmo siempre partirá de una población inicial diferente.

De las Tablas 4.1.1 y 4.1.2 se toma el mejor valor para los costos (valor mínimo), el cual se obtuvo con probabilidad de mutación de 0.3, que en este caso corresponde a la corrida número 1, para ambas topologías del sistema. En las siguientes tablas se muestran la totalidad de los resultados obtenidos en esa corrida, esto define las condiciones iniciales para el sistema DAE y puntos de partida para las variables de diseño.

**Tabla 4.1.3. Reporte de resultados del programa OptEstatPar\_AG.m (Anexo K) para el estado estacionario. Topología de reactores en paralelo.**

<b>Probabilidad de mutación 0.3</b>	
<b>Optimización en estado estacionario</b>	
Costo capital (\$)	294706,758
Costo utilidad (\$)	6787,97513
Costo total (\$)	301494,733
<b>Parámetros óptimos de diseño</b>	
Hr1 (ft)	7,91E+00
Hr2 (ft)	7,63E-01
Fb (ft <sup>3</sup> /h)	1,00E+00
Dr1 (ft)	2,36E+01
Dr2 (ft)	1,02E+00
Fjn1 (ft <sup>3</sup> /h)	3,93E+01
Fjn2 (ft <sup>3</sup> /h)	1,24E+01

**Tabla 4.1.4. Reporte de resultados del programa OptEStatSer\_AG.m (Anexo L) para el estado estacionario. Topología de reactores en serie. Probabilidad de mutación 0.3**

<b>Optimización en estado estacionario</b>	
Costo capital (\$)	177958,851
Costo utilidad (\$)	5444,7729
Costo total (\$)	183403,624
<b>Parámetros óptimos de diseño</b>	
Hr1 (ft)	4,36E+00
Hr2 (ft)	4,10E+00
Fb (ft <sup>3</sup> /h)	4,50E-01
Dr1 (ft)	1,29E+01
Dr2 (ft)	1,18E+01
Fjn1 (ft <sup>3</sup> /h)	3,68E+01
Fjn2 (ft <sup>3</sup> /h)	4,77E+00

**Determinación del intervalo de estudio para  $\varepsilon$  y de los puntos de partida de los parámetros del controlador del planteamiento matemático original**

Dado que la estructura algorítmica es descompuesta en tres partes (el problema entero mixto, la naturaleza multiobjetivo y el problema de control óptimo), es necesario para resolver la naturaleza multiobjetivo de la aplicación de la técnica de la restricción, la cual reduce el problema a uno de optimización de objetivo único.

Para ello, se convierte uno de los objetivos en restricción  $(obj_j \leq \varepsilon)$ , en este caso el error cuadrático integral pasa a formar parte de las restricciones no lineales del problema, lo que implica resolver el problema NLP-DAE a diferentes valores de

$\varepsilon$  que van desde el óptimo controlable hasta el óptimo económico, en este punto el problema multiobjetivo se convierte en un problema de programación no lineal con ecuaciones algebraico diferenciales NLP-DAE.

Para determinar el intervalo de estudio para  $\varepsilon$  en donde la controlabilidad tiene un impacto económico apreciable sobre el diseño, se emplea la estructura de control al diseño del proceso reactivo encontrado mediante la minimización de la función costo sujeto al planteamiento en estado estacionario, como se muestra en la programación no lineal (NLP) del planteamiento original topología de reactores en serie y paralelo.

Se diseña un controlador tipo PI (proporcional-integral), utilizando como técnica el ISE, minimizando  $\mu$  para encontrar los puntos de partidas de los parámetros del controlador del planteamiento matemático original, ganancia proporcional y término integral ( $K_p$  y  $\tau_i$ , respectivamente). El problema es alimentado con los valores obtenidos en la optimización en estado estacionario (Ver Tablas 4.1.3 y 4.1.4), las variables de control serán  $k_{j1}$ ,  $k_{j2}$ ,  $\tau_{j1}$ ,  $\tau_{j2}$  y  $\mu$ .

La definición de los programas en MATLAB R2012a con el uso de algoritmos genéticos son mostrados en los Anexos G (OnlycontrolPar\_AG.m) y H (OnlycontrolSer\_AG.m) para la optimización mediante algoritmos genéticos de las variables de control en estado estacionario para topología de reactores en paralelo y en serie, respectivamente. Estos programas hacen uso de la función Costo.m definida en el Anexo A. Los resultados de las corridas se muestran en las tablas siguientes.

**Tabla 4.1.5. Valores de  $\varepsilon$  (ypsilón) en varias corridas del programa OnlyControlPar\_AG.m del Anexo G, topología de reactores en paralelo.**

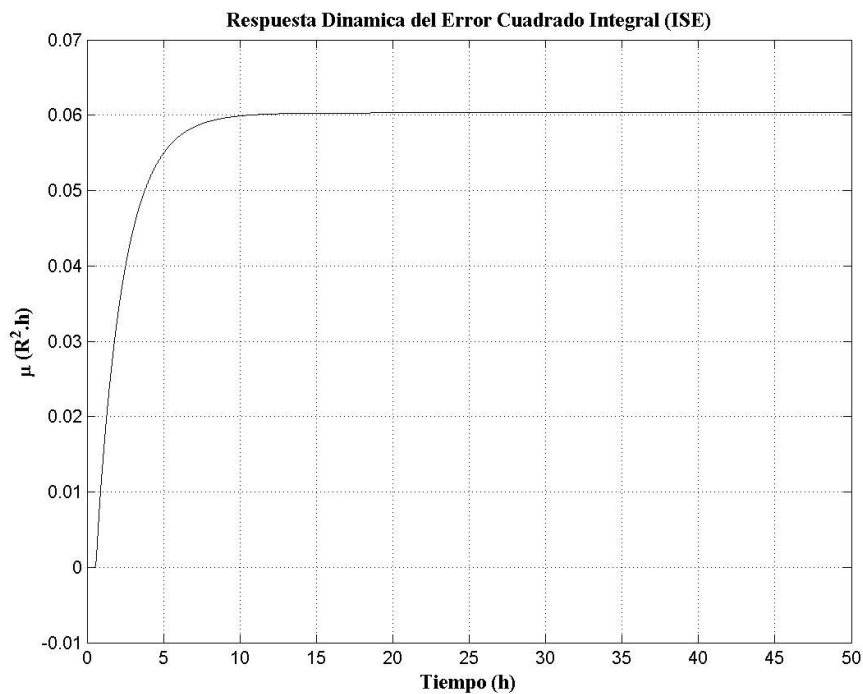
<b>Nº de corrida</b>	<b><math>\varepsilon</math> (ypsilón)</b>
1	0,0603
2	0,0770
3	0,1070
4	0,2025
5	1,2001

**Tabla 4.1.6. Valores de  $\varepsilon$  (ypsilón) en varias corridas del programa OnlyControlSer\_AG.m del Anexo H, topología de reactores en serie.**

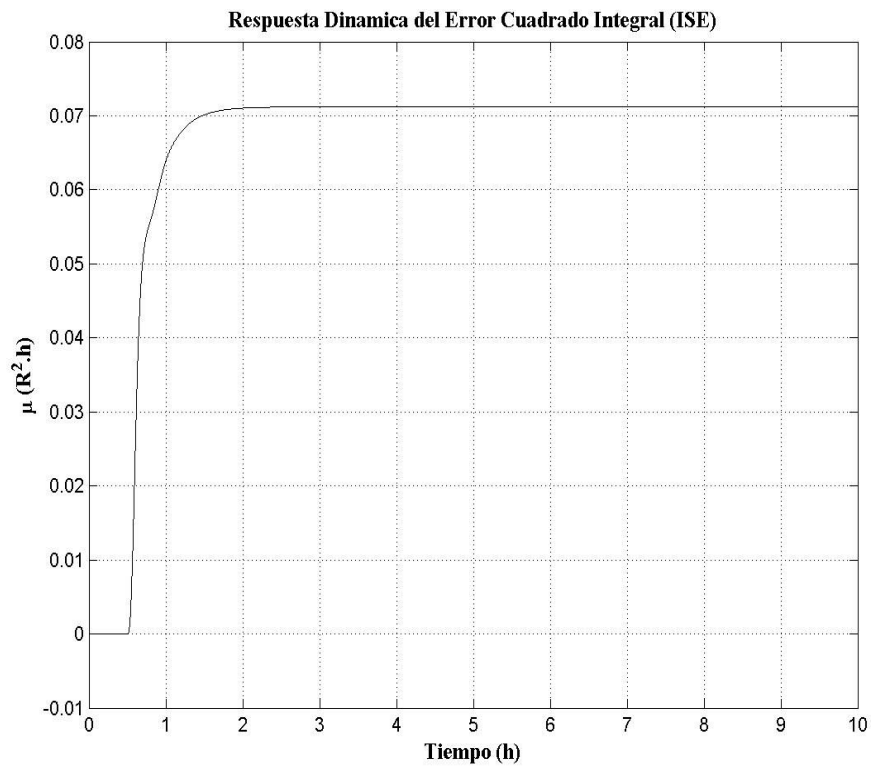
<b>Nº de corrida</b>	<b><math>\varepsilon</math> (ypsilón)</b>
1	0,0712
2	0,1352
3	0,2070
4	0,2310
5	0,2426

El comportamiento dinámico del ISE se muestra en la Gráfica 4.1.1 obtenido de la corrida del programa presentado en el Anexo G (Programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario mediante algoritmos genéticos, para topología de reactores en paralelo, para la corrida n°1 de la Tabla 4.1.5 (menor valor de  $\mu$ ).

Se observa que la variación de la controlabilidad a lo largo del periodo de estudio ocurre en el rango que va de  $[0,0001-0,0603]$ , para la distribución de reactores en paralelo, los cuales representan los valores donde la controlabilidad tiene impacto sobre los costos al momento de realizar el diseño integrado. Similarmente se reporta en la Gráfica 4.1.2 el comportamiento dinámico del error cuadrático integral (ISE) obtenido de la corrida del programa del Anexo H (Programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario mediante algoritmos genéticos, para topología de reactores en serie, para la corrida n°1 de la Tabla 4.1.6 (menor valor de  $\mu$ ), obteniéndose el rango  $[0,0001-0,0712]$ .



**Gráfica 4.1.1. Respuesta dinámica del Error Cuadrático Integral (ISE) para topología en paralelo.**



**Gráfica 4.1.2. Respuesta dinámica del error cuadrático integral (ISE) para topología en serie.**

Las siguientes tablas muestran los resultados obtenidos de la optimización con algoritmos genéticos de las variables de control en estado estacionario que se emplearán como condiciones iniciales para el problema de diseño integrado.



**Tabla 4.1.7. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo. Optimización de control mediante algoritmos genéticos. (Anexo G)**

<b>Parámetros óptimos de control</b>	
kj1 (ft <sup>3</sup> /h.R)	5.21620000e+000
kj2 (ft <sup>3</sup> /h.R)	3.01291600e+002
Tauj1 (h)	2.51520000e+000
Tauj2 (h)	4.13070000e+000
<b>Resultados estado estacionario</b>	
Fr (ft <sup>3</sup> /h)	2.88000000e-002
Cri1 (lbmol/ft <sup>3</sup> )	6.41016970e-001
Tri1 (R)	5.31758517e+002
Cro1 (lbmol/ft <sup>3</sup> ):	5.31644638e+002
Tro1 (R)	9.99712000e+001
Tjo1 (R)	2.45831302e-002
Cp (lbmol/ft <sup>3</sup> )	6.19977865e+002
Tp (R)	6.18757350e+002
Tjo2 (R)	2.47606631e-002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>	
Cro1 (lbmol/ft <sup>3</sup> )	6.40999128e-001
Tro1 (R)	5.31758518e+002
Tjo1 (R)	5.31574170e+002
Tro2 (R)	6.19977866e+002
Tjo2 (R)	6.17495955e+002
Itro1 (R.h)	3.40661563e-001
Itro2 (R.h)	5.73518093e-001
$\mu$ (R <sup>2</sup> .h)	6.03055467e-002
Cp (lbmol/ft <sup>3</sup> )	2.47606571e-002
Tp (R)	6.19952459e+002

**Tabla 4.1.8. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. Optimización del control mediante algoritmos genéticos.**

<b>Parámetros óptimos de control</b>	
kj1 (ft <sup>3</sup> /h.R)	1.05319634e+002
kj2 (ft <sup>3</sup> /h.R)	1.05319634e+002
Tauj1 (h)	5.60541854e-001
Tauj2 (h)	1.04464570e+000
<b>Resultados estado estacionario</b>	
Fr (ft <sup>3</sup> /h)	1.00130800e+002
Cri1 (lbmol/ft <sup>3</sup> )	9.98725930e-001
Tri1 (R)	6.00026051e+002
Cro1 (lbmol/ft <sup>3</sup> )	1.57049607e-001
Tro1 (R)	6.19971485e+002
Tjo1 (R)	6.16201836e+002
Cp (lbmol/ft <sup>3</sup> )	2.46633515e-002
Tp (R)	6.19942749e+002
Tjo2 (R)	6.19106039e+002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>	
Cro1 (lbmol/ft <sup>3</sup> )	1.57049585e-001
Tro1 (R)	6.19971485e+002
Tjo1 (R)	6.11435224e+002
Cp (lbmol/ft <sup>3</sup> )	2.46633170e-002
Tp (R)	6.19942751e+002
Tjo2 (R)	6.19106041e+002
Itro1 (R.h)	2.46359346e-001
Itro2 (R.h)	-2.12185070e-006
$\mu$ (R <sup>2</sup> .h)	7.12056202e-002

**Tabla 4.1.9. Resumen de los resultados obtenidos para los puntos de partida al diseño integrado. Topología de reactores en paralelo**

<b>Parámetros de diseño</b>	
Hr1 (ft)	7,91E+00
Hr2 (ft)	7,63E-01
Fb (ft <sup>3</sup> /h)	1,00E+00
Dr1 (ft)	2,36E+01
Dr2 (ft)	1,02E+00
Fjn1 (ft <sup>3</sup> /h)	3,93E+01
Fjn2 (ft <sup>3</sup> /h)	1,24E+01
<b>Parámetros de control</b>	
kj1 (ft <sup>3</sup> /h.R)	5.21620000e+000
kj2 (ft <sup>3</sup> /h.R)	3.01291600e+002
Tauj1 (h)	2.51520000e+000
Tauj2 (h)	4.13070000e+000

**Tabla 4.1.10. Resumen de los resultados obtenidos para los puntos de partida al diseño integrado. Topología de reactores en serie**

<b>Parámetros de diseño</b>	
Hr1 (ft)	4,36E+00
Hr2 (ft)	4,10E+00
Fb (ft <sup>3</sup> /h)	4,50E-01
Dr1 (ft)	1,29E+01
Dr2 (ft)	1,18E+01
Fjn1 (ft <sup>3</sup> /h)	3,68E+01
Fjn2 (ft <sup>3</sup> /h)	4,77E+00
<b>Parámetros de control</b>	
kj1 (ft <sup>3</sup> /h.R)	1.05319634e+002
kj2 (ft <sup>3</sup> /h.R)	1.05319634e+002
Tauj1 (h)	5.60541854e-001
Tauj2 (h)	1.04464570e+000

Asegurando que con el algoritmo genético se empezará la optimización local desde la región donde se encuentra el óptimo global, entonces se empleará el potencial de la función “fmincon” de rápida convergencia. En los Anexos I (OptDynPar.m) y J (OptDynSer.m) se muestran las programaciones en MATLAB R2012a para las dos topologías propuestas al sistema usando fmincon y alimentando el problema de diseño integrado allí presentado con los datos de las tablas 4.1.9 y 4.1.10. Estos programas manejan a su vez las funciones que se definen en MATLAB R2012a en “scripts”.

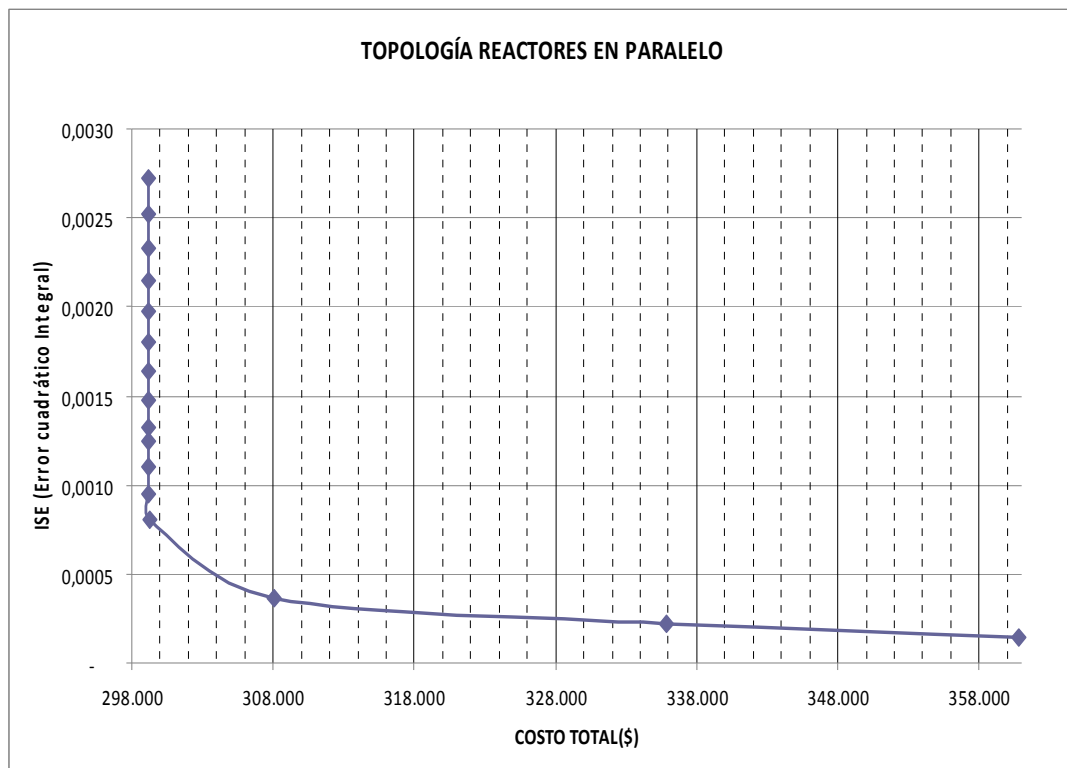
Estas funciones que son llamadas por el programa OptDynPar.m son: la función “Costo” (Anexo A), “DynReacPar” (Anexo B),  $T_i$  (Anexo F) y “StatReacPar” (Anexo D), que calcula la función costo, hace la simulación del sistema de reactores en estado dinámico, introduce la perturbación al sistema y simula el sistema de reactores en estado estacionario para la topología paralelo, respectivamente.

Similarmente se definen las funciones que son llamadas por el programa OptDynSer.m: la función “Costo” (Anexo A), “DynReacSer” (Anexo C),  $T_i$  (Anexo F) y “StatReacSer” (Anexo E), que calcula la función costo, hace la simulación del sistema de reactores en estado dinámico, introduce la perturbación al sistema y simula el sistema de reactores en estado estacionario para la topología serie, respectivamente.

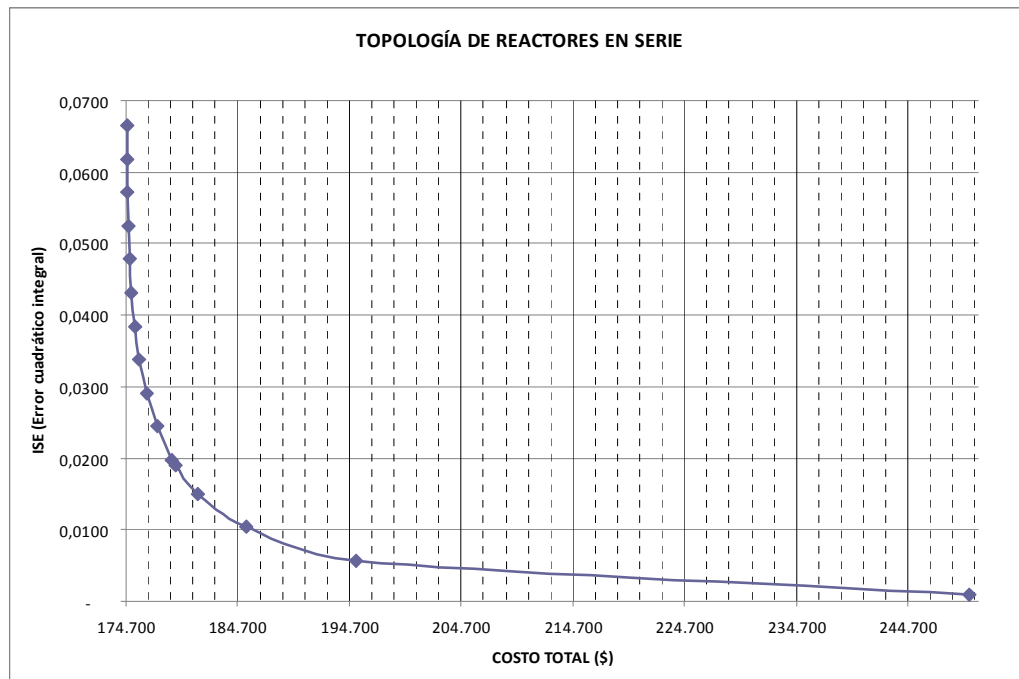
Para la corrida de la programación de los Anexos I y J se evalúan distintos valores de  $\varepsilon$  (ypsilon) de controlabilidad tanto para la topología de reactores en paralelo como para la topología de reactores en serie, dentro de los rangos ya encontrados. Es posible construir las siguientes gráficas de la relación entre el costo y la

controlabilidad del sistema (Ver Graficas 4.1.3 y 4.1.4), que van desde el valor óptimo económico de  $\varepsilon$  hasta el valor óptimo de controlabilidad.

Esto se hace para encontrar los valores que proporcionen un equilibrio tanto en los objetivos económicos como en los objetivos de control. Es obvio que a menor error cuadrático integral los costos totales del diseño final se incrementan, porque ejercer un mejor control sobre el sistema involucra aumento en los costos de utilidad (costos por refrigerante) que lógicamente repercuten en el costo total.



**Gráfica 4.1.3. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en paralelo**



**Gráfica 4.1.4. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en serie**

Los costos totales, en el caso de la topología de reactores en paralelo, se comienzan a elevar considerablemente cuando  $\varepsilon=0,0004$ , siendo el valor óptimo considerado  $\varepsilon=0,0008$  y para el caso en serie los costos totales se incrementan a partir de  $\varepsilon=0.0057$  aproximadamente, tomándose como óptimo  $\varepsilon = 0.019$ .

Como se observa, para el caso de la topología de reactores en paralelo es posible tener un control más riguroso al lograr seleccionar un valor de  $\varepsilon$  más bajo que en el caso de la disposición de reactores en serie, esto se debe principalmente a que en el resultado del diseño integrado en paralelo la dimensión de uno de los reactores es prácticamente despreciable en comparación con el diseño en serie.

**Tabla 4.1.11. Reporte de resultados del programa OptDynPar.m**

<b>Optimización en estado dinámico</b>	
Costo capital (\$)	293.932,1
Costo utilidad (\$)	5.286,6
Costo total (\$)	299.218,8
<b>Parámetros óptimos de diseño</b>	
Hr1 (ft)	5.00000000e-001
Hr2 (ft)	7.88680000e+000
Fb (ft <sup>3</sup> /h)	0.00000000e+000
Dr1 (ft)	1.00000000e+000
Dr2 (ft)	2.36603000e+001
Fjn1 (ft <sup>3</sup> /h)	1.00000000e+000
Fjn2 (ft <sup>3</sup> /h)	3.93312000e+001
<b>Parámetros óptimos de control</b>	
kj1 (ft <sup>3</sup> /h.R)	2,29432285e+002
kj2 (ft <sup>3</sup> /h.R)	2,81841776e+001
Tauj1 (h)	4,47488870e-001
Tauj2 (h)	7,90873617e-001
<b>Resultados estado estacionario</b>	
Fr (ft <sup>3</sup> /h)	1,00000160e+002
Cri1 (lbmol/ft <sup>3</sup> )	9,99998436e-001
Tri1 (R)	6,00000032e+002
Cro1 (lbmol/ft <sup>3</sup> )	1,01558575e-001
Tro1 (R)	6,20000000e+002
Tjo1 (R)	6,17026241e+002
Cp (lbmol/ft <sup>3</sup> )	2,50025123e-002
Tp (R)	6,20000000e+002
Tjo2 (R)	6,19287804e+002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>	
Cro1 (lbmol/ft <sup>3</sup> )	1,01555745e-001
Tro1 (R)	6,20000011e+002
Tjo1 (R)	6,13586850e+002
Cp (lbmol/ft <sup>3</sup> )	2,50000529e-002
Tp (R)	6,20000010e+002
Tjo2 (R)	6,19287885e+002
Itr1 (R.h)	8,71444634e-002
Itr2 (R.h)	-1,88166298e-005
mu (R <sup>2</sup> .h)	1,30006441e-002
Cp (lbmol/ft <sup>3</sup> )	2.50018856e-002
Tp (R)	6.19999965e+002

**Tabla 4.1.12. Reporte de resultados del programa OptDynSer.m**

<b>Optimización en estado dinámico</b>	
Costo capital (\$)	173.895,2
Costo utilidad (\$)	5.251,1
Costo total (\$)	179.146,3
<b>Parámetros óptimos de diseño</b>	
Hr1 (ft)	4.65621549e+000
Hr2 (ft)	3.50804785e+000
Fb (ft <sup>3</sup> /h)	1.52131422e-006
Dr1 (ft)	1.39686465e+001
Dr2 (ft)	1.05241435e+001
Fjn1 (ft <sup>3</sup> /h)	3.54312801e+001
Fjn2 (ft <sup>3</sup> /h)	4.62895994e+000
<b>Parámetros óptimos de control</b>	
kj1 (ft <sup>3</sup> /h.R)	1.74691406e+002
kj2 (ft <sup>3</sup> /h.R)	9.46012566e+001
Tauj1 (h)	3.57286347e-001
Tauj2 (h)	9.78506014e-001
<b>Resultados estado estacionario</b>	
Fr (ft <sup>3</sup> /h)	1.00000002e+002
Cri1 (lbmol/ft <sup>3</sup> )	9.99999985e-001
Tri1 (R)	6.00000000e+002
Cro1 (lbmol/ft <sup>3</sup> )	1.10806377e-001
Tro1 (R)	6.20000000e+002
Tjo1 (R)	6.16871795e+002
Cp (lbmol/ft <sup>3</sup> )	2.50022588e-002
Tp (R)	6.20000000e+002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>	
Cro1 (lbmol/ft <sup>3</sup> )	1.10806372e-001
Tro1 (R)	6.20000000e+002
Tjo1 (R)	6.13201306e+002
Cp (lbmol/ft <sup>3</sup> )	2.50022521e-002
Tp (R)	6.20000000e+002
Tjo2 (R)	6.19260214e+002
Itro1 (R.h)	9.19756541e-002
Itro2 (R.h)	-1.20823869e-007
mu (R <sup>2</sup> .h)	1.90000000e-002



**Tabla 4.1.13. Tabla comparativa resultados de las topologías de rectores estudiadas**

Variables/Parámetros	Topología	
	Serie	Paralelo
<b>Optimización en estado dinámico</b>		
Costo capital (\$)	173.895,2	293.932,1
Costo utilidad (\$)	5.251,1	5.286,6
Costo total (\$)	179.146,3	299.218,8
<b>Parámetros óptimos de diseño</b>		
Hr1 (ft)	4.65621549e+000	5.00000000e-001
Hr2 (ft)	3.50804785e+000	7.88680000e+000
Fb (ft <sup>3</sup> /h)	1.52131422e-006	0.00000000e+000
Dr1 (ft)	1.39686465e+001	1.00000000e+000
Dr2 (ft)	1.05241435e+001	2.36603000e+001
Fjn1 (ft <sup>3</sup> /h)	3.54312801e+001	1.00000000e+000
Fjn2 (ft <sup>3</sup> /h)	4.62895994e+000	3.93312000e+001
<b>Parámetros óptimos de control</b>		
kj1 (ft <sup>3</sup> /h.R)	1.74691406e+002	2,29432285e+002
kj2 (ft <sup>3</sup> /h.R)	9.46012566e+001	2,81841776e+001
Tauj1 (h)	3.57286347e-001	4,47488870e-001
Tauj2 (h)	9.78506014e-001	7,90873617e-001
<b>Resultados estado estacionario</b>		
Fr (ft <sup>3</sup> /h)	1.00000002e+002	1.00000002e+002
Cr1l (lbmol/ft <sup>3</sup> )	9.99999985e-001	9.99999985e-001
Tri1 (R)	6.00000000e+002	6.00000000e+002
Cro1 (lbmol/ft <sup>3</sup> )	1.10806377e-001	1.10806377e-001
Tro1 (R)	6.20000000e+002	6.20000000e+002
Tjo1 (R)	6.16871795e+002	6.16871795e+002
Cp (lbmol/ft <sup>3</sup> )	2.50022588e-002	2.50022588e-002
Tp (R)	6.20000000e+002	6.20000000e+002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>		
Cro1 (lbmol/ft <sup>3</sup> )	1.10806372e-001	1,01555745e-001
Tro1 (R)	6.20000000e+002	6,20000011e+002
Tjo1 (R)	6.13201306e+002	6,13586850e+002
Cp (lbmol/ft <sup>3</sup> )	2.50022521e-002	2,50000529e-002
Tp (R)	6.20000000e+002	6,20000010e+002
Tjo2 (R)	6.19260214e+002	6,19287885e+002
Itro1 (R.h)	9.19756541e-002	8,71444634e-002
Itro2 (R.h)	-1.20823869e-007	-1,88166298e-005
mu (R <sup>2</sup> .h)	1.90000000e-002	1,30006441e-002

Las Tablas 4.1.11 y 4.1.12, que presentan el reporte generado por el programa OptDynPar.m (Anexo I) para el valor de  $\varepsilon=0,0008$  y el reporte del programa OptDynSer.m (Anexo J) para  $\varepsilon = 0.019$ . Estos valores representan el resultado del diseño integrado del sistema reactivo objeto de estudio.

**Tabla 4.1.14. Resumen de las dimensiones óptimas de los reactores**

Reactor	Dimensiones de los reactores	Reactores en paralelo	Reactores en serie
Primer reactor	Hr1 (ft)	5.00000000e-001	4.65621549e+000
	Dr1 (ft)	1.00000000e+000	1.39686465e+001
Segundo Reactor	Hr2 (ft)	7.88680000e+000	3.50804785e+000
	Dr2 (ft)	2.36603000e+001	1.05241435e+001

La razón por la que el arreglo de reactores en paralelo tiene mejor controlabilidad que en serie se debe a los resultados obtenidos en cuanto a las dimensiones entre los reactores de cada topología.

El área de transferencia de calor de los reactores calculados a partir del conjunto de ecuaciones 129 y 130 se muestra en la Tabla 4.1.15.

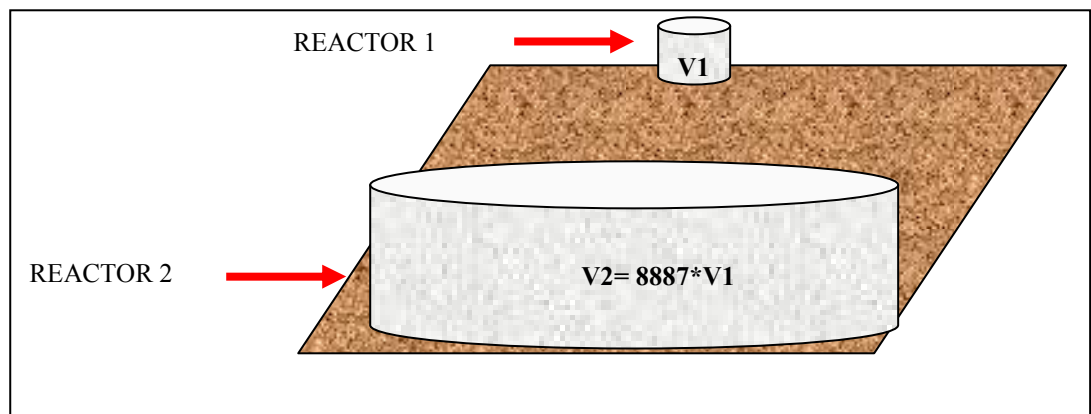
$$A_1 = \pi \cdot D_{r1} \cdot H_{r1} \quad (129)$$

$$A_2 = \pi \cdot D_{r2} \cdot H_{r2} \quad (130)$$

**Tabla 4.1.15. Volumen de los reactores y área de transferencia de calor**

Reactor	Volumen del Reactor (ft <sup>3</sup> )		Área de intercambio de calor (ft <sup>2</sup> )	
	Paralelo	Serie	Paralelo	Serie
	Primer reactor	0,39	713,20	1,57
Segundo Reactor	3465,86	305,00	585,94	115,93

Se observa que para el arreglo en paralelo el segundo reactor es mucho más grande que el primero, la relación entre los volúmenes  $V_2/V_1$  de los dos reactores es tan grande que prácticamente el primer reactor no existe, originando que la disposición en paralelo no sea óptima porque se dispone en sí de un solo reactor en donde casi toda el área de transferencia de calor la soporta el segundo reactor.



**Figura 4.1.1. Ilustración de la relación entre las dimensiones de los dos (2) reactores en el arreglo en paralelo**

Como el volumen de reacción es muy grande puede tolerar las perturbaciones sin que el sistema sufra mayor desestabilización. Al comparar este fenómeno con la disposición de reactores en serie se observa que tanto los volúmenes como el área de transferencia de calor están distribuidos entre los dos reactores en una relación  $V_2=0,43V_1$ .

Desde el punto de vista del proceso químico se tiene que las perturbaciones para el diseño en serie impactarán en mayor magnitud que en paralelo, lo cual es lógico ya que una variación en la temperatura de entrada o en el flujo de entrada de los reactivos modificará más rápidamente y en mayor magnitud bien sea la temperatura del reactor o concentración de algún compuesto en la masa reaccionante menor.

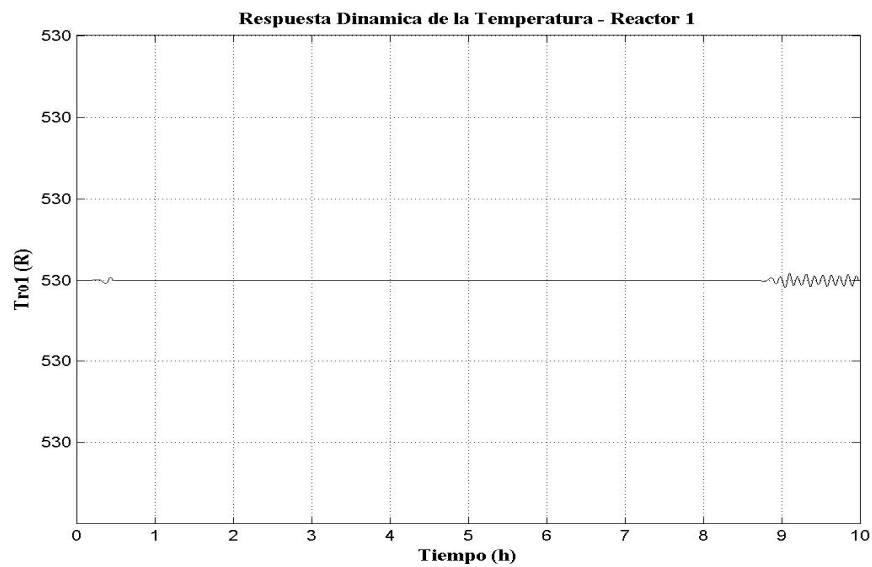
#### **Comparación de topologías y selección de la mejor disposición de los reactores obtenidas por el uso de algoritmos genéticos**

Comparando inicialmente los costos totales según la Tabla 4.1.16, se establece que el arreglo de los reactores en serie es óptimo, por presentar un diseño más económico.

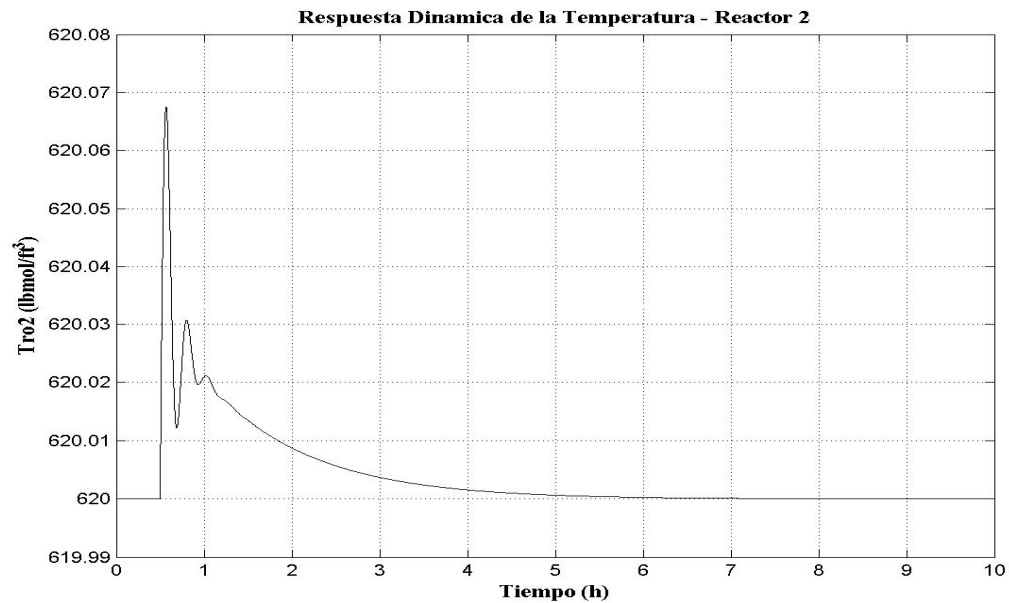
**Tabla 4.1.16. Comparación de los costos totales según el arreglo de los reactores**

<b>Topología de los reactores</b>	<b>Costo total (\$)</b>
Paralelo	299.218,8
Serie	179.146,3

Comparando la respuesta dinámica frente a las perturbaciones de ambos arreglos, se analizan las Gráficas 4.1.5 y 4.1.6. En el arreglo en paralelo la dinámica del primer reactor es prácticamente nula en comparación con el segundo reactor del mismo arreglo.

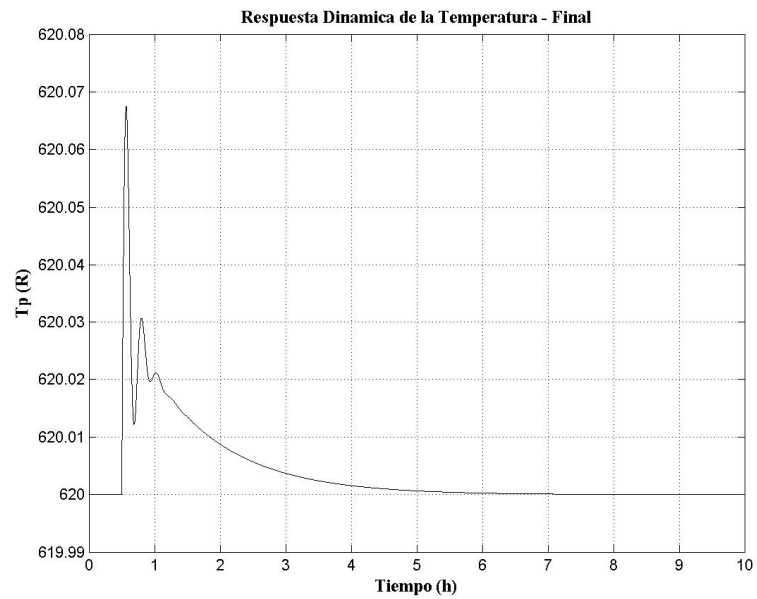


**Gráfica 4.1.5. Respuesta Dinámica de la temperatura de salida del primer reactor. Diseño Integrado para el arreglo de reactores en paralelo**

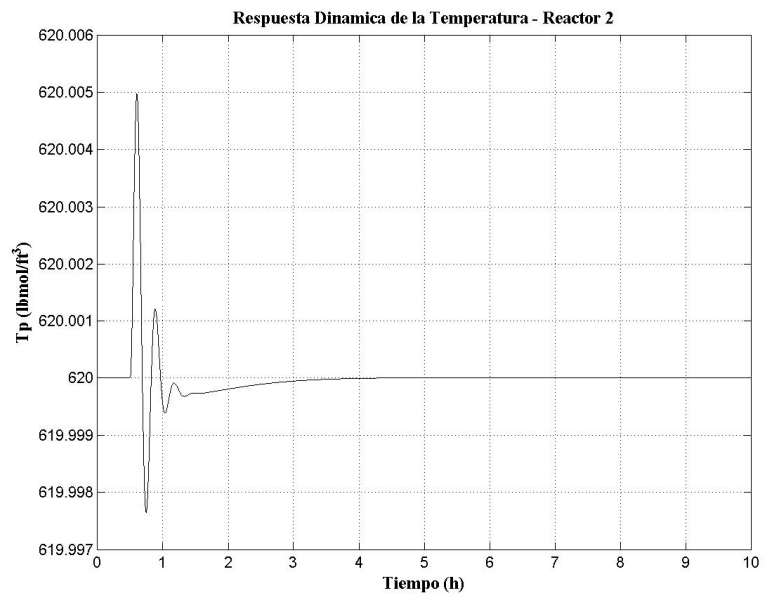


**Gráfica 4.1.6. Respuesta dinámica de la temperatura de salida del segundo reactor. Diseño integrado para el arreglo de reactores en paralelo**

Como ya se ha mencionado, el arreglo de reactores en paralelo tiene mejor controlabilidad que en serie por los resultados relativos a las dimensiones entre los reactores de cada topología. Pero aunque la totalidad del proceso en la topología en paralelo puede ser más controlable, la disposición en serie tiene un mejor efecto en la amortiguación del efecto de la perturbación en el producto. Esto se verifica en el comportamiento dinámico del sistema, específicamente en el estudio de la dinámica en la temperatura de salida en ambos diseños (ver Gráficas 4.1.7 y 4.1.8) se establece que el comportamiento del arreglo en serie presenta un mejor desempeño que el diseño en paralelo, ya que el primero tiene un tiempo de asentamiento de 4 y el último de 6 horas, así como menor sobre impulso.



**Gráfica 4.1.7. Respuesta dinámica de la temperatura final. Diseño integrado para el arreglo de reactores en paralelo**



**Gráfica 4.1.8. Respuesta dinámica de la temperatura a la salida del segundo reactor. Diseño integrado para el arreglo de reactores en serie**

Finalmente se observa que el resultado del arreglo en paralelo no es viable técnicamente porque se está llevando a cabo toda la reacción en un solo reactor. Se genera una inversión adicional por ese equipo que elevará los costos totales (ver Tabla 4.1.16 de comparación de los costos totales).

La mejor topología de dos reactores mezcla completa es en serie. Obviamente el método de optimización para resolver el problema planteado es válido en cuanto a la síntesis del sistema ya que se obtienen los mismos resultados reflejados por Luyben (1993) quien estudió el problema de síntesis de reactores para analizar el compromiso entre el diseño económico y el diseño controlado. Luyben encontró que el mejor sistema, según criterios económicos de diseño en estado estacionario y controlabilidad, corresponde al caso de dos reactores conectados en serie, pues se requiere un menor volumen para llegar al 95% de conversión requerida. Los resultados finales de la síntesis y diseño integrado de reactores mezcla completa obtenidos a través del diagrama de bloques del algoritmo de solución del problema, se muestran en la Tabla 4.1.17.

**Tabla 4.1.17. Resultados finales de la síntesis y diseño integrado de reactores mezcla completa usando algoritmos genéticos**

<b>Arreglo</b>		<b>SERIE</b>	
<b>Dimensiones</b>	Primer Reactor	Hr1 (ft)	4,7
		Dr1 (ft)	14
	Segundo Reactor	Hr2 (ft)	3,6
		Dr2 (ft)	10,6
<b>Parámetros del Controlador</b>	Primer Reactor	$kj1$ (ft <sup>3</sup> /h.R)	174,69
		$kj2$ (ft <sup>3</sup> /h.R)	94,6
	Segundo Reactor	$\tau_{uj1}$ (h)	0,357
		$\tau_{uj2}$ (h)	0,979



## 4.2 RECOCIDO SIMULADO (SA)

### Implementación de la estructura algorítmica en MATLAB R2012a

Aplicando la metodología expuesta en el Capítulo III, se procede con el método del recocido simulado, de aquí en adelante “SA” y se obtienen las condiciones iniciales del problema matemático del diseño integrado. Se optimiza en estado estacionario por separado y en forma secuencial primeramente las variables del diseño de los reactores. Se optimizan también en el estado estacionario las variables de control que se relacionan con los parámetros del controlador. Se realiza una optimización de la función costos sujeta al planteamiento matemático en estado estacionario para cada topología por separado definiendo la programación en MATLAB R2012a, usando SA.

La definición de la programación en MATLAB R2012a se presenta en el Anexo O: Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en paralelo, usando SA, (OptEstatPar\_SA.m) y en el Anexo P: Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie, usando SA, (OptEstatSer\_SA.m).

Similarmente a la forma en que se estructuró la programación con AG, se establece para programa del Anexo O, OptEstatPar\_SA.m, el conjunto de funciones que se usan dentro de su bucle de programación:

- “Costo.m” (Anexo A),
- “DynReacPar.m” (Anexo B) y

- “StatReacPar.m” (Anexo D).

También se definen las funciones que son llamadas por el programa OptEstatSer\_SA.m (Anexo P):

- “Costo.m” (Anexo A), que calcula la función costo,
- “DynReacSer” (Anexo C), que hace la simulación del sistema de reactores en estado dinámico y
- “StatReacSer” (Anexo E), que simula el sistema de reactores en estado estacionario para la topología serie.

A continuación se resumen las opciones empleadas

**Tabla 4.2.1. Opciones del SA para el diseño integrado del sistema**

<b>Opciones</b>	<b>Valor asignado/Función asignada</b>
options.MaxIter	20000
options.Display	'iter'
options.MaxFunEvals	Inf
options.InitialTemperature	1000000
options.TemperatureFcn	@temperaturexp
options.PlotFcns	@saplotbestf
options.ReannealInterval	500

Se imponen los límites del problema especificando un rango de:

$$LB = [0.5, 0.5, 0, 1, 1, 1, 1]; UB = [100, 100, 1, 100, 100, 500, 500];$$

Para este trabajo se especifica una temperatura inicial muy elevada, de acuerdo a las características inherentes al proceso de recocido de un sólido, que especifica que inicialmente se debe incrementar la temperatura del “baño térmico” a un valor máximo.

**Tabla 4.2.2. Reporte de resultados del programa OptEstatPar\_SA.m (Anexo O) para el estado estacionario. Topología de reactores en paralelo**

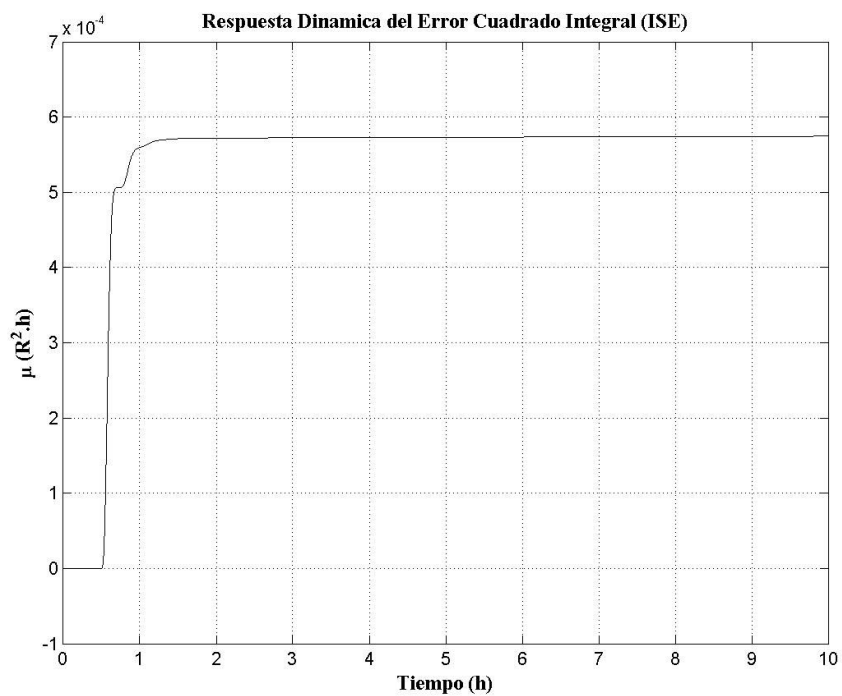
<b>Optimización en estado estacionario</b>	
Costo capital (\$)	304.578,123
Costo utilidad (\$)	15.468,062
Costo total (\$)	320.046,186
<b>Parámetros óptimos de diseño</b>	
Hr1 (ft)	8,31
Hr2 (ft)	1,99
Fr1 (ft <sup>3</sup> /h)	0,1
Dr1 (ft)	23,06
Dr2 (ft)	2,12
Fjn1 (ft <sup>3</sup> /h)	39,34
Fjn2 (ft <sup>3</sup> /h)	78,66

**Tabla 4.2.3. Reporte de resultados del programa OptEstatSer\_SA.m (Anexo P) para el estado estacionario. Topología de reactores en serie**

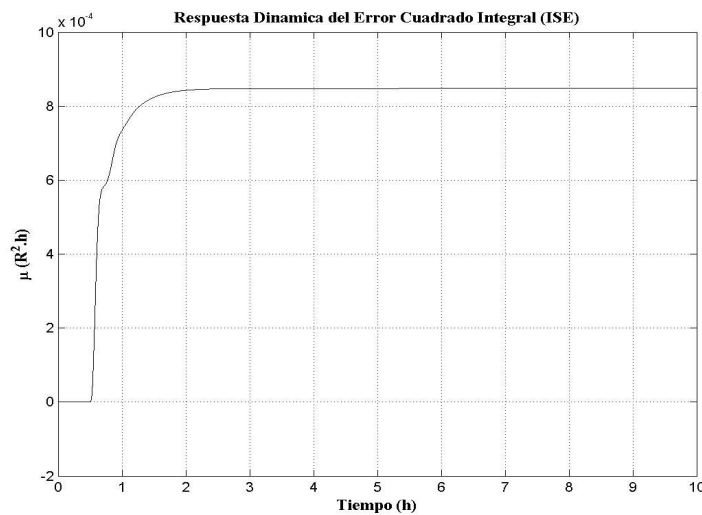
<b>Optimización en estado estacionario</b>	
Costo capital (\$)	333606,8
Costo utilidad (\$)	28064,96
Costo total (\$)	361671,76
<b>Parámetros óptimos de diseño</b>	
Hr1 (ft)	8.83188238e+000
Hr2 (ft)	3.27852631e+001
Fb (ft <sup>3</sup> /h)	2.95944797e-001
Dr1 (ft)	2.23746448e+001
Dr2 (ft)	1.00001073e+00
Fjn1 (ft <sup>3</sup> /h)	3.97724619e+001
Fjn2 (ft <sup>3</sup> /h)	1.74333159e+002

### Determinación del intervalo de estudio para $\varepsilon$ y de los puntos de partida de los parámetros del controlador

La definición de los programas en MATLAB R2012a con el uso de SA son mostrados en los Anexos S (OnlyControlPar\_SA.m) y R (OnlyControlSer\_SA.m) para la optimización mediante SA de las variables de control en estado estacionario para topología de reactores en paralelo y en serie, respectivamente. Los resultados se muestran en las gráficas siguientes.



Gráfica 4.2.1. Respuesta dinámica del error cuadrático integral (ISE) para topología en paralelo



**Gráfica 4.2.2. Respuesta dinámica del error cuadrático integral (ISE) para topología en serie**

El comportamiento dinámico del error cuadrático integral (ISE) se muestra en la Gráfica 4.2.1 obtenido de la corrida del programa presentado en el Anexo O. Se observa que la variación de la controlabilidad a lo largo del periodo de estudio ocurre en el rango  $[0,0001- 0,0006]$ , para la distribución de reactores en paralelo, en los cuales la controlabilidad tiene impacto sobre los costos al momento de realizar el diseño integrado.

Similarmente se reporta en la Gráfica 4.2.2 el comportamiento dinámico del error cuadrático integral (ISE) obtenido de la corrida del programa del Anexo Q, es decir, para la programación en MATLAB R2012a del cálculo de las variables de control en estado estacionario mediante SA, para topología de reactores en serie, para el menor valor de  $\mu$ , obteniéndose el rango  $[0,0001-0,0008]$ . Las siguientes tablas muestran los resultados obtenidos de la optimización con SA de las variables de control en estado estacionario que se emplearán como condiciones iniciales para el problema de diseño integrado.

**Tabla 4.2.4. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo. Optimización de control mediante SA**

<b>Parámetros óptimos de control</b>	
kj1 (ft <sup>3</sup> /h.R)	9.96089920e+002
kj2 (ft <sup>3</sup> /h.R)	9.98599501e+002
Tauj1 (h)	3.64321937e-001
Tauj2 (h)	6.23636646e-001
<b>Resultados estado estacionario</b>	
Fr (ft <sup>3</sup> /h)	9.99994000e+001
Cri1 (lbmol/ft <sup>3</sup> )	2.50001191e-002
Tri1 (R)	6.19975582e+002
Cro1 (lbmol/ft <sup>3</sup> ):	6.18771195e+002
Tro1 (R)	6.00000000e-004
Tjo1 (R)	4.71153359e-003
Cp (lbmol/ft <sup>3</sup> )	5.30001536e+002
Tp (R)	5.30000687e+002
Tjo2 (R)	2.49999974e-002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>	
Cro1 (lbmol/ft <sup>3</sup> )	2.50001186e-002
Tro1 (R)	6.19975582e+002
Tjo1 (R)	6.17525801e+002
Tro2 (R)	5.30002055e+002
Tjo2 (R)	5.30000865e+002
Itro1 (R.h)	1.52966077e-002
Itro2 (R.h)	5.25998631e-003
$\mu$ (R <sup>2</sup> .h)	5.74538718e-004

**Tabla 4.2.5. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. Optimización de control mediante SA**

<b>Parámetros óptimos de control</b>	
kj1 (ft <sup>3</sup> /h.R)	9.88351478e+002
kj2 (ft <sup>3</sup> /h.R)	1.70884356e+001
Tauj1 (h)	6.67535256e-001
Tauj2 (h)	9.86521037e+000
<b>Resultados estado estacionario</b>	
Fr (ft <sup>3</sup> /h)	1.00295945e+002
Cri1 (lbmol/ft <sup>3</sup> )	9.97123052e-001
Tri1 (R)	5.99877635e+002
Cro1 (lbmol/ft <sup>3</sup> )	2.54969120e-002
Tro1 (R)	6.19450897e+002
Tjo1 (R)	6.18276446e+002
Cp (lbmol/ft <sup>3</sup> )	2.50000000e-002
Tp (R)	5.58530331e+002
Tjo2 (R)	5.51110272e+002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>	
Cro1 (lbmol/ft <sup>3</sup> )	2.54969096e-002
Tro1 (R)	6.19450897e+002
Tjo1 (R)	6.17068349e+002
Cp (lbmol/ft <sup>3</sup> )	2.49999999e-002
Tp (R)	5.58530199e+002
Tjo2 (R)	5.51110123e+002
Itr1 (R.h)	2.83881108e-002
Itr2 (R.h)	2.22614434e-003
$\mu$ (R <sup>2</sup> .h)	8.47858313e-004

**Tabla 4.2.6. Resumen de los resultados obtenidos para los puntos de partida al Diseño Integrado. Topología de reactores en paralelo**

<b>Parámetros de diseño</b>	
Hr1 (ft)	8.31416200e+000
Hr2 (ft)	1.98531500e+000
Fb (ft <sup>3</sup> /h)	9.99994000e-001
Dr1 (ft)	2.30559870e+001
Dr2 (ft)	2.11871200e+000
Fjn1 (ft <sup>3</sup> /h)	3.93440690e+001
Fjn2 (ft <sup>3</sup> /h)	7.86606780e+001
<b>Parámetros de control</b>	
kj1 (ft <sup>3</sup> /h.R)	9.96089920e+002
kj2 (ft <sup>3</sup> /h.R)	9.98599501e+002
Tauj1 (h)	3.64321937e-001
Tauj2 (h)	6.23636646e-001

**Tabla 4.2.7. Resumen de los resultados obtenidos para los puntos de partida al Diseño Integrado. Topología de reactores en serie**

<b>Parámetros de diseño</b>	
Hr1 (ft)	8.83188238e+000
Hr2 (ft)	3.27852631e+001
Fb (ft <sup>3</sup> /h)	2.95944797e-001
Dr1 (ft)	2.23746448e+001
Dr2 (ft)	1.00001073e+000
Fjn1 (ft <sup>3</sup> /h)	3.97724619e+001
Fjn2 (ft <sup>3</sup> /h)	1.74333159e+002
<b>Parámetros de control</b>	
kj1 (ft <sup>3</sup> /h.R)	9.88351478e+002
kj2 (ft <sup>3</sup> /h.R)	1.70884356e+001
Tauj1 (h)	6.67535256e-001
Tauj2 (h)	9.86521037e+000



Hasta este punto se observan las cualidades de este método de optimización heurístico o global en la determinación de las condiciones iniciales. El SA presenta las ventajas de simplicidad y aplicabilidad, como se muestra en la determinación de las opciones y en los “scripts” mostrados en los anexos.

Ahora, siendo consistente en la metodología empleada con algoritmos genéticos, se hace uso del potencial de los métodos híbridos de optimización empezando la optimización local desde la región donde se encuentra el óptimo global proporcionado por la minimización usando SA, entonces se emplea la función “fmincon” de rápida convergencia.

En los Anexos I (OptDynPar.m) y J (OptDynSer.m) se muestran las programaciones en MATLAB R2012a para las dos topologías propuestas al sistema usando “fmincon” y alimentando el problema de diseño integrado allí presentado con los datos de las Tablas 4.2.6 y 4.2.7.

Estos programas manejan a su vez las funciones que se definen en MATLAB R2012a en “scripts”.

Funciones o “scripts” requeridas por OptDynPar.m:

- 1) Función “Costo” (Anexo A), que calcula la función costo.
- 2) “DynReacPar” (Anexo B), hace la simulación del sistema de reactores en paralelo en el estado dinámico.
- 3) Ti (Anexo F), introduce la perturbación al sistema.
- 4) “StatReacPar” (Anexo D), que simula el sistema de reactores en estado estacionario para la topología paralelo.

Funciones o “scripts” llamadas por el programa OptDynSer.m:

- 1) Función “Costo” (Anexo A).
- 2) “DynReacSer” (Anexo C), hace la simulación del sistema de reactores en serie en el estado dinámico.
- 3) Ti (Anexo F).
- 4) “StatReacSer” (Anexo E), simula el sistema de reactores en estado estacionario para la topología serie.

Para la corrida de la programación de los Anexos I y J se evalúan distintos valores de  $\varepsilon$  (ypsilón) de controlabilidad tanto para la topología de reactores en paralelo como para la topología de reactores en serie, dentro de los rangos encontrados.

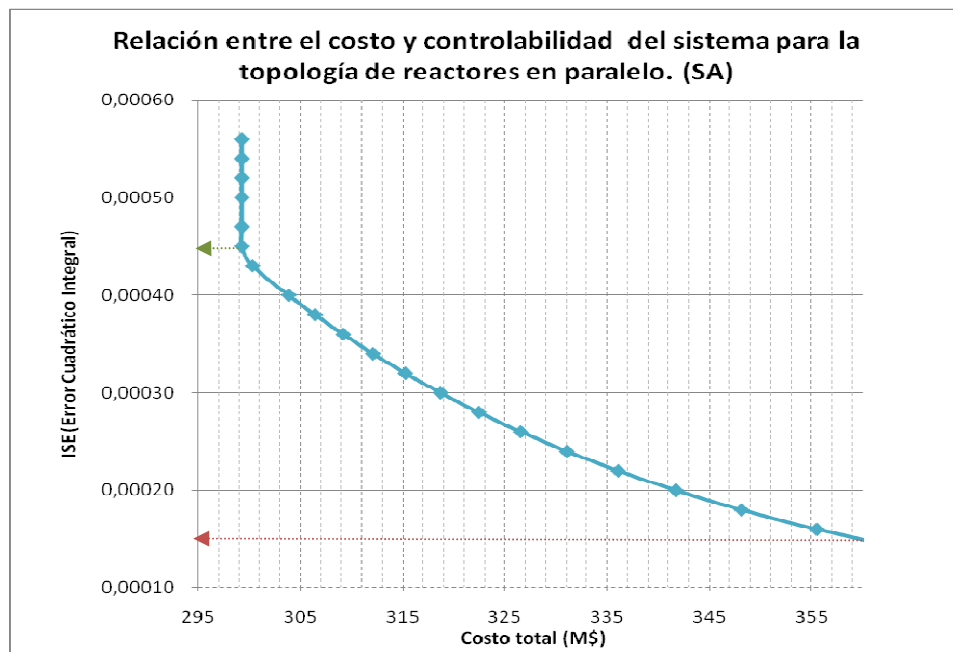
Se construyen las siguientes gráficas de la relación entre el costo y la controlabilidad del sistema (Ver Graficas 4.2.3 y 4.2.4), que van desde el valor óptimo económico de  $\varepsilon$  hasta el valor óptimo de controlabilidad.

La razón por la que esto se efectúa, como ya se mencionó en la sección anterior, es para encontrar los valores que proporcionen un equilibrio tanto en los objetivos económicos como en los objetivos de control.

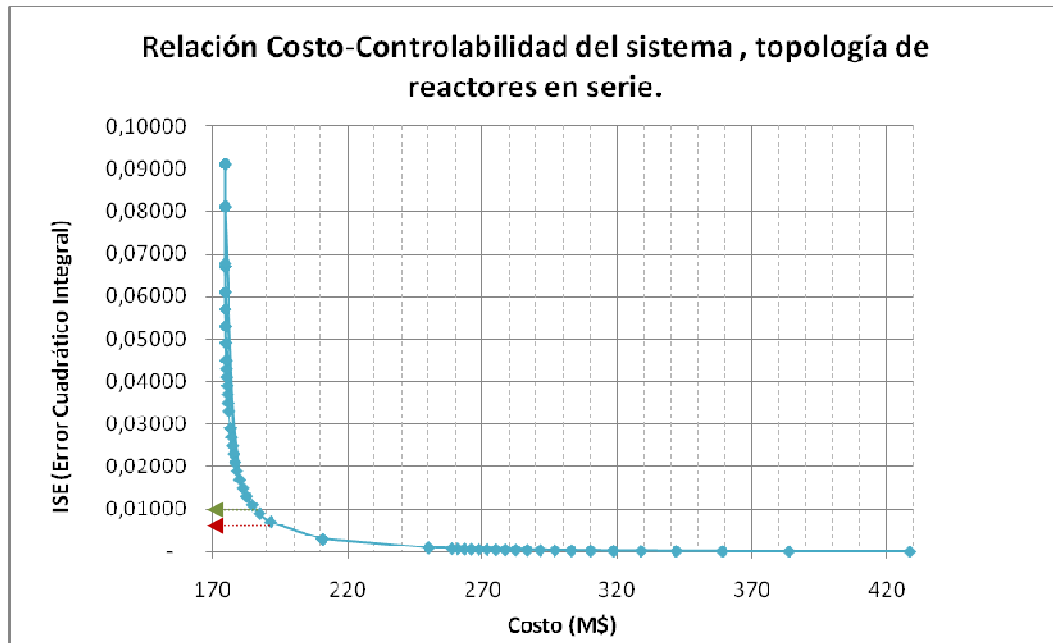
Es importante hacer mención del contenido presentado en el Capítulo II, en el que se establece según Gutiérrez y de Prada (2003) que el “Diseño Integrado” es una técnica o enfoque de diseño de plantas mediante el cual las características de controlabilidad son consideradas previamente en las etapas de diseño, lo que permite fijar especificaciones dinámicas del sistema que facilitará la operación y flexibilidad de las mismas que pueden ser gobernadas por un sistema de control automático.

La metodología presentada para la resolución del problema diseño integrado del sistema estudiado permite la evaluación de los parámetros de la planta y el sistema de control al mismo tiempo, haciendo que el sistema diseñado sea más fácilmente controlable.

Al evaluar los diversos índices de controlabilidad junto con consideraciones económicas para producir una planta óptima, se observa que a menor error cuadrático integral los costos totales del diseño final se incrementan. Esto se debe a que existe un mejor control sobre el sistema, lo cual involucra el aumento en los costos de utilidad (costos por refrigerante) que repercute en el costo total.



**Gráfica 4.2.3. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en paralelo. Hibridación SA y SQP**



**Gráfica 4.2.4. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en serie. Hibridación SA y SQP**

Los costos totales, en el caso de la topología de reactores en paralelo, se elevan sin observar mejora en la controlabilidad del sistema cuando  $\varepsilon=0,00015$  (ver línea roja punteada, Gráfica 4.2.3). El valor óptimo considerado es de  $\varepsilon=0,00045$  (ver línea verde punteada, Gráfica 4.2.3), que es el punto de inflexión de la curva a partir del cual los costos se incrementan en relación al error cuadrático integral.

Para el caso en serie los costos totales se incrementan a partir de  $\varepsilon=0,0010$  aproximadamente (ver línea roja punteada, Gráfica 4.2.4), tomándose como óptimo  $\varepsilon = 0,011$  (ver línea verde punteada, Gráfica 4.2.4).

Es por esta razón que estos valores representan el resultado del diseño integrado del sistema reactivo objeto de estudio. Como se observa, para el caso de la topología de reactores en paralelo es posible tener un control más riguroso al lograr seleccionar un valor de  $\varepsilon$  más bajo que en el caso de la disposición de reactores en serie, esto se debe principalmente a que en el resultado del diseño integrado en paralelo la dimensión de uno de los reactores es prácticamente despreciable en comparación con el diseño en serie. Véase Tabla 4.2.8.

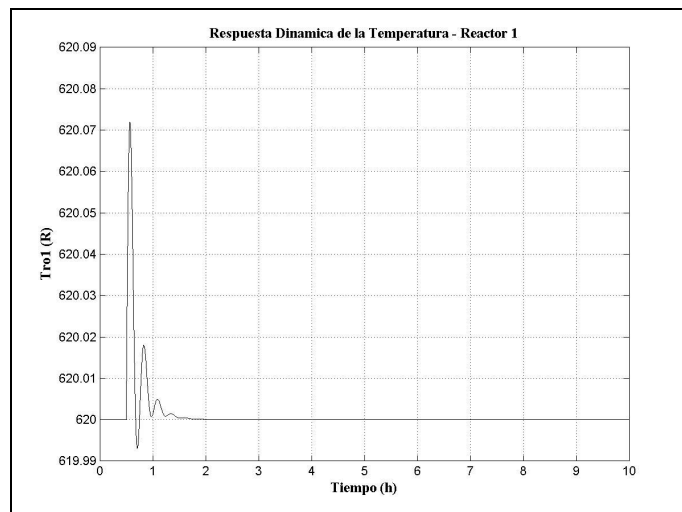
**Tabla 4.2.8. Resultados del diseño integrado mediante método híbrido (SA + SQP). Comparación de topologías serie y paralelo**

Variables/Parámetros	Topología	
	Paralelo	Serie
<b>Optimización en estado dinámico</b>		
Costo capital (\$)	293.945,26	179.467,32
Costo utilidad (\$)	5.286,62	5.237,29
Costo total (\$)	299.231,88	184.704,61
<b>Parámetros óptimos de diseño</b>		
Hr1 (ft)	7,89	4,98
Hr2 (ft)	0,50	3,24
Fb (ft <sup>3</sup> /h)	0	0
Dr1 (ft)	23,66	14,92
Dr2 (ft)	1,00	9,72
Fjn1 (ft <sup>3</sup> /h)	39,33	36,31
Fjn2 (ft <sup>3</sup> /h)	1,00	9,72
<b>Parámetros óptimos de control</b>		
kj1 (ft <sup>3</sup> /h.R)	1.077,94	228,40
kj2 (ft <sup>3</sup> /h.R)	998,60	264,23
Tauj1 (h)	0,273	0,348
Tauj2 (h)	1,72	0,797
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>		
Cro1 (lbmol/ft <sup>3</sup> )	2.50000865e-002	9.26603247e-002
Tro1 (R)	6.20000000e+002	6,20000011e+002
Tjo1 (R)	6.17484059e+002	6.13968407e+002
Cp (lbmol/ft <sup>3</sup> )	2.50000865e-002	2.50015892e-002
Tp (R)	6.20000000e+002	6,20000010e+002
Tjo2 (R)	5.30000000e+002	6.19316555e+002
Itr01 (R.h)	1.05980454e-002	6.77100659e-002
Itr02 (R.h)	-4.34457804e-007	-1.17763357e-008
mu (R <sup>2</sup> .h)	4.49980532e-004	1.10000004e-002

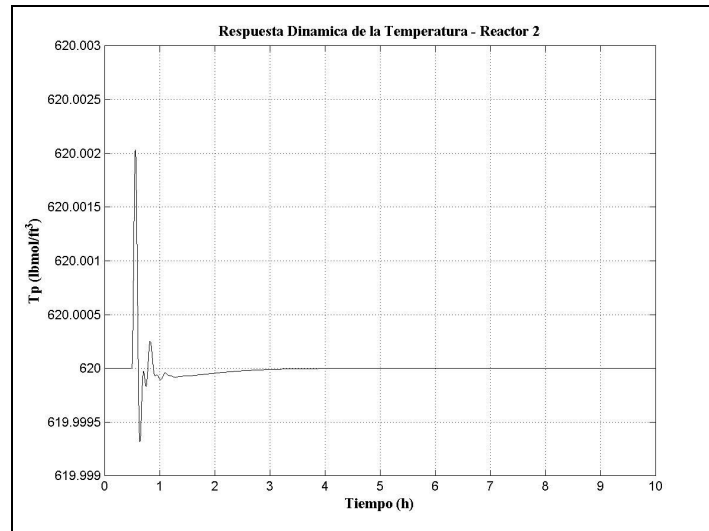
### Comparación de ambas topologías y selección de la mejor disposición de los reactores

Comparando inicialmente los costos totales según la Tabla 4.2.8, se establece que el arreglo de los reactores en serie es económicamente óptimo, por presentar un diseño más barato.

Para estudiar la controlabilidad del sistema, se deben considerar las siguientes gráficas:



**Gráfica 4.2.5. Respuesta dinámica de la temperatura a la salida del segundo reactor. Topología de reactores en paralelo. Hibridación SA/SQP**



**Gráfica 4.2.6. Respuesta dinámica de la temperatura a la salida del segundo reactor. Topología de reactores en serie. Hibridación SA/SQP**

Como se observa, para el caso de la topología de reactores en paralelo es posible tener un control más riguroso al lograr seleccionar un valor de  $\varepsilon$  más bajo que en el caso de la disposición de reactores en serie, esto se debe principalmente a que en el resultado del diseño integrado en paralelo la dimensión de uno de los reactores es prácticamente despreciable en comparación con el diseño en serie, siendo la relación entre los volúmenes  $V_1/V_2$  de los dos reactores tan grande para el caso en paralelo que prácticamente el segundo reactor no existe, originando que la disposición en paralelo sea más controlable porque se dispone en sí de un solo reactor en donde casi toda el área de transferencia de calor la soporta el primer reactor, este gran volumen de reacción (caso paralelo) permite tolerar las perturbaciones sin que el sistema sufra mayor desestabilización.

Al comparar este fenómeno con la disposición de reactores en serie se observa que tanto los volúmenes como el área de transferencia de calor están distribuidos entre



los dos reactores en una relación  $V_1=3,62V_2$ , a diferencia de la disposición en paralelo en donde  $V_1=8849,35V_2$ .

Desde el punto de vista del proceso químico se tiene que las perturbaciones para el diseño en serie impactarán en mayor magnitud que en paralelo, lo cual es lógico ya que una variación en la temperatura de entrada o en el flujo de entrada de los reactivos modificará más rápidamente y en mayor magnitud bien sea la temperatura del reactor o concentración de algún compuesto en la masa reaccionante menor.

Todo lo expuesto anteriormente tiene un impacto económico originando que la disposición en paralelo (\$ 299.231,88) presente mayor costo total que la disposición en serie (\$ 184.704,61). Véase Tabla 4.2.9.

**Tabla 4.2.9. Volumen de los reactores y área de transferencia de calor diseño integrado método híbrido (SA/SQP)**

Reactor	Volumen del Reactor (ft <sup>3</sup> )		Área de intercambio de calor (ft <sup>2</sup> )	
	Paralelo	Serie	Paralelo	Serie
	Primer reactor	3468,94	870,68	586,47
Segundo Reactor	0,392	240,42	1,57	98,938

**Tabla 4.2.10. Resultados finales de la síntesis y diseño integrado de reactores mezcla completa usando SA**

<b>Arreglo</b>		<b>SERIE</b>
<b>Dimensiones</b>	Primer Reactor	Hr1 (ft)
		4,98
		Dr1 (ft)
		14,92
<b>Parámetros del Controlador</b>	Segundo Reactor	Hr2 (ft)
		3,24
		Dr2 (ft)
		9,72
<b>Parámetros del Controlador</b>	Primer Reactor	$kj1$ (ft <sup>3</sup> /h.R)
		228,40
		$kj2$ (ft <sup>3</sup> /h.R)
		264,23
<b>Controlador</b>	Segundo Reactor	Tauj1 (h)
		0,348
	Reactor	Tauj2 (h)
		0,797

### 4.3 ALGORITMO DIRECT (GBLSOLVE)

#### Implementación de la estructura algorítmica en MATLAB R2012a

La resolución del diseño integrado del sistema objeto de estudio, a través de la función “gblsolve” (GBL), del método de optimización DIRECT, es presentada a continuación.

Este método de optimización se encuentra dentro de la clasificación tanto de los métodos globales como locales, ya que es una modificación del algoritmo Lipschitziano, en el que se realizan búsquedas simultáneas utilizando todas las constantes posibles, y por lo tanto opera tanto a nivel global como local.

A diferencia de los métodos empleados en las secciones precedentes, es un método de optimización determinístico y no estocástico, es decir, que no envuelve ningún elemento aleatorio, por lo que recibe el nombre de método confiable. Es por estas características y por su facilidad de implementación que se evalúa su desempeño para el diseño integrado de reactores mezcla completa.

Aplicando la metodología expuesta en el Capítulo III, se procede con el método DIRECT, usando la programación independiente de “glbolve” de MATLAB R2012a, de aquí en adelante “GBL”, para obtener las condiciones iniciales del problema matemático del diseño integrado.

Se optimiza en estado estacionario por separado y en forma secuencial las variables del diseño de los reactores. Se optimizan también en el estado estacionario las variables de control que se relacionan con los parámetros del controlador.

Posteriormente se realiza una optimización de la función costos sujeta al planteamiento matemático en estado estacionario para cada topología por separado definiendo la programación en MATLAB R2012a, usando GBL.

La definición de la programación en MATLAB R2012a se presenta en el Anexo S: Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en paralelo, usando GBL, (OptEstatPar\_GBL.m) y en el Anexo T: Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie, usando GBL, (OptEstatSer\_GBL.m).

Se establece para el programa del anexo S, OptEstatPar\_GBL.m, el conjunto de funciones que se usan dentro de su bucle de programación.

Para el sistema estudiado se establecen las opciones de programación, de acuerdo a los criterios presentados en el Capítulo II, para GBL. El resto de las opciones requeridas presentan los valores por defecto de MATLAB2012a. A continuación se resumen las opciones más importantes empleadas:

**Tabla 4.3.1. Opciones del GBL para el Diseño Integrado del Sistema**

<b>Opciones</b>	<b>Valor asignado/Función asignada</b>
GL.iterations	70
GL.epsilon	1e-8
GL.tolerance	1.0e-3
GL.NR	2
GL.Nmax	50

**Tabla 4.3.2. Reporte de resultados del programa OptEstatPar\_GBL.m (Anexo S) para el estado estacionario. Topología de reactores en paralelo**

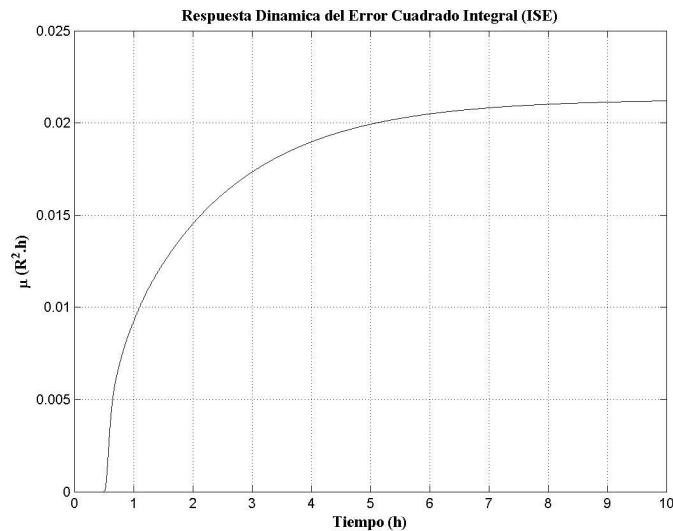
<b>Optimización en estado estacionario</b>	
Costo capital (\$)	326.06,.34
Costo utilidad (\$)	5.361,22
Costo total (\$)	331.424,56
<b>Parámetros óptimos de diseño</b>	
Hr1 (ft)	8.39992288e+000
Hr2 (ft)	2.54981224e+000
Fr1 (ft <sup>3</sup> /h)	9.81481481e-001
Dr1 (ft)	2.40139460e+001
Dr2 (ft)	3.00817473e+000
Fjn1 (ft <sup>3</sup> /h)	4.01638216e+001
Fjn2 (ft <sup>3</sup> /h)	7.36542314e-001

**Tabla 4.3.3. Reporte de resultados del programa OptEstatSer\_GBL.m (Anexo T) para el estado estacionario. Topología de reactores en serie**

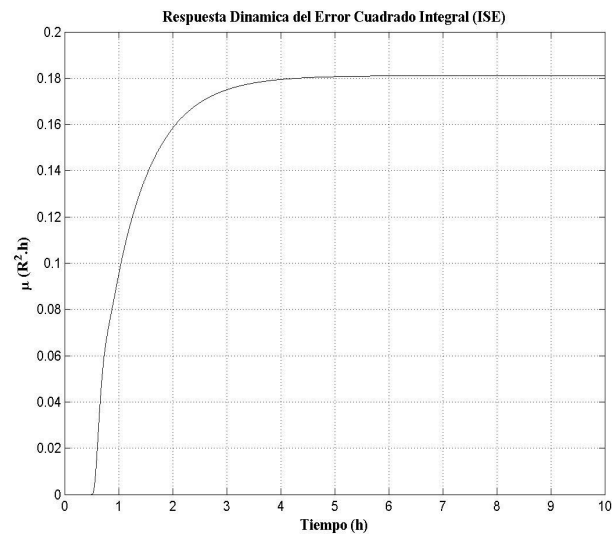
<b>Optimización en estado estacionario</b>	
Costo capital (\$)	192.792,84
Costo utilidad (\$)	5.473,95
Costo total (\$)	198.266,80
<b>Parámetros óptimos de diseño</b>	
Hr1 (ft)	4.53876018e+000
Hr2 (ft)	6.00928633e+000
Fb (ft <sup>3</sup> /h)	7.07808267e+000
Dr1 (ft)	7.07808267e+000
Dr2 (ft)	9.93096411e+000
Fjn1 (ft <sup>3</sup> /h)	3.71336639e+001
Fjn2 (ft <sup>3</sup> /h)	4.62674861e+000

### Determinación del intervalo de estudio para $\varepsilon$ y de los puntos de partida de los parámetros del controlador

La definición de los programas en MATLAB R2012a con el uso de GBL son mostrados en los Anexos U (OnlyControlPar\_GBL.m) y V (OnlyControlSer\_GBL.m) para la optimización mediante GBL de las variables de control en estado estacionario para topología de reactores en paralelo y en serie, respectivamente. Los resultados se muestran en las gráficas siguientes.



**Gráfica 4.3.1. Respuesta dinámica del error cuadrático integral (ISE) para topología en paralelo. GBL**



**Gráfica 4.3.2. Respuesta dinámica del error cuadrático integral (ISE) para topología en serie. GBL**

El comportamiento dinámico del error cuadrático integral (ISE) se muestra en la Gráfica 4.3.1 obtenido de la corrida del programa presentado en el Anexo U, para el menor valor de  $\mu$ . La variación de la controlabilidad es de [0,0001- 0,021193], para reactores en paralelo.

Se reporta en la Gráfica 4.3.2 el comportamiento dinámico del error cuadrático integral (ISE) obtenido de la corrida del programa del Anexo V, para la topología de reactores en serie, obteniéndose el rango [0,0001-0,1811].

La Tabla 4.3.4. muestra los resultados de la optimización con GBL de las variables de control en estado estacionario que se emplearán como condiciones iniciales.

**Tabla 4.3.4. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo. Optimización de control mediante GBL**

<b>Parámetros óptimos de control</b>	
kj1 (ft <sup>3</sup> /h.R)	4.68593479e+002
kj2 (ft <sup>3</sup> /h.R)	6.18853801e+000
Tauj1 (h)	3.79636879e+000
Tauj2 (h)	5.21998270e-001
<b>Resultados estado estacionario</b>	
Fr (ft <sup>3</sup> /h)	9.81481000e+001
Cri1 (lbmol/ft <sup>3</sup> )	2.36325976e-002
Tri1 (R)	6.18634478e+002
Cro1 (lbmol/ft <sup>3</sup> ):	6.17483049e+002
Tro1 (R)	1.85190000e+000
Tjo1 (R)	9.65774884e-002
Cp (lbmol/ft <sup>3</sup> )	6.15883962e+002
Tp (R)	6.15342251e+002
Tjo2 (R)	2.49834641e-002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>	
Cro1 (lbmol/ft <sup>3</sup> )	2.36240483e-002
Tro1 (R)	6.18641592e+002
Tjo1 (R)	6.16327407e+002
Tro2 (R)	5.30002055e+002
Tjo2 (R)	6.14765860e+002
Itro1 (R.h)	3.10370286e-001
Itro2 (R.h)	6.69760874e-002
$\mu$ (R <sup>2</sup> .h)	2.11933043e-002



**Tabla 4.3.5. Reporte de resultados de la programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. Optimización de control mediante GBL**

<b>Parámetros óptimos de control</b>	
kj1 (ft <sup>3</sup> /h.R)	1.02127047e+002
kj2 (ft <sup>3</sup> /h.R)	9.96071706e+002
Tauj1 (h)	1.55123457e+000
Tauj2 (h)	1.01988856e-001
<b>Resultados estado estacionario</b>	
Fr (ft <sup>3</sup> /h)	1.07078083e+002
Cri1 (lbmol/ft <sup>3</sup> )	9.35501855e-001
Tri1 (R)	6.01307937e+002
Cro1 (lbmol/ft <sup>3</sup> )	1.41897235e-001
Tro1 (R)	6.16788439e+002
Tjo1 (R)	6.13327512e+002
Cp (lbmol/ft <sup>3</sup> )	2.42643770e-002
Tp (R)	6.19786621e+002
Tjo2 (R)	6.19328827e+002
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>	
Cro1 (lbmol/ft <sup>3</sup> )	1.41886902e-001
Tro1 (R)	6.16789411e+002
Tjo1 (R)	6.09288928e+002
Cp (lbmol/ft <sup>3</sup> )	2.42610398e-002
Tp (R)	6.19786621e+002
Tjo2 (R)	6.19328850e+002
Itr01 (R.h)	7.19094052e-001
Itr02 (R.h)	-2.45024660e-008
$\mu$ (R <sup>2</sup> .h)	1.81094512e-001

**Tabla 4.3.6. Resumen de los resultados obtenidos para los puntos de partida al Diseño Integrado. Topología de reactores en paralelo. GBL**

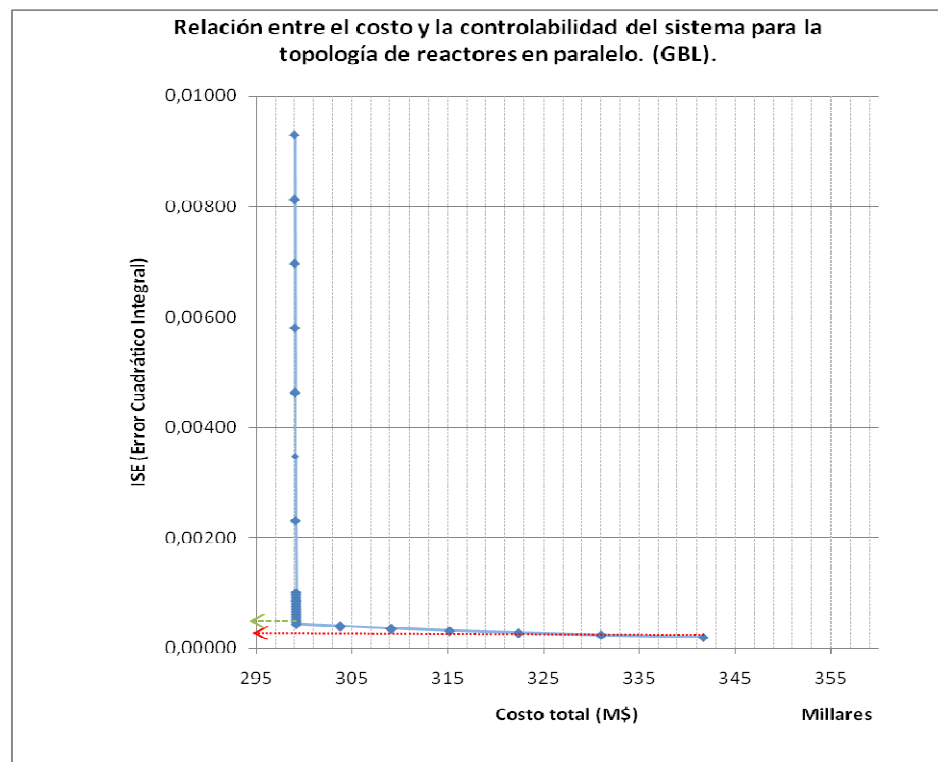
<b>Parámetros de diseño</b>	
Hr1 (ft)	8.39992300e+000
Hr2 (ft)	2.54981200e+000
Fb (ft <sup>3</sup> /h)	9.81481000e-001
Dr1 (ft)	2.40139460e+001
Dr2 (ft)	3.00817500e+000
Fjn1 (ft <sup>3</sup> /h)	4.01638220e+001
Fjn2 (ft <sup>3</sup> /h)	7.36542000e-001
<b>Parámetros de control</b>	
kj1 (ft <sup>3</sup> /h.R)	4.68593479e+002
kj2 (ft <sup>3</sup> /h.R)	6.18853801e+000
Tauj1 (h)	3.79636879e+000
Tauj2 (h)	5.21998270e-001

**Tabla 4.3.7. Resumen de los resultados obtenidos para los puntos de partida al Diseño Integrado. Topología de reactores en serie. GBL**

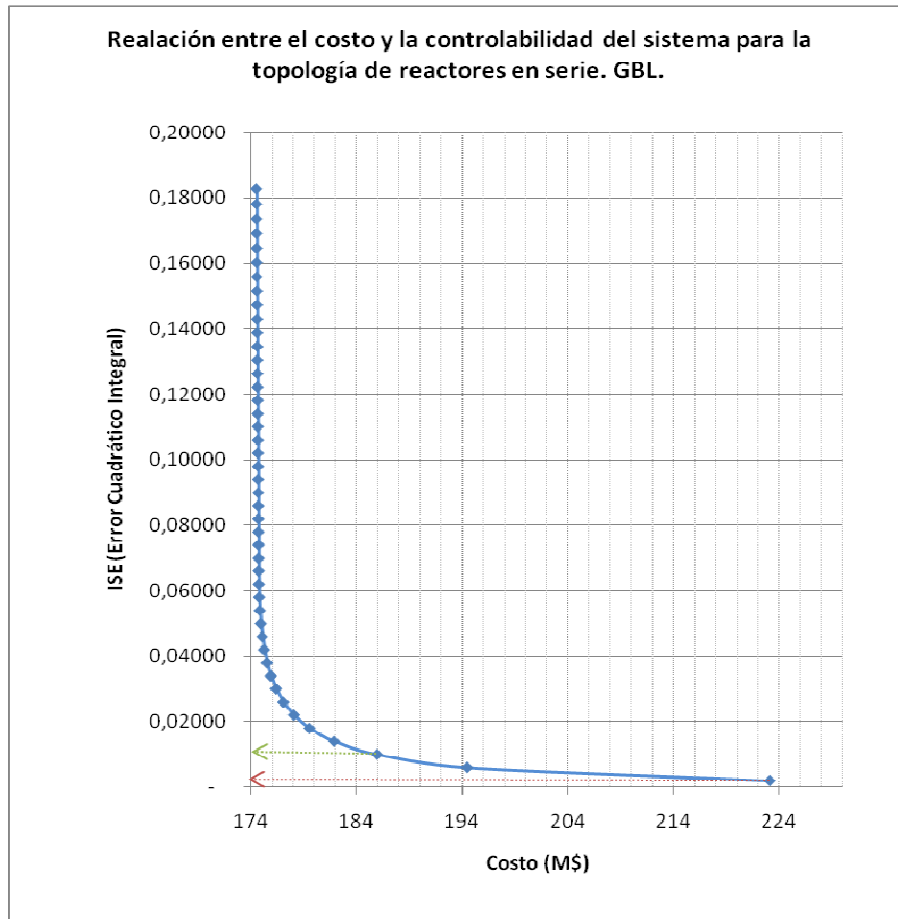
<b>Parámetros de diseño</b>	
Hr1 (ft)	4.53876000e+000
Hr2 (ft)	6.00928600e+000
Fb (ft <sup>3</sup> /h)	7.07808300e+000
Dr1 (ft)	1.30209740e+001
Dr2 (ft)	9.93096400e+000
Fjn1 (ft <sup>3</sup> /h)	3.71336640e+001
Fjn2 (ft <sup>3</sup> /h)	4.62674900e+000
<b>Parámetros de control</b>	
kj1 (ft <sup>3</sup> /h.R)	1.02127047e+002
kj2 (ft <sup>3</sup> /h.R)	9.96071706e+002
Tauj1 (h)	1.55123457e+000
Tauj2 (h)	1.01988856e-001

Se muestran los resultados de la relación entre el costo y la controlabilidad del sistema en las Gráficas 4.3.3 y 4.3.4, que van desde el valor óptimo económico de  $\varepsilon$  hasta el valor óptimo de controlabilidad.

La razón por la que esto se efectúa, como ya se mencionó en la sección anterior, es para encontrar los valores que proporcionen un equilibrio tanto en los objetivos económicos como en los objetivos de control. Se obtienen tanto los parámetros de la planta como los del controlador en forma simultánea.



**Gráfica 4.3.3. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en paralelo. Hibridación GBL y SQP**



**Gráfica 4.3.4. Relación entre el costo y la controlabilidad del sistema para la topología de reactores en serie. Hibridación GBL y SQP**

Los costos totales, en el caso de la topología de reactores en paralelo, se elevan sin observar mejora en la controlabilidad del sistema cuando  $\varepsilon=0,00020$  (ver línea roja punteada, Gráfica 4.3.3). El valor óptimo considerado es de  $\varepsilon=0,00044$  (ver línea verde punteada, Gráfica 4.3.3), que es el punto de inflexión de la curva a partir del cual los costos se incrementan en relación al error cuadrático integral.

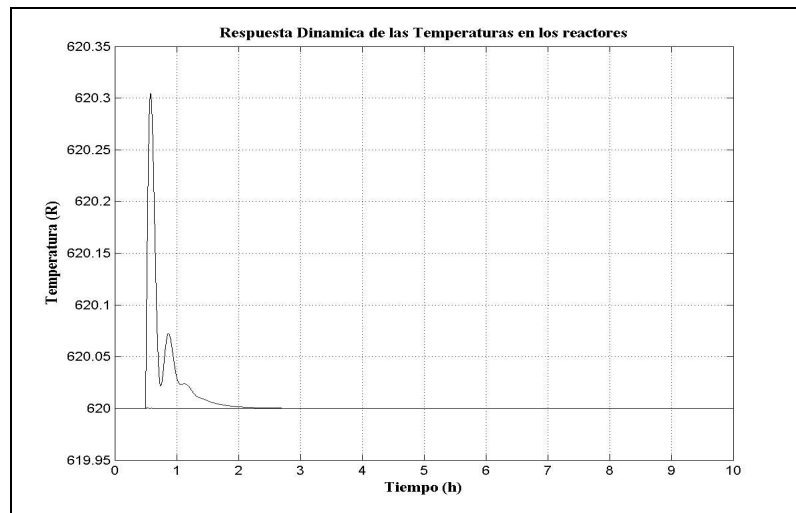
Para el caso en serie los costos totales se incrementan a partir de  $\varepsilon=0,0020$  aproximadamente (ver línea roja punteada, Gráfica 4.3.4), tomándose como óptimo  $\varepsilon = 0,01000$  (ver línea verde punteada, Gráfica 4.3.4).

Como ocurre con el empleo de los métodos anteriores, para el caso de la topología de reactores en paralelo es posible tener un control más riguroso al lograr seleccionar un valor de  $\varepsilon$  más bajo que en el caso de la disposición de reactores en serie, por el hecho de que en el arreglo en paralelo la dimensión de uno de los reactores es despreciable en comparación con el diseño en serie. Véase Tabla 4.3.8. Comparando inicialmente los costos totales según la Tabla 4.3.8, se establece que el arreglo de los reactores en serie es económicamente óptimo, por presentar un diseño más barato.

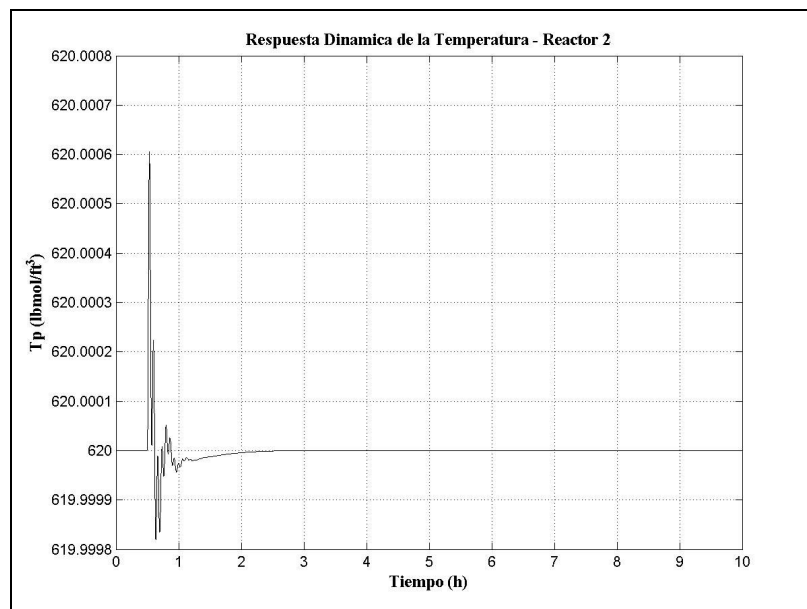
Considerando la controlabilidad del sistema, se deben considerar las siguientes gráficas 4.3.5 y 4.3.6.

**Tabla 4.3.8. Resultados del Diseño Integrado mediante método híbrido (GBL + SQP). Comparación de topologías serie y paralelo**

Variables/Parámetros	Topología	
	Paralelo	Serie
<b>Optimización en estado dinámico</b>		
Costo capital (\$)	293.944,99	180.733,67
Costo utilidad (\$)	5.286,62	5.234,78
Costo total (\$)	299.231,61	185.968,45
<b>Parámetros óptimos de diseño</b>		
Hr1 (ft)	7.88696683e+000	5.03559478e+000
Hr2 (ft)	5.00000000e-001	3.19271134e+000
Fb (ft <sup>3</sup> /h)	0	0
Dr1 (ft)	2.36609005e+001	1.51067843e+001
Dr2 (ft)	1.00000000e+000	9.57813403e+000
Fjn1 (ft <sup>3</sup> /h)	3.93312235e+001	3.64489655e+001
Fjn2 (ft <sup>3</sup> /h)	1.00000000e+000	3.48681669e+000
<b>Parámetros óptimos de control</b>		
kj1 (ft <sup>3</sup> /h.R)	1.16356913e+003	2.39431346e+002
kj2 (ft <sup>3</sup> /h.R)	1.16384324e+002	9.80589209e+002
Tauj1 (h)	3.59528067e-001	3.47210879e-001
Tauj2 (h)	1.26942022e-001	3.63017403e-001
<b>Resultados estado dinámico (al final del periodo de de estudio)</b>		
Cro1 (lbmol/ft <sup>3</sup> )	2.50000865e-002	8.96825893e-002
Tro1 (R)	6.20000000e+002	6,20000011e+002
Tjo1 (R)	6.17484057e+002	6.14098748e+002
Cp (lbmol/ft <sup>3</sup> )	2.50001228e-002	2.50014937e-002
Tp (R)	6.20000000e+002	6,20000010e+002
Tjo2 (R)	5.30000000e+002	6.19326734e+002
Itr1 (R.h)	1.29334675e-002	6.42479079e-002
Itr2 (R.h)	5.10240124e-007	-3.36607100e-010
mu (R <sup>2</sup> .h)	4.40000061e-004	1.00000003e-002



**Gráfica 4.3.5. Respuesta dinámica de la temperatura a la salida de los reactores. Topología de reactores en paralelo. Hibridación GBL/SQP**



**Gráfica 4.3.6. Respuesta dinámica de la temperatura a la salida del segundo reactor. Topología de reactores en serie. Hibridación GBL/SQP**

Como era de esperarse, para el caso de la topología de reactores en paralelo, ocurre el mismo fenómeno usando este método de optimización. Es posible tener un control más riguroso al lograr seleccionar un valor de  $\varepsilon$  más bajo que en el caso de la disposición de reactores en serie, debido a que en el resultado del diseño integrado en paralelo la dimensión de uno de los reactores es prácticamente despreciable. La razón de este comportamiento está expuesta en las secciones precedentes.

Las perturbaciones para el diseño en serie impactan en mayor magnitud que en paralelo, por la variación en la temperatura de entrada o en el flujo de entrada de los reactivos que modifica más rápidamente y en mayor magnitud a la temperatura del reactor y la concentración de algún compuesto en la menor masa reaccionante, respectivamente. Este impacto económico ocasiona que la disposición en paralelo (\$ 299.231,88) presente mayores costos totales que la disposición en serie (\$ 184.704,61). Véase tabla 4.3.9.

**Tabla 4.3.9. Resultado final de la síntesis y diseño integrado de reactores mezcla completa usando GBL**

		Arreglo	SERIE	
<b>Dimensiones</b>	Primer Reactor	Hr1 (ft) Dr1 (ft)	5.03559478e+000 1.51067843e+001	
	Segundo Reactor	Hr2 (ft) Dr2 (ft)	3.19271134e+000 9.57813403e+000	
	<b>Parámetros del Controlador</b>	Primer Reactor	kj1 (ft <sup>3</sup> /h.R)	2.39431346e+002
		Segundo Reactor	kj2 (ft <sup>3</sup> /h.R)	9.80589209e+002
		Tauj1 (h)	3.47210879e-001	



#### **4.4. DESEMPEÑO DE LAS TÉCNICAS DE OPTIMIZACIÓN EN EL DISEÑO INTEGRADO DEL SISTEMA**

El estudio sistemático de la aplicación de diferentes técnicas de optimización al diseño integrado de reactores mezcla completa produce resultados que describen cuantitativamente las técnicas de optimización y su desempeño en el diseño integrado de reactores mezcla completa. Estos resultados sirven para establecer cuantitativamente cual de las dos áreas (clásicos y heurísticos) dentro del campo de optimización presenta mejor desempeño en el diseño integrado de reactores mezcla completa.

Mediante la metodología de trabajo expuesta en el Capítulo III, es posible alcanzar el objetivo principal que se busca en la síntesis y diseño de procesos en la ingeniería química, indistintamente del método de optimización empleado. Se obtienen: las dimensiones físicas óptimas (diámetro y altura) de los reactores mezcla completa para la reacción presentada y los parámetros de entonación del controlador asociados al control de la temperatura del sistema para que se lleve a cabo efectivamente la reacción, obteniendo un diseño óptimo tanto a los criterios económicos como a la controlabilidad del proceso garantizando una operación factible. (Ver Tabla 4.4.1).

**Tabla 4.4.1. Resultados finales de la síntesis y diseño integrado de una red de reactores mezcla completa en serie**

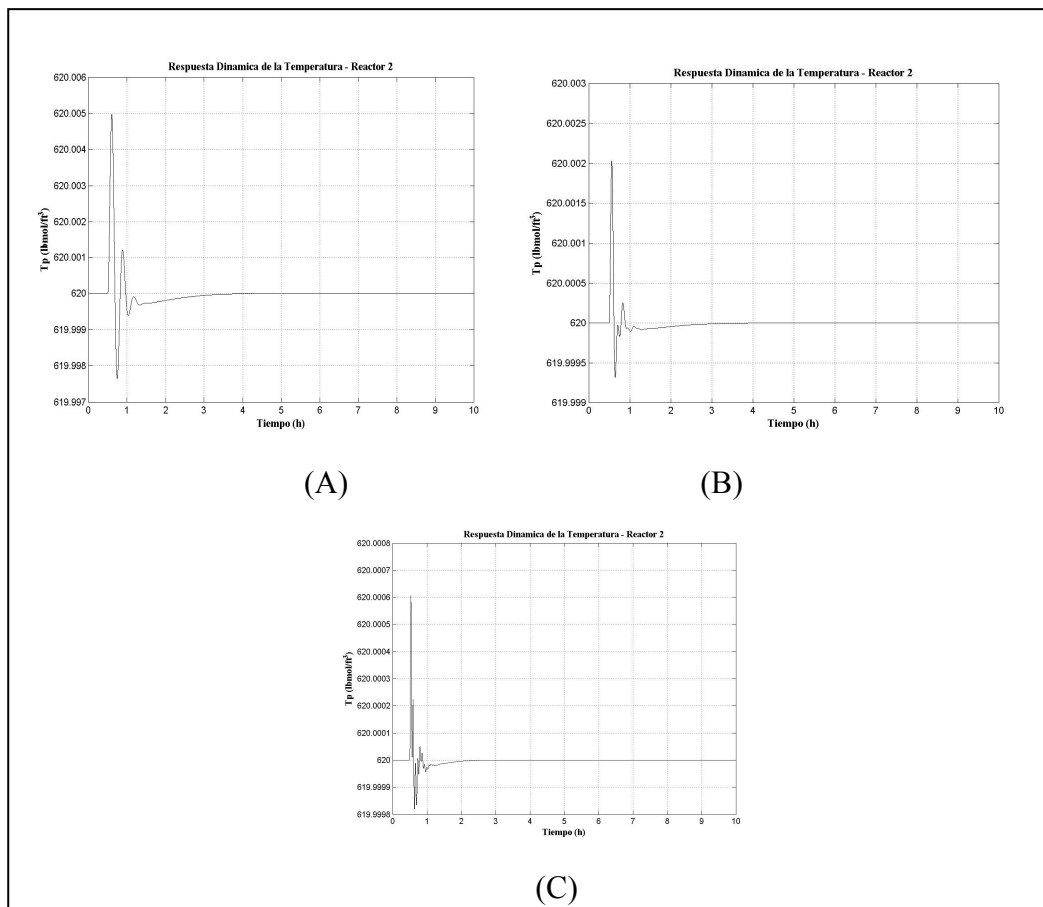
<b>Método de Optimización</b>			<b>AG</b>	<b>SA</b>	<b>GLB</b>
<b>Costos T. (\$)</b>			179.146,3	184.704,61	185.968,45
<b>Dimensiones</b>	Primer	Hr1 (ft)	4,7	4,98	5,04
	Reactor	Dr1 (ft)	14	14,92	15,1
	Segundo	Hr2 (ft)	3,6	3,24	3,19
	Reactor	Dr2 (ft)	10,6	9,72	9,58
<b>Parámetros del</b>	Primer	$kj1$ (ft <sup>3</sup> /h.R)	174,69	228,40	239,43
	Reactor	$kj2$ (ft <sup>3</sup> /h.R)	94,6	264,23	980,589
<b>Controlador</b>	Segundo	Tauj1 (h)	0,357	0,348	0,347
	Reactor	Tauj2 (h)	0,979	0,797	0,363

Esto se debe a que en el algoritmo propuesto y empleado para el diseño integrado del sistema estudiado, se incluyó el uso de métodos híbridos de optimización en los que se empieza la optimización local desde la región donde se encuentra el óptimo global. La metodología propuesta usando métodos globales o heurísticos para obtener las condiciones iniciales del algoritmo, redujo el espacio de búsqueda, acercándose al óptimo global, pudiendo servirse posteriormente de un método local o clásico para alcanzar convergencia.

Es posible analizar de lo antes expuesto, que los distintos métodos estudiados inciden en el diseño integrado del sistema desde la obtención de las condiciones iniciales que alimentan el algoritmo matemático.

Se observa en la Tabla 4.4.1, que usando algoritmos genéticos (AG), el costo total del diseño es menor (179.146,3\$), siguiéndole el uso de recocido simulado (SA), con un costo total de 184.704,61\$ y en último lugar el uso del método DIRECT o GBL (185.968,45\$).

Por lo tanto desde el punto de vista económico resulta factible el empleo de algoritmos genéticos, como método global o heurístico, en el diseño integrado de dos reactores mezcla completa en serie, para las condiciones de reacción establecidas. Desde el punto de vista de controlabilidad se puede observar la siguiente gráfica comparativa.



**Gráfica 4.4.1. Respuesta dinámica de la temperatura a la salida del segundo reactor. Diseño integrado para el arreglo de reactores en serie. (A)algoritmos genéticos, (B)recocido simulado, (C)GBL.**

Desde el punto de vista de controlabilidad se observa que el tiempo de estabilización de la temperatura en 620 R, a la salida del segundo reactor, no varía sustancialmente entre los métodos de optimización.

Desde el punto de vista del costo computacional asociado, los métodos estudiados se pueden comparar en el número de iteraciones o de evaluaciones y la facilidad de programación o número de opciones a definir en el programa. Estos aspectos también pueden influir en la elección de alguno de los métodos de optimización.

De manera cuantitativa, se aprecia en la siguiente tabla que los aspectos mencionados favorecen el uso de recocido simulado en la metodología de resolución del problema, ya que tanto en complejidad de programación como en número de evaluaciones, los valores son menores que para los otros dos casos.

De manera contradictoria el costo computacional del algoritmo genético es mayor porque se requiere establecer más opciones a la hora de programar, mientras que GLB se ve eclipsado por el elevado número de iteraciones que resultaría en un tiempo de obtención de resultados mayor. El resultado entra en concordancia con lo establecido por Bjorkman y Holmström (2012), quienes establecen que una de las limitaciones del algoritmo DIRECT es que requiere, relativamente, gran cantidad de evaluaciones de la función objetivo, en particular cuando la solución óptima se encuentra cerca de la frontera del espacio de búsqueda. Esto es debido a que le da mayor énfasis a la búsqueda global.

**Tabla 4.4.2. Costo computacional asociado a los métodos estudiados**

<b>Método</b>	<b>AG</b>	<b>SA</b>	<b>GBL</b>
Número de Iteraciones	12.100	10.643	42.941
Opciones de programación (número)	30	6	23

Es importante destacar que desde el punto de vista de diseño y síntesis de procesos la metodología propuesta y estudiada a través del uso de algoritmos genéticos es la alternativa más viable. Cuando es necesario tomar en cuenta el costo computacional, usando recocido simulado se obtiene un equilibrio, ya que es óptimo desde este punto de vista y no difiere sustancialmente en los costos totales obtenidos por diseño y síntesis de procesos, respecto de los otros métodos.

Por ello, queda desde la perspectiva del lector, ingeniero o ejecutor del diseño a un sistema real, seleccionar la alternativa que más se ajuste a sus necesidades, resultando el presente trabajo una contribución al diseño integrado de reactores mezcla completa.

# CAPÍTULO V

## CONCLUSIONES Y RECOMENDACIONES

## CAPÍTULO V

### CONCLUSIONES

- Los aspectos técnicos que describen el proceso están dados por una estructura de dos reactores mezcla completa con chaquetas de enfriamiento de fluido perfectamente mezclado, a través de una reacción de primer orden elemental irreversible y exotérmica. El control del sistema reactivo se ejerce por la temperatura del mismo, manipulando el flujo de refrigerante a través de las chaquetas de enfriamiento. La optimización del diseño del sistema está basada en la minimización del costo total de diseño y del error cuadrático integral para el control.
- La modelación matemática consiste en un modelo dinámico determinado por los balances de materia y energía en la disposición en serie y paralelo de la red de reactores. Se emplean restricciones lógicas de dimensión de los equipos, restricciones de controlabilidad, ley de control proporcional-integral y el conjunto de perturbaciones.
- Las técnicas de optimización aplicadas al diseño integrado de reactores mezcla completa, involucran el uso de algoritmos genéticos (AG), Recocido Simulado (SA) y Algoritmo DIRECT (GBL), cada uno en asociación con el método del gradiente.

- La primera estrategia metodológica aplicada al diseño integrado de reactores mezcla completa, comprende el uso de la función “ga” de algoritmos genéticos (método heurístico de optimización). La segunda estrategia metodológica involucra el uso de la función “SIMULANNEALBND” de recocido simulado, que pertenece también a los métodos heurísticos de optimización. En la tercera estrategia se usa la función “gblSolve” del Algoritmo DIRECT, con la operación simultánea de métodos locales y globales de optimización. En las tres estrategias se emplea la función “fmincon”, en representación de los métodos clásicos o estocásticos. Todas estas funciones pertenecen a MATLAB R2012a.
- Del estudio sistemático de las tres estrategias metodológicas se obtiene que la mejor solución es la hibridación de los métodos clásicos y heurísticos. Esto es debido a la exploración exhaustiva el espacio de soluciones y la garantía de rápida convergencia del algoritmo de minimización de los costos totales de la síntesis, diseño, control y operación del proceso.
- La disposición de los reactores en serie es la óptima, por presentar un diseño más económico en todos los casos estudiados, que el arreglo de los reactores en paralelo. El resultado del arreglo en paralelo no es viable técnicamente porque se lleva a cabo toda la reacción en un solo reactor. Se genera una inversión adicional por ese equipo que eleva los costos totales. El arreglo en serie presenta un mejor desempeño que el diseño en paralelo, con un tiempo de asentamiento de 4 y el último de 6 horas, así como menor sobre impulso. En la topología de reactores en serie tanto los volúmenes como el área de transferencia de calor quedan distribuidos entre los dos reactores en una relación  $V_2=0,43V_1$ .



- De el punto de vista económico la técnica de optimización de mejor desempeño es el algoritmo genético (AG), con el menor costo total de diseño (\$ 179.146,3), siguiéndole el uso de recocido simulado (SA), con un costo total de \$ 184.704,61 y en último lugar el uso del método DIRECT o GBL (\$ 185.968,45).
- Desde el punto de vista de controlabilidad se observa que el tiempo de estabilización de la temperatura del sistema en 620 R, para el arreglo en serie, no varía sustancialmente entre los diferentes métodos de optimización estudiados para el diseño integrado de reactores mezcla completa.
- Desde el punto de vista del costo computacional asociado al diseño integrado de reactores mezcla completa, es más favorable el uso del recocido simulado (SA) en la metodología de resolución del problema, por su menor complejidad de programación y menor número de evaluaciones de la función.
- Para el sistema estudiado los métodos heurísticos exhiben su mejor propiedad al aproximarse lo más cercanamente posible a la solución óptima hallando las condiciones iniciales del problema NLP-DAE y la dirección de movimiento proporcionada por el método del gradiente garantiza la mejora en el objetivo acelerando la convergencia del algoritmo hacia el óptimo global.
- En función a los resultados obtenidos, el presente trabajo representa una contribución al diseño integrado de reactores mezcla completa, por lo que

el diseñador de un sistema real, deberá seleccionar la alternativa que más se ajuste a sus necesidades.

### **RECOMENDACIONES**

- Explorar nuevas estrategias metodológicas que involucren diferentes métodos clásicos de optimización.
- Ampliar el diseño integrado a redes de reactores más complejas, tanto en número, como en tipo de reactores.
- Aplicar la metodología desarrollada del diseño integrado de reactores mezcla completa en distintas condiciones de reacción.
- Se recomienda utilizar otras herramientas computacionales mas especializadas en el área de optimización, como el GAMS y combinarlas con paquetes para la resolución de ecuaciones diferenciales.
- Aplicar la metodología del diseño integrado para la construcción a escala piloto de un proceso reactivo similar al caso de estudio, para validar los resultados con datos reales.

## REFERENCIAS BIBLIOGRÁFICAS

- ANDERSON, J. (1996). **A Practical Problem in Dynamic Heat Transfer**". Chem. Engineer, CE-97-CE103.
- ALANDER, J. (1992). **On optimal population size of genetic algorithms. Proceedings.** Comp Euro Computer Systems and Software Engineering, 6th Annual European Computer Conference, pp. 65-70.
- ARIAS, F. (2000). **El Proyecto de Investigación. La metodología científica.** Caracas. Editorial Episteme, pp. 19-24.
- BARNHART, J. y otros (1998). **Branch and Price: Column generation for solving huge integer programs.** Operations Research. pp. 316-329.
- BJORKMAN, M. y Holmström, K. (2012). **Global Optimization Using the DIRECT Algorithm in Matlab 1.** Center for Mathematical Modeling Department of Mathematics and Physics Mälardalen University. Sweden.
- BORCHES, R. y Michell, C. (1994). **An improved branch and bound algorithm for mixed integer non lineal programming.** USA. Revista "Mathematical Programming". 359-366.
- BRENGEL D. y Sieder W. (1992). **"Coordinated design and control optimization of nonlinear processes"**. Comp. Chem. Eng. 16: 861-886.
- BRISTOL, E., (1966). **On a new measure of interaction for multivariable process control. Automatic control.** London: Imperial College Press.
- BRITO, A. y otros. (2003). **"Síntesis y Diseño Integrado de un Sistema de Reactores en serie utilizando Algoritmos Genéticos"**. Caracas, Venezuela. Universidad Simón Bolívar.
- CONN, A. y otros (2000). **Trust region methods.** Philadelphia. USA. Revista "Series on optimization".

- DAVIS, L. (1985). **Applying adaptive algorithms to epistatic domains**. New York. Proceedings of the International Joint Conference on Artificial intelligence, 162-164.
- DARWIN, C. (1859). **On the Origin of Species by Means of Natural Selection**. Murray, Londres.
- DE JONG, K. (1975) . **An analysis of the behavior of a class of genetic adaptive systems**. Tesis doctoral. USA.University of Michigan.
- DOUGLAS, F. (1988). **Conceptual Design of Chemical Processes**. USA. McGraw Hill.
- ELLIOT, T. y Luyben, W. (1995). **Capacity based approach for the quantitative assessment of process controllability during the conceptual design stage**. Ind. Eng. Chem. Res. 34; 3907-3915.
- ELHOSSINI, A. y otros. (2010). **Strength Pareto Particle Swarm Optimization and Hybrid EA-PSO for Multi-Objective Optimization**. School of Engineering, University of Guelph, Guelph, ON, N1G 2W1. Canada. Massachusetts Institute of Technology.
- FLETCHER, R. (1980). **Practical Methods of Optimization, Vol. 1, Unconstrained Optimization**, John Wiley and Sons, 1980.
- Fondo Editorial de la Universidad Pedagógica Experimental Libertador, (2005), **Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales**. Caracas.
- GEORGIU, A. y Floudas C., (1989). **Simultaneous Process Synthesis and Control: Minimization of Disturbance Propagation in Heat Recovery Systems**. In: Foundations of Computer Aided Process.
- GOLDBERG, D. (1989). **Genetic algorithms in search, optimization, and machine learning**, New York. Addison-Wesley.
- GROSSMANN, I. (1985). **Mixed Integer programming Approach for the Synthesis of Integrated Process Flowsheets**, USA. Comp and Chem. Engng., 9, 463-482

- GUTIÉRREZ A. y otros (1998). **Optimización con recocido simulado para el problema de conjunto independiente**. Universidad Autónoma Metropolitana. México.
- GUTIÉRREZ, G., de Prada, C. (2003), “**Diseño Integrado y síntesis de procesos químicos**”, España, Departamento de Ingeniería de Sistemas y Automática. Facultad de Ciencias. Universidad de Valladolid.
- HANG, S. (1976). **Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems**. USA. Revista “Mathematical Programming”. pp 263-288.
- HERNÁNDEZ, G. (2006). **Métodos Clásicos de Optimización para Problemas No-Lineales sin Restricciones**. Departamento de Ingeniería Matemática. Universidad de Chile. Chile.
- HOLLAND J. (1975). **Adaptation in Natural and Artificial Systems**, USA. University of Michigan Press, Ann Arbor.
- HOLT, B. (1984). **The Assesment of dynamic resiliency: The effect of nonminimum Phase element**, Doctoral Dissertation, University of Wisconsin. Chemical Engineering Plants.
- HONGFENG, X. y otros. (2009). **Large Scale Function Optimization or High-Dimension Function Optimization in Large Using Simplex-based Genetic Algorithm**. School of Information Science and Engineering, Central South University, Shangai China.
- JACOBSEN, E., (1991). **Modelling and Identification for Robust Control of III- Conditioned Plants – a Distillation Case Study**. American Control Conference. IEEE. Pages 242-248. Boston, MA, USA.
- JONES, D. y otros. (1993). **Lipschitzian optimization without the Lipschitz constant**. Journal of Optimization Theory and Applications archive. Volume 79 Issue 1, Oct. 1993. Pages 157-181. Plenum Press New York, NY, USA.
- KIRKPATRICK, S. y otros (1983). **Optimization by simulated annealing**, Revista “Science”, Vol. 220, No. 4598, pp. 671-680.

- LARRAÑAGA P. y otros (1999). **Algoritmos Genéticos**. Departamento de Ciencias de la Computación e Inteligencia Artificial Universidad del País Vasco Euskal Herriko Unibertsitatea.
- LENHOF, A., Morari, M., (1982), **Design of resilient processing plants I. Process design under consideration of dynamic aspects**. Chemical Engineering Science, 37(2), 245-258.
- LEVENSPIEL, O. (1998). **Ingeniería de las reacciones químicas**. USA. Tercera Edición, Editorial Reverté.
- LEYFER S. (2001). **Integrating SQP and Branch and Bound for Mixed Integer non Linear Programming**. USA. Revista "Computational Optimization and Application. pp 295-309.
- LINNHOF, B., y Kotjabasakis, E. (1986). **Downstream path for operable process design**. Chem. Eng. Progress, 82, 23-28.
- LINARES, P. y otros (2001). **Modelos Matemáticos de Optimización**. España. Universidad Pontificia Comillas Madrid. Escuela Técnica Superior de Ingeniería. Departamento de organización Industrial.
- LIU, C. y otros. (2009). **Image based Reconstruction using Hybrid Optimization of Simulated Annealing and Genetic Algorithm**. Shanghai University.
- LUYBEN, W. (1993). **"Trade offs between design and control in chemical reactor systems"**. Department of Chemical Engineering, Iacocca Hall, Lenhigh University, USA.
- LUYBEN, W. y Floudas, C. (1994). **Analyzing the Interaction of Design and Control**, USA. Comp. Chem. Eng. Vol. 18, Nro. 10, 933-969.
- MARSELLE, D. (1982). **Design of resilient processing plants II. Design and control of energy management systems**. Chem. Eng. Sci. 37, 259-270.
- MARTINEZ, T. y Behar, A. (2011). **Estrategia Híbrida AG-SIMPLEX para la identificación experimental de sistemas**. Departamento de Control Automático, Instituto de Cibernética, Matemática y Física (ICIMAF) CIE. México.
- MICHALEWICZ, Z. (1992). **Genetic Algorithms + Data Structures = Evolution Programs**. Springer-Verlag, Berlin Heidelberg.

- MORARI, M. (1983). **Design of resilient processing plants III. A general framework for the assessment of dynamic resilience.** Ind. Chem. Eng. Sci. 38: 1881-1891.
- MORARI, M. (1992). **Effect of design on the controllability of chemical plants.** Pergamon Press. Oxford, RU, pp. 3-16
- MORARI y Perkins (1994).”**Design for Operations**”. USA. FODCAPD Proceedings.
- MORARI, M. y Zafiriou, E. (1989). **Robust Process Control.** Prentice-Hall, Englewood Cliffs, NJ, USA. 488 pp.
- NARRAWAY, L., y Perkins J. (1993). **Selection of process control structure based on linear dynamic economics.** Ind. Eng. Chem. Res. Vol. 32, 2681-2692.
- NEMHAUSER , G. y Wolsey L. (1998). **Integer and Combinatorial Optimization.** New York Wiley Interscience.
- NISHIDA N., e Ichikawa, A. (1975). **Synthesis of Optimal Dynamic Process System by a Gradient Method.** Ind. Eng. Chem. Process. 13: 236-242.
- NISHIDA N., e Ichikawa, A. (1976). **Studies in Chemical Process Design and Synthesis II. Optimal Synthesis of Dynamic Process Systems with Uncertainty.** AIChE J. 22: 539-549.
- NISHIDA N. y otros (1981). “**A Review of Process Synthesis**”, *AIChE J.*, 27, pp. 321.
- PERKINS, J. (1989). **Interaction between Process Design and Process Control.** Prep. IFAC Symp. On Dynamics and Control of Chemical.
- PÉREZ, J. y Basterrechea, J. (2005). **Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo.** Universidad de Cantabria. Santander. Tesis Doctoral presentada en la Universidad de Cantabria para la obtención del título de Doctor.
- PÉREZ, G. y otros, J. (2007). **Hibridación de métodos exactos y heurísticos para el problema multiobjetivo.** Universidad de Málaga.

- PETCU, A. y Faltings, B. (2007). **A Hybrid of Inference and Local Search for Distributed Combinatorial Optimization**. EPFL, Switzerland.
- PIRONA, K (2009) **Diseño integrado de una red de reactores mezcla completa mediante algoritmos genéticos**. Universidad Nacional Experimental Politécnica “Antonio J. de Sucre”. Barquisimeto. Venezuela.
- POWELL, M. (1978). **A fast algorithm for nonlinear constrained optimization Calculation**. Springle Verlag. Berlin. Revista “In Numerical Analysis.
- PSARRIS, P., y Floudas, C., (1991). **Dynamic Operability of mimo systems with time delays and transmissions zeros, 1. Assesment**. Chemical Engineering Science, 46(10), 2691-2707.
- REEVES, C. (1993). **Modern Heuristic Techniques for Combinatorial Problems**, Blackwell Scientific Publications.
- ROWE y col. , (2002). **Integrated Design of responsive chemical manufacturing facilities**. Chemical Egeineering Science.
- SCENNA, N. (1999). **Modelado, Simulación y Optimización de Procesos Químicos**.
- SCHWEIGER C., Floudas C.,(1997), “**Interaction of Desing and Control**” (Optimization with Dynamic Models), Princeton, USA, W.W. Hager and P.M. Pardalos, Editors.
- SEGURA, C. (2007). **Diseño Integrado de Reactores Mezcla Completa**. Universidad Nacional Experimental Politécnica “Antonio J. de Sucre”. Barquisimeto. Venezuela.
- SHEFFIELD, R. (1992), **Integrate process and control system design**. Ind. Chem. Eng. Prog. 88: 30-35.
- SKOGESTAD, S., (1991), **Consistency of steady-state models using insight about extensive variables**. Ind. Eng. Chem. Res. 30: 654-661.
- SKOGESTAD, S., Morari, M., (1987b), **Design of Resilient Processing plants IX. Effect of Model Uncertainty on Dynamic Resilience**. Chem. Eng. Sci, 42: 1765-1780.
- SKOGESTAD, S., Postlethwaite, I., (1997), “**Multivariable Feedback Control, Analysis and Design**”, John Wiley & Sons.



- SKOGESTAD, S., Wolff, E., (1992), **Controllability Measures for Disturbance Rejection**. Preprints IFAC-workshop on interactions between design and control. Edited by Perkins S, Pergamon Press. London. 127-132.
- SMITH C., Corripio, A., (1991), “**Control Automático de Procesos**”, (Teoría y Práctica), México, Editorial LIMUSA S. A, Primera edición, Traductor: Manzanares, S.
- STUBBS R. y Mehrotra S. (1999). **A branch and cut method for 0-1 mixed convex programming**. USA: Revista “Mathematical Programming”. pp. 515-532.
- TAKAMA N. y Umeda T. (1980) “**Multi-level, Multi-objective Optimization in Process Engineering**”, Chem. Engng. Sci., 36, pp. 129-136.
- UMEDA, T. (1982). “**Computer Aided Process Synthesis**”, Proceedings of Process Systems Engineering Symposium, p. 79, Kyoto, Japón.
- UMEDA, T. y Kuriyama, T.(1978). “**A logical Structure for Process Control Synthesis**”, Proc. Of IFAC Congress, Helsinki, Finlandia.
- VIZCAÍNO, G., (2006). **Automatización del proceso de diseño de un circuito electrónico con la herramienta MATLAB y el simulador de circuitos SPECTRE**. Escuela Superior de Ingenieros Universidad de Sevilla. España.
- WALSH y Perkins (1996) “**Operability and Control and Process Synthesis and Design**”.USA. In John I. Anderson, editor, Advances in Chemical engineering, vol 23, 301-402, Academic Press.
- Find global minima for bounded nonlinear problems. MathWorks** © 1994-2014 The MathWorks, Inc. Disponible: <http://www.mathworks.com/discovery/simulated-annealing.html>.
- ZAPATA, J. (2012). **Métodos de optimización sin restricciones**. Instituto Universitario Politécnico "Santiago Mariño" (IUPSM), Extensión Maracay, Estado Aragua. Venezuela.
- ZHOU, Q. y otros. (2009). **A Hybrid Optimization Algorithm for the Job-shop Scheduling Problem**. Department of Computer Science and Technology Chuzhou University. Chuzhou, China.

## ANEXOS

**Tabla 5.1. Identificación de los Anexos**

Identificación del Anexo	Nombre del Anexo	Códigos de programación	Función en la programación
<b>A</b>	Programación en MATLAB R2012a para calcular la función costo	Costo.m	Calcula la función costo
<b>B</b>	Programación en MATLAB R2012a para la simulación del sistema de reactores en estado Dinámico paralelo	DynReacPar.m	Hace la simulación del sistema de reactores en estado Dinámico paralelo
<b>C</b>	Programación en MATLAB R2012a para la simulación del sistema de reactores en estado Dinámico serie	DynReacSer.m	Hace la simulación del sistema de reactores en estado Dinámico serie
<b>D</b>	Programación en MATLAB R2012a para la simulación del sistema de reactores en estado Estacionario paralelo	StatReacPar.m	Hace la simulación del sistema de reactores en estado Estacionario paralelo
<b>E</b>	Programación en MATLAB R2012a para la simulación del sistema de reactores en estado Estacionario serie	StatReacSer.m	Hace la simulación del sistema de reactores en estado Estacionario serie
<b>F</b>	Programación en MATLAB R2012a para la Función de perturbación	Ti.m	Función de perturbación
<b>G</b>	Programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo Optimización de control mediante Algoritmos Genéticos	OnlyControlPar_AG.m	Optimización de parámetros de control, topología de reactores en paralelo
<b>H</b>	Programación en MATLAB R2012a para el cálculo de las variables de control en estado	OnlyControlSer_AG.m	Optimización de parámetros de control, topología de reactores en serie

	estacionario para topología de reactores en serie. Optimización de control mediante Algoritmos Genéticos		
<b>I</b>	Programación en MATLAB R2012a para el Diseño Integrado. Topología de reactores en paralelo	OptDynPar.m	Optimización integrada diseño + control, topología de reactores en paralelo
<b>J</b>	Programación en MATLAB R2012a para el Diseño Integrado. Topología de reactores en serie	OptDynSer.m	Optimización integrada diseño + control, topología de reactores en serie
<b>K</b>	Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en paralelo, AG	OptEstatPar_AG.m	Optimización de diseño, topología de reactores en paralelo
<b>L</b>	Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie, AG	OptEstatSer_AG.m	Optimización de diseño, topología de reactores en serie
<b>M</b>	Cálculos Típicos del Balance de masa y energía de la de dos reactores mezcla completa en paralelo	-----	-----
<b>N</b>	Cálculos Típicos del Balance de masa y energía de la de dos reactores mezcla completa en serie	-----	-----
<b>O</b>	Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en paralelo, SA	OptEstatPar_SA.m	Optimización de diseño, topología de reactores en paralelo
<b>P</b>	Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie, SA	OptEstatSer_SA.m	Optimización de diseño, topología de reactores en serie
<b>Q</b>	Programación en MATLAB	OnlyControlPar_SA.m	Optimización de parámetros

	R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo Optimización de control mediante SA		de control, topología de reactores en paralelo
<b>R</b>	Programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. Optimización de control mediante SA	OnlyControlSer_SA.m	Optimización de parámetros de control, topología de reactores en serie
<b>S</b>	Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en paralelo, GLB	OptEstatPar_GLB.m	Optimización de diseño, topología de reactores en paralelo
<b>T</b>	Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie, GLB	OptEstatSer_GLB.m	Optimización de diseño, topología de reactores en serie
<b>U</b>	Programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo Optimización de control mediante GBL	OnlyControlPar_GBL.m	Optimización de parámetros de control, topología de reactores en paralelo
<b>V</b>	Programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. Optimización de control mediante GBL	OnlyControlSer_GBL.m	Optimización de parámetros de control, topología de reactores en serie



## ANEXO B. Programación en MATLAB R2012a para la simulación del sistema de reactores en estado dinámico paralelo

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% BALANCE DE MASA Y ENERGIA - ESTADO DINAMICO
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [C, Ceq] = DynReac(v,pe)
    global E I2 D1
    % EVITA RESOLVER PARA VALORES NEGATIVOS DE DISEÑO
    v = abs(v);
    % RESUELVE EL SISTEMA EN ESTADO ESTACIONARIO
    Ce = StatReac(v, pe);
    % CONDICIONES INICIALES DEL SISTEMA DE ECUACIONES DIFERENCIALES
    Ce1 = [Ce(2:4), Ce(6:8), zeros(1,3)];
    vv = [v(1:7), Ce1(2), Ce1(5), v(8:11)];
    if I2 == 'S'
        Te = 0:0.001:5;
    else
        Te = 0:0.001:50;
    end
    opt = odeset('RelTol', 1e-12, 'AbsTol', 1e-12, 'NormControl',
    'on');
    % RESUELVE EL SISTEMA DE ECUACIONES DIFERENCIALES
    [T,Ct] = ode23(@dSeg,Te,Ce1,opt,vv,pe);
    Ct(:,10)=v(3).*Ct(:,1)+(1-v(3)).*Ct(:,4);
    Ct(:,11)=v(3).*Ct(:,2)+(1-v(3)).*Ct(:,5);
    N = length(T); % POSICIÓN DEL ÚLTIMO VALOR EN LA MATRIZ DE
DATOS
    Cp = Ct(N,10); % CONCENTRACIÓN DE SALIDA
    ISE = Ct(N,9); % CONTROLABILIDAD

    % CÁLCULO DEL FLUJO MÁXIMO DE REFRIGERANTE UTILIZADO
    Fj1 = vv(6) + vv(10).*(Ct(:,2)-vv(8))+vv(10).*Ct(:,7)./vv(12);
    Fj2 = vv(7) + vv(11).*(Ct(:,5)-vv(9))+vv(11).*Ct(:,8)./vv(13);
    F1max = max(Fj1);
    F2max = max(Fj2);
    F1min = min(Fj1);
    F2min = min(Fj2);
    if I2 == 'S'
    % SALIDAS DEL SISTEMA PARA LA OPTIMIZACIÓN DINAMICA
        C = [1000.*(Cp-0.025),...
            Ce1(2)-620, Ce1(5)-620, ISE-1000.*E,...
            0.30.*F1max - vv(6), 0.30.*F2max-vv(7),...
            -F1min, -F2min];
        Ceq = [];
    else

```

```

% SALIDAS DEL SISTEMA PARA LA SIMULACIÓN Y REPORTE
Ct(:,9) = Ct(:,9)/1000;
if I2 == 'Sim'
    C = Ct;
    Ceq = T;
else
    C = Ct(N,:);
    Ceq = [];
end
% GENERACIÓN DE GRAFICOS
close all
% GRÁFICOS SENCILLOS
graficar(T,Ct(:,1),'Respuesta Dinamica de la Concentración -
Reactor 1',...
    'Tiempo (h)', 'Cro1 (lbmol/ft^3)',1,0,0)
graficar(T,Ct(:,4),'Respuesta Dinamica de la Concentración -
Reactor 2',...
    'Tiempo (h)', 'Cro2 (lbmol/ft^3)',2,0,0)
graficar(T,Ct(:,2),'Respuesta Dinamica de la Temperatura -
Reactor 1',...
    'Tiempo (h)', 'Tro1 (R)',3,0,0)
graficar(T,Ct(:,5),'Respuesta Dinamica de la Temperatura -
Reactor 2',...
    'Tiempo (h)', 'Tro2 (lbmol/ft^3)',4,0,0)
graficar(T,Ct(:,3),'Respuesta Dinamica de la Temperatura -
Chaqueta 1',...
    'Tiempo (h)', 'Tjo1 (R)',5,0,0)
graficar(T,Ct(:,6),'Respuesta Dinamica de la Temperatura -
Chaqueta 2',...
    'Tiempo (h)', 'Tjo2 (lbmol/ft^3)',6,0,0)
graficar(T,Ct(:,7),'Respuesta Dinamica del Termino Integral
para el Controlador PI - Reactor 1',...
    'Tiempo (h)', 'Itro1 (R.h)',7,0,0)
graficar(T,Ct(:,8),'Respuesta Dinamica del Termino Integral
para el Controlador PI - Reactor 2',...
    'Tiempo (h)', 'Itro2 (R.h)',8,0,0)
graficar(T,Ct(:,9),'Respuesta Dinamica del Error Cuadrado
Integral (ISE)',...
    'Tiempo (h)', '\mu (R^{2}.h)',9,0,0)
graficar(T,Ct(:,10),'Respuesta Dinamica de la Concentración -
Final',...
    'Tiempo (h)', 'Cp (lbmol/ft^3)',10,0,0)
graficar(T,Ct(:,11),'Respuesta Dinamica de la Temperatura -
Final',...
    'Tiempo (h)', 'Tp (R)',11,0,0);
% DATOS APAREADOS
graficar(T,[Ct(:,1),Ct(:,4)],'Respuesta Dinamica de las
Concentraciones',...
    'Tiempo (h)', 'Concentracion
(lbmol/ft^3)',12,'Cro1','Cro2')
graficar(T,[Ct(:,2),Ct(:,5)],'Respuesta Dinamica de las
Temperaturas en los reactores',...

```

```

        'Tiempo (h)', 'Temperatura (R)',13,'Trol','Tp')
    graficar(T,[Ct(:,3),Ct(:,6)],'Respuesta Dinamica de las
Temperaturas en las chaquetas',...
        'Tiempo (h)', 'Temperatura (R)',14,'Tjo1','Tjo2')
    graficar(T,[Ct(:,7),Ct(:,8)],'Respuesta Dinamica del Termino
Integral para el Controlador PI',...
        'Tiempo (h)', 'Termino Integral
(R.h)',15,'Itrol','Itro2')
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIN
DE DYNREAC

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%           FUNCION AUXILIAR PARA FORMATO DE GRAFICAS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function graficar (XX,YY,S1,S2,S3,B,AA, BB)
% FORMATEA LA PRESENTACIÓN DE LOS DATOS GRAFICADOS
global D1 I2
if B >= 12
    plot(XX,YY(:,1),'k-',XX,YY(:,2),'b--')
    ss1 = ['\fontname{Times New Roman}\fontsize{12}\bf{' AA '}''];
    ss2 = ['\fontname{Times New Roman}\fontsize{12}\bf{' BB '}''];
    legend(ss1,ss2,'location','SouthEast')
else
    plot(XX,YY,'k-')
end
grid on
title(['\fontname{Times New Roman}\fontsize{12}\bf{' S1 '}''])
xlabel(['\fontname{Times New Roman}\fontsize{12}\bf{' S2 '}''])
ylabel(['\fontname{Times New Roman}\fontsize{12}\bf{' S3 '}''])
set(gcf,'NextPlot','replace');
filename1 = [D1 '\ ' I2 '_' num2str(B)];
print('-djpeg',filename1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIN DE
GRAFICAR

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%           ECUACIONES DIFERENCIALES - MODELO DINAMICO
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function dSeg = dSeg(t,c,ve, p)
% ORDEN DE LAS VARIABLES EN C (ECUACIONES DIFERENCIALES)
% Cro1, Tro1, Tjo1, Cro2, Tro2, Tjo2, Itro1, Itro2, mu
H = ve(1:2);
D = ve(4:5);
ve(1:2) = pi.*D.^2.*H./4; % VOLUMENES
ve(4:5) = pi.*D.*H; % AREAS

```





## ANEXO C. Programación en MATLAB R2012a para la simulación del sistema de reactores en estado dinámico serie

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% BALANCE DE MASA Y ENERGIA - ESTADO DINAMICO
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [C, Ceq] = DynReac(v,pe)
    global E I2 D1
% EVITA RESOLVER PARA VALORES NEGATIVOS DE DISEÑO
    v = abs(v);
% RESUELVE EL SISTEMA EN ESTADO ESTACIONARIO
    Ce = StatReac(v, pe);
% CONDICIONES INICIALES DEL SISTEMA DE ECUACIONES DIFERENCIALES
    Ce1 = [Ce(4:9), zeros(1,3)];
    vv = [v(1:7), Ce1(2), Ce1(5), v(8:11)];
    if I2 == 'S'
        Te = 0:0.001:5;
    else
        Te = 0:0.001:10;
    end
    opt = odeset('RelTol', 1e-10, 'AbsTol', 1e-10, 'NormControl',
'On');
% RESUELVE EL SISTEMA DE ECUACIONES DIFERENCIALES
    [T,Ct] = ode23(@dSeg,Te,Ce1,opt,vv,pe);

    N = length(T); % POSICIÓN DEL ÚLTIMO VALOR EN LA MATRIZ DE
DATOS
    Cp = Ct(N,4); % CONCENTRACIÓN DE SALIDA
    ISE = Ct(N,9); % CONTROLABILIDAD

% CÁLCULO DEL FLUJO MÁXIMO DE REFRIGERANTE UTILIZADO
    Fj1 = vv(6) + vv(10).*(Ct(:,2)-vv(8))+vv(10).*Ct(:,7)./vv(12);
    Fj2 = vv(7) + vv(11).*(Ct(:,5)-vv(9))+vv(11).*Ct(:,8)./vv(13);
    F1max = max(Fj1);
    F2max = max(Fj2);
    F1min = min(Fj1);
    F2min = min(Fj2);
    if I2 == 'S'
% SALIDAS DEL SISTEMA PARA LA OPTIMIZACIÓN DINAMICA
        C = [1000.*(Cp-0.025),...
            Ce1(2)-620, Ce1(5)-620, ISE-1000.*E,...
            0.30.*F1max - vv(6), 0.30.*F2max-vv(7),...
            -F1min, -F2min];
        Ceq = [];
    else
% SALIDAS DEL SISTEMA PARA LA SIMULACIÓN Y REPORTE

```

```

Ct(:,9) = Ct(:,9)/1000;
if I2 == 'Sim'
    C = Ct;
    Ceq = T;
else
    C = Ct(N,:);
    Ceq = [];
end
% GENERACIÓN DE GRAFICOS
close all
% GRÁFICOS SENCILLOS
graficar(T,Ct(:,1),'Respuesta Dinamica de la Concentración
- Reactor 1',...
    'Tiempo (h)', 'Cro1 (lbmol/ft^3)',1,0,0)
graficar(T,Ct(:,4),'Respuesta Dinamica de la Concentración
- Reactor 2',...
    'Tiempo (h)', 'Cp (lbmol/ft^3)',2,0,0)
graficar(T,Ct(:,2),'Respuesta Dinamica de la Temperatura -
Reactor 1',...
    'Tiempo (h)', 'Trol (R)',3,0,0)
graficar(T,Ct(:,5),'Respuesta Dinamica de la Temperatura -
Reactor 2',...
    'Tiempo (h)', 'Tp (lbmol/ft^3)',4,0,0)
graficar(T,Ct(:,3),'Respuesta Dinamica de la Temperatura -
Chaqueta 1',...
    'Tiempo (h)', 'Tjo1 (R)',5,0,0)
graficar(T,Ct(:,6),'Respuesta Dinamica de la Temperatura -
Chaqueta 2',...
    'Tiempo (h)', 'Tjo2 (lbmol/ft^3)',6,0,0)
graficar(T,Ct(:,7),'Respuesta Dinamica del Termino
Integral para el Controlador PI - Reactor 1',...
    'Tiempo (h)', 'Itrol (R.h)',7,0,0)
graficar(T,Ct(:,8),'Respuesta Dinamica del Termino
Integral para el Controlador PI - Reactor 2',...
    'Tiempo (h)', 'Intro2 (R.h)',8,0,0)
graficar(T,Ct(:,9),'Respuesta Dinamica del Error Cuadrado
Integral (ISE)',...
    'Tiempo (h)', '\mu (R^{2}.h)',9,0,0)
% DATOS APAREADOS
graficar(T,[Ct(:,1),Ct(:,4)],'Respuesta Dinamica de las
Concentraciones',...
    'Tiempo (h)', 'Concentracion
(lbmol/ft^3)',10,'Cro1','Cro2')
graficar(T,[Ct(:,2),Ct(:,5)],'Respuesta Dinamica de las
Temperaturas en los reactores',...
    'Tiempo (h)', 'Temperatura (R)',11,'Trol','Tp')
graficar(T,[Ct(:,3),Ct(:,6)],'Respuesta Dinamica de las
Temperaturas en las chaquetas',...
    'Tiempo (h)', 'Temperatura (R)',12,'Tjo1','Tjo2')
graficar(T,[Ct(:,7),Ct(:,8)],'Respuesta Dinamica del
Termino Integral para el Controlador PI',...

```

```

                                'Tiempo (h)', 'Termino Integral
(R.h)',13,'Itro1','Itro2')
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIN
DE DYNREAC

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               FUNCION AUXILIAR PARA FORMATO DE GRAFICAS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function graficar (XX,YY,S1,S2,S3,B,AA, BB)
% FORMATEA LA PRESENTACIÓN DE LOS DATOS GRAFICADOS
global D1 I2
if B >= 12
    plot(XX,YY(:,1),'k-',XX,YY(:,2),'b--')
    ss1 = ['\fontname{Times New Roman}\fontsize{12}\bf{ ' AA ' }'];
    ss2 = ['\fontname{Times New Roman}\fontsize{12}\bf{ ' BB ' }'];
    legend(ss1,ss2,'location', 'SouthEast')
else
    plot(XX,YY,'k-')
end
grid on
title(['\fontname{Times New Roman}\fontsize{12}\bf{ ' S1 ' }'])
xlabel(['\fontname{Times New Roman}\fontsize{12}\bf{ ' S2 ' }'])
ylabel(['\fontname{Times New Roman}\fontsize{12}\bf{ ' S3 ' }'])
set(gcf,'NextPlot','replace');
filename1 = [D1 '\ ' I2 '_' num2str(B)];
print('-djpeg',filename1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIN DE
GRAFICAR

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               ECUACIONES DIFERENCIALES - MODELO DINAMICO
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function dSeg = dSeg(t,c,ve, p)
% ORDEN DE LAS VARIABLES EN C (ECUACIONES DIFERENCIALES)
% Cro1, Tro1, Tjo1, Cro2, Tro2, Tjo2, Itro1, Itro2, mu
H = ve(1:2);
D = ve(4:5);
ve(1:2) = pi.*D.^2.*H./4;    % VOLUMENES
ve(4:5) = pi.*D.*H;        % AREAS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARA Cro1 y Tro1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Fr    = p(4) + ve(3);
Cri1  = (p(4).*p(3) + ve(3).*c(4))./Fr;
kl    = p(1).*exp(-p(2)./c(2));

```















**ANEXO G. Programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo optimización de control mediante algoritmos genéticos**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%      OPTIMIZACION - SOLO CONTROL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function OnlyControlPar
% Variables de Diseño (ve)
% 1   2   3   4   5   6   7   8   9   10  11  12
13
% Hr1, Hr2, Fb, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1   2   3   4   5   6   7   8   9   10  11  12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global I2 D1
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S';           % INDICADOR
tiempo = clock;    % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_OC'];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN DEL CONTROL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      PARAMETROS DE DISEÑO OBTENIDO DE LA OPTIMIZACION ESTACIONARIA
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

vd = [0.922139, 7.964433, 0.000288, 1.058047, 23.756373,...
1.022018, 39.360178];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
  LB = [0, 0, 1e-1, 1e-1, 0];
  UB = [5000, 5000, 5, 5, 30000];
% Función Objetivo
  Problem.fitnessfcn=@Costo2;
% Restricciones no-lineales
  Problem.nonlcon= @(v)DynReac2(v,vd,pe);
%Number of Variables
  Problem.nvars= 5;
% RESTRICCIONES LINEALES DE DESIGUALDAD A*X <= B
  Problem.Aineq= [ ];
  Problem.Bineq= [ ];
% RESTRICCIONES LINEALES DE IGUALDAD Aeq*X = Beq
  Problem.Aeq= [ ];
  Problem.Beq= [ ];
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
  Problem.LB= LB;
  Problem.UB= UB;
% OPCIONES DE OPTIMIZACION
  disp(' ')
% OPCIONES AG
  options = gaoptimset;
  options = gaoptimset(options,'Display','iter');
  options = gaoptimset(options,'StallGenLimit',Inf);
  options = gaoptimset(options,'StallTimeLimit',Inf);
  options = gaoptimset(options,'TolFun',1e-2);
  options = gaoptimset(options,'TolCon',1e-4);
  options = gaoptimset(options,'EliteCount',2);
  options = gaoptimset(options,'PopulationSize',10);
  options = gaoptimset(options,'FitnessScalingFcn'
,@fitscalingprop);
  options = gaoptimset(options,'SelectionFcn',@selectionuniform);
  options = gaoptimset(options,'MutationFcn',
{@mutationadaptfeasible 0.1});
  options =
gaoptimset(options,'PopInitRange',[Problem.LB;Problem.UB]);
  Problem.options= options;
  disp('Opciones del AG')
  disp(options)
% OPTIMIZADOR ---> ALGORITMO GENETICO
  [X,FVAL,REASON,OUTPUT,POPULATION,SCORES] = ga(Problem);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
% PUBLICACION DE RESULTADOS
  R = input('¿Desea Generar los Resultados?(s/n)','s');

```

```

if or(R == 'S', R == 's')
    mkdir(D1);
    copyfile('temp.txt',D1,'f')
    I2 = 'Estat';
    diary([D1 'out.txt'])
    clc
    disp(' ')
    disp('%%%%%%%%%% RESULTADOS %%%%%%%%%%%')
    disp(['FECHA: ' date])
    disp(' ');
    disp('OPTIMIZACION CONTROL')
    disp(' ')
    I2 = 'OC';
    disp('%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%')
    disp(['Hr1 (ft)           : ' sprintf('%1.8e ',vd(1))]);
    disp(['Hr2 (ft)           : ' sprintf('%1.8e ',vd(2))]);
    disp(['Fb (ft^3/h)        : ' sprintf('%1.8e ',vd(3))]);
    disp(['Dr1 (ft)           : ' sprintf('%1.8e ',vd(4))]);
    disp(['Dr2 (ft)           : ' sprintf('%1.8e ',vd(5))]);
    disp(['Fjn1 (ft^3/h)      : ' sprintf('%1.8e ',vd(6))]);
    disp(['Fjn2 (ft^3/h)      : ' sprintf('%1.8e ',vd(7))]);
    disp('%%%%%%%%%% PARAMETROS OPTIMOS DE CONTROL
%%%%%%%%%%')
    disp(['kj1 (ft^3/h.R)     : ' sprintf('%1.8e ',X(1))]);
    disp(['kj2 (ft^3/h.R)     : ' sprintf('%1.8e ',X(2))]);
    disp(['Tauj1 (h)          : ' sprintf('%1.8e ',X(3))]);
    disp(['Tauj2 (h)          : ' sprintf('%1.8e ',X(4))]);
    disp('%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO
%%%%%%%%%%')
    C3 = StatReac([vd, X(1:4)],pe);
    disp(['Fr (ft^3/h)        : ' sprintf('%1.8e ',C3(1))]);
    disp(['Cri1 (lbmol/ft^3) : ' sprintf('%1.8e ',C3(2))]);
    disp(['Tri1 (R)           : ' sprintf('%1.8e ',C3(3))]);
    disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C3(4))]);
    disp(['Tro1 (R)           : ' sprintf('%1.8e ',C3(5))]);
    disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C3(6))]);
    disp(['Cp (lbmol/ft^3)   : ' sprintf('%1.8e ',C3(7))]);
    disp(['Tp (R)            : ' sprintf('%1.8e ',C3(8))]);
    disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C3(9))]);
    disp('%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%')
    disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
    C4 = DynReac2(X,vd,pe);
    disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C4(1))]);
    disp(['Tro1 (R)           : ' sprintf('%1.8e ',C4(2))]);
    disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C4(3))]);
    disp(['Cp (lbmol/ft^3)   : ' sprintf('%1.8e ',C4(4))]);
    disp(['Tp (R)            : ' sprintf('%1.8e ',C4(5))]);
    disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C4(6))]);
    disp(['Itro1 (R.h)        : ' sprintf('%1.8e ',C4(7))]);
    disp(['Itro2 (R.h)        : ' sprintf('%1.8e ',C4(8))]);
    disp(['mu (R^2.h)         : ' sprintf('%1.8e ',C4(9))]);
    disp(' ');

```



**ANEXO H. Programación en MATLAB R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie. optimización de control mediante algoritmos genéticos**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%      OPTIMIZACION - SOLO CONTROL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function OnlyControlSer
% Variables de Diseño (ve)
% 1   2   3   4   5   6   7   8   9   10  11  12
13
% Hr1, Hr2, Fb, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1   2   3   4   5   6   7   8   9   10  11  12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global I2 D1
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S';           % INDICADOR
tiempo = clock;    % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_OC'];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN DEL CONTROL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      PARAMETROS DE DISEÑO OBTENIDO DE LA OPTIMIZACION ESTACIONARIA
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

vd = [4.124120, 4.192577, 0.130800, 12.144237, 12.060946,...
      33.133623, 7.183125];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
  LB = [0, 0, 1e-1, 1e-1, 0];
  UB = [250, 250, 5, 5, 250];
% Función Objetivo
  Problem.fitnessfcn=@Costo2;
% Restricciones no-lineales
  Problem.nonlcon= @(v)DynReac2(v,vd,pe);
%Number of Variables
  Problem.nvars= 5;
% RESTRICCIONES LINEALES DE DESIGUALDAD A*X <= B
  Problem.Aineq= [ ];
  Problem.Bineq= [ ];
% RESTRICCIONES LINEALES DE IGUALDAD Aeq*X = Beq
  Problem.Aeq= [ ];
  Problem.Beq= [ ];
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
  Problem.LB= LB;
  Problem.UB= UB;
% OPCIONES DE OPTIMIZACION
  disp(' ')
% OPCIONES AG
  options = gaoptimset;
  options = gaoptimset(options,'Display' , 'iter');
  options = gaoptimset(options,'StallGenLimit' , Inf);
  options = gaoptimset(options,'StallTimeLimit' , Inf);
  options = gaoptimset(options,'TolFun' , 1e-2);
  options = gaoptimset(options,'TolCon' , 1e-4);
  options = gaoptimset(options,'EliteCount',2);
  options = gaoptimset(options,'PopulationSize',10);
  options = gaoptimset(options,'FitnessScalingFcn'
,@fitscalingprop);
  options = gaoptimset(options,'SelectionFcn' ,@selectionuniform);
  options = gaoptimset(options,'MutationFcn' ,
{@mutationadaptfeasible 0.1});
  options =
gaoptimset(options,'PopInitRange',[Problem.LB;Problem.UB]);
  Problem.options= options;
  disp('Opciones del AG')
  disp(options)
% OPTIMIZADOR ---> ALGORITMO GENETICO
  [X,FVAL,REASON,OUTPUT,POPULATION,SCORES] = ga(Problem);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  diary off
% PUBLICACION DE RESULTADOS
  R = input('¿Desea Generar los Resultados?(s/n) ','s');

```



```

if or(R == 'S', R == 's')
    mkdir(D1);
    copyfile('temp.txt',D1,'f')
    I2 = 'Estat';
    diary([D1 'out.txt'])
    clc
    disp(' ')
    disp('%%%%%%%%%% RESULTADOS %%%%%%%%%%%')
    disp(['FECHA: ' date])
    disp(' ');
    disp('OPTIMIZACION CONTROL')
    disp(' ')
    I2 = 'OC';
    disp('%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%')
    disp(['Hr1 (ft)           : ' sprintf('%1.8e ',vd(1))]);
    disp(['Hr2 (ft)           : ' sprintf('%1.8e ',vd(2))]);
    disp(['Fb (ft^3/h)        : ' sprintf('%1.8e ',vd(3))]);
    disp(['Dr1 (ft)           : ' sprintf('%1.8e ',vd(4))]);
    disp(['Dr2 (ft)           : ' sprintf('%1.8e ',vd(5))]);
    disp(['Fjn1 (ft^3/h)      : ' sprintf('%1.8e ',vd(6))]);
    disp(['Fjn2 (ft^3/h)      : ' sprintf('%1.8e ',vd(7))]);
    disp('%%%%%%%%%% PARAMETROS OPTIMOS DE CONTROL
%%%%%%%%%%')
    disp(['kj1 (ft^3/h.R)    : ' sprintf('%1.8e ',X(1))]);
    disp(['kj2 (ft^3/h.R)    : ' sprintf('%1.8e ',X(2))]);
    disp(['Tauj1 (h)         : ' sprintf('%1.8e ',X(3))]);
    disp(['Tauj2 (h)         : ' sprintf('%1.8e ',X(4))]);
    disp('%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO
%%%%%%%%%%')
    C3 = StatReac([vd, X(1:4)],pe);
    disp(['Fr (ft^3/h)       : ' sprintf('%1.8e ',C3(1))]);
    disp(['Cri1 (lbmol/ft^3) : ' sprintf('%1.8e ',C3(2))]);
    disp(['Tri1 (R)           : ' sprintf('%1.8e ',C3(3))]);
    disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C3(4))]);
    disp(['Tro1 (R)           : ' sprintf('%1.8e ',C3(5))]);
    disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C3(6))]);
    disp(['Cp (lbmol/ft^3)   : ' sprintf('%1.8e ',C3(7))]);
    disp(['Tp (R)            : ' sprintf('%1.8e ',C3(8))]);
    disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C3(9))]);
    disp('%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%')
    disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
    C4 = DynReac2(X,vd,pe);
    disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C4(1))]);
    disp(['Tro1 (R)           : ' sprintf('%1.8e ',C4(2))]);
    disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C4(3))]);
    disp(['Cp (lbmol/ft^3)   : ' sprintf('%1.8e ',C4(4))]);
    disp(['Tp (R)            : ' sprintf('%1.8e ',C4(5))]);
    disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C4(6))]);
    disp(['Itro1 (R.h)        : ' sprintf('%1.8e ',C4(7))]);
    disp(['Itro2 (R.h)        : ' sprintf('%1.8e ',C4(8))]);
    disp(['mu (R^2.h)         : ' sprintf('%1.8e ',C4(9))]);
    disp(' ');

```



## ANEXO I. Programación en MATLAB R2012a para el diseño integrado.

### Topología de reactores en paralelo

```

function OptDynPar
%
% Script para el cálculo de las variables de diseño en estado
% DINAMICO de dos reactores mezcla completa con chaqueta de
% enfriamiento dispuestos en forma PARALELA. Se procesa un
compuesto A
% que reacciona a una velocidad de reacción de primer orden. Éste
cálculo
% es obtenido a partir de la optimización minimizando el costo de
diseño
% (en base al diámetro-altura de los reactores y el flujo de
refrigerante).
%
% Éste script utiliza la función 'FMINCON' del toolbox de
optimización de
% MatLab. Además consta de las siguientes funciones:
%
% Costo(VOD):          Calcula el costo de diseño de los reactores
para las
%                      variables de diseño VOD.
% DynReac(v,pe):      Realiza la simulación dinámica de los
reactores
%                      para las variables de diseño v y los
parámetros del
%                      sistema pe.
% dSeg(t,c,ve, p):    Calcula los valores de la ecuaciones
diferenciales
%                      correspondiente al modelo de simulación. t
(tiempo), c
%                      (variables dinámicas), ve (variables de
diseño), p
%                      (parámetros del sistema)
% StatReac(vs,ps):    Simula el reactor en estado estacionario.
%                      vs (variables de diseño), ps (parámetros del
sistema)
% error(c):           Función utilizada para resolver el sistema de
ecuaciones algebraicas en la simulación en
estado
%                      estacionario.
% Ti(tp):             Función de perturbación para la temperatura de
alimentación al reactor.
%
%
% Variables de Diseño (ve)
% 1    2    3    4    5    6    7    8    9    10   11   12
13

```

```

% Hr1, Hr2, Fb, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global E I2 D1
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% VALOR MAXIMO DE CONTROLABILIDAD MODIFICAR MANUALMENTE
%
E = 8e-4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I2 = 'S'; % INDICADOR
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_P'
num2str(fix(1000.*E))];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
% OPCIONES DE OPTIMIZACION
disp(' ')
Options = optimset('MaxFunEvals', 500000, 'TolCon', 1e-5,...
'LevenbergMarquardt','on','LargeScale','off','Display',...
'iter','DiffMinChange',1e-2,'TolFun',1e-
2,'DiffMaxChange',...
1,'MaxIter',250);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULO OPTIMIZACIÓN EN ESTADO TRANSITORIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN EN ESTADO TRANSITORIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(' ')
disp(['Controlabilidad : ', num2str(E) ])
disp(' ')
% VALORES INICIALES DE V OBTENIDOS CON AG (SS) y AG (OC)
ve = [0.5000 7.8868 -0.0000 1.0000 23.6605 1.0000 39.3312
50.5460 1174.1819 0.7408 0.9848 ];

```

```

% RESTRICCIONES LINEALES DE DESIGUALDAD A*X <= B
A =[-300,    0,    0,   100,    0, 0, 0, 0, 0, 0, 0
      0, -300,    0,    0,   100, 0, 0, 0, 0, 0, 0
      100,    0,    0, -100,    0, 0, 0, 0, 0, 0, 0
      0,   100,    0,    0, -100, 0, 0, 0, 0, 0, 0];
B = [0; 0; 0; 0];
% RESTRICCIONES LINEALES DE IGUALDAD Aeq*X = Beq
Aeq = [ ];
Beq = [ ];
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
LB = [0.5, 0.5, 0, 1, 1, 1, 1, 0, 0, 1e-1, 1e-1];
UB = [100, 100, 1, 100, 100, 500, 500, 5000, 5000, 5, 5];
% OPTIMIZADOR ---> FMINCON
[X,FVAL,EXITFLAG] = fmincon( @Costo, ve, A, B, Aeq, Beq, LB,
UB, ...
                            @(v) DynReac(v,pe), Options);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp(sprintf('%6.4f ',X));           % VARIABLES DE DISEÑO
disp(sprintf('%6.4f ',FVAL));       % FUNCIÓN COSTO

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
R = input('¿Desea Generar los Resultados?(s/n)', 's');
if or(R == 'S', R == 's')
    mkdir(D1);
    copyfile('temp.txt',D1,'f')
    I2 = 'Est';
    diary([D1 '\out.txt'])
    clc
    disp(' ')
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS OPTCONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['FECHA: ' date])
    disp(' ');
    disp('OPTIMIZACION EN ESTADO DINAMICO')
    disp(' ')
    I2 = 'Din';
    cost2 = Costo(X);
    disp(['COSTO CAPITAL ($) : ' sprintf('%10.12g ',cost2(1))]);
    disp(['COSTO UTILIDAD($) : ' sprintf('%10.12g ',cost2(2))]);
    disp(['COSTO TOTAL ($) : ' sprintf('%10.12g ',cost2(3))]);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%')
    disp(['Hr1 (ft) : ' sprintf('%1.8e ',X(1))]);
    disp(['Hr2 (ft) : ' sprintf('%1.8e ',X(2))]);
    disp(['Fr1/Ff : ' sprintf('%1.8e ',X(3))]);
    disp(['Dr1 (ft) : ' sprintf('%1.8e ',X(4))]);
    disp(['Dr2 (ft) : ' sprintf('%1.8e ',X(5))]);

```



**ANEXO J. Programación en MATLAB R2012a para el diseño integrado.**  
**Topología de reactores en serie**

```

function OptDynSer
%
% Script para el cálculo de las variables de diseño en estado
% DINAMICO de dos reactores mezcla completa con chaqueta de
% enfriamiento dispuestos en forma SERIE. Se procesa un compuesto
A
% que reacciona a una velocidad de reacción de primer orden. Éste
cálculo
% es obtenido a partir de la optimización minimizando el costo de
diseño
% (en base al diámetro-altura de los reactores y el flujo de
refrigerante).
%
% Éste script utiliza la función 'FMINCON' del toolbox de
optimización de
% MatLab. Además consta de las siguientes funciones:
%
% Costo(VOD):          Calcula el costo de diseño de los reactores
para las
%                      variables de diseño VOD.
% DynReac(v,pe):      Realiza la simulación dinámica de los
reactores
%                      para las variables de diseño v y los
parámetros del
%                      sistema pe.
% dSeg(t,c,ve, p):    Calcula los valores de la ecuaciones
diferenciales
%                      correspondiente al modelo de simulación. t
(tiempo), c
%                      (variables dinámicas), ve (variables de
diseño), p
%                      (parámetros del sistema)
% StatReac(vs,ps):    Simula el reactor en estado estacionario.
%                      vs (variables de diseño), ps (parámetros del
sistema)
% error(c):           Función utilizada para resolver el sistema de
ecuaciones algebraicas en la simulación en
estado
%                      estacionario.
% Ti(tp):            Función de perturbación para la temperatura de
alimentación al reactor.
%
%
% Variables de Diseño (ve)
% 1    2    3    4    5    6    7    8    9    10   11   12
13

```

```

% Hr1, Hr2, Fb, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global E I2 D1
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% VALOR MAXIMO DE CONTROLABILIDAD MODIFICAR MANUALMENTE
%
E = 1.90e-002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I2 = 'S'; % INDICADOR
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_P'
num2str(fix(1000.*E))];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
% OPCIONES DE OPTIMIZACION
disp(' ')
Options = optimset('MaxFunEvals', 500000, 'TolCon', 1e-5,...
'LevenbergMarquardt','on','LargeScale','off','Display',...
'iter','DiffMinChange',1e-3,'TolFun',1e-
3,'DiffMaxChange',...
1,'MaxIter',250);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULO OPTIMIZACIÓN EN ESTADO TRANSITORIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN EN ESTADO TRANSITORIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(' ')
disp(['Controlabilidad : ', num2str(E) ])
disp(' ')
% VALORES INICIALES DE V OBTENIDOS CON AG (SS) y AG (OC)
ve = [ 4.63027386e+000, 3.53054128e+000, 1.80702148e-004,...

```



```

    1.38908216e+001, 1.05916238e+001, 3.53500963e+001,
4.71887211e+000,...
    1.75264888e+002, 9.36063527e+001, 3.81969011e-001,
9.57276646e-001];
% RESTRICCIONES LINEALES DE DESIGUALDAD A*X <= B
A =[-300,    0,    0,   100,    0, 0, 0, 0, 0, 0, 0, 0
     0, -300,    0,    0,   100, 0, 0, 0, 0, 0, 0, 0
     100,    0,    0, -100,    0, 0, 0, 0, 0, 0, 0, 0
     0,   100,    0,    0, -100, 0, 0, 0, 0, 0, 0, 0];
B =[0; 0; 0; 0];
% RESTRICCIONES LINEALES DE IGUALDAD Aeq*X = Beq
Aeq =[ ];
Beq =[ ];
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
LB = [1.5, 1.5, 0, 1.5, 1.5, 1e-2, 1e-2, 0, 0, 1e-1, 1e-1];
UB = [100, 100, 1000, 100, 100, 1000, 1000, 5000, 5000, 5, 5];
% OPTIMIZADOR ---> FMINCON
[X,FVAL,EXITFLAG] = fmincon( @Costo, ve, A, B, Aeq, Beq, LB,
UB,...
                                @(v)DynReac(v,pe), Options);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp(sprintf('%6.4f ',X));           % VARIABLES DE DISEÑO
disp(sprintf('%6.4f ',FVAL));       % FUNCIÓN COSTO

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
R = input('¿Desea Generar los Resultados?(s/n)', 's');
if or(R == 'S', R == 's')
    mkdir(D1);
    copyfile('temp.txt',D1,'f')
    I2 = 'Est';
    diary([D1 '\out.txt'])
    clc
    disp(' ')
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS OPTCONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['FECHA: ' date])
    disp(' ');
    disp('OPTIMIZACION EN ESTADO DINAMICO')
    disp(' ')
    I2 = 'Din';
    cost2 = Costo(X);
    disp(['COSTO CAPITAL ($) : ' sprintf('%10.12g ',cost2(1))]);
    disp(['COSTO UTILIDAD($) : ' sprintf('%10.12g ',cost2(2))]);
    disp(['COSTO TOTAL ($) : ' sprintf('%10.12g ',cost2(3))]);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['Hr1 (ft) : ' sprintf('%1.8e ',X(1))]);

```

```

disp(['Hr2 (ft)           : ' sprintf('%1.8e ',X(2))]);
disp(['Fb (ft^3/h)       : ' sprintf('%1.8e ',X(3))]);
disp(['Dr1 (ft)          : ' sprintf('%1.8e ',X(4))]);
disp(['Dr2 (ft)          : ' sprintf('%1.8e ',X(5))]);
disp(['Fjn1 (ft^3/h)     : ' sprintf('%1.8e ',X(6))]);
disp(['Fjn2 (ft^3/h)     : ' sprintf('%1.8e ',X(7))]);
disp('%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE CONTROL %%%%%%%%%%%%%%%%%')
disp(['kj1 (ft^3/h.R)    : ' sprintf('%1.8e ',X(8))]);
disp(['kj2 (ft^3/h.R)    : ' sprintf('%1.8e ',X(9))]);
disp(['Tauj1 (h)         : ' sprintf('%1.8e ',X(10))]);
disp(['Tauj2 (h)         : ' sprintf('%1.8e ',X(11))]);
disp('%%%%%%%%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%%%%%%%%')
C3 = StatReac(X,pe);
disp(['Fr (ft^3/h)       : ' sprintf('%1.8e ',C3(1))]);
disp(['Cri1 (lbmol/ft^3) : ' sprintf('%1.8e ',C3(2))]);
disp(['Tri1 (R)          : ' sprintf('%1.8e ',C3(3))]);
disp(['Crol (lbmol/ft^3) : ' sprintf('%1.8e ',C3(4))]);
disp(['Trol (R)          : ' sprintf('%1.8e ',C3(5))]);
disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C3(6))]);
disp(['Cp (lbmol/ft^3)   : ' sprintf('%1.8e ',C3(7))]);
disp(['Tp (R)            : ' sprintf('%1.8e ',C3(8))]);
disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C3(9))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%%%%%%%%%%%%%%%')
disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
C4 = DynReac(X,pe);
disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C4(1))]);
disp(['Trol (R)          : ' sprintf('%1.8e ',C4(2))]);
disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C4(3))]);
disp(['Cp (lbmol/ft^3)   : ' sprintf('%1.8e ',C4(4))]);
disp(['Tp (R)            : ' sprintf('%1.8e ',C4(5))]);
disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C4(6))]);
disp(['Itro1 (R.h)       : ' sprintf('%1.8e ',C4(7))]);
disp(['Itro2 (R.h)       : ' sprintf('%1.8e ',C4(8))]);
disp(['mu (R^2.h)        : ' sprintf('%1.8e ',C4(9))]);
disp(' ');
disp(' *Nota: La respuesta dinámica en todo el periodo de
estudio')
disp(['          se encuentran graficados en los archivos ' I2
'_. . . .jpeg'])
disp('          en la carpeta:')
disp(['          ' cd '\\ ' D1])
disp(' ')
close all
toc
disp(' ')
disp('****FIN DEL REPORTE')
diary off
save([D1 'out'])
end
%%%%%%%%%%%%%%%%%%%%%%%%%FIN DE
POPTCONTROL

```

**ANEXO K. Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en paralelo**

```

function OptEstatPar
%
% Script para el cálculo de las variables de diseño en estado
% ESTACIONARIO de dos reactores mezcla completa con chaqueta de
% enfriamiento dispuestos en forma PARALELA. Se procesa un
compuesto A
% que reacciona a una velocidad de reacción de primer orden. Éste
cálculo
% es obtenido a partir de la optimización minimizando el costo de
diseño
% (en base al diámetro-altura de los reactores y el flujo de
refrigerante).
%
%
% Éste script utiliza la función GA del toolbox de optimización de
% MatLab. Además consta de las siguientes funciones:
%
% Costo(VOD):          Calcula el costo de diseño de los reactores
para las
%                      variables de diseño VOD.
% DynReac(v,pe):      Realiza la simulación dinámica de los
reactores
%                      para las variables de diseño v y los
parámetros del
%                      sistema pe.
% dSeg(t,c,ve, p):    Calcula los valores de la ecuaciones
diferenciales
%                      correspondiente al modelo de simulación. t
(tiempo), c
%                      (variables dinámicas), ve (variables de
diseño), p
%                      (parámetros del sistema)
% StatReac(vs,ps):    Simula el reactor en estado estacionario.
%                      vs (variables de diseño), ps (parámetros del
sistema)
% error(c):           Función utilizada para resolver el sistema de
%                      ecuaciones algebraicas en la simulación en
estado
%                      estacionario.
% Ti(tp):             Función de perturbación para la temperatura de
%                      alimentación al reactor.
%
%
% Variables de Diseño (ve)
% 1    2    3    4    5    6    7    8    9    10   11   12
13

```

```

% Hr1, Hr2, Rf, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global E I2 D1 pe
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S'; % INDICADOR
E = 0.001; % VALOR MAXIMO DE CONTROLABILIDAD
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_'
num2str(fix(1000.*E))];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%% REACTORES EN PARALELO
%%%%%%%%%%')
disp('%%%%%%%%%% OPTIMIZACIÓN EN ESTADO ESTACIONARIO
%%%%%%%%%%')
% Función Objetivo
Problem.fitnessfcn=@Costo;
% Restricciones no-lineales
Problem.nonlcon= @(v)StatReac(v,pe);
%Number of Variables
Problem.nvars= 7;
% RESTRICCIONES LINEALES DE DESIGUALDAD A*X <= B
Problem.Aineq= [-300, 0, 0, 100, 0, 0, 0
0, -300, 0, 0, 100, 0, 0];
Problem.Bineq= [ 0; 0];
% RESTRICCIONES LINEALES DE IGUALDAD Aeq*X = Beq
Problem.Aeq= [ ];
Problem.Beq= [ ];
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
Problem.LB= [0.5, 0.5, 0, 1, 1, 1, 1];
Problem.UB= [100, 100, 1, 100, 100, 500, 500];
% OPCIONES DE OPTIMIZACION
disp(' ')
% OPCIONES AG
options = gaoptimset;
options = gaoptimset(options,'Display' , 'iter');
options = gaoptimset(options,'StallGenLimit' ,Inf);
options = gaoptimset(options,'StallTimeLimit' ,Inf);
options = gaoptimset(options,'EliteCount',2);

```

```

options = gaoptimset(options, 'FitnessScalingFcn'
,@fitscalingprop);
options = gaoptimset(options, 'SelectionFcn' ,@selectionuniform);
options = gaoptimset(options, 'MutationFcn' ,
{@mutationadaptfeasible 0.1});
options =
gaoptimset(options, 'PopInitRange', [Problem.LB;Problem.UB]);

Problem.options= options;
disp('Opciones del AG')
disp(options)
% OPTIMIZADOR ---> ALGORITMO GENETICO
[X1,FVAL,REASON,OUTPUT,POPULATION,SCORES] = ga(Problem);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp('RESULTADOS AG')
disp(sprintf('%10.6f ',X1));           % VARIABLES DE DISEÑO
disp(sprintf('%10.6f ',FVAL));         % FUNCIÓN COSTO
disp(REASON);                          % CONDICIÓN DE SALIDA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

diary off
R = input('¿Desea Generar los Resultados?(s/n)', 's');
% Datos Anteriores Estado Estacionario
ve = [X1, 0, 0, 1.2,1.2];

if or(R == 'S', R == 's')
mkdir(D1);
copyfile('temp.txt',D1,'f')
I2 = 'Est';
diary([D1 '\out.txt'])
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS OPTCONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['FECHA: ' date])
disp(' ');
disp('OPTIMIZACION EN ESTADO ESTACIONARIO')
disp(' ')
cost = Costo(ve);
disp(['COSTO CAPITAL ($) : ' sprintf('%10.12g ',cost(1))]);
disp(['COSTO UTILIDAD($) : ' sprintf('%10.12g ',cost(2))]);
disp(['COSTO TOTAL ($) : ' sprintf('%10.12g ',cost(3))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['Hr1 (ft) : ' sprintf('%1.8e ',ve(1))]);
disp(['Hr2 (ft) : ' sprintf('%1.8e ',ve(2))]);

```

```

disp(['Fr1/Ff          : ' sprintf('%1.8e ',ve(3))]);
disp(['Dr1 (ft)       : ' sprintf('%1.8e ',ve(4))]);
disp(['Dr2 (ft)       : ' sprintf('%1.8e ',ve(5))]);
disp(['Fjn1 (ft^3/h)  : ' sprintf('%1.8e ',ve(6))]);
disp(['Fjn2 (ft^3/h)  : ' sprintf('%1.8e ',ve(7))]);
disp('%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%')
C1 = StatReac(ve,pe);
disp(['Fr1 (ft^3/h)   : ' sprintf('%1.8e ',C1(1))]);
disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C1(2))]);
disp(['Tro1 (R)       : ' sprintf('%1.8e ',C1(3))]);
disp(['Tjo1 (R)       : ' sprintf('%1.8e ',C1(4))]);
disp(['Fr2 (ft^3/h)   : ' sprintf('%1.8e ',C1(5))]);
disp(['Cro2 (lbmol/ft^3) : ' sprintf('%1.8e ',C1(6))]);
disp(['Tro2 (R)       : ' sprintf('%1.8e ',C1(7))]);
disp(['Tjo2 (R)       : ' sprintf('%1.8e ',C1(8))]);
disp(['Cp (lbmol/ft^3) : ' sprintf('%1.8e ',C1(9))]);
disp(['Tp (R)         : ' sprintf('%1.8e ',C1(10))]);
disp(' ');
disp('%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%')
disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
C2 = DynReac(ve,pe);
disp(['Cro1 (lbmol/ft^3): ' sprintf('%1.8e ',C2(1))]);
disp(['Tro1 (R)         : ' sprintf('%1.8e ',C2(2))]);
disp(['Tjo1 (R)         : ' sprintf('%1.8e ',C2(3))]);
disp(['Cro2 (lbmol/ft^3): ' sprintf('%1.8e ',C2(4))]);
disp(['Tro2 (R)         : ' sprintf('%1.8e ',C2(5))]);
disp(['Tjo2 (R)         : ' sprintf('%1.8e ',C2(6))]);
disp(['Itro1 (R.h)      : ' sprintf('%1.8e ',C2(7))]);
disp(['Itro2 (R.h)      : ' sprintf('%1.8e ',C2(8))]);
disp(['mu (R^2.h)       : ' sprintf('%1.8e ',C2(9))]);
disp(['Cp (lbmol/ft^3)  : ' sprintf('%1.8e ',C2(10))]);
disp(['Tp (R)           : ' sprintf('%1.8e ',C2(11))]);
disp(' ');
disp('*Nota: La respuesta dinámica en todo el periodo de
estudio')
disp([' se encuentran graficados en los archivos ' I2
'_.jpg'])
disp('          en la carpeta:')
disp(['          ' cd '\ ' D1])
close all
toc
disp(' ')
disp('****FIN DEL REPORTE')
diary off
save([D1 'out.mat'])
end
%%%%%%%%%%FIN DE
OPTSTATPAR

```

## ANEXO L. Programación en MATLAB 2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie

```

function OptEstatPar
%
% Script para el cálculo de las variables de diseño en estado
% ESTACIONARIO de dos reactores mezcla completa con chaqueta de
% enfriamiento dispuestos en forma SERIE. Se procesa un compuesto
A
% que reacciona a una velocidad de reacción de primer orden. Éste
cálculo
% es obtenido a partir de la optimización minimizando el costo de
diseño
% (en base al diámetro-altura de los reactores y el flujo de
refrigerante).
%
%
% Éste script utiliza la función GA del toolbox de optimización de
% MatLab. Además consta de las siguientes funciones:
%
% Costo(VOD):          Calcula el costo de diseño de los reactores
para las
%                      variables de diseño VOD.
% DynReac(v,pe):      Realiza la simulación dinámica de los
reactores
%                      para las variables de diseño v y los
parámetros del
%                      sistema pe.
% dSeg(t,c,ve, p):    Calcula los valores de la ecuaciones
diferenciales
%                      correspondiente al modelo de simulación. t
(tiempo), c
%                      (variables dinámicas), ve (variables de
diseño), p
%                      (parámetros del sistema)
% StatReac(vs,ps):    Simula el reactor en estado estacionario.
%                      vs (variables de diseño), ps (parámetros del
sistema)
% error(c):           Función utilizada para resolver el sistema de
ecuaciones algebraicas en la simulación en
estado
%                      estacionario.
% Ti(tp):             Función de perturbación para la temperatura de
alimentación al reactor.
%
%
% Variables de Diseño (ve)
% 1    2    3    4    5    6    7    8    9    10   11   12
13

```

```

% Hr1, Hr2, Rf, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global E I2 D1 pe
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S'; % INDICADOR
E = 0.001; % VALOR MAXIMO DE CONTROLABILIDAD
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_'
num2str(fix(1000.*E))];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%% REACTORES EN PARALELO
%%%%%%%%%%')
disp('%%%%%%%%%% OPTIMIZACIÓN EN ESTADO ESTACIONARIO
%%%%%%%%%%')
% Función Objetivo
Problem.fitnessfcn=@Costo;
% Restricciones no-lineales
Problem.nonlcon= @(v)StatReac(v,pe);
%Number of Variables
Problem.nvars= 7;
% RESTRICCIONES LINEALES DE DESIGUALDAD A*X <= B
Problem.Aineq= [-300, 0, 0, 100, 0, 0, 0
0, -300, 0, 0, 100, 0, 0];
Problem.Bineq= [ 0; 0];
% RESTRICCIONES LINEALES DE IGUALDAD Aeq*X = Beq
Problem.Aeq= [ ];
Problem.Beq= [ ];
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
Problem.LB= [0.5, 0.5, 0, 1, 1, 1, 1];
Problem.UB= [100, 100, 500, 100, 100, 500, 500];
% OPCIONES DE OPTIMIZACION
disp(' ')
% OPCIONES AG
options = gaoptimset;
options = gaoptimset(options,'Display' , 'iter');
options = gaoptimset(options,'StallGenLimit' ,Inf);
options = gaoptimset(options,'StallTimeLimit' ,Inf);
options = gaoptimset(options,'EliteCount',2);

```



```

options = gaoptimset(options, 'FitnessScalingFcn'
,@fitscalingprop);
options = gaoptimset(options, 'SelectionFcn' ,@selectionuniform);
options = gaoptimset(options, 'MutationFcn' ,
{@mutationadaptfeasible 0.1});
options =
gaoptimset(options, 'PopInitRange', [Problem.LB;Problem.UB]);

Problem.options= options;
disp('Opciones del AG')
disp(options)
% OPTIMIZADOR ---> ALGORITMO GENETICO
[X1,FVAL,REASON,OUTPUT,POPULATION,SCORES] = ga(Problem);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp('RESULTADOS AG')
disp(sprintf('%10.6f ',X1));           % VARIABLES DE DISEÑO
disp(sprintf('%10.6f ',FVAL));         % FUNCIÓN COSTO
disp(REASON);                          % CONDICIÓN DE SALIDA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

diary off
R = input('¿Desea Generar los Resultados?(s/n)', 's');
% Datos Anteriores Estado Estacionario
ve = [X1, 0, 0, 1.2,1.2];

if or(R == 'S', R == 's')
mkdir(D1);
copyfile('temp.txt',D1,'f')
I2 = 'Est';
diary([D1 '\out.txt'])
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS OPTCONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['FECHA: ' date])
disp(' ');
disp('OPTIMIZACION EN ESTADO ESTACIONARIO')
disp(' ')
cost = Costo(ve);
disp(['COSTO CAPITAL ($) : ' sprintf('%10.12g ',cost(1))]);
disp(['COSTO UTILIDAD($) : ' sprintf('%10.12g ',cost(2))]);
disp(['COSTO TOTAL ($) : ' sprintf('%10.12g ',cost(3))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['Hr1 (ft) : ' sprintf('%1.8e ',ve(1))]);
disp(['Hr2 (ft) : ' sprintf('%1.8e ',ve(2))]);

```

```

disp(['Fb (ft^3/h)      : ' sprintf('%1.8e ',ve(3))]);
disp(['Dr1 (ft)        : ' sprintf('%1.8e ',ve(4))]);
disp(['Dr2 (ft)        : ' sprintf('%1.8e ',ve(5))]);
disp(['Fjn1 (ft^3/h)   : ' sprintf('%1.8e ',ve(6))]);
disp(['Fjn2 (ft^3/h)   : ' sprintf('%1.8e ',ve(7))]);
disp('%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%')
C1 = StatReac(ve,pe);
disp(['Fr (ft^3/h)     : ' sprintf('%1.8e ',C1(1))]);
disp(['Cri1 (lbmol/ft^3): ' sprintf('%1.8e ',C1(2))]);
disp(['Tri1 (R)        : ' sprintf('%1.8e ',C1(3))]);
disp(['Cro1 (lbmol/ft^3): ' sprintf('%1.8e ',C1(4))]);
disp(['Tro1 (R)        : ' sprintf('%1.8e ',C1(5))]);
disp(['Tjo1 (R)        : ' sprintf('%1.8e ',C1(6))]);
disp(['Cp (lbmol/ft^3) : ' sprintf('%1.8e ',C1(7))]);
disp(['Tp (R)          : ' sprintf('%1.8e ',C1(8))]);
disp(['Tjo2 (R)        : ' sprintf('%1.8e ',C1(9))]);
disp(' ');
disp('%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%')
disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
C2 = DynReac(ve,pe);
disp(['Cro1 (lbmol/ft^3): ' sprintf('%1.8e ',C2(1))]);
disp(['Tro1 (R)          : ' sprintf('%1.8e ',C2(2))]);
disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C2(3))]);
disp(['Cp (lbmol/ft^3)  : ' sprintf('%1.8e ',C2(4))]);
disp(['Tp (R)           : ' sprintf('%1.8e ',C2(5))]);
disp(['Tjo2 (R)         : ' sprintf('%1.8e ',C2(6))]);
disp(['Itro1 (R.h)      : ' sprintf('%1.8e ',C2(7))]);
disp(['Itro2 (R.h)      : ' sprintf('%1.8e ',C2(8))]);
disp(['mu (R^2.h)       : ' sprintf('%1.8e ',C2(9))]);
disp(' ');
disp('*Nota: La respuesta dinámica en todo el periodo de
estudio')
disp([' se encuentran graficados en los archivos ' I2
' _...jpeg'])
disp('          en la carpeta:')
disp(['          ' cd '\ ' D1])
close all
toc
disp(' ')
disp('****FIN DEL REPORTE')
diary off
save([D1 '\out.mat'])
end
%%%%%%%%%%FIN DE
OPTSTATSER

```

**ANEXO M. Cálculos típicos del balance de masa y energía de la de dos reactores mezcla completa en paralelo**

Balance de masa en p

$$F_{r1} \cdot \rho_{r1} + F_{r2} \cdot \rho_{r2} = F_p \cdot \rho_p \quad (\rho_{r1} \cong \rho_{r2} \cong \rho_p)$$

$$\Rightarrow F_{r1} = F_p - F_{r2}$$

Balance de masa en f

$$F_{r1} \cdot \rho_{r1} + F_{r2} \cdot \rho_{r2} = F_f \cdot \rho_f \quad (\rho_{r1} \cong \rho_{r2} \cong \rho_f)$$

$$\Rightarrow F_{r1} = F_f - F_{r2}$$

Balance de molar de a en p

$$F_{r1} \cdot C_{ro1} + F_{r2} \cdot C_{ro2} = F_p \cdot C_p$$

Balance energetico en p:

$$F_{r1} \cdot h_{ro1} \cdot \rho_{ro1} + F_{r2} \cdot h_{ro2} \cdot \rho_{ro2} = F_p \cdot h_p \cdot \rho_p \Rightarrow F_{r1} \cdot h_{ro1} + F_{r2} \cdot h_{ro2} = F_p \cdot h_p$$

Donde:

$$h_{ro1} = C_{pro1} \cdot (T_{ro1} - T_{ref}); \quad h_{ro2} = C_{pro2} \cdot (T_{ro2} - T_{ref}); \quad h_p = C_p \cdot (T_p - T_{ref})$$

Sustituyendo se tiene:

$$F_{r1} \cdot Cp_{ro1} \cdot (T_{ro1} - T_{ref}) + F_{r2} \cdot Cp_{ro2} \cdot (T_{ro2} - T_{ref}) = F_p \cdot Cp \cdot (T_p - T_{ref})$$

Por encontrarse en fase líquida se tiene

$$Cp_{ro1} \cong Cp_{ro2} \cong Cp$$

Entonces:

$$T_{ro1} \cdot F_{r1} + T_{ro2} \cdot F_{r2} = T_p \cdot F_p + T_{ref} \cdot (F_{r1} + F_{r2} - F_p)$$

Del balance de masa en p :

$$F_{r1} + F_{r2} - F_p = 0$$

Finalmente tenemos :

$$T_{ro1} \cdot F_{r1} + T_{ro2} \cdot F_{r2} = T_p \cdot F_p$$

Balance molar por componente de a en el reactor 1

$$\frac{d(C_{ro1} \cdot V_{r1})}{dt} = F_{r1} \cdot C_f - F_{r1} \cdot C_{ro1} - r_{A1} \cdot V_{r1} \quad \text{Donde } r_{A1} = k_1 \cdot C_A$$

$$\Rightarrow V_{r1} \cdot \frac{dC_{ro1}}{dt} = F_{r1} \cdot C_f - F_{r1} \cdot C_{ro1} - k_1 \cdot C_{ro1} \cdot V_{r1}$$

Balance molar por componente de A en el reactor 2

$$\frac{d(C_{ro2} \cdot V_{r2})}{dt} = F_{r2} \cdot C_f - F_{r2} \cdot C_{ro2} - r_{A2} \cdot V_{r2} \quad \text{Donde } r_{A2} = k_2 \cdot C_A$$

$$\Rightarrow V_{r2} \cdot \frac{dC_{ro2}}{dt} = F_{r2} \cdot C_f - F_{r2} \cdot C_{ro2} - k_2 \cdot C_{ro2} \cdot V_{r2}$$

Balance de energía en el reactor 1

$$\frac{dU}{dt} = F_{r1} \cdot \rho_f \cdot h_f - F_{r1} \cdot \rho_{ro1} \cdot h_{ro1} - \Delta H \cdot r_A \cdot V_{r1} - Q$$

Donde:

$$U = V_{r1} \cdot \rho \cdot Cv \cdot (T_{ro1} - T_{ref}); \quad h_{ro1} = Cp \cdot (T_{ro1} - T_{ref}); \quad h_f = Cp \cdot (T_f - T_{ref})$$

$$Q = U \cdot A_1 \cdot (T_{ro1} - T_{jo1}); \quad r_A = k_1 \cdot C_{ro1}$$

Sustituyendo tenemos :

$$\Rightarrow V_{r1} \cdot \rho \cdot Cv \cdot \frac{dT_{ro1}}{dt} = F_{r1} \cdot \rho \cdot Cp \cdot (T_f - T_{ref}) - F_{r1} \cdot \rho \cdot Cp \cdot (T_{ro1} - T_{ref}) - U \cdot A_1 \cdot (T_{ro1} - T_{jo1}) - \Delta H \cdot k_1 \cdot C_{ro1} \cdot V_{r1}$$

$$\Rightarrow V_{r1} \cdot \frac{dT_{ro1}}{dt} = F_{r1} \cdot T_f - F_{r1} \cdot T_{ro1} + F_{r1} \cdot T_{ref} - F_{r1} \cdot T_{ref} - \frac{U \cdot A_1}{\rho \cdot Cp} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_1 \cdot C_{ro1} \cdot V_{r1}$$

$$\Rightarrow V_{r1} \cdot \frac{dT_{ro1}}{dt} = F_{r1} \cdot T_f - F_{r1} \cdot T_{ro1} - \frac{U \cdot A_1}{\rho \cdot Cp} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_1 \cdot C_{ro1} \cdot V_{r1}$$

Balance de energía en el reactor 2

$$\frac{dU}{dt} = F_{r2} \cdot \rho_f \cdot h_f - F_{r2} \cdot \rho_{ro2} \cdot h_{ro2} - \Delta H \cdot r_A \cdot V_{r2} - Q$$

Donde:

$$U = V_{r2} \cdot \rho \cdot C_v \cdot (T_{ro2} - T_{ref}); \quad h_f = Cp \cdot (T_f - T_{ref}); \quad h_{ro2} = Cp \cdot (T_{ro2} - T_{ref})$$

$$Q = U \cdot A_2 \cdot (T_{ro2} - T_{jo2}); \quad r_A = k_2 \cdot C_{ro2}$$

Sustituyendo tenemos :

$$V_{r2} \cdot \frac{dT_{ro2}}{dt} = F_{r2} \cdot T_f - F_{r2} \cdot T_{ro2} - \frac{U \cdot A_2}{\rho \cdot Cp} \cdot (T_{ro2} - T_{jo2}) - \frac{\Delta H}{\rho \cdot Cp} k_2 \cdot C_{ro2} \cdot V_{r2}$$

Balance de energía en la chaqueta 1

$$\frac{dU}{dt} = F_{j1} \cdot \rho_{j1} \cdot h_{ji1} - F_{j1} \cdot \rho_{j1} \cdot h_{jo1} + Q$$

Donde:

$$U = V_{j1} \cdot \rho_j \cdot C_v \cdot (T_{jo1} - T_{ref}); \quad h_{ji1} = Cp_j \cdot (T_{ji1} - T_{ref}); \quad h_{jo1} = C_{pj} \cdot (T_{jo1} - T_{ref})$$

$$Q = U \cdot A_1 \cdot (T_{ro1} - T_{jo1})$$

Sustituyendo tenemos :

$$\Rightarrow V_{j1} \cdot \rho_j \cdot C_v \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot \rho_j \cdot Cp_j \cdot (T_{ji1} - T_{ref}) - F_{j1} \cdot \rho_j \cdot C_{pj} \cdot (T_{jo1} - T_{ref}) + U \cdot A_1 \cdot (T_{ro1} - T_{jo1})$$

$$\Rightarrow V_{j1} \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + F_{j1} \cdot T_{ref} - F_{j1} \cdot T_{ref} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1})$$

$$\Rightarrow V_{j1} \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1})$$

Balance de energía en la chaqueta 2

$$\frac{dU}{dt} = F_{j2} \cdot \rho_j \cdot h_{ji2} - F_{j2} \cdot \rho_j \cdot h_{jo2} + Q$$

Donde:

$$U = V_{j2} \cdot \rho_j \cdot C_{vj} \cdot (T_{jo2} - T_{ref}); \quad h_{ji2} = C_{pj} \cdot (T_{ji2} - T_{ref}); \quad h_{jo1} = C_{pj} \cdot (T_{jo2} - T_{ref})$$

$$Q = U \cdot A_2 \cdot (T_{ro2} - T_{jo2})$$

Sustituyendo tenemos:

$$\Rightarrow V_{j2} \cdot \rho_j \cdot C_{vj} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot \rho_j \cdot C_{pj} \cdot (T_{ji1} - T_{ref}) - F_{j1} \cdot \rho_j \cdot C_{pj} \cdot (T_{jo2} - T_{ref}) + U \cdot A_2 \cdot (T_{ro2} - T_{jo2})$$

$$\Rightarrow V_{j2} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + F_{j2} \cdot T_{ref} - F_{j2} \cdot T_{ref} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_{ro2} - T_{jo2})$$

$$\Rightarrow V_{j2} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_{ro2} - T_{jo2})$$

### Resumen de ecuaciones del modelado matemático para la disposición de los reactores en paralelo

- 1)  $F_{r1} = F_p - F_{r2}$
- 2)  $F_{r1} = F_f - F_{r2}$
- 3)  $F_{r1} \cdot C_{ro1} + F_{r2} \cdot C_{ro2} = F_p \cdot C_p$
- 4)  $T_{ro1} \cdot F_{r1} + T_{ro2} \cdot F_{r2} = T_p \cdot F_p$

$$5) V_{r1} \cdot \frac{dC_{ro1}}{dt} = F_{r1} \cdot C_f - F_{r1} \cdot C_{ro1} - k_1 \cdot C_{ro1} \cdot V_{r1}$$

$$6) V_{r2} \cdot \frac{dC_{ro2}}{dt} = F_{r2} \cdot C_f - F_{r2} \cdot C_{ro2} - k_2 \cdot C_{ro2} \cdot V_{r2}$$

$$7) V_{r1} \cdot \frac{dT_{ro1}}{dt} = F_{r1} \cdot T_f - F_{r1} \cdot T_{ro1} - \frac{U \cdot A_1}{\rho \cdot Cp} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot Cp} k_1 \cdot C_{ro1} \cdot V_{r1}$$

$$8) V_{r2} \cdot \frac{dT_{ro2}}{dt} = F_{r2} \cdot T_f - F_{r2} \cdot T_{ro2} - \frac{U \cdot A_2}{\rho \cdot Cp} \cdot (T_{ro2} - T_{jo2}) - \frac{\Delta H}{\rho \cdot Cp} k_2 \cdot C_{ro2} \cdot V_{r2}$$

$$9) V_{j1} \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1})$$

$$10) V_{j2} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_{ro2} - T_{jo2})$$

$$11) k_1 = k_0 \cdot e^{\frac{-E}{R \cdot T_{ro1}}}$$

$$12) k_2 = k_0 \cdot e^{\frac{-E}{R \cdot T_{ro2}}}$$

## ANEXO N. Cálculos típicos del balance de masa y energía de la de dos reactores mezcla completa en serie

Balance de masa en f



$$F_f \cdot \rho_f + F_b \cdot \rho_p = F_r \cdot \rho_r \quad (\rho_r \cong \rho_f \cong \rho_p)$$

$$\Rightarrow F_r = F_f + F_b$$

Balance de masa en p

$$F_r \cdot \rho_r = F_p \cdot \rho_p + F_b \cdot \rho_b \quad (\rho_r \cong \rho_f \cong \rho_p)$$

$$\Rightarrow F_r = F_p + F_b$$

Balance de molar de a en f

$$F_f \cdot C_f + F_b \cdot C_p = F_r \cdot C_{ri1}$$

Balance energetico en f

$$F_f \cdot h_f \cdot \rho_f + F_b \cdot h_b \cdot \rho_b = F_r \cdot h_{r1} \cdot \rho_r \Rightarrow F_f \cdot h_f + F_b \cdot h_b = F_r \cdot h_{r1}$$

$$F_f \cdot h_f \cdot \rho_f + F_b \cdot h_b \cdot \rho_b = F_r \cdot h_{r1} \cdot \rho_r \Rightarrow F_f \cdot h_f + F_b \cdot h_b = F_r \cdot h_{r1}$$

Donde:

$$h_f = C_{pf} \cdot (T_f - T_{ref}); \quad h_b = C_{pb} \cdot (T_p - T_{ref}); \quad h_{r1} = C_{pr} \cdot (T_{ri1} - T_{ref})$$

Sustituyendo se tiene:

$$F_f \cdot Cp_f \cdot (T_f - T_{ref}) + F_b \cdot Cp_b \cdot (T_p - T_{ref}) = F_r \cdot Cp_r \cdot (T_{ri1} - T_{ref})$$

Por encontrarse en fase liquida se tiene

$$Cp_b \cong Cp_f \cong Cp_r$$

Entonces:

$$T_f \cdot F_f + T_p \cdot F_b = T_{ri1} \cdot F_r + T_{ref} \cdot (F_f + F_b - F_r)$$

Del balance de masa en F :

$$F_f + F_b - F_r = 0$$

Finalmente tenemos :

$$T_f \cdot F_f + T_p \cdot F_b = T_{ri1} \cdot F_r$$

Balance molar por componente de a en el reactor 1

$$\frac{d(C_{ro1} \cdot V_{r1})}{dt} = F_r \cdot C_{ri1} - F_r \cdot C_{ro1} - r_{A1} \cdot V_{r1} \quad \text{Donde } r_A = k_1 \cdot C_A$$

$$\Rightarrow V_{r1} \cdot \frac{dC_{ro1}}{dt} = F_r \cdot C_{ri1} - F_r \cdot C_{ro1} - k_1 \cdot C_{ro1} \cdot V_{r1}$$

Balance molar por componente de a en el reactor 2

$$\frac{d(C_p \cdot V_{r2})}{dt} = F_r \cdot C_{ro1} - F_r \cdot C_p - r_{A2} \cdot V_{r2} \quad \text{Donde } r_A = k_1 \cdot C_A$$

$$\Rightarrow V_{r2} \cdot \frac{dC_p}{dt} = F_r \cdot C_{ro1} - F_r \cdot C_p - k_2 \cdot C_p \cdot V_{r2}$$

Balance de energia en el reactor 1

$$\frac{dU}{dt} = F_r \cdot \rho_r \cdot h_{ri1} - F_r \cdot \rho_r \cdot h_{ro1} - \Delta H \cdot r_A \cdot V_{r1} - Q$$

Donde:

$$U = V_{r1} \cdot \rho \cdot C_v \cdot (T_{ro1} - T_{ref}); \quad h_{ro1} = Cp \cdot (T_{ro1} - T_{ref}); \quad h_{ri1} = Cp \cdot (T_{ri1} - T_{ref})$$

$$Q = U \cdot A_1 \cdot (T_{ro1} - T_{jo1}); \quad r_A = k \cdot C_{ro1}$$

Sustituyendo tenemos :

$$\Rightarrow V_{r1} \cdot \rho \cdot C_v \cdot \frac{dT_{ro1}}{dt} = F_r \cdot \rho \cdot Cp \cdot (T_{ri1} - T_{ref}) - F_r \cdot \rho \cdot Cp \cdot (T_{ro1} - T_{ref}) - U \cdot A_1 \cdot (T_{ro1} - T_{jo1}) - \Delta H \cdot k_1 \cdot C_{ro1} \cdot V_{r1}$$

$$\Rightarrow V_{r1} \cdot \frac{dT_{ro1}}{dt} = F_r \cdot T_{ri1} - F_r \cdot T_{ro1} + F_r \cdot T_{ref} - F_r \cdot T_{ref} - \frac{U \cdot A_1}{\rho \cdot Cp} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_1 \cdot C_{ro1} \cdot V_{r1}$$

$$\Rightarrow V_{r1} \cdot \frac{dT_{ro1}}{dt} = F_r \cdot T_{ri1} - F_r \cdot T_{ro1} - \frac{U \cdot A_1}{\rho \cdot Cp} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_1 \cdot C_{ro1} \cdot V_{r1}$$

Balance de energia en el reactor 2

$$\frac{dU}{dt} = F_r \cdot \rho_r \cdot h_{ro1} - F_r \cdot \rho_r \cdot h_{ro2} - \Delta H \cdot r_A \cdot V_{r2} - Q$$

Donde:

$$U = V_{r2} \cdot \rho \cdot C_v \cdot (T_p - T_{ref}); \quad h_{ro1} = Cp \cdot (T_{ro1} - T_{ref}); \quad h_{ro2} = Cp \cdot (T_p - T_{ref})$$

$$Q = U \cdot A_1 \cdot (T_p - T_{jo2}); \quad r_A = k_2 \cdot C_p$$

Sustituyendo tenemos :

$$\Rightarrow V_{r2} \cdot \rho \cdot C_v \cdot \frac{dT_p}{dt} = F_r \cdot \rho \cdot Cp \cdot (T_{ro1} - T_{ref}) - F_r \cdot \rho \cdot Cp \cdot (T_p - T_{ref}) - U \cdot A_2 \cdot (T_p - T_{jo2}) - \Delta H \cdot k_2 \cdot C_p \cdot V_{r2}$$

$$\Rightarrow V_{r2} \cdot \frac{dT_p}{dt} = F_r \cdot T_{ro1} - F_r \cdot T_p + F_r \cdot T_{ref} - F_r \cdot T_{ref} - \frac{U \cdot A_2}{\rho \cdot Cp} \cdot (T_p - T_{jo2}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_2 \cdot C_p \cdot V_{r2}$$

$$\Rightarrow V_{r2} \cdot \frac{dT_p}{dt} = F_r \cdot T_{ro1} - F_r \cdot T_p - \frac{U \cdot A_2}{\rho \cdot Cp} \cdot (T_p - T_{jo2}) - \frac{\Delta H}{\rho \cdot Cp} \cdot k_2 \cdot C_p \cdot V_{r2}$$

Balance de energia en la chaqueta 1

$$\frac{dU}{dt} = F_{j1} \cdot \rho_{j1} \cdot h_{ji1} - F_{j1} \cdot \rho_{j1} \cdot h_{jo1} + Q$$

Donde:

$$U = V_{j1} \cdot \rho_j \cdot C_v \cdot (T_{jo1} - T_{ref}); \quad h_{ji1} = Cp_j \cdot (T_{ji1} - T_{ref}); \quad h_{jo1} = C_{pj} \cdot (T_{jo1} - T_{ref})$$

$$Q = U \cdot A_1 \cdot (T_{ro1} - T_{jo1})$$

Sustituyendo tenemos :

$$\Rightarrow V_{j1} \cdot \rho_j \cdot C_v \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot \rho_j \cdot Cp_j \cdot (T_{ji1} - T_{ref}) - F_{j1} \cdot \rho_j \cdot Cp_j \cdot (T_{jo1} - T_{ref}) + U \cdot A_1 \cdot (T_{ro1} - T_{jo1})$$

$$\Rightarrow V_{j1} \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + F_{j1} \cdot T_{ref} - F_{j1} \cdot T_{ref} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1})$$

$$\Rightarrow V_{j1} \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1})$$

Balance de energia en la chaqueta 2

$$\frac{dU}{dt} = F_{j2} \cdot \rho_j \cdot h_{ji2} - F_{j2} \cdot \rho_j \cdot h_{jo2} + Q$$

Donde:

$$U = V_{j2} \cdot \rho_j \cdot C_{vj} \cdot (T_{jo2} - T_{ref}); \quad h_{ji2} = C_{pj} \cdot (T_{ji2} - T_{ref}); \quad h_{jo1} = C_{pj} \cdot (T_{jo2} - T_{ref})$$

$$Q = U \cdot A_2 \cdot (T_p - T_{jo2})$$

Sustituyendo tenemos :

$$\Rightarrow V_{j2} \cdot \rho_j \cdot C_{vj} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot \rho_j \cdot C_{pj} \cdot (T_{ji1} - T_{ref}) - F_{j1} \cdot \rho_j \cdot C_{pj} \cdot (T_{jo2} - T_{ref}) + U \cdot A_2 \cdot (T_{ro2} - T_{jo2})$$

$$\Rightarrow V_{j2} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + F_{j2} \cdot T_{ref} - F_{j2} \cdot T_{ref} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_p - T_{jo2})$$

$$\Rightarrow V_{j2} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_p - T_{jo2})$$

**Resumen de ecuaciones del modelado matemático para la disposición de los reactores en serie**

$$1) F_r = F_f + F_b$$

$$2) F_r = F_p + F_b$$

$$3) F_f \cdot C_f + F_b \cdot C_p = F_r \cdot C_{ri1}$$

$$4) T_f \cdot F_f + T_p \cdot F_b = T_{ri1} \cdot F_r$$

$$5) V_{r1} \cdot \frac{dC_{ro1}}{dt} = F_r \cdot C_{ri1} - F_r \cdot C_{ro1} - k_1 \cdot C_{ro1} \cdot V_{r1}$$

$$6) V_{r2} \cdot \frac{dC_p}{dt} = F_r \cdot C_{ro1} - F_r \cdot C_p - k_2 \cdot C_p \cdot V_{r2}$$

$$7) V_{r1} \cdot \frac{dT_{ro1}}{dt} = F_r \cdot T_{ri1} - F_r \cdot T_{ro1} - \frac{U \cdot A_1}{\rho \cdot C_p} \cdot (T_{ro1} - T_{jo1}) - \frac{\Delta H}{\rho \cdot C_p} k_1 \cdot C_{ro1} \cdot V_{r1}$$

$$8) V_{r2} \cdot \frac{dT_p}{dt} = F_r \cdot T_{ro1} - F_r \cdot T_p - \frac{U \cdot A_2}{\rho \cdot C_p} \cdot (T_p - T_{jo2}) - \frac{\Delta H}{\rho \cdot C_p} k_2 \cdot C_p \cdot V_{r2}$$

$$9) V_{j1} \cdot \frac{dT_{jo1}}{dt} = F_{j1} \cdot T_{j1} - F_{j1} \cdot T_{jo1} + \frac{U \cdot A_1}{\rho_j \cdot C_{pj}} \cdot (T_{ro1} - T_{jo1})$$

$$10) V_{j2} \cdot \frac{dT_{jo2}}{dt} = F_{j2} \cdot T_{j2} - F_{j2} \cdot T_{jo2} + \frac{U \cdot A_2}{\rho_j \cdot C_{pj}} \cdot (T_p - T_{jo2})$$

$$11) k_1 = k_0 \cdot e^{\frac{-E}{R \cdot T_{ro1}}}$$

$$12) k_2 = k_0 \cdot e^{\frac{-E}{R \cdot T_p}}$$

**ANEXO O. Programación en MATLAB para el cálculo de las variables de  
diseño en estado estacionario para topología de reactores en paralelo, usando  
SA.**

```

function OptEstatPar_SA
warning off
%
% Script para el cálculo de las variables de diseño en estado
% ESTACIONARIO de dos reactores mezcla completa con chaqueta de
% enfriamiento dispuestos en forma PARALELA. Se procesa un
% compuesto A
% que reacciona a una velocidad de reacción de primer orden. Éste
% cálculo
% es obtenido a partir de la optimización minimizando el costo de
% diseño
% (en base al diámetro-altura de los reactores y el flujo de
% refrigerante).
%
%
% Éste script utiliza la función SA del toolbox de optimización de
% MatLab. Además consta de las siguientes funciones:
%
% Costo(VOD):          Calcula el costo de diseño de los reactores
% para las variables de diseño VOD.
% DynReac(v,pe):      Realiza la simulación dinámica de los
% reactores
%                      para las variables de diseño v y los
% parámetros del
%                      sistema pe.
% dSeg(t,c,ve, p):    Calcula los valores de la ecuaciones
% diferenciales
%                      correspondiente al modelo de simulación. t
% (tiempo), c
%                      (variables dinámicas), ve (variables de
% diseño), p
%                      (parámetros del sistema)
% StatReac(vs,ps):    Simula el reactor en estado estacionario.
%                      vs (variables de diseño), ps (parámetros del
% sistema)
% error(c):           Función utilizada para resolver el sistema de
%                      ecuaciones algebraicas en la simulación en
% estado
%                      estacionario.
% Ti(tp):             Función de perturbación para la temperatura de
%                      alimentación al reactor.
%
%
% Variables de Diseño (ve)
% 1    2    3    4    5    6    7    8    9    10   11   12
% 13

```

```

% Hr1, Hr2, Rf, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global E I2 D1 pe
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S'; % INDICADOR
E = 0.001; % VALOR MAXIMO DE CONTROLABILIDAD
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_']
num2str(fix(1000.*E))];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% REACTORES EN PARALELO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN EN ESTADO ESTACIONARIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
% Rango del problema
LB = [0.5, 0.5, 0, 1, 1, 1, 1];
UB = [100, 100, 1, 100, 100, 500, 500];
% Función Objetivo
Problem.objective=@(v) CostoRest(v,pe,LB,UB);
% Number of Variables
Problem.nvars= 7;
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
Problem.lb= LB; %[0.05, 0.05, 0, 0.1, 0.1, 0.1, 0.1];
Problem.ub= UB; %[100, 100, 1, 100, 100, 500, 500];
Problem.x0= Problem.lb + (Problem.ub-
Problem.lb).*rand(1,Problem.nvars);
Problem.solver='simulannealbnd';
% OPCIONES DE OPTIMIZACION SA
disp(' ')
options = saoptimset;
options.MaxIter = 20000;
options.Display='iter';
options.MaxFunEvals = Inf;
options.InitialTemperature = 1000000;
options.TemperatureFcn=@temperatureexp;
options.PlotFcns=@saplotbestf;
options.ReannealInterval = 500;
Problem.options= options;

```



```

disp('Opciones del SA')
disp(options)
% OPTIMIZADOR ---> Simulated Annealing
[X1,FVAL,REASON,OUTPUT] = simulannealbnd(Problem);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp('RESULTADOS SA')
disp(sprintf('%10.6f ',X1));          % VARIABLES DE DISEÑO
disp(sprintf('%10.6f ',FVAL));        % FUNCIÓN COSTO
disp(REASON);                        % CONDICIÓN DE SALIDA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MODULO          PUBLICACIÓN          DE          RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
R = input('¿Desea Generar los Resultados?(s/n) ','s');
ve = [X1, 0, 0, 1.2,1.2];

if or(R == 'S', R == 's')
    mkdir(D1);
    copyfile('temp.txt',D1,'f')
    I2 = 'Est';
    diary([D1 '\out.txt'])
    clc
    disp(' ')
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%          RESULTADOS          OPTCONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['FECHA: ' date])
    disp(' ');
    disp('OPTIMIZACION EN ESTADO ESTACIONARIO')
    disp(' ')
    cost = Costo(ve);
    disp(['COSTO CAPITAL ($) : ' sprintf('%10.12g ',cost(1))]);
    disp(['COSTO UTILIDAD($) : ' sprintf('%10.12g ',cost(2))]);
    disp(['COSTO TOTAL ($) : ' sprintf('%10.12g ',cost(3))]);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['Hr1 (ft)          : ' sprintf('%1.8e ',ve(1))]);
    disp(['Hr2 (ft)          : ' sprintf('%1.8e ',ve(2))]);
    disp(['Fr1/Ff           : ' sprintf('%1.8e ',ve(3))]);
    disp(['Dr1 (ft)          : ' sprintf('%1.8e ',ve(4))]);
    disp(['Dr2 (ft)          : ' sprintf('%1.8e ',ve(5))]);
    disp(['Fjn1 (ft^3/h)     : ' sprintf('%1.8e ',ve(6))]);
    disp(['Fjn2 (ft^3/h)     : ' sprintf('%1.8e ',ve(7))]);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    C1 = StatReac(ve,pe);
    disp(['Fr1 (ft^3/h)      : ' sprintf('%1.8e ',C1(1))]);
    disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C1(2))]);
    disp(['Trol (R)          : ' sprintf('%1.8e ',C1(3))]);
    disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C1(4))]);
    disp(['Fr2 (ft^3/h)      : ' sprintf('%1.8e ',C1(5))]);

```

```

disp(['Cro2 (lbmol/ft^3) : ' sprintf('%1.8e ',C1(6))]);
disp(['Tro2 (R)          : ' sprintf('%1.8e ',C1(7))]);
disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C1(8))]);
disp(['Cp (lbmol/ft^3)  : ' sprintf('%1.8e ',C1(9))]);
disp(['Tp (R)           : ' sprintf('%1.8e ',C1(10))]);
disp(' ');
disp('%%%%%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%%%%%');
disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
C2 = DynReac(ve,pe);
disp(['Cro1 (lbmol/ft^3): ' sprintf('%1.8e ',C2(1))]);
disp(['Tro1 (R)          : ' sprintf('%1.8e ',C2(2))]);
disp(['Tjo1 (R)          : ' sprintf('%1.8e ',C2(3))]);
disp(['Cro2 (lbmol/ft^3): ' sprintf('%1.8e ',C2(4))]);
disp(['Tro2 (R)          : ' sprintf('%1.8e ',C2(5))]);
disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C2(6))]);
disp(['Itro1 (R.h)       : ' sprintf('%1.8e ',C2(7))]);
disp(['Itro2 (R.h)       : ' sprintf('%1.8e ',C2(8))]);
disp(['mu (R^2.h)        : ' sprintf('%1.8e ',C2(9))]);
disp(['Cp (lbmol/ft^3)  : ' sprintf('%1.8e ',C2(10))]);
disp(['Tp (R)           : ' sprintf('%1.8e ',C2(11))]);
disp(' ');
disp('*Nota:  La respuesta dinámica en todo el periodo de
estudio')
disp(['          se encuentran graficados en los archivos ' I2
' _...jpeg'])
disp('          en la carpeta:')
disp(['          ' cd '\ D1])
close all
toc
disp(' ')
disp('****FIN DEL REPORTE')
diary off
save([D1 'out.mat'])
end
%%%%%%%%%%%%%%FIN
OPTSTATPAR
DE

```

**ANEXO P: Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie, usando SA**

```

function OptEstatSer_SA
warning off
%
% Script para el cálculo de las variables de diseño en estado
% ESTACIONARIO de dos reactores mezcla completa con chaqueta de
% enfriamiento dispuestos en SERIE. Se procesa un compuesto A
% que reacciona a una velocidad de reacción de primer orden. Éste
% cálculo
% es obtenido a partir de la optimización minimizando el costo de
% diseño
% (en base al diámetro-altura de los reactores y el flujo de
% refrigerante).
%
%
% Éste script utiliza la función SA del toolbox de optimización de
% MatLab. Además consta de las siguientes funciones:
%
% Costo(VOD):          Calcula el costo de diseño de los reactores
para las
%                      variables de diseño VOD.
% DynReac(v,pe):      Realiza la simulación dinámica de los
reactores
%                      para las variables de diseño v y los
parámetros del
%                      sistema pe.
% dSeg(t,c,ve, p):    Calcula los valores de la ecuaciones
diferenciales
%                      correspondiente al modelo de simulación. t
(tiempo), c
%                      (variables dinámicas), ve (variables de
diseño), p
%                      (parámetros del sistema)
% StatReac(vs,ps):    Simula el reactor en estado estacionario.
%                      vs (variables de diseño), ps (parámetros del
sistema)
% error(c):           Función utilizada para resolver el sistema de
%                      ecuaciones algebraicas en la simulación en
estado
%                      estacionario.
% Ti(tp):             Función de perturbación para la temperatura de
%                      alimentación al reactor.
%
%
% Variables de Diseño (ve)
% 1    2    3    4    5    6    7    8    9    10   11   12
% 13

```

```

% Hr1, Hr2, Rf, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global E I2 D1 pe
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S'; % INDICADOR
E = 0.001; % VALOR MAXIMO DE CONTROLABILIDAD
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_']
num2str(fix(1000.*E))];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% REACTORES EN SERIE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN EN ESTADO ESTACIONARIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
% Rango del problema
LB = [0.5, 0.5, 0, 1, 1, 1, 1];
UB = [100, 100, 1, 100, 100, 500, 500];
% Función Objetivo
Problem.objective=@(v) CostoRest(v,pe, LB,UB);
%Number of Variables
Problem.nvars= 7;
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
Problem.lb= LB;
Problem.ub= UB;
% VALOR INICIAL OBTENIDO ALEATORIAMENTE
Problem.x0= Problem.lb + (Problem.ub-
Problem.lb).*rand(1,Problem.nvars);
Problem.solver='simulannealbnd';
% OPCIONES DE OPTIMIZACION SA
disp(' ')
options = saoptimset;
options.MaxIter = 200000;
options.Display='iter';
options.MaxFunEvals = Inf;
options.InitialTemperature = 1000000;
options.TemperatureFcn=@temperatureexp;
options.PlotFcns=@saplotbestf;
options.ReannealInterval = 500;

```

```

Problem.options= options;
disp('Opciones del SA')
disp(options)
% OPTIMIZADOR ---> Simulated Annealing
[X1,FVAL,REASON,OUTPUT] = simulannealbnd(Problem);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp('RESULTADOS SA')
disp(sprintf('%10.6f ',X1));           % VARIABLES DE DISEÑO
disp(sprintf('%10.6f ',FVAL));         % FUNCIÓN COSTO
disp(REASON);                          % CONDICIÓN DE SALIDA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MODULO          PUBLICACIÓN          DE          RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
R = input('¿Desea Generar los Resultados?(s/n) ','s');
% Datos Anteriores Estado Estacionario
ve = [X1, 0, 0, 1.2,1.2];

if or(R == 'S', R == 's')
mkdir(D1);
copyfile('temp.txt',D1,'f')
I2 = 'Est';
diary([D1 '\out.txt'])
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%          RESULTADOS          OPTCONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['FECHA: ' date])
disp(' ');
disp('OPTIMIZACION EN ESTADO ESTACIONARIO')
disp(' ')
cost = Costo(ve);
disp(['COSTO CAPITAL ($) : ' sprintf('%10.12g ',cost(1))]);
disp(['COSTO UTILIDAD($) : ' sprintf('%10.12g ',cost(2))]);
disp(['COSTO TOTAL ($) : ' sprintf('%10.12g ',cost(3))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['Hr1 (ft) : ' sprintf('%1.8e ',ve(1))]);
disp(['Hr2 (ft) : ' sprintf('%1.8e ',ve(2))]);
disp(['Fb (ft^3/h) : ' sprintf('%1.8e ',ve(3))]);
disp(['Dr1 (ft) : ' sprintf('%1.8e ',ve(4))]);
disp(['Dr2 (ft) : ' sprintf('%1.8e ',ve(5))]);
disp(['Fjn1 (ft^3/h) : ' sprintf('%1.8e ',ve(6))]);
disp(['Fjn2 (ft^3/h) : ' sprintf('%1.8e ',ve(7))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
C1 = StatReac(ve,pe);
disp(['Fr (ft^3/h) : ' sprintf('%1.8e ',C1(1))]);
disp(['Cri1 (lbmol/ft^3): ' sprintf('%1.8e ',C1(2))]);
disp(['Tri1 (R) : ' sprintf('%1.8e ',C1(3))]);

```

```

disp(['Crol (lbmol/ft^3): ' sprintf('%1.8e ',C1(4))]);
disp(['Trol (R)           : ' sprintf('%1.8e ',C1(5))]);
disp(['Tjo1 (R)           : ' sprintf('%1.8e ',C1(6))]);
disp(['Cp (lbmol/ft^3)   : ' sprintf('%1.8e ',C1(7))]);
disp(['Tp (R)            : ' sprintf('%1.8e ',C1(8))]);
disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C1(9))]);
disp(' ');
disp('%%%%%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%%%%%')
disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
C2 = DynReac(ve,pe);
disp(['Crol (lbmol/ft^3): ' sprintf('%1.8e ',C2(1))]);
disp(['Trol (R)           : ' sprintf('%1.8e ',C2(2))]);
disp(['Tjo1 (R)           : ' sprintf('%1.8e ',C2(3))]);
disp(['Cp (lbmol/ft^3)   : ' sprintf('%1.8e ',C2(4))]);
disp(['Tp (R)            : ' sprintf('%1.8e ',C2(5))]);
disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C2(6))]);
disp(['Itrol (R.h)        : ' sprintf('%1.8e ',C2(7))]);
disp(['Itro2 (R.h)        : ' sprintf('%1.8e ',C2(8))]);
disp(['mu (R^2.h)         : ' sprintf('%1.8e ',C2(9))]);
disp(' ');
disp('*Nota:  La respuesta dinámica en todo el periodo de
estudio')
disp([' se encuentran graficados en los archivos ' I2
'_.jpg'])
disp(' en la carpeta:')
disp([' cd '\ D1])
close all
toc
disp(' ')
disp('****FIN DEL REPORTE')
diary off
save([D1 '\out.mat'])
end
end
%%%%%%%%%%%%%%FIN
OPTSTATSER
DE

```

**ANEXO Q. Programación en Matlab R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo.**

**Optimización de control mediante SA**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%
%                                OPTIMIZACION      -      SOLO      CONTROL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
function OnlyControlPar_SA
% Variables de Diseño (ve)
% 1 2 3 4 5 6 7 8 9 10 11 12
13
% Hr1, Hr2, Fb, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
warning off
global I2 D1
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S'; % INDICADOR
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_OC'];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% REACTORES EN SERIE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN DEL CONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% PARAMETROS DE DISEÑO OBTENIDO DE LA OPTIMIZACION ESTACIONARIA
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
vd = [8.314162 1.985315 0.999994 23.055987 2.118712
...
39.344069 78.660678];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rango del problema
LB = [0, 0, 1e-1, 1e-1, 0.1];
UB = [1000, 1000, 10, 10, 10000];
% Función Objetivo
Problem.objective=@(v) Costo2(v,vd,pe, LB, UB);
%Number of Variables
Problem.nvars= 5;
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
Problem.lb= LB;
Problem.ub= UB;
Problem.x0= Problem.lb + (Problem.ub-
Problem.lb).*rand(1,Problem.nvars);
Problem.solver='simulannealbnd';
% OPCIONES DE OPTIMIZACION SA
disp(' ')
options = saoptimset;
options.MaxIter = 200000;
options.Display='iter';
options.MaxFunEvals = Inf;
options.InitialTemperature = 1000000;
options.TemperatureFcn=@temperatureexp;
options.PlotFcns=@saplotbestf;
options.ReannealInterval = 500;
Problem.options= options;
disp('Opciones del SA')
disp(options)
% OPTIMIZADOR ---> Simulated Annealing
[X,FVAL,REASON] = simulannealbnd(Problem);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp('RESULTADOS SA')
disp(sprintf('%10.6f ',X)); % VARIABLES DE DISEÑO
disp(sprintf('%10.6f ',FVAL)); % FUNCIÓN COSTO
disp(REASON); % CONDICIÓN DE SALIDA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
% PUBLICACION DE RESULTADOS
R = input('¿Desea Generar los Resultados?(s/n) ','s');
if or(R == 'S', R == 's')
mkdir(D1);
copyfile('temp.txt',D1,'f')
I2 = 'Estat';
diary([D1 '\out.txt'])
clc
disp(' ')

```



```

disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['FECHA: ' date])
disp(' ');
disp('OPTIMIZACION CONTROL')
disp(' ');
I2 = 'Dim';
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['Hr1 (ft)           : ' sprintf('%1.8e ',vd(1))]);
disp(['Hr2 (ft)           : ' sprintf('%1.8e ',vd(2))]);
disp(['Fb (ft^3/h)        : ' sprintf('%1.8e ',vd(3))]);
disp(['Dr1 (ft)           : ' sprintf('%1.8e ',vd(4))]);
disp(['Dr2 (ft)           : ' sprintf('%1.8e ',vd(5))]);
disp(['Fjn1 (ft^3/h)      : ' sprintf('%1.8e ',vd(6))]);
disp(['Fjn2 (ft^3/h)      : ' sprintf('%1.8e ',vd(7))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE CONTROL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['kj1 (ft^3/h.R)     : ' sprintf('%1.8e ',X(1))]);
disp(['kj2 (ft^3/h.R)     : ' sprintf('%1.8e ',X(2))]);
disp(['Tauj1 (h)           : ' sprintf('%1.8e ',X(3))]);
disp(['Tauj2 (h)           : ' sprintf('%1.8e ',X(4))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
C3 = StatReac([vd, X(1:4)],pe);
disp(['Fr (ft^3/h)        : ' sprintf('%1.8e ',C3(1))]);
disp(['Cril (lbmol/ft^3) : ' sprintf('%1.8e ',C3(2))]);
disp(['Tril (R)            : ' sprintf('%1.8e ',C3(3))]);
disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C3(4))]);
disp(['Tro1 (R)            : ' sprintf('%1.8e ',C3(5))]);
disp(['Tjo1 (R)            : ' sprintf('%1.8e ',C3(6))]);
disp(['Cp (lbmol/ft^3)    : ' sprintf('%1.8e ',C3(7))]);
disp(['Tp (R)              : ' sprintf('%1.8e ',C3(8))]);
disp(['Tjo2 (R)            : ' sprintf('%1.8e ',C3(9))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
C4 = DynReac2(X,vd,pe);
disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ',C4(1))]);
disp(['Tro1 (R)            : ' sprintf('%1.8e ',C4(2))]);
disp(['Tjo1 (R)            : ' sprintf('%1.8e ',C4(3))]);
disp(['Cp (lbmol/ft^3)    : ' sprintf('%1.8e ',C4(4))]);
disp(['Tp (R)              : ' sprintf('%1.8e ',C4(5))]);
disp(['Tjo2 (R)            : ' sprintf('%1.8e ',C4(6))]);
disp(['Itro1 (R.h)         : ' sprintf('%1.8e ',C4(7))]);
disp(['Itro2 (R.h)         : ' sprintf('%1.8e ',C4(8))]);
disp(['mu (R^2.h)          : ' sprintf('%1.8e ',C4(9))]);
disp(' ');
disp(' *Nota: La respuesta dinámica en todo el periodo de estudio')
disp(['          se encuentran graficados en los archivos ' I2
' _...jpeg'])
disp('          en la carpeta:')
disp(['          ' cd '\ D1])
disp(' ')

```

```

        close all
        toc
        disp(' ')
        disp('****FIN DEL REPORTE')
        diary off
        save([D1 '\out'])
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIN            DR
ONLYCONTROLPAR

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               FUNCION   OBJETIVO   -   SOLO   CONTROL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Fun, Costo, Res] = Costo2(xe,vd,pe,xl,xu)

%Tolerancia en la evaluación de las restricciones
TolCon = 1e-6;
% Factor de penalización
Penal = 1e6;
% Asegura que X, xl, xu sean vector columna
[dummy, n] = size(xe);
if n == 1
    xe = xe';
end
[dummy, n] = size(xl);
if n == 1
    xl = xl';
end
[dummy, n] = size(xu);
if n == 1
    xu = xu';
end
% Evalua las restricciones de frontera
Rbl = 1000*max(xl-xe);
Rbu = 1000*max(xe-xu);
% Evalua las restricciones lineales
% No hay restricciones de Desigualdad A*X - B <= 0
Rld = -1000.0;

% No hay restriccion lineal de igualdad
Rle = -1000.0;

% Evalua las restricciones No lineales
[C, Ceq] = feval('DynReac2',xe, vd, pe);

% Desigualdad
if isempty(C)
    Rnld = -1000.0;

```



**ANEXO R. Programación en Matlab R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en serie.**

**Optimización de control mediante SA**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                                OPTIMIZACION      -      SOLO      CONTROL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function OnlyControlSer_SA
% Variables de Diseño (ve)
% 1    2    3    4    5    6    7    8    9    10    11    12
13
% Hr1, Hr2, Fb, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1    2    3    4    5    6    7    8    9    10    11    12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
warning off
global I2 D1
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S';           % INDICADOR
tiempo = clock;    % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_SA_OC'];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      REACTORES      EN      SERIE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      OPTIMIZACIÓN      DEL      CONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PARAMETROS DE DISEÑO OBTENIDO DE LA OPTIMIZACION ESTACIONARIA
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
vd = [8.83188238e+000    3.27852631e+001    2.95944797e-001
2.23746448e+001 ...

```

```

1.00001073e+000 3.97724619e+001 1.74333159e+002];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rango del problema
LB = [0, 0, 1e-1, 1e-1, 0.1];
UB = [1000, 1000, 10, 10, 10000];
% Función Objetivo
Problem.objective=@(v) Costo2(v,vd,pe,LB,UB);
%Number of Variables
Problem.nvars= 5;
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
Problem.lb= LB;
Problem.ub= UB;
Problem.x0= Problem.lb + (Problem.ub-
Problem.lb).*rand(1,Problem.nvars);
Problem.solver='simulannealbnd';
% OPCIONES DE OPTIMIZACION
disp(' ')
options = saoptimset;
options.MaxIter = 200000;
options.Display='iter';
options.MaxFunEvals = Inf;
options.InitialTemperature = 1000000;
options.TemperatureFcn=@temperatureexp;
options.PlotFcns=@saplotbestf;
options.ReannealInterval = 500;
Problem.options= options;
disp('Opciones del SA')
disp(options)
% OPTIMIZADOR ---> Simulated Annealing
[X,FVAL,REASON] = simulannealbnd(Problem);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp('RESULTADOS SA')
disp(sprintf('%10.6f ',X)); % VARIABLES DE DISEÑO
disp(sprintf('%10.6f ',FVAL)); % FUNCIÓN COSTO
disp(REASON); % CONDICIÓN DE SALIDA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
% PUBLICACION DE RESULTADOS
R = input('¿Desea Generar los Resultados?(s/n) ','s');
if or(R == 'S', R == 's')
mkdir(D1);
copyfile('temp.txt',D1,'f')
I2 = 'Estat';
diary([D1 '\out.txt'])
clc

```

```

disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['FECHA: ' date])
disp(' ');
disp('OPTIMIZACION CONTROL')
disp(' ')
I2 = 'Dim';
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['Hr1 (ft)           : ' sprintf('%1.8e ',vd(1))]);
disp(['Hr2 (ft)           : ' sprintf('%1.8e ',vd(2))]);
disp(['Fb (ft^3/h)        : ' sprintf('%1.8e ',vd(3))]);
disp(['Dr1 (ft)           : ' sprintf('%1.8e ',vd(4))]);
disp(['Dr2 (ft)           : ' sprintf('%1.8e ',vd(5))]);
disp(['Fjn1 (ft^3/h)       : ' sprintf('%1.8e ',vd(6))]);
disp(['Fjn2 (ft^3/h)       : ' sprintf('%1.8e ',vd(7))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE CONTROL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp(['kj1 (ft^3/h.R)      : ' sprintf('%1.8e ',X(1))]);
disp(['kj2 (ft^3/h.R)      : ' sprintf('%1.8e ',X(2))]);
disp(['Tauj1 (h)           : ' sprintf('%1.8e ',X(3))]);
disp(['Tauj2 (h)           : ' sprintf('%1.8e ',X(4))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
C3 = StatReac([vd, X(1:4)],pe);
disp(['Fr (ft^3/h)         : ' sprintf('%1.8e ',C3(1))]);
disp(['Cri1 (lbmol/ft^3)   : ' sprintf('%1.8e ',C3(2))]);
disp(['Tri1 (R)             : ' sprintf('%1.8e ',C3(3))]);
disp(['Cro1 (lbmol/ft^3)   : ' sprintf('%1.8e ',C3(4))]);
disp(['Trol (R)             : ' sprintf('%1.8e ',C3(5))]);
disp(['Tjo1 (R)             : ' sprintf('%1.8e ',C3(6))]);
disp(['Cp (lbmol/ft^3)     : ' sprintf('%1.8e ',C3(7))]);
disp(['Tp (R)                : ' sprintf('%1.8e ',C3(8))]);
disp(['Tjo2 (R)             : ' sprintf('%1.8e ',C3(9))]);
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
C4 = DynReac2(X,vd,pe);
disp(['Cro1 (lbmol/ft^3)   : ' sprintf('%1.8e ',C4(1))]);
disp(['Trol (R)             : ' sprintf('%1.8e ',C4(2))]);
disp(['Tjo1 (R)             : ' sprintf('%1.8e ',C4(3))]);
disp(['Cp (lbmol/ft^3)     : ' sprintf('%1.8e ',C4(4))]);
disp(['Tp (R)                : ' sprintf('%1.8e ',C4(5))]);
disp(['Tjo2 (R)             : ' sprintf('%1.8e ',C4(6))]);
disp(['Itro1 (R.h)          : ' sprintf('%1.8e ',C4(7))]);
disp(['Itro2 (R.h)          : ' sprintf('%1.8e ',C4(8))]);
disp(['mu (R^2.h)          : ' sprintf('%1.8e ',C4(9))]);
disp(' ');
disp(' *Nota: La respuesta dinámica en todo el periodo de estudio')
disp(['          se encuentran graficados en los archivos ' I2
'_{...}.jpeg'])
disp('          en la carpeta:')
disp(['          ' cd '\ D1])

```

```

disp(' ')
close all
toc
disp(' ')
disp('****FIN DEL REPORTE')
diary off
save([D1 '\out'])
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIN            DR
ONLYCONTROLPAR

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                FUNCION  OBJETIVO  -  SOLO  CONTROL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Fun, Costo, Res] = Costo2(xe,vd,pe,xl,xu)

%Tolerancia en la evaluación de las restricciones
TolCon = 1e-6;
% Factor de penalización
Penal = 1e6;
% Asegura que X, xl, xu sean vector columna
[dummy, n] = size(xe);
if n == 1
    xe = xe';
end
[dummy, n] = size(xl);
if n == 1
    xl = xl';
end
[dummy, n] = size(xu);
if n == 1
    xu = xu';
end
% Evalua las restricciones de frontera
Rbl = 1000*max(xl-xe);
Rbu = 1000*max(xe-xu);
% Evalua las restricciones lineales
% No hay restricciones de Desigualdad A*X - B <= 0
Rld = -1000.0;

% No hay restriccion lineal de igualdad
Rle = -1000.0;

% Evalua las restricciones No lineales
[C, Ceq] = feval('DynReac2',xe, vd, pe);

% Desigualdad
if isempty(C)

```





**ANEXO S. Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en paralelo, usando GLB.**

```

function OptEstatPar_GB
warning off
%
% Script para el cálculo de las variables de diseño en estado
% ESTACIONARIO de dos reactores mezcla completa con chaqueta de
% enfriamiento dispuestos en forma PARALELA. Se procesa un
% compuesto A
% que reacciona a una velocidad de reacción de primer orden. Éste
% cálculo
% es obtenido a partir de la optimización minimizando el costo de
% diseño
% (en base al diámetro-altura de los reactores y el flujo de
% refrigerante).
%
%
% Éste script utiliza la función gblsolve de MatLab. Además consta
% de las siguientes funciones:
%
% Costo(VOD):          Calcula el costo de diseño de los reactores
para las
%                      variables de diseño VOD.
% DynReac(v,pe):      Realiza la simulación dinámica de los
reactores
%                      para las variables de diseño v y los
parámetros del
%                      sistema pe.
% dSeg(t,c,ve, p):    Calcula los valores de la ecuaciones
diferenciales
%                      correspondiente al modelo de simulación. t
(tiempo), c
%                      (variables dinámicas), ve (variables de
diseño), p
%                      (parámetros del sistema)
% StatReac(vs,ps):    Simula el reactor en estado estacionario.
%                      vs (variables de diseño), ps (parámetros del
sistema)
% error(c):           Función utilizada para resolver el sistema de
%                      ecuaciones algebraicas en la simulación en
estado
%                      estacionario.
% Ti(tp):             Función de perturbación para la temperatura de
%                      alimentación al reactor.
%
%
% Variables de Diseño (ve)

```

```

% 1 2 3 4 5 6 7 8 9 10 11 12
13
% Hr1, Hr2, Rf, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
close all
tic
global E I2 D1 pe
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S'; % INDICADOR
E = 0.001; % VALOR MAXIMO DE CONTROLABILIDAD
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_']
num2str(fix(1000.*E))];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('***** REACTORES EN PARALELO
*****')
disp('***** OPTIMIZACIÓN EN ESTADO ESTACIONARIO
*****')
% RESTRICCIONES DE FRONTERAS LB <= X <= UB
LB = [0.5, 0.5, 0, 1, 1, 1, 1];
UB = [100, 100, 1, 100, 100, 500, 500];
% OPCIONES DE OPTIMIZACION GBL SOLVE
GL.iterations = 70;
GL.epsilon = 1e-8;
GL.tolerance = 1.0e-3;
GL.NR = 2;
GL.Nmax = 50;
disp('Opciones de GBL SOLVE')
disp(GL)
% OPTIMIZADOR ---> GBL SOLVE
Result = gb('CostoRest',LB,UB,GL,2,pe, LB,UB);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
disp('RESULTADOS GBL SOLVE')
X1=Result.x_k';
disp(sprintf('%10.6f ',X1)); % VARIABLES DE DISEÑO
disp(sprintf('%10.6f ',Result.f_k)); % FUNCIÓN COSTO

*****
*****

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULO PUBLICACIÓN DE RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
R = input('¿Desea Generar los Resultados?(s/n)', 's');
% Datos Anteriores Estado Estacionario
ve = [X1, 0, 0, 1.2, 1.2];

if or(R == 'S', R == 's')
    mkdir(D1);
    copyfile('temp.txt', D1, 'f')
    I2 = 'Est';
    diary([D1 'out.txt'])
    clc
    disp(' ')
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS OPTCONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['FECHA: ' date])
    disp(' ');
    disp('OPTIMIZACION EN ESTADO ESTACIONARIO')
    disp(' ')
    cost = Costo(ve);
    disp(['COSTO CAPITAL ($) : ' sprintf('%10.12g ', cost(1))]);
    disp(['COSTO UTILIDAD($) : ' sprintf('%10.12g ', cost(2))]);
    disp(['COSTO TOTAL ($) : ' sprintf('%10.12g ', cost(3))]);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['Hr1 (ft) : ' sprintf('%1.8e ', ve(1))]);
    disp(['Hr2 (ft) : ' sprintf('%1.8e ', ve(2))]);
    disp(['Fr1/Ff : ' sprintf('%1.8e ', ve(3))]);
    disp(['Dr1 (ft) : ' sprintf('%1.8e ', ve(4))]);
    disp(['Dr2 (ft) : ' sprintf('%1.8e ', ve(5))]);
    disp(['Fjn1 (ft^3/h) : ' sprintf('%1.8e ', ve(6))]);
    disp(['Fjn2 (ft^3/h) : ' sprintf('%1.8e ', ve(7))]);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    C1 = StatReac(ve, pe);
    disp(['Fr1 (ft^3/h) : ' sprintf('%1.8e ', C1(1))]);
    disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ', C1(2))]);
    disp(['Tro1 (R) : ' sprintf('%1.8e ', C1(3))]);
    disp(['Tjo1 (R) : ' sprintf('%1.8e ', C1(4))]);
    disp(['Fr2 (ft^3/h) : ' sprintf('%1.8e ', C1(5))]);
    disp(['Cro2 (lbmol/ft^3) : ' sprintf('%1.8e ', C1(6))]);
    disp(['Tro2 (R) : ' sprintf('%1.8e ', C1(7))]);
    disp(['Tjo2 (R) : ' sprintf('%1.8e ', C1(8))]);
    disp(['Cp (lbmol/ft^3) : ' sprintf('%1.8e ', C1(9))]);
    disp(['Tp (R) : ' sprintf('%1.8e ', C1(10))]);
    disp(' ');
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(' RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
    C2 = DynReac(ve, pe);
    disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ', C2(1))]);
    disp(['Tro1 (R) : ' sprintf('%1.8e ', C2(2))]);

```



**ANEXO T. Programación en MATLAB R2012a para el cálculo de las variables de diseño en estado estacionario para topología de reactores en serie, usando GBL.**

```

function OptEstatSer_GB
warning off
%
% Script para el cálculo de las variables de diseño en estado
% ESTACIONARIO de dos reactores mezcla completa con chaqueta de
% enfriamiento dispuestos en forma PARALELA. Se procesa un
% compuesto A
% que reacciona a una velocidad de reacción de primer orden. Éste
% cálculo
% es obtenido a partir de la optimización minimizando el costo de
% diseño
% (en base al diámetro-altura de los reactores y el flujo de
% refrigerante).
%
%
% Éste script utiliza la función gBLSOLVE DE MatLab. Además consta
% de las siguientes funciones:
%
% Costo(VOD):          Calcula el costo de diseño de los reactores
para las
%                      variables de diseño VOD.
% DynReac(v,pe):      Realiza la simulación dinámica de los
reactores
%                      para las variables de diseño v y los
parámetros del
%                      sistema pe.
% dSeg(t,c,ve, p):    Calcula los valores de la ecuaciones
diferenciales
%                      correspondiente al modelo de simulación. t
(tiempo), c
%                      (variables dinámicas), ve (variables de
diseño), p
%                      (parámetros del sistema)
% StatReac(vs,ps):    Simula el reactor en estado estacionario.
%                      vs (variables de diseño), ps (parámetros del
sistema)
% error(c):           Función utilizada para resolver el sistema de
%                      ecuaciones algebraicas en la simulación en
estado
%                      estacionario.
% Ti(tp):             Función de perturbación para la temperatura de
%                      alimentación al reactor.
%
%
% Variables de Diseño (ve)

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULO      PUBLICACIÓN      DE      RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
R = input('¿Desea Generar los Resultados?(s/n)', 's');
% Datos Anteriores Estado Estacionario
ve = [X1, 0, 0, 1.2, 1.2];
if or(R == 'S', R == 's')
    mkdir(D1);
    copyfile('temp.txt', D1, 'f')
    I2 = 'Est';
    diary([D1 ' \out.txt'])
    clc
    disp(' ')
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      RESULTADOS      OPTCONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['FECHA: ' date])
    disp(' ');
    disp('OPTIMIZACION EN ESTADO ESTACIONARIO')
    disp(' ')
    cost = Costo(ve);
    disp(['COSTO CAPITAL ($) : ' sprintf('%10.12g ', cost(1))]);
    disp(['COSTO UTILIDAD($) : ' sprintf('%10.12g ', cost(2))]);
    disp(['COSTO TOTAL ($) : ' sprintf('%10.12g ', cost(3))]);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['Hr1 (ft) : ' sprintf('%1.8e ', ve(1))]);
    disp(['Hr2 (ft) : ' sprintf('%1.8e ', ve(2))]);
    disp(['Fb (ft^3/h) : ' sprintf('%1.8e ', ve(3))]);
    disp(['Dr1 (ft) : ' sprintf('%1.8e ', ve(4))]);
    disp(['Dr2 (ft) : ' sprintf('%1.8e ', ve(5))]);
    disp(['Fjn1 (ft^3/h) : ' sprintf('%1.8e ', ve(6))]);
    disp(['Fjn2 (ft^3/h) : ' sprintf('%1.8e ', ve(7))]);
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    C1 = StatReac(ve, pe);
    disp(['Fr (ft^3/h) : ' sprintf('%1.8e ', C1(1))]);
    disp(['Cri1 (lbmol/ft^3) : ' sprintf('%1.8e ', C1(2))]);
    disp(['Tril (R) : ' sprintf('%1.8e ', C1(3))]);
    disp(['Crol (lbmol/ft^3) : ' sprintf('%1.8e ', C1(4))]);
    disp(['Trol (R) : ' sprintf('%1.8e ', C1(5))]);
    disp(['Tjo1 (R) : ' sprintf('%1.8e ', C1(6))]);
    disp(['Cp (lbmol/ft^3) : ' sprintf('%1.8e ', C1(7))]);
    disp(['Tp (R) : ' sprintf('%1.8e ', C1(8))]);
    disp(['Tjo2 (R) : ' sprintf('%1.8e ', C1(9))]);
    disp(' ');
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp('          RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
    C2 = DynReac(ve, pe);
    disp(['Cro1 (lbmol/ft^3) : ' sprintf('%1.8e ', C2(1))]);
    disp(['Tro1 (R) : ' sprintf('%1.8e ', C2(2))]);
    disp(['Tjo1 (R) : ' sprintf('%1.8e ', C2(3))]);
    disp(['Cp (lbmol/ft^3) : ' sprintf('%1.8e ', C2(4))]);

```





**ANEXO U. Programación en Matlab R2012a para el cálculo de las variables de control en estado estacionario para topología de reactores en paralelo.  
Optimización de control mediante GBL.**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%              OPTIMIZACION - SOLO CONTROL - GBSOLVE
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function OnlyControlPar_GB
% Variables de Diseño (ve)
% 1 2 3 4 5 6 7 8 9 10 11 12
13
% Hr1, Hr2, Fb, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1 2 3 4 5 6 7 8 9 10 11 12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
warning off
close all
tic
global I2 D1
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S'; % INDICADOR
tiempo = clock; % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_OC'];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% REACTORES EN PARALELO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN DEL CONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PARAMETROS DE DISEÑO OBTENIDO DE LA OPTIMIZACION ESTACIONARIA
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
vd = [8.399923 2.549812 0.981481 24.013946 3.008175
...

```

```

        40.163822    0.736542];
    pe(2,1:7) = vd;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rango del problema
    LB = [0, 0, 1e-2, 1e-2, 0.1];
    UB = [500, 500, 5, 5, 100000];

% OPCIONES DE OPTIMIZACION GBL SOLVE
    GL.iterations = 70;
    GL.epsilon = 1e-6;
    GL.tolerance = 1.0e-3;
    GL.NR = 2;
    GL.Nmax = 20;
    disp('Opciones de GBL SOLVE')
    disp(GL)
% OPTIMIZADOR ---> GBL SOLVE
    Result = gb('Costo2',LB,UB,GL,2,pe,UB,UB);
% MUESTRA LOS VALORES OBTENIDOS EN ESTADO ESTACIONARIO
    disp('RESULTADOS GBL SOLVE')
    X=Result.x_k';
    disp(sprintf('%10.6f ',X));          % VARIABLES DE DISEÑO
    disp(sprintf('%10.6f ',Result.f_k)); % FUNCIÓN COSTO

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MODULO      PUBLICACIÓN      DE      RESULTADOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diary off
% PUBLICACION DE RESULTADOS
R = input('¿Desea Generar los Resultados?(s/n) ','s');
if or(R == 'S', R == 's')
    mkdir(D1);
    copyfile('temp.txt',D1,'f')
    I2 = 'Estat';
    diary([D1 '\out.txt'])
    clc
    disp(' ')
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESULTADOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['FECHA: ' date])
    disp(' ');
    disp('OPTIMIZACION CONTROL')
    disp(' ')
    I2 = 'Dim';
    disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS OPTIMOS DE DISEÑO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
    disp(['Hr1 (ft)           : ' sprintf('%1.8e ',vd(1))]);
    disp(['Hr2 (ft)           : ' sprintf('%1.8e ',vd(2))]);
    disp(['Fb (ft^3/h)        : ' sprintf('%1.8e ',vd(3))]);
    disp(['Dr1 (ft)           : ' sprintf('%1.8e ',vd(4))]);
    disp(['Dr2 (ft)           : ' sprintf('%1.8e ',vd(5))]);

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               FUNCION   OBJETIVO   -   SOLO   CONTROL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Fun = Costo2(xe,pe,xl,xu)
vd = pe(2,1:7);
ps = pe(1,:);
%Tolerancia en la evaluación de las restricciones
TolCon = 1e-6;
% Factor de penalización
Penal = 1e6;
% Asegura que X, xl, xu sean vector columna
[dummy, n] = size(xe);
if n == 1
    xe = xe';
end
[dummy, n] = size(xl);
if n == 1
    xl = xl';
end
[dummy, n] = size(xu);
if n == 1
    xu = xu';
end
% Evalua las restricciones de frontera
Rbl = 1000*max(xl-xe);
Rbu = 1000*max(xe-xu);
% Evalua las restricciones lineales
% No hay restricciones de Desigualdad A*X - B <= 0
Rld = -1000.0;

% No hay restriccion lineal de igualdad
Rle = -1000.0;

% Evalua las restricciones No lineales
[C, Ceq] = feval('DynReac2',xe, vd, ps);

% Desigualdad
if isempty(C)
    Rnld = -1000.0;
else
    Rnld = 1000*max(C);
end

% Igualdad
if isempty(Ceq)
    Rnle = -1000.0;
else
    Rnle = 1000*max(abs(Ceq) - TolCon);
end

```



**ANEXO V. Programación en Matlab R2012a para el cálculo de las variables  
de control en estado estacionario para topología de reactores en serie.  
Optimización de control mediante GBL.**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           OPTIMIZACION - SOLO CONTROL - GBSOLVE
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function OnlyControlSer_GB
% Variables de Diseño (ve)
% 1   2   3   4   5   6   7   8   9   10  11  12
13
% Hr1, Hr2, Fb, D1, D2, Fjn1, Fjn2, Tro1*, Tro2*, kj1, kj2, tauj1,
tauj2
% Parametros (pe)
% 1   2   3   4   5   6   7   8   9   10  11  12
% k0, ER, Cf, Ff, DH, rho, Cp, U, Tji1, Tji2, rhoj, Cj
%
warning off
close all
tic
global I2 D1
% PARAMETROS DEL SISTEMA
pe = [4.08e10, 15075, 1, 100, -3000, 50, 0.75, 300, 530, 530,
62.3, 1.0];
I2 = 'S';           % INDICADOR
tiempo = clock;    % HORA DEL SISTEMA
% NOMBRE DE LA CARPETA PARA EL REPORTE
D1=[date num2str(tiempo(4)) num2str(tiempo(5)) '_OC'];
% REGISTRO DE EVENTOS
diary off
delete 'temp.txt'
diary('temp.txt')
clc
disp(' ')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% REACTORES EN SERIE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZACIÓN DEL CONTROL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           PARAMETROS DE DISEÑO OBTENIDO DE LA OPTIMIZACION ESTACIONARIA
%

```



```

disp(['Hr1 (ft)           : ' sprintf('%1.8e ',vd(1))]);
disp(['Hr2 (ft)           : ' sprintf('%1.8e ',vd(2))]);
disp(['Fb (ft^3/h)        : ' sprintf('%1.8e ',vd(3))]);
disp(['Dr1 (ft)           : ' sprintf('%1.8e ',vd(4))]);
disp(['Dr2 (ft)           : ' sprintf('%1.8e ',vd(5))]);
disp(['Fjn1 (ft^3/h)      : ' sprintf('%1.8e ',vd(6))]);
disp(['Fjn2 (ft^3/h)      : ' sprintf('%1.8e ',vd(7))]);
disp('%%%%%%%%%% PARAMETROS OPTIMOS DE CONTROL
%%%%%%%%%%');
disp(['kj1 (ft^3/h.R)     : ' sprintf('%1.8e ',X(1))]);
disp(['kj2 (ft^3/h.R)     : ' sprintf('%1.8e ',X(2))]);
disp(['Tauj1 (h)          : ' sprintf('%1.8e ',X(3))]);
disp(['Tauj2 (h)          : ' sprintf('%1.8e ',X(4))]);
disp('%%%%%%%%%% RESULTADOS ESTADO ESTACIONARIO
%%%%%%%%%%');
ps = pe(1,:);
C3 = StatReac([vd, X(1:4)],ps);
disp(['Fr (ft^3/h)        : ' sprintf('%1.8e ',C3(1))]);
disp(['Cri1 (lbmol/ft^3) : ' sprintf('%1.8e ',C3(2))]);
disp(['Tril (R)           : ' sprintf('%1.8e ',C3(3))]);
disp(['Crol (lbmol/ft^3) : ' sprintf('%1.8e ',C3(4))]);
disp(['Trol (R)           : ' sprintf('%1.8e ',C3(5))]);
disp(['Tjo1 (R)           : ' sprintf('%1.8e ',C3(6))]);
disp(['Cp (lbmol/ft^3)    : ' sprintf('%1.8e ',C3(7))]);
disp(['Tp (R)             : ' sprintf('%1.8e ',C3(8))]);
disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C3(9))]);
disp('%%%%%%%%%% RESPUESTA DINAMICA %%%%%%%%%%%');
disp('      RESPUESTA AL FINAL DEL PERIODO DE ESTUDIO')
C4 = DynReac2(X,vd,ps);
disp(['Crol (lbmol/ft^3) : ' sprintf('%1.8e ',C4(1))]);
disp(['Trol (R)           : ' sprintf('%1.8e ',C4(2))]);
disp(['Tjo1 (R)           : ' sprintf('%1.8e ',C4(3))]);
disp(['Cp (lbmol/ft^3)    : ' sprintf('%1.8e ',C4(4))]);
disp(['Tp (R)             : ' sprintf('%1.8e ',C4(5))]);
disp(['Tjo2 (R)          : ' sprintf('%1.8e ',C4(6))]);
disp(['Itrol (R.h)        : ' sprintf('%1.8e ',C4(7))]);
disp(['Itro2 (R.h)        : ' sprintf('%1.8e ',C4(8))]);
disp(['mu (R^2.h)         : ' sprintf('%1.8e ',C4(9))]);
disp(' ');
disp(' *Nota: La respuesta dinámica en todo el periodo de
estudio')
disp(['      se encuentran graficados en los archivos ' I2
'_.jpg'])
disp('      en la carpeta:')
disp(['      ' cd '\ D1])
disp(' ');
close all
toc
disp(' ');
disp('****FIN DEL REPORTE')
diary off
save([D1 'out'])

```



```

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FIN DR
ONLYCONTROLPAR

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           FUNCION OBJETIVO - SOLO CONTROL
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Fun = Costo2(xe,pe,xl,xu)

vd = pe(2,1:7);
ps = pe(1,:);

%Tolerancia en la evaluación de las restricciones
TolCon = 1e-6;
% Factor de penalización
Penal = 1e6;
% Asegura que X, xl, xu sean vector columna
[dummy, n] = size(xe);
if n == 1
    xe = xe';
end
[dummy, n] = size(xl);
if n == 1
    xl = xl';
end
[dummy, n] = size(xu);
if n == 1
    xu = xu';
end
% Evalua las restricciones de frontera
Rbl = 1000*max(xl-xe);
Rbu = 1000*max(xe-xu);
% Evalua las restricciones lineales
% No hay restricciones de Desigualdad A*X - B <= 0
Rld = -1000.0;

% No hay restriccion lineal de igualdad
Rle = -1000.0;

% Evalua las restricciones No lineales
[C, Ceq] = feval('DynReac2',xe, vd, ps);

% Desigualdad
if isempty(C)
    Rnld = -1000.0;
else
    Rnld = 1000*max(C);
end

```

