



UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA QUÍMICA

APROXIMACIÓN EN TIEMPO REAL DEL CONTROL ÓPTIMO DE PROCESOS VÍA COMPUTACIÓN EN PARALELO

Tesis doctoral presentada como parte de los requisitos de la Universidad Nacional
del Litoral para la obtención del grado académico de
DOCTOR EN TECNOLOGÍA QUÍMICA

Por:
John Anderson Gómez Múnica

Instituto Tecnológico para el Desarrollo de la Industria Química (INTEC)
Departamento de Ingeniería Química (FIQ-UNL)



UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA QUÍMICA

APROXIMACIÓN EN TIEMPO REAL DEL CONTROL ÓPTIMO DE PROCESOS VÍA COMPUTACIÓN EN PARALELO

Tesis doctoral presentada como parte de los requisitos de la Universidad Nacional
del Litoral para la obtención del grado académico de
DOCTOR EN TECNOLOGÍA QUÍMICA

Instituto Tecnológico para el Desarrollo de la Industria Química (INTEC)
Departamento de Ingeniería Química (FIQ-UNL)

Doctorando

Ing. John Anderson Gómez Múnera

Director

Dr. Vicente Costanza

Codirector

Dr. Mario Storti

Santa Fe, abril de 2017

Aproximación en tiempo real del control óptimo de procesos vía computación en paralelo

Autor: Ing. John Anderson Gómez Múnera

Jurado: Dra. Marta Basualdo

Jurado: Dr. Pío Aguirre

Jurado: Dra. María Inés Troparevsky

Texto impreso en Santa Fe

Email: jagomezmunera@santafe-conicet.gov.ar

Primera edición, abril 2017

*A la memoria de mi padre Adolfo León Gómez,
a mi Madre y mi hermano,
y por supuesto a mi ángel, mi compañera de vida.
A todas las personas que creen y luchan por la educación pública.*

Resumen

El objetivo principal de esta tesis es establecer nuevos desarrollos científicos y tecnológicos sobre control óptimo de procesos, cuando existe o se imponen restricciones sobre la variable manipulada, especialmente utilizando estrategias basadas en el formalismo Hamiltoniano y en el Principio del Máximo de Pontryagin. El énfasis es puesto en generar esquemas de control óptimo implementables en tiempo real, que tengan la capacidad de generar trayectorias óptimas en paralelo con el proceso. Estos problemas son analizados también explorando e implementando el uso de computación en paralelo para su resolución.

Al comienzo de esta tesis se presentan, de manera concisa y metódica, los elementos y definiciones más importantes de la literatura relacionada, con el fin de sentar el fundamento teórico necesario para la comprensión del presente trabajo.

Se aborda por ende el problema de control óptimo en dos contextos distintos, siempre asumiendo la existencia de restricciones: *(i)* el tratamiento de problemas lineales mediante ecuaciones exactas, recurriendo a la actualización de los parámetros involucrados en la ley de control obtenida a través de la solución de la ecuación diferencial de Riccati, y explorando la aparición de varios períodos regulares, y *(ii)* el tratamiento para sistemas no lineales, para el cual se saca ventaja inicialmente de una fórmula existente para la actualización del control cuando no se tienen restricciones, y posteriormente la incorporación de diferentes variaciones espaciales a la trayectoria de control.

Los tratamientos y estrategias de control obtenidas fueron verificadas a través de varios casos de estudio, incluyendo un proceso ficticio, aplicaciones en procesos de ingeniería, y a un problema biomédico: *(i)* el

proceso ficticio hace referencia al frenado óptimo de un tren, con dos estados, modelo que fue modificado para agregar rozamiento y ser tratado dentro de un ámbito no lineal (en este caso un sistema bilineal), *(ii)* en un proceso de laminación de metales, discretizado y modelado inicialmente con diez estados, *(iii)* a un conjunto de calderas operando en paralelo, a las cuales se les asigna dinámicamente consignas de producción total de vapor, donde además se busca abatir eficazmente perturbaciones aleatorias en los parámetros de cada dispositivo, asimismo los ruidos en las señales de entrada y salida del sistema, y *(iv)* a la administración de insulina para el control de glucemia a pacientes diabéticos Tipo 1.

Las estrategias obtenidas para problemas lineales son analizadas y tratadas por intermedio de computación en paralelo para permitir aproximaciones en tiempo real del control óptimo de los procesos involucrados, su aplicación es presentada mediante aplicaciones numéricas. La primera aplicación corresponde al proceso ficticio del frenado óptimo del tren y la segunda aplicación comprende el proceso de laminación. Su implementación contempla diferentes alternativas tanto a nivel de máquinas locales multinúcleo como en un cluster de procesadores.

La aplicación de estrategias de control basadas estrictamente en modelos en tiempo continuo no siempre es posible. Por ejemplo, el estudio de administración de insulina para el control de la glucemia presenta las siguientes propiedades: *(i)* algunos estados no son observables en tiempo real, *(ii)* existen retardos en la evaluación del paciente y en la velocidad del médico para tomar decisiones terapéuticas, y *(iii)* la administración discreta de insulina exógena (los valores de control sólo pueden variar en un conjunto finito de fraccionamientos posibles). Resulta entonces conveniente elegir esquemas diferentes para tratar el problema de control y optimización que plantea la reducción de glucosa. En este caso se aplicó la metodología de “programación dinámica híbrida” para determinar la administración de insulina del paciente.

Agradecimientos

Ante todo, mi mayor agradecimiento es a mi madre y hermano por su amor incondicional, quienes son mi mayor orgullo y las personas capaces de mover mi mundo sólo con sus abrazos, de igual manera quiero dedicar el trabajo de la tesis a la memoria de mi padre; sé que desde donde se encuentre me envía constantemente la fuerza necesaria.

Esta tesis va dedicada de manera especial a mi compañera de vida Ángela, quien me enseña y demuestra cada día el amor, a ella por su paciencia, su fuerza y su comprensión. También la dedicación es para esos pequeños seres inocentes que alegran mis días, mi prima Valentina y mis sobrinas Paulina y Mariangel. A Dios por acompañarme y guiarme siempre.

Estoy infinitamente agradecido a mi Director, mi profesor, mi maestro, el Doctor Vicente Costanza. Su sabiduría, imaginación y entrega me han permitido alcanzar hoy esta meta tan anhelada. Sus sugerencias acertadas, su amable y cortés forma de señalar mis errores, su inagotable paciencia, me han formado no sólo académicamente, sino que también creo yo, a ser mejor persona.

A mi codirector, el Doctor Mario Storti por siempre tener esa buena predisposición antes mis dudas, su compromiso y su amplio conocimiento permitieron abordar temas mas allá del control óptimo.

Le estoy también muy agradecido a mi compañero, el Doctor Santiago Rivadeneira por su inconmensurable ayuda y paciencia, a las horas de discusión y a sus grandes enseñanzas.

Quiero agradecer también a una persona que es un maestro para mí, que ha sido gran pilar de mi formación académica, y que gracias a él

estoy inmerso en el apasionante mundo del control, el maestro Amado Tavera Crespo. A todos esos amigos que me enseñaron a crecer dentro de la academia, con los que compartí en mi alma mater en la etapa de pregrado, en especial: Andrés, Mario y Alejo, mi admiración por cada uno de ustedes es inmensa.

A las personas que han sabido soportarme durante todos estos años alejados de mis raíces, aquellos que me han cobijado desde el primer día: mi tío John Múnera, su esposa Ana, a Sergio (a quien considero un hermano), Enzo, Yohana, Milyer, Adriana, Vanessa, Nacho, a la colonia de Colombianos en Santa Fe, y a una infinidad de amigos que son injustamente omitidos. A los compañeros del INTEC, personal de apoyo, seguridad, con quienes he compartido los últimos años, les deseo lo mejor.

Por último mi agradecimiento es a este hermoso país Argentina por acogerme y enseñarme tanto, al CONICET por haberme brindado esta bonita oportunidad y permitir que mi sueño se materializara.

De todos y cada uno de ustedes es este logro.

Muchas gracias,

John Anderson Gómez Múnera

abril 2017

Índice general

Índice de figuras	xiii
Índice de tablas	xvii
Glosario	xix
1 Introducción	1
1.1 Motivación	2
1.2 Valor científico-técnico del trabajo	3
1.3 Breve Análisis de la principal bibliografía relacionada con la tesis	6
1.4 Aspectos novedosos de los resultados obtenidos	10
1.5 Publicaciones o en vías de publicación relacionadas al desarrollo de la tesis	11
1.6 Estructura de la tesis	12
2 Formalismo del problema de control óptimo	15
2.1 Sistemas de control y el problema de control óptimo.	16
2.2 La función de valor y el formalismo Hamiltoniano	21
2.2.1 Formalismo Hamiltoniano en problemas lineales	26
2.3 Controles acotados y el Principio de Pontriagyn	29
3 Tratamiento lineal	31
3.1 Valores de Fase Finales	33
3.2 Problema LQR Regular para sistemas lineales	35
3.3 Control con Restricciones	37

ÍNDICE GENERAL

3.3.1	Un hecho teórico relevante para la actualización de los parámetros	38
3.3.1.1	Fórmulas algebraicas	40
3.3.1.2	Objetos auxiliares	40
3.3.1.3	Derivadas parciales del costo	43
3.3.1.4	Actualizando los parámetros	46
3.3.1.5	Descripción esquemática para el procedimiento numérico	47
3.4	Aplicaciones Numéricas	48
3.4.1	Caso de estudio: Frenado óptimo de un tren	48
3.4.2	Una típica situación de un modelo linealizado: un proceso de laminación.	50
3.5	Varios períodos regulares	60
3.5.1	Dos períodos regulares	60
3.5.1.1	Semilla para Dos Períodos regulares	61
3.5.1.2	Optimización por medio del proceso “oculto”, “relacionado” o “fantasma”	62
3.5.1.2.1	Optimización desde t_0 hasta τ_2	63
3.5.1.2.2	Optimización desde τ_2 hasta τ_4	67
3.5.1.2.3	Optimización desde τ_4 hasta t_f	73
3.6	Conclusiones, perspectivas y discusión	77
3.6.1	Un período regular	77
3.6.2	Dos períodos regulares	78
4	Tratamiento no lineal	81
4.1	Estrategia simple	83
4.1.1	Búsqueda del proceso “oculto”, “fantasma” o irrestricto subyacente para sistemas no lineales	84
4.1.2	Linealización del modelo	85
4.1.3	Variaciones basadas en la ecuación diferencial de Riccati	86
4.1.4	Actualización de parámetros	87
4.1.5	Aplicación numérica simple	88
4.1.5.1	Solución óptima del problema irrestricto	89

ÍNDICE GENERAL

4.1.5.2	Método del gradiente para el problema restringido	90
4.2	Estrategia mixta	96
4.2.1	Métodos de reducción del costo	97
4.2.1.1	Reducción del costo en problemas regulares	97
4.2.1.1.1	Actualización de las trayectorias de control	99
4.2.1.2	Criterios de alerta de falla durante el procedimiento regular	100
4.2.2	Cómo proceder cuando el tratamiento regular está próximo a fallar	104
4.2.2.1	Otros tipos de variaciones de control	104
4.2.3	Aplicaciones numéricas estrategia mixta	105
4.3	Discusión, conclusiones y perspectivas	115
4.3.1	Estrategia simple	115
4.3.2	Estrategia mixta	116
5	Computación en paralelo para control óptimo basado en modelos	119
5.1	Computación en paralelo	120
5.1.1	Aspectos prácticos	121
5.1.1.1	MATLAB	122
5.1.1.2	Lenguaje de nivel medio: Uso de un cluster	125
5.2	Computación en paralelo aplicada a problemas de control	127
5.2.1	Primer acercamiento a una aplicación de control óptimo	128
5.2.1.1	Caso de estudio en el frenado óptimo de un tren de la sección 3.4.1	128
5.3	Otro acercamiento al control óptimo	136
5.3.1	Computación en paralelo para el tratamiento numérico de la sección 3.1	136
5.3.1.1	Paralelización del programa a través de secciones	136
5.3.1.2	Paralelización del programa a través de los bucles	137
5.4	Discusión de los resultados	143

ÍNDICE GENERAL

6	Otros Problemas adicionales de control óptimo	145
6.1	Programación dinámica	147
6.1.1	Modelo de pacientes con diabetes mellitus tipo 1	150
6.1.2	Costo asociado a la acción terapéutica	152
6.1.2.1	La necesidad de discretizar	153
6.1.3	Aplicación numérica al problema de diabétes tipo I	155
6.1.4	Conclusiones	160
6.2	Asignación dinámica en calderas	162
6.2.1	Configuración teórica	164
6.2.1.1	Modelado en espacio de estados para las unidades	164
6.2.1.2	Asignación óptima como un problema LQR	167
6.2.1.2.1	Manejo de los cambios en la demanda total α	171
6.2.1.3	Filtrado óptimo	172
6.2.1.4	Control estocástico	175
6.2.2	Aplicación numérica: Dos calderas en paralelo	179
6.2.2.1	Parámetros del costo determinista	180
6.2.2.2	Simulaciones de control estocástico.	180
6.2.2.3	Confirmaciones relevantes adicionales	185
6.2.2.3.1	Evaluando optimalidad	185
6.2.2.3.2	Eficiencia dinámica	185
6.2.3	Conclusiones	187
6.3	Discusión, conclusiones y perspectivas	190
7	Conclusiones y perspectivas	191
7.1	Conclusiones generales	192
7.2	Algunas perspectivas	195
	Bibliografía	197
A	Cálculo de Integrales Expresadas con Matrices Exponenciales	213
A.1	Integral 3.20	213

B	Códigos de cálculo en paralelo	217
B.1	Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren	217
B.1.1	Código en <code>MATLAB</code> con las divisiones de los intervalos de muestra	217
B.1.2	Código en <code>C++</code> para estudiar la evolución del costo ante el refinamiento en la división de los intervalos de muestra	224
B.1.3	Código en <code>MATLAB</code> para el problema fantasma	231
B.2	Paralelización con <code>OpenMP</code> . Problema de reducción del costo por actualización de los valores de fase finales 3.1	239
B.2.1	Programa principal para el problema de dimensión $n = 100$	240
B.2.2	Código fuente para las funciones	243
B.2.3	Makefile para la compilación de manera local	250
B.2.4	Porción de código paralelizado con ‘ <code>omp parallel sections</code> ’ subsección 5.3.1.1	251
B.2.5	Porción de código paralelizado con ‘ <code>omp parallel for</code> ’ subsección 5.3.1.2	253
B.2.5.1	Makefile para compilar en el cluster ‘coyote’ y agregar armadillo	254
B.2.5.2	‘script’ para correr el programa en el cluster ‘coyote’	255

Índice de figuras

2.1	Clasificación de sistemas.	16
2.2	Sistema de control general	17
2.3	Control saturado para dimensión $m = 1$	29
3.1	Comportamiento cualitativo típico de trayectorias de fase óptimas para el problema original restringido $\{x_{x_0}^*, \lambda_{x_0}^*\}$, y para el proceso relacionado irrestricto $\{\hat{x}, \hat{\lambda}\}$	38
3.2	Como actualizar los tiempos de conmutación bajo diferentes situaciones en las que el método del gradiente es aplicado.	49
3.3	Estrategias de control resultantes de la aplicación del método del gradiente al ejemplo del tren cuando se realizan modificaciones en los valores de fase finales, problema bidimensional.	51
3.4	Evolución del costo total con modificaciones en los valores de fase finales resultante del método del gradiente, problema bidimensional.	51
3.5	Evolución de los τ_s con modificaciones en los valores de fase finales resultante del método del gradiente, problema bidimensional.	52
3.6	Evolución de los ρ_s y μ_s con modificaciones en los valores de fase finales resultante del método del gradiente, problema bidimensional.	52
3.7	Evolución de las trayectorias de control con las iteraciones hasta alcanzar el control óptimo.	57
3.8	Evolución del costo total cuando se aplica el método del gradiente conjugado.	58
3.9	Evolución de los τ_s cuando se aplica el método del gradiente conjugado.	58

ÍNDICE DE FIGURAS

3.10 Evolución de los ρ y μ cuando se aplica el método del gradiente conjugado.	59
3.11 Control semilla	62
3.12 Optimización del primer período regular. (a) Perturbaciones del costo, (b) gradiente.	65
3.13 Comparación de controles en el primer período regular optimizado. (a) Sin restricciones, (b) con restricciones.	66
3.14 Gradiente de τ_2	66
3.15 Variación de \tilde{S} con las iteraciones	67
3.16 Gradientes del costo, optimización del segundo período regular	70
3.17 Comparación de controles, optimización segundo período regular	70
3.18 Gradiente de τ_4	71
3.19 Variación de S con las iteraciones	72
3.20 Gradientes del costo en la tercera etapa de optimización	74
3.21 Tercera etapa optimizada	75
3.22 Control final obtenido	76
4.1 Modelo relacionado.	91
4.2 Solución de la función de valor en $t = 0$	92
4.3 Trayectorias de control.	94
4.4 Variación del costo y de los estados $X(t)$ para el sistema linealizado.	94
4.5 Evolución de los taus con las iteraciones.	95
4.6 Truncamiento cuando la restricción es inferior.	101
4.7 Truncamiento cuando la restricción es superior.	102
4.8 Variaciones ε_τ con los δ_i constantes para otro tipo de variaciones de control. $\delta_1 = \delta_2 = 0,01$	105
4.9 Variaciones δ_i con el ε_τ constante para otros tipos de variaciones de control. $\varepsilon_\tau = 1 \times 10^3$	106
4.10 a) Evolución del costo total resultante del método regular, b) evolución del costo total resultante del método numérico.	107
4.11 ΔJ obtenido de la formulación 4.2.1.2.	107
4.12 Variaciones de control desde la estrategia regular (4.30).	108

ÍNDICE DE FIGURAS

4.13	Variaciones de control para todas las variaciones de los estados, a) variaciones positivas, b) variaciones negativas.	109
4.14	Variaciones de control generadas solucionando la RDE	110
4.15	Estrategia de control mixta para un ejemplo bidimensional.	110
4.16	Evolución del costo total resultante del método regular.	112
4.17	Falla del método, y evolución del costo total resultante del método numérico.	112
4.18	ΔJ obtenidos de la formulación de las ecuaciones (4.32, 4.43) . .	113
4.19	Estrategia de control mixta para el ejemplo bidimensional modificado.	114
5.1	Clasificación basada en arreglo de memorias y comunicación entre PE [7].	121
5.2	Sesión interactiva <code>parpool</code> o <code>parcluster</code> [108].	123
5.3	<code>pmode</code> con 3 workers.	125
5.4	En la figura superior se encuentran los estados y en la figura inferior las trayectorias de control. Salidas de MATLAB y el cluster.	129
5.5	Comparación de controles procesador 2:3 con MATLAB.	131
5.6	Gradiente del costo. Los parametros de la iteración usados fueron $h = 0,1$, $k = 0,5$, $\gamma = 2 * 10^{-3}$. Corrida con 30 iteraciones.	133
5.7	Control semilla e iterado	134
5.8	Speedup y ganancia total de tiempo para el problema de actualización de fase finales paralelizado con secciones y $p = 2$ procesadores locales.	138
5.9	Corrida en el cluster ‘coyote’.	140
5.10	Speedup para el problema de actualización de fase finales paralelizado con bucles for hasta con $p = 8$ procesadores.	141
5.11	Eficiencia de la arquitectura paralela para el problema de actualización de fase finales paralelizado con bucles for hasta con $p = 8$ procesadores.	141
6.1	Trayectorias discretas de nivel de glucosa en sangre x_1 obtenidas por programación dinámica.	157

ÍNDICE DE FIGURAS

6.2	Trayectorias discretas de insulina en sangre x_2 obtenidas por programación dinámica.	157
6.3	Trayectorias discretas de insulina subcutánea x_3 obtenidas por programación dinámica.	158
6.4	Trayectorias discretas de control obtenidas por programación dinámica.	158
6.5	Trayectoria continua de la glucosa en sangre x_1 después de la aplicación de la secuencia de control obtenida por programación dinámica.	159
6.6	Trayectoria continua de insulina en sangre x_2 después de la aplicación de la secuencia de control obtenida por programación dinámica.	159
6.7	Estrategia de control u para un horizonte de 600 min obtenida por programación dinámica.	160
6.8	Esquema de control para la supervisión con la asignación de la demanda óptima de una planta [92].	163
6.9	η versus q_1 y q_2	181
6.10	η versus q_1 y r	181
6.11	Trayectorias de estados.	182
6.12	Trayectorias de control.	182
6.13	Trayectorias de estados, estocástico con líneas solidas, y trayectorias con ruido con líneas discontinuas.	183
6.14	Trayectorias de control para el problema determinístico y el estocástico con los mismos coeficientes.	184
6.15	Marco principal: Variaciones alrededor del control óptimo. Marco pequeño: Diferencias relativas $(\mathcal{J}_\beta - \mathcal{J}^*) / \mathcal{J}^*$ entre el valor del costo generado por cada variación de control \mathcal{J}_β y el costo óptimo \mathcal{J}^* . El comportamiento parabólico del costo bajo combinaciones lineales de controles refleja la naturaleza lineal-cuadrática del problema.	186
6.16	Eficiencias dinámicas correspondientes a varios miembros de la familia de control $\{u = \gamma u^*, 1 \leq \gamma \leq 1,4\}$ pero manteniendo una demanda total de $\alpha = 150$ para las dos calderas.	187

Índice de tablas

3.1	Valores semilla y finales para los parámetros	56
3.2	Costo para el problema regular inicial	62
3.3	Valores utilizados en la primera optimización	64
3.4	Valores obtenidos con la primera optimización	68
3.5	Valores utilizados en el segundo intervalo de optimización	69
3.6	Valores obtenidos de la segunda optimización	72
3.7	Valores usados en el tercer intervalo de optimización	74
3.8	Valores obtenidos de la optimización completa	75
4.1	Valores numéricos finales estrategia simple	93
4.2	Valores numéricos finales estrategia mixta ejemplo 1	111
4.3	Valores numéricos finales estrategia mixta ejemplo 2	114
5.1	Medidas por iteración del programa paralelizado con omp sections	137
5.2	Medidas por iteración del programa paralelizado con omp for en el cluster con $n = 10$ y medidas de tiempo en segundos	138
5.3	Medidas por iteración del programa paralelizado con omp for en el cluster con $n = 20$ y medidas de tiempos en segundos	139
5.4	Medidas por iteración del programa paralelizado con omp for en el cluster con $n = 50$ y medidas de tiempo en segundos	139
5.5	Medidas por iteración del programa paralelizado con omp for en el cluster con $n = 100$ y medidas de tiempo en segundos	139

Glosario

Simbolos Griegos

α	Demanda en la sección 6.2
α, β	Matrices auxiliares
η	Eficiencia en la sección 6.2
γ	Tamaño de paso del vector gradiente
Λ_0	Covarianza inicial
λ	coestado
μ	Coestados finales del proceso Fantasma
ρ	Estados finales del proceso Fantasma
Σ	Sistema
σ	Factor escalar introducido por el término afín en la función de valor
τ	Tiempo de conmutación
θ	Temperatura de la banda metálica
φ	Función de transición
ϑ	Variables del sistema redefinido en 6.1.1
ξ	Vector columna introducido por el término afín en la función de valor

GLOSARIO

Notación

'	Transpuesta
#	Sistema aumentado
\mathbb{E}	Estimación o valor esperado
\mathbb{R}	Reales
\mathbb{R}_+	Reales no-negativos
\mathcal{H}	Hamiltoniano
\mathcal{J}	Costo, o índice de desempeño
\mathcal{K}	Penalización final
\mathcal{O}	Subconjunto abierto del espacio de Euclídeo
\mathcal{U}	Subconjunto compacto
\mathcal{V}	Función de valor
∇	Gradiente
A	Matriz de estado para sistemas lineales
B	Matriz de entrada para sistemas lineales
Cov	Covarianza
E_p	Eficiencia
I	Matriz identidad
L	Lagrangiano
m	Dimensión del espacio de entradas
n	Dimensión del espacio de estados
$P(\cdot)$	Solución de la RDE

S	Condición de borde final de la RDE
S_p	Speedup
S_{pA}	Speedup de Amdahl
S_{pG}	Speedup de Gustafson
t_f	Tiempo final
Tr	Traza
u	Entrada
x	Estado
y	Valores de salida

Acrónimos

AP	Artificial Pancreas
CF	Correction Factor
CGM	Continuous Glucose Monitoring
CSII	Continuous Subcutaneous Insulin Infusion
EGP	Endogenous Glucose Production
GPU	Graphics Processor Unit
HCE	Hamilton Canonical Equation
HJB	Hamilton Jacobi Bellman
IC	Initial Condition
ISF	Insulin Sensitivity Factor
IVP	Initial Value Problem
LQR	Lineal Quadratic Regulator

GLOSARIO

LTV	Linear-Time Variant
MDCS	MATLAB Distributed computed server
MDII	Multiple Daily Insulin Injection
MIMD	Multiple Instruction Multiple Date
MPC	Model Predictive Control
MPI	Message Passing Interface
ODE	Ordinary Differential Equation
OpenMP	Open Multi-Processing
PCT	Parallel Computing Toolbox
PDE	Partial Differential Equation
PE	Processing Elements
PID	Proportional-Integral-Derivative
PMP	Pontryagin's Maximum Principle
RDE	Riccati Differential Equation
RHS	Right Hand Side
SIMD	Single Instruction Multiple Date
SPMD	Simple Program Multiple Date
T1DM	Type 1 Diabetes Mellitus
UBIBO	Uniformly Bounded-Input Bounded-Output

*Tú, no te detengas ante ningún reto.
Y no pases a formar parte de ningún
gremio. Que nunca te puedan defi-
nir ni encasillar.*

Andrés Caicedo

CAPÍTULO

1

Introducción

En este capítulo se expone inicialmente la motivación del trabajo desarrollado, se da luego una visión general de la tesis doctoral, se realiza un análisis del valor técnico científico, y de la principal bibliografía involucrada. Se señalan los aspectos novedosos en esta tesis, se los contrasta con resultados ya existentes sobre los temas en cuestión, y por último se da una descripción breve de los capítulos que la componen.

1. INTRODUCCIÓN

1.1 Motivación

El objetivo principal de esta tesis es la obtención de nuevos resultados teóricos y desarrollos computacionales sobre los problemas de control óptimo con restricciones en la variable manipulada, tanto en sistemas lineales como en sistemas no lineales, con un especial énfasis en la exploración del uso de computación en paralelo para el tratamiento numérico de problemas que surgen en las aplicaciones.

La mayoría de sistemas se pueden considerar no lineales, aunque muchas veces pueden analizarse a través de aproximaciones lineales con matrices variantes en el tiempo $A(t), B(t)$. Los modelos lineales variantes en el tiempo resultantes ofrecen una buena relación entre simplicidad y capacidad de descripción del comportamiento del proceso.

Esta tesis también pretende explorar la instrumentación de las estrategias de control óptimo por intermedio del cálculo en paralelo, como una herramienta que posibilite disponer de las soluciones en tiempo real. En teoría de control el número de ecuaciones o el orden del proceso estudiado pueden generar grandes cantidades de cálculo, y en muchos casos imposibilitar que las soluciones sean obtenidas en los instantes en que se necesite enviar la estrategia de control al proceso. En principio el cálculo en paralelo puede manejar estas limitaciones al reducir el tiempo de cálculo del modelo completo.

La tesis está escrita desarrollando los dos tópicos antes expuestos, buscando la integración de ellos, explorando nuevos métodos y extrayendo sus ventajas.

1.2 Valor científico-técnico del trabajo

Esta tesis exhibe nuevos desarrollos científicos y técnicos que permiten encarar los problemas de control óptimo desde diferentes perspectivas, dependiendo del tipo de sistema que esté involucrado (sistemas lineales o sistemas no lineales). Muchos de los modelos relacionados a los procesos que se encuentran en ingeniería o algunas de las disciplinas ligadas, responden a modelos dominados por comportamientos no lineales. Estos sistemas dinámicos presentan comportamientos cualitativamente complejos tales como: múltiples equilibrios, bifurcación, atractores extraños, ciclos límites, oscilación subarmónica, caos, histéresis. Además de analizar los comportamientos se debe tener en cuenta que en la mayoría de los procesos se presentan restricciones en las variables (deberán trabajar en rangos de operación seguros y para valores de entrada alcanzables y compatibles con las limitaciones físicas), ruidos en las mediciones de los sistemas (señales de entrada y salida), y perturbaciones ambientales o efectos inesperados en el modelo. Eso implica que en general las herramientas del álgebra lineal y las ecuaciones diferenciales lineales resultan insuficientes para describir la estructura de las trayectorias, excepto en la cercanía de los puntos de equilibrio. Por otro lado, se detecta también en el entorno industrial la creciente preocupación y necesidad no sólo de controlar adecuadamente el proceso sino también de optimizar su funcionamiento. Se trabaja generalmente con un funcional de costo al que se busca minimizar o maximizar según el caso.

Las operaciones de “puesta en funcionamiento” y “apagado” de procesos continuos pueden analizarse como procesos batch, y dado que deben realizarse en un tiempo acotado, la optimización de dichos procesos conduce a problemas de control óptimo con horizonte finito. En general se buscan nuevas estrategias de control cuando se tienen restricciones en la variable de entrada y cuando deben ser implementadas en tiempo real. Por ejemplo se hizo uso para ello del ‘formalismo hamiltoniano’ y de técnicas tales como la ‘programación dinámica híbrida’ en el control y optimización de medicación de pacientes con diabetes mellitus tipo I, enmarcado como un problema de la ingeniería biomédica. También se consideran problemas para los cuales se presentan ruidos en las variables de estado (perturbaciones del ambiente) y ruidos en la medición de las variables de salida, que llevan

1. INTRODUCCIÓN

necesariamente al control estocástico. En el caso lineal se aprovechó el uso del filtro de Kalman-Bucy para realizar predicciones óptimas. Esto es empleado (como otro ejemplo de aplicación de la tesis) en la asignación dinámica de consignas a unidades de producción de vapor, utilizados ampliamente en procesos industriales.

Las herramientas generalizadas para obtener estrategias de control en los tratamientos lineales y no lineales fueron adaptadas para resolver diferentes situaciones en tiempo real (en algunos casos a través de la implementación con estrategias de cálculo en paralelo), como regulaciones, perturbaciones, cambios en las referencias o seguimiento de trayectorias, garantizando criterios estándar de funcionamiento, robustez y optimalidad. Con esto en mente se proyectaron las siguientes hipótesis y se propusieron algunas metas:

- Para procesos lineales la robustez está garantizada en casos irrestrictos, al ser posible el uso de la ecuación diferencial de Riccati para el cálculo de una realimentación de estados.
- Para procesos no lineales es posible desarrollar (o transformar) técnicas provenientes del enfoque Hamiltoniano de control óptimo (linealizando a lo largo de las trayectorias para evitar trabajar con las ecuaciones canónicas que son habitualmente inestables), que pueden ser implementadas en tiempo real, mejorando robustez y flexibilidad.
- Estas técnicas deben contemplar el filtrado y manejo de restricciones, considerando que existen perturbaciones en las entradas y salidas del proceso, y otras debido a las interacciones del sistema con el ambiente.
- Permitir el empleo de modelos de procesos en espacio de estados de cualquier dimensión. Admitir sistemas no lineales con funcionales de costo y penalización final arbitrarios.
- Integrar herramientas de cálculo en paralelo para problemas de control óptimo, obteniendo mejoras en el rendimiento de tiempos de cálculo e implementaciones en tiempo real. Exhibir posibilidades de implementaciones de computación en paralelo y en problemas de control óptimo.

1.2 Valor científico-técnico del trabajo

- Analizar diferentes procesos de ingeniería y de biomedicina empleando la teoría de control óptimo y el formalismo Hamiltoniano.
- Incorporación de filtrado, observación y retroalimentación óptimos, a fin de abatir eventuales perturbaciones del proceso y reducir ruidos en las señales de entrada y salida del sistema.

El incremento en la velocidad de modificación de estrategias en la optimización dinámica y el control óptimo de los procesos de ingeniería se ha visto restringida por la capacidad computacional disponible. Por tal motivo se han buscado alternativas que permitan obtener soluciones adecuadas dentro de las medidas de tiempo deseadas (“sampling periods”). Se busca una reducción total del tiempo de cálculo del problema, de acuerdo a la necesidad de los modelos a trabajar. Muchos de los problemas que se trabajan en la actualidad requieren una gran cantidad de operaciones con un alto número de datos significativos, que crecen con la dimensionalidad del modelo. En algunos casos, como en ciertas aplicaciones de “Model Predictive Control” (MPC), no se logran los tiempos y las precisiones adecuadas debido a la complejidad del modelo.

Para esto la computación en paralelo constituye en sí una herramienta ampliamente utilizada en diversos campos, que van desde la bioinformática hasta la economía, para la ejecución de los cálculos. Pensada en un principio para operar con problemas de dimensiones altas y volverlos realizables, en algunos casos se le termina utilizando para acelerar tareas específicas. Teniendo en cuenta que la cantidad de operaciones que se ven involucradas al simular los modelos matemáticos (lineales y no lineales) crece rápidamente con la dimensión, la computación en paralelo utiliza simultáneamente múltiples elementos de procesamiento para evaluar dichos modelos; lo que posibilita resolver de manera más perenne problemas de mayor dimensión, más complejos, en intervalos de tiempo más reducidos y con mayor precisión.

La necesidad de tener presente las restricciones en los valores del control, puede generar la existencia de varios períodos de saturación, que pueden ser encarados con técnicas que involucren optimización del modelo a través de desplazamientos

1. INTRODUCCIÓN

en los horizontes de cálculo. Esto implica, para el tratamiento de sistemas lineales, la existencia de diferentes soluciones de la matriz de Riccati de acuerdo a la cantidad de períodos regulares.

1.3 Breve Análisis de la principal bibliografía relacionada con la tesis

La teoría clásica se encargó de abordar los problemas de optimización de sistemas lineales con una función de costo cuadrático asociado [21, 98, 102, 134]. Sin embargo, los sistemas lineales resultan inadecuados para aproximar todas las trayectorias generadas por estrategias de control admisibles durante un tiempo arbitrario en los procesos industriales (o tecnológicos) no lineales. La necesidad de una herramienta eficiente para controlar dichos procesos ha sido ampliamente reconocida en la literatura. La “linealización global” permite tratar modelos no lineales con herramientas conocidas para lineales, pero su aplicabilidad depende de que se cumplan las condiciones del teorema de Frobenius [83], siendo las fórmulas de transformación complicadas.

Otros intentos por utilizar modelos no lineales han recurrido a estructuras con coeficientes desconocidos, los cuales deben ajustarse adaptativamente durante el proceso [89, 90]. Pero estos intentos dejan aún sin resolver el problema de controlar sistemas con una estructura no-lineal generalizada, y con un costo asociado por optimizar, como es a veces necesario en un proceso industrial.

Los sistemas bilineales fueron considerados de interés desde un punto de vista práctico por su similitud aparente con los sistemas lineales [20, 96]. Se conocen resultados sobre los sistemas bilineales con un funcional de costo cuadrático para el caso de regulación y cambio de set-point, que conducen a soluciones sub-óptimas del problema dada la validez de la aproximación realizada. Estos modelos han sido empleados en estrategias de control donde se combina la actualización de parámetros, observadores no lineales y el filtro óptimo de Kalman – Bucy para reducir ruidos externos (ver [34]).

Para sistemas no lineales en general y con funcionales de costo arbitrarios, en problemas de control óptimo no se ha obtenido una solución estándar y efectiva.

1.3 Breve Análisis de la principal bibliografía relacionada con la tesis

El formalismo Hamiltoniano ha sido el centro del desarrollo de la teoría de control óptimo clásica [117]. Cuando el problema para un sistema n -dimensional con un funcional de costo dado es regular [44, 88], es decir, cuando su Hamiltoniano puede ser minimizado por un valor único de la acción de control u dependiente de las variables restantes (t, x, λ) , entonces se debe resolver un conjunto de $2n$ ecuaciones diferenciales ordinarias (ecuaciones Hamiltonianas), con condiciones de borde mixtas. A menudo éste es un problema numérico difícil de resolver, por sus condiciones de borde mixtas y por su estructura no lineal.

Para el regulador cuadrático lineal (LQR) con horizonte finito existen varios métodos que transforman el problema de condiciones de borde mixtas en un problema de condiciones iniciales [134]. Un nuevo método fue desarrollado en [36], para resolver el problema de las condiciones mixtas para el LQR en tiempo finito y con penalización final $x'(t_f)Sx(t_f)$. Esta nueva forma de abordar el problema muestra la ventaja de no tener que resolver la ecuación diferencial de Riccati cada vez que se cambie la penalización o el tiempo de optimización, ya que se resuelven una sola vez las ecuaciones propuestas para un rango de valores de t_f y S y se las guarda en memoria. Si se quiere un cambio en cualquiera de las dos condiciones de diseño solamente se extraen de ella los datos que se necesiten (condición inicial del coestado y final del estado). Además estas ecuaciones posibilitan el cálculo del costo a priori, es decir, sin tener que evaluar las ecuaciones Hamiltonianas.

Los sistemas dinámicos Hamiltonianos (modelados por un conjunto de $2n$ ecuaciones diferenciales ordinarias, cuyo campo vectorial puede ser expresado en términos de las derivadas parciales de una función de energía total, normalmente llamada el Hamiltoniano del sistema) son objetos importantes de estudio tanto en la Matemática como en la Física. Las ecuaciones diferenciales ordinarias (ODEs) para el estado y el coestado referidos al problema de control óptimo constituyen el sistema Hamiltoniano. Richard Bellman contribuyó en los dos campos, particularmente en aplicaciones a sistemas simplécticos provenientes de la Física (ver para una introducción [1]). Él diseñó una ecuación en derivadas parciales (PDE) para el valor final del estado $x(t_f)$, donde t_f es el horizonte de optimización, como función de la duración del proceso $t_f - t$, y del valor final impuesto para el coestado $\lambda(t_f) = c$ (una de las condiciones de borde mixtas, [14]). Bellman desarrolló estas ideas en torno a una técnica numérica llamada “invariant imbedding”.

1. INTRODUCCIÓN

En [35], esta técnica ha sido generalizada y probada para sistemas unidimensionales con funcional de costo cuadrático y ley de control explícita. Se trata de obtener el valor del estado final $x(t_f) = \rho(t_f, S)$ y el coestado inicial $\lambda(0) = \sigma(t_f, S)$, con los cuales es posible iniciar la integración numérica de las $2n$ ecuaciones Hamiltonianas. Se llega a dos ecuaciones en derivadas parciales (PDEs) que dependen exclusivamente de dos parámetros: el horizonte de optimización t_f y la matriz S involucrada en la penalización cuadrática final $x'(t_f)Sx(t_f)$. Estas ecuaciones resultan ser cuasilineales de primer orden y pueden ser integradas simultáneamente. Los resultados obtenidos deben ser guardados en memoria para su posterior uso en la puesta en marcha del proceso a controlar. La información suministrada no sólo es útil en recuperar las condiciones de borde del sistema Hamiltoniano, sino que también puede ser utilizada en la etapa de diseño del controlador.

En [36], se realizó una extensión de las ecuaciones PDE para el caso multidimensional, donde se arriba a unas ecuaciones similares para $\sigma(t_f, S)$ y $\rho(t_f, S)$.

En [43] se trabaja también con limitaciones en el control, que puede tomar valores dentro de ciertos límites físicos (controles acotados). Se ilustra en la tesis para el problema LQR, en el que se involucra penalización cuadrática final, y del que se obtienen soluciones numéricas partiendo de unos valores semillas y calculando nuevas soluciones con el control acotado a través del método del gradiente y una nueva penalización final apropiada para el nuevo problema. Utilizar una implementación en tiempo real del esquema implicaría también un mayor uso de cálculo computacional, por lo que se vuelve necesario pensar en implementaciones con computación en paralelo, de manera que se obtengan soluciones en períodos de tiempo acotados y con las exigencias de rendimiento del modelo.

En [45], se trabaja en lo concerniente a la aplicación a PDEs discretizadas por el método de línea, lo que implica modelos de estados de dimensión alta.

En los últimos años de aplicación del control óptimo en la práctica, el incremento de cálculo numérico ha desbordado las capacidades computacionales existentes, llevando a buscar alternativas que permitan afrontar problemas para lograr tener soluciones en tiempo real [6, 113]. Estas estrategias de tiempo real deben responder y tomar decisiones frente a estímulos externos (tales como perturbaciones) dentro de períodos de tiempo acotados pequeños. Cuando se quieren satisfacer mayores exigencias de rendimiento, los períodos de tiempo necesitan ser menores

1.3 Breve Análisis de la principal bibliografía relacionada con la tesis

y la complejidad en los algoritmos de control aumenta. Al no ser alcanzadas estas exigencias dentro de los períodos de tiempo establecidos, se ha debido recurrir a estrategias más eficientes vía el procesamiento en paralelo [139, 140].

En el estudio de cálculo numérico en paralelo, y con el fin de aprovechar de una mejor manera el uso de nuevas herramientas computacionales, es necesario un entendimiento pleno del funcionamiento, de la arquitectura y de la distribución de las máquinas disponibles (hardware). Por esto es importante comprender tanto las limitaciones físicas a las está sometido el equipo computacional, como tomar las decisiones apropiadas sobre la arquitectura para realizar programación en alto nivel, ya que la mayoría de las instrucciones ejecutadas en el programa son salidas desde un compilador. Por lo tanto, el rendimiento de un computador es afectado por la calidad del compilador, por lo cual es muy complicado separar las evaluaciones de la arquitectura de la de su implementación [74].

El uso necesario de los lenguajes de alto nivel para programación de métodos numéricos comúnmente hace referencia a lenguajes como FORTRAN, PASCAL, o C/C++, los cuales necesitan compiladores adecuados antes de llegar a la ejecución de las instrucciones programadas. En los últimos años, para lograr una mayor capacidad de cálculo en estas condiciones, se empezó a implementar el cálculo en paralelo, y a profundizar el conocimiento de la arquitectura a la cual pertenece. La programación en paralelo se puede dar a través de memoria compartida o memoria distribuida, y cada una de éstas requiere técnicas diferentes para ser paralelizada. Se han estudiado técnicas para memoria compartida “OpenMP”, concepto que fue adoptado como un estándar en 1997 [25]. En el uso de memoria distribuida se hace necesario hablar de “cluster” (procesadores independientes que poseen un solo núcleo), por lo que es necesario dominar el paso de mensajes. Ante la gran cantidad de soluciones comerciales propuestas por las diferentes compañías que obligaba a los usuarios a tomar una serie de decisiones poniendo en compromiso ‘portabilidad’, ‘rendimiento’ y ‘prestaciones’, en 1994 se creó un estándar denominado MPI (en inglés “Message Passing Interface”) [132].

Dentro del estudio de los lenguajes de programación también se han utilizado soluciones con MATLAB como plataforma para resolver métodos numéricos, aprovechando toolboxes como PCT (Parallel Computing Toolbox), que fue instrumentado en distintas versiones desde 2007b en adelante [108].

1. INTRODUCCIÓN

En [64] se puede observar un primer acercamiento al uso de computación en paralelo para obtener soluciones en tiempo real.

1.4 Aspectos novedosos de los resultados obtenidos

1. La incorporación de diferentes estrategias para la obtención del control óptimo cuando se tienen restricciones en la variable manipulada, ha sido hecha a través de enfoques diferentes tanto en sistemas lineales como en los sistemas no lineales [49, 50, 51, 65].
2. En general las restricciones generan problemas numéricos al perderse la regularidad, lo que llevó a la inserción de herramientas computacionales “en paralelo” para optimizar los cálculos y trabajar con sistemas en tiempo real sin limitaciones, garantizando el cumplimiento de criterios de desempeño: precisión en el modelado del sistema, abatimiento de perturbaciones, robustez, estabilidad, filtrado, y adecuación a restricciones físicas. Este enfoque es nuevo en la literatura de control óptimo. Se aplicó a el problema del frenado óptimo de un tren y al proceso de laminación.
3. Esta tesis, además de contemplar la existencia de restricciones en sus salidas y variables manipuladas, analiza comportamientos en los que se presentan varios períodos de saturación para las variables de entrada, siempre apuntando a la minimización del funcional del costo total y al trabajo en tiempo real.
4. En los sistemas no lineales, en cuyo ámbito están enmarcados la mayoría de procesos de ingeniería, se ha conseguido un aprovechamiento creativo de una formulación (para problemas irrestrictos) descrita por Sontag, así como el uso de nuevos planteamientos de estrategias tales como las variaciones de “Riccati”, “variaciones espaciales” de Pontryagin o las variaciones en los tiempos de conmutación (ver [37, 48, 65]).

1.5 Publicaciones o en vías de publicación relacionadas al desarrollo de la tesis

Como consecuencia del trabajo exhibido en esta tesis, se logró la documentación y exposición a través de

- Publicación de artículos en revistas internacionales con referato:
 1. Costanza. V., Rivadeneira, P. S. y Gómez Múnera, J. A. (2016). An Efficient Cost Reduction Procedure for Bounded-Control LQR Problems. *Computational and Applied Mathematics*, Springer (ISSN: 1807–0302), doi: 10.1007/s40314-016-0393-x [50].
 2. Costanza. V., Rivadeneira, P. S. y Gómez Múnera, J. A. (2016). Numerical Treatment of the Bounded-Control LQR Problem By Updating the Final Phase Value. *IEEE Latin America Transactions*, 14(6), 2687 – 2692, doi: 10.1109/TLA.2016.7555239 [51].
 3. Rivadeneira. Pablo S., Gómez Múnera. John A. y Costanza. V. Dynamic Allocation of Industrial Utilities as an Optimal Stochastic Tracking Problem. *Chemical Engineering Science*, Elsevier, Vol. 160, 120 – 130, doi: 10.1016/j.ces.2016.11.022 [125].
- Trabajos en proceso de revisión:
 1. Gómez Múnera. John A., Rivadeneira. Pablo S. y Costanza. V. (2017). A Cost Reduction Procedure for Control Restricted Nonlinear Systems. *Computational Optimization and Applications*, enviado [65].
- Trabajos en preparación:
 1. Gómez Múnera, John. A., Costanza, V. y Storti M. (2017). Cálculo en paralelo para el proceso del Rolling mill.
- Trabajos presentados en congresos:

1. INTRODUCCIÓN

1. Gómez Múnera, J. A. y Costanza, V. (2014). Computación en paralelo para control óptimo basado en modelos. XXIV Congreso Asociación Argentina de Control Automático - AADECA, Buenos Aires, Argentina [64].
2. Costanza, V., Rivadeneira, P. S. y Gómez Múnera, J. A. (2015). Suboptimal control of nonlinear systems under restrictions in the manipulated variable. XVI Reunión de trabajo en Procesamiento de la Información y Control - RPIC, doi: 10.1109/RPIC.2015.7497061, Córdoba, Argentina [48].
3. Costanza, V., Rivadeneira, P. S. y Gómez Múnera, J. A. (2015). Numerical Treatment of the Bounded-Control LQR Problem By Updating The Final Phase Value. XVI Reunión de trabajo en Procesamiento de la Información y Control - RPIC, Córdoba, Argentina [49].
4. Costanza, V. y Gómez Múnera, J. A. (2016). Cost Reduction for Nonlinear Systems Under Restrictions on the Manipulated Variables. 5th International Conference on Engineering Optimization, Foz de Iguazú, Brasil [37].
5. Rivadeneira, P. S., Gómez Múnera, John. A. y Costanza, V. (2016). Programación Híbrida para el Manejo de la Glucemia en Pacientes Diabéticos Tipo I. XXV Congreso Asociación Argentina de Control Automático - AADECA, Buenos Aires, Argentina [124].

1.6 Estructura de la tesis

Los capítulos posteriores de la tesis adoptan la siguiente disposición:

- En el capítulo 2 se presenta y se fundamenta el formalismo del problema de control óptimo, base del desarrollo de la investigación. En primer lugar se realiza una revisión de los objetos de control y el problema de control óptimo, luego es introducida la función de valor y el formalismo Hamiltoniano resultante, haciendo también una breve exposición del problema LQR para

sistemas lineales. Por último se consideran restricciones en la variable manipulada y se describe la importancia del Principio de Pontriagyn para esta clase de problemas.

- En el capítulo 3 se substancia una estrategia para el tratamiento de sistemas lineales con restricciones: en primer lugar para problemas con un solo período regular, para lo cual no es necesaria la integración de la trayectoria de estados porque se hace uso de ecuaciones algebraicas. Se utiliza el método del gradiente, modificando los valores de fase finales, que permiten obtener la matriz de penalización final de la ecuación de Riccati por intermedio de un problema relacionado irrestricto. Posteriormente se desarrolla un análisis para problemas que presentan varios períodos regulares, exponiéndose la metodología a través de un ejemplo en el que aparecen dos períodos regulares.
- En el capítulo 4 se plantean dos nuevas estrategias para resolver problemas de control óptimo con la variable de control restringida para sistemas no lineales. En la primera de ellas se utiliza un procedimiento sencillo, que realiza modificaciones por intermedio del método del gradiente en los valores finales para la linealización por trayectoria del modelo. En la segunda, denominada estrategia mixta, se aprovecha un tratamiento ya expuesto para problemas regulares para empezar la disminución de costo total a través de una modificación en la trayectoria de control, y luego, cuando las variaciones del control puedan provocar fallas en el método, son utilizadas otro tipo de variaciones. Dichas variaciones son basadas en diferentes estrategias que proponen pequeñas acciones controladas que contribuyan a la reducción sistemática del costo, variaciones tales como soluciones de la RDE, variaciones espaciales de Pontryagin u oscilaciones en el interior de los períodos regulares generadas por la modificación de los tiempos de conmutación.
- En el capítulo 5 se da en primer lugar una descripción de la computación en paralelo y sus aplicaciones actuales, así como de algunos aspectos prácticos (esto incluye entornos y lenguajes de programación). Una relación con respecto a control óptimo y su aplicación para computación en paralelo es abor-

1. INTRODUCCIÓN

dada y presentada mediante aplicaciones numéricas. La primera aplicación presentada corresponde al frenado óptimo de un tren y la segunda aplicación comprende un proceso de laminación, para los cuales son estudiadas e implementadas diferentes alternativas tanto a nivel de máquinas locales multinúcleo como en un cluster, programadas por medio de software privativos y libres.

- En el capítulo 6 se estudian problemas especiales de control óptimo aplicados, el primero de ellos desarrollando una estrategia de control con programación dinámica híbrida para el manejo de glucemia en pacientes con diabetes tipo I, y el segundo resolviendo la asignación (distribución) dinámica de servicios suministrados a unidades industriales, en respuesta a la demanda de servicios globales de una planta. La naturaleza del modelado también admite un manejo óptimo estocástico de ruido y perturbaciones sistémicas.
- El capítulo 7 está destinado a exponer las conclusiones obtenidas a lo largo del trabajo elaborado, además de plantear las líneas de trabajo a futuro relacionadas con los temas abordados.

*La ciencia no nos ha enseñado aún
si la locura es o no lo más sublime
de la inteligencia.*

Edgar Allan Poe

CAPÍTULO

2

Formalismo del problema de control óptimo

Este capítulo expone brevemente las características básicas de los sistemas de control a tratar, y fija la notación de las variables principales y sus dominios respectivos. También define el problema de control óptimo que se considerará en todas las aplicaciones de la tesis y los objetos matemáticos relacionados. Luego, se introduce la función de valor del sistema, el formalismo Hamiltoniano y se describe cómo el problema de control óptimo puede ser resuelto a través de varias metodologías. Por último se lleva a cabo una discusión sobre el tipo de restricciones físicas que aparecen en la variable manipulada u y su implicancia en el formalismo del problema.

2. FORMALISMO DEL PROBLEMA DE CONTROL ÓPTIMO

2.1 Sistemas de control y el problema de control óptimo.

Los sistemas pueden ser clasificados de acuerdo al tipo de ecuaciones con las que son descritas, en la figura 2.1 se puede apreciar una clasificación de los sistemas (ver [85]), en el cual existe para cada nivel un par de categorías. Las líneas punteadas indican la posibilidad de derivación, similar a aquéllas en el mismo nivel. La zona demarcada es la correspondiente al énfasis elegido en esta tesis.

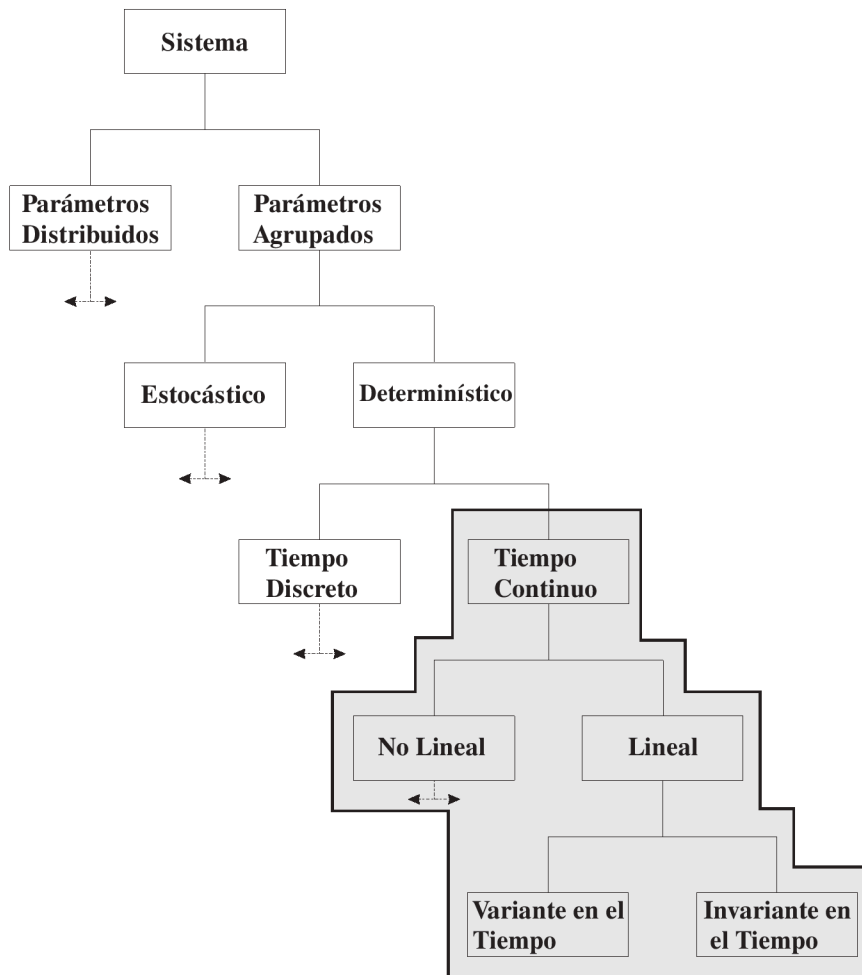


Figura 2.1: Clasificación de sistemas.

Una forma sencilla de visualizar un sistema de control (Σ) es mediante la figura 2.2,

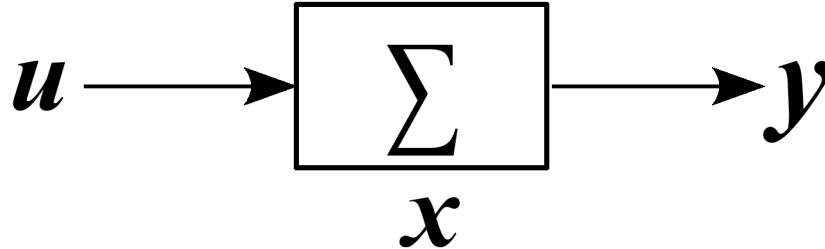


Figura 2.2: Sistema de control general

donde la variable de entrada, variable manipulada o “control” $u(t)$ (se entiende como una trayectoria de vectores de \mathbb{R}^m) fuerza al proceso a lograr cierto comportamiento (reflejado en los estados del sistema (x) o en las salidas (y)). Esta variable de control es manejada directamente por un agente externo al proceso. Por ejemplo, en el llenado de un tanque, la apertura o cierre de la válvula que fija el nivel del mismo es movida expresamente desde fuera del tanque. De ese modo, la entrada influencia a las variables a controlar o “estados” del sistema (representados por x , donde $x(t)$ es un vector del espacio euclídeo n -dimensional denotado por \mathbb{R}^n para cada t). El estado de un sistema es el conjunto más pequeño de variables (denominadas variables de estado) tal que el conocimiento de esas variables en $t = t_0$ en conjunto con el valor de la variable de entrada para $t \geq t_0$ determinan completamente el comportamiento del proceso para cualquier tiempo $t \geq t_0$. Una idea intuitiva de los estados está relacionada con la memoria necesaria del proceso para poder predecir su futuro. Los estados por lo general son variables físicas como la temperatura, concentración, nivel, distancias, o simplemente entes matemáticos abstractos. Se debe aclarar que a lo largo de esta tesis sólo se tratarán procesos físicos representados por el formalismo del “espacio de estados” (ver [134]). La salida $y(t)$ del sistema de control resume las observaciones del proceso físico. Por lo general, es un conjunto de variables físicamente medibles y de dimensión menor o igual a n , con n la dimensión de los estados.

En general, el objetivo de un problema de control es lograr que las salidas se comporten de alguna forma prescrita mediante la manipulación de las entradas del sistema de control.

2. FORMALISMO DEL PROBLEMA DE CONTROL ÓPTIMO

Los estados x se mueven en una región \mathcal{O}_{x_0} de \mathbb{R}^n , que contiene a las condiciones iniciales x_0 . La variable manipulable u toma valores en un conjunto $\mathcal{U} \subseteq \mathbb{R}^m$. Sin pérdida de generalidad, las trayectorias de control admisibles serán todas las funciones $u : [0, t_f] \rightarrow \mathbb{R}^m$ dependientes del tiempo t de duración $t_f \leq \infty$ y continuas a trozos. La relación entre el control, los estados y las salidas del proceso estará dada mediante una ecuación diferencial llamada “La dinámica del sistema” y otra de observación:

$$(\Sigma) \quad \dot{x}(t) = f(t, x, u), \quad x(t_0) = x_0, \quad (2.1)$$

$$y(t) = g(t, x, u), \quad (2.2)$$

con regularidad tal que están garantizadas la existencia y unicidad de la solución de la ecuación diferencial (2.1) [29, 76].

El uso de entradas externas o controles u “fuerzan” a que los estados del sistema x partan de sus condiciones iniciales x_0 y se dirijan a otros estados deseados (target) \bar{x} . Sin embargo, dicha transición, si es posible, en general no es única y depende fuertemente de los controles aplicados. En este contexto, naturalmente surge la inquietud de hallar la mejor manera de realizar dicha transición, así que el “problema de control óptimo” consiste en buscar una trayectoria o señal de control $u(t)$ que le permita al proceso llegar lo más cercano posible a su estado “deseado” minimizando una función denominada “costo” o “índice de desempeño”. Dicho costo estará descrito por un funcional generalizado, al cual llamaremos funcional de costo, criterio de optimización o función objetivo:

$$\mathcal{J}(t_f, t_0, x_0, u) = \int_{t_0}^{t_f} L(\tau, x_u(\tau), u(\tau)) d\tau + \mathcal{K}(x(t_f)), \quad (2.3)$$

donde L es el “Lagrangiano” del problema de control óptimo y $\mathcal{K}(x(t_f))$ representa a una penalización final. El horizonte de optimización $\Delta t = t_f - t_0$ del problema es finito si $t_f < \infty$ y $\tau \in [t_0, t_f]$. La minimización del funcional de costo con respecto a u está sujeto a la restricción impuesta por la “dinámica” del sistema de control, es decir $x_u(\tau)$ debe ser la solución de (2.1) correspondiente a la trayectoria de control $u(\tau)$, donde t_0 es el tiempo inicial y t_f es el tiempo final. Dado que

2.1 Sistemas de control y el problema de control óptimo.

normalmente se tomará $t_0 = 0$ el horizonte de optimización Δt se reduce a t_f . Si existe una trayectoria de control que minimiza (2.3) será denotada por u^* , verificará

$$u^*(\cdot) \triangleq \arg \inf_{u(\cdot)} \mathcal{J}(t_f, 0, x_0, u(\cdot)), \quad (2.4)$$

y una vez aplicada al sistema (2.1) generará la trayectoria óptima de estado $x^*(t)$ de tal modo que $x^*(\cdot)$ sea solución de (2.1) con $u(\cdot) = u^*(\cdot)$.

El Lagrangiano $L(t, x, u)$ de la ecuación 2.3 es una función escalar no negativa ($L \geq 0$), es decir, transforma elementos de $\mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ a elementos de \mathbb{R}_0^+ . Esta función se debería entender como un diferencial del costo de trayectoria. Por ejemplo, en el caso donde $L = x'Qx + u'Ru$, se cuantifica el esfuerzo de llevar los estados x a cero aplicando la acción de control u suficientemente pequeño a lo largo del horizonte de optimización. Cuando L es diferente a cero, su integral sigue creciendo al ser una función positiva. En un típico “costo cuadrático”, el término $x'Qx$ indica el costo instantáneo de los estados y análogamente $u'Ru$ nos da una idea del esfuerzo de control ejercido. En cada aplicación se debe elegir qué tipo concreto de Lagrangiano utilizar, sin embargo, lo más usado son las formas cuadráticas para x y u , aunque también es posible trabajar con otras formas, como por ejemplo la “potencia” del sistema definida como el producto de los estados por el control, los de mínima energía del sistema definidas solo a partir del cuadrado de las acciones de control, o aquellas de tiempo mínimo, donde $L \equiv 1$. El término $\mathcal{K}(x(t_f)) : \mathbb{R}^n \rightarrow \mathbb{R}_+$ se entiende como una penalización final sobre los estados. Claramente, introduce un costo adicional al final del horizonte de optimización t_f . Si por ejemplo, se define como $\mathcal{K}(x(t_f)) = x'(t_f)Sx(t_f)$, el efecto que se busca del proceso es que éste intente terminar lo más cercano de cero (o del punto al cual se quiere regular, llevado a cero mediante alguna transformación de coordenada).

El sistema también puede ser visto, no como una ecuación diferencial, sino a través de su función de transición o mapeo de transición, la cual define a qué estado evolucionará el sistema $\{x(t) = \varphi(t, \sigma, x, w), t \in [0, t_f]\}$ (ver [88, 134]), donde el miembro derecho puede leerse como el estado al tiempo t de la transición, desde un tiempo inicial σ , con x como el estado al tiempo σ y w la trayectoria de control aplicada entre los pares de tiempos inicial y final σ y t . Definida φ de esta forma, cuando la aplicamos nos brinda el valor exacto del estado en t . De esta manera

2. FORMALISMO DEL PROBLEMA DE CONTROL ÓPTIMO

podemos reescribir el funcional de costo:

$$\mathcal{J}(t_f, t_0, x_0, u) = \int_{t_0}^{t_f} L(t, \varphi(t, t_0, x_0, u), u(t)) dt + \mathcal{K}(x(t_f)), \quad (2.5)$$

Esta función que realiza una transición de elementos $\varphi : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, también se le conoce como el “flujo” de la ecuación diferencial (ver [76]).

2.2 La función de valor y el formalismo Hamiltoniano

En programación dinámica, que es una visión clásica de los problemas de optimización el papel central es protagonizado por la “función de valor” o función de Bellman \mathcal{V} (llamada así dado que Richard Bellman fue pionero en el desarrollo y el uso de esta función), que se define [18, 134] como:

$$\mathcal{V}(t, x) \triangleq \inf_u \mathcal{J}(t_f, t, x, u), \quad (2.6)$$

en $[t_0, t_f] \times \mathbb{R}^n$ y con valores en \mathbb{R}_+ , donde t_f sigue siendo el tiempo final del horizonte de optimización, $t \in [t_0, t_f]$, x el estado desde el cual se desea optimizar a partir de t , y u es la trayectoria de control aplicada. Al igual que \mathcal{J} , la función de valor \mathcal{V} es un costo, sin embargo, ésta nos da el costo mínimo entre todas las posibles trayectorias que comienzan en (t, x) , por lo que $\mathcal{V}(t_0, x_0)$ daría el costo óptimo total del proceso.

La función de valor tiene la siguientes propiedades:

- (i) Para cada par s, t tal que $\sigma \leq t \leq s \leq \tau$ y cada $x \in \mathbb{R}^n$, para todos los controles w ,

$$\mathcal{V}(t, x) \leq \mathcal{J}(s, t, x, w) + \mathcal{V}(s, \varphi(s, t, x, w)). \quad (2.7)$$

Si la función de Bellman es alcanzada, entonces

$$\mathcal{V}(t, x) = \mathcal{J}(s, t, x, w |_{[t,s]}) + \mathcal{V}(s, \varphi(s, t, x, w |_{[t,s]})) \quad (2.8)$$

para un control w óptimo en (t, x) . El análisis principal es que a lo largo de cualquier trayectoria óptima, se debe considerar el costo óptimo como la diferencia entre la función de valor en un punto y la función de valor obtenida al realizar una transición.

- (ii) Para cada $s \in [t_0, t_f]$ y $x \in \mathbb{R}^n$

$$\mathcal{V}(s, x) \leq \int_s^t L(\tau, x_u(\tau), u(\tau)) d\tau + \mathcal{V}(t, x_u(t)), \quad \forall s < t \leq t_f, x, \quad \forall u. \quad (2.9)$$

2. FORMALISMO DEL PROBLEMA DE CONTROL ÓPTIMO

Esta propiedad es sencilla pero debe ser bien entendida, el término $\mathcal{V}(t, x_u)$ dice que a partir de las condiciones (t, x_u) hasta el tiempo final t_f el costo óptimo es conocido. Por lo tanto, para un $u \neq u^*$ dado, el costo correspondiente $\mathcal{V}(s, x)$ para el horizonte $\Delta t = t_f - s$ tiene que ser menor a la suma de la integral del Lagrangiano y $\mathcal{V}(t, x_u)$, a menos que dicho u sea el óptimo. En este último caso resulta ser la igualdad. En [134] se puede consultar una explicación más rigurosa de esta propiedad.

(iii) Condición de borde, en $t = t_f$,

$$\mathcal{V}(t_f, x(t_f)) = \mathcal{K}(x(t_f)), \quad (2.10)$$

la condición de borde es obtenida inductivamente de manera que satisface la ecuación 2.6.

En el caso discreto la programación dinámica puede ser implementada mediante el siguiente algoritmo (para una adecuada fundamentación teórica de la técnica consultar [15, 88, 93, 102, 134]):

1. En primer lugar ha de plantearse el problema de forma que pueda ser descompuesto en k subproblemas. Es decir, dividir el problema con un tamaño n_k de forma tal que $1 \leq n_k \leq n$, sea el problema completo.
2. Discretizar los estados en $t_k = t_f$,
3. luego por la propiedad (iii) se calcula el costo, el cual es $\mathcal{K}(x(t_k))$ para todos los puntos $x(t_k)$ dentro de la discretización de x . Éste es el costo óptimo en el tiempo final.
4. Se pasa al tiempo t_{k-1} , hacia atrás en la discretización del tiempo. Ahí, para cada punto discreto de x en t_{k-1} se aplican todas las acciones de control permitidas en $[t_{k-1}, t_f]$, y con sus respectivas trayectorias de estado, se calcula la integral del Lagrangiano. Por las propiedades (i), (ii) y suponiendo que \mathcal{V} está asumida (es decir, existe un control u^* tal que $\mathcal{V}(t, x) = \min_u \mathcal{J}(t_f, s, x, u)$), se escoge la de mínimo costo. Para este tiempo t_{k-1} , dicho valor de control será el óptimo. En este momento se guarda el valor del costo $\mathcal{V}(t_{k-1}, x(t_{k-1}))$ óptimo, el control óptimo $u^*(t_{k-1})$ que lo produce y el estado óptimo $x^*(t_f)$ final alcanzado.

2.2 La función de valor y el formalismo Hamiltoniano

5. Luego, nuevamente se va hacia el paso siguiente hacia atrás t_{k-2} , o de forma general a un tiempo t_{k-i} con $i = 2, 3, \dots, k$, se aplica lo explicado en el ítem anterior y se guarda el costo óptimo $\mathcal{V}(t_{k-i}, x(t_{k-i}))$, el control óptimo $u^*(t_{k-i})$ y el estado óptimo $x^*(t_{k-1})$ correspondientes.
6. El procedimiento se repite hasta llegar a las condiciones iniciales en t_0 , donde $x(t_0) = x_0$ son conocidos.

Ahora si queremos saber cuál es la trayectoria óptima desde la condición inicial x_0 hasta el final del horizonte de optimización t_f , es suficiente con concatenar todas las soluciones óptimas guardadas, ya que por el enunciado de Bellman, (“En una secuencia de decisiones óptima toda subsecuencia ha de ser también óptima”) se ve reflejada en la propiedad (i) sabemos que una solución calculada como la suma de soluciones óptimas será óptima también. Este procedimiento (de “programación dinámica”) describe la forma de encontrar trayectorias óptimas para problemas generales: el sistema de control es de cualquier tipo (es decir, podría ser lineal, no lineal, autónomo, variante con el tiempo), el Lagrangiano es una función generalizada, y los estados y el control pueden tener restricciones y estar acotados.

Retomando la formulación en tiempo continuo y suponiendo que existe una trayectoria óptima de control $u^*(t)$ que produce la trayectoria óptima de estados $x^*(t)$, a lo largo de esas trayectorias será

$$v(t) \triangleq \mathcal{V}(t, x^*(t)) = \int_t^{t_f} L(\tau, x^*(\tau), u^*(\tau)) d\tau + \mathcal{V}(t_f, x^*(t_f)) \quad (2.11)$$

la función v es absolutamente continua y tiene derivada con respecto al tiempo:

$$\dot{v}(t) = \frac{dv}{dt}(t) = -L(t, x^*(t), u^*(t)). \quad (2.12)$$

Por tanto, a lo largo de una trayectoria óptima la velocidad de decrecimiento de la función de Bellman es igual al costo instantáneo negativo. Por otro lado, también es cierto que la derivada de $v(t)$ con respecto al tiempo es (asumiendo que \mathcal{V} resulta una función lo suficientemente suave de sus variables) $\dot{v} = \mathcal{V}_t + \mathcal{V}_x \dot{x}(t)$ (donde la notación \mathcal{V}_x indica la derivada parcial de \mathcal{V} con respecto a su variable x), por lo que al igualarla con la ecuación 2.12 y reemplazando $\dot{x}(t)$ por su respectiva derivada óptima ($\dot{x}^*(t)$), se tiene

$$\mathcal{V}_t(t, x^*(t)) = -L(t, x^*(t), u^*(t)) - \mathcal{V}_x(t, x^*(t)) f(t, x^*(t), u^*(t)). \quad (2.13)$$

2. FORMALISMO DEL PROBLEMA DE CONTROL ÓPTIMO

L es una función escalar, por lo que las dimensiones del gradiente ∇_x de \mathcal{V} deberían ser las de un vector fila. Se convendrá que todas las derivadas parciales de una función escalar con respecto a un vector (sea este fila o columna) serán consideradas como vector fila.

El Hamiltoniano del sistema asociado al problema de control óptimo es definido como una función escalar (véase [44]), continuamente diferenciable que depende de las variables (t, x, λ, u)

$$\mathcal{H}(t, x, \lambda, u) \triangleq L(t, x, u) + \lambda' f(t, x, u), \quad (2.14)$$

y visto como una función que mapea $[t_0, t_f] \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$. En la literatura, esta variable auxiliar λ es conocida como el “coestado” del sistema, o covector a partir de su interpretación geométrica en manifolds (o variedades diferenciales en español), ya que para cada punto x , λ es un elemento que pertenece al plano cotangente T_x^*M del espacio de estados en la variedad M y se define como un funcional lineal perteneciente al espacio dual del espacio tangente T_xM en x . También, el coestado podría ser matemáticamente entendido como un “multiplicador de Lagrange” (usado para encontrar valores extremos de una función de n variables), el cual cumple la tarea de asociar el conjunto de ecuaciones que representan las restricciones a la función que reproduce el modelo del sistema dinámico para el problema de optimización del costo asociado con el Langrangiano.

Se dice que el problema (o el Hamiltoniano \mathcal{H}) de control es regular (véase [88]), cuando para cada terna (t, x, λ) arbitrarios existe un único valor de control

$$u^0(t, x, \lambda) \triangleq \arg \min_u \mathcal{H}(t, x, \lambda, u), \quad (2.15)$$

que minimice el Hamiltoniano del sistema 2.14, es decir, se cumple que

$$\mathcal{H}(t, x, \lambda, u^0(t, x, \lambda)) \leq \mathcal{H}(t, x, \lambda, u) \quad \forall u \in \mathbb{U}. \quad (2.16)$$

Justificados en esta minimización de \mathcal{H} , en este caso podemos definir el Hamiltoniano “óptimo” \mathcal{H}^0 como

$$\mathcal{H}^0(t, x, \lambda) \triangleq \mathcal{H}(t, x, \lambda, u^0(t, x, \lambda)), \quad (2.17)$$

2.2 La función de valor y el formalismo Hamiltoniano

y en el caso que \mathcal{H} resulte una función suave con respecto a todas sus variables, la definición de u^0 implica claramente

$$\frac{\partial \mathcal{H}}{\partial u}(t, x, \lambda, u^0(t, x, \lambda)) \equiv 0, \quad (2.18)$$

cuya relación con la función de valor es (al combinar las ecuaciones (2.14 - 2.17)), da lugar a una ecuación en derivadas parciales asociada al problema de control óptimo conocida como la “Hamilton-Jacobi-Bellman” (HJB)(ver [44]):

$$\frac{\partial \mathcal{V}}{\partial t}(t, x) + \mathcal{H}^0\left(t, x, \left[\frac{\partial \mathcal{V}}{\partial x}\right]'(t, x)\right) = 0, \quad \mathcal{V}(t_f, x) = K(x(t_f)). \quad (2.19)$$

De (2.15 - 2.19) resulta claro que la trayectoria de coestado óptima λ^* verifica

$$\lambda^*(t) = \left(\frac{\partial \mathcal{V}}{\partial x}(t, x^*(t))\right)', \quad t \in [0, t_f], \quad (2.20)$$

y en particular, para $t = t_f$ la ecuación (2.20) nos brinda la siguiente condición final,

$$\lambda^*(t_f) = \left(\frac{\partial \mathcal{V}}{\partial x}(t_f, x^*(t_f))\right)'. \quad (2.21)$$

Además, el control óptimo Hamiltoniano puede ser expresado como (ver [134], p. 361)

$$u^*(t) = u^0(x^*(t), \lambda^*(t)). \quad (2.22)$$

La importancia de este resultado es que todo el problema de control óptimo se reduce a hallar la solución de la HJB. Sin embargo, ésta resulta difícil de resolver inclusive numéricamente, y si no hay regularidad no existe solución a menos que se permita una noción generalizada de la solución [21, 134]. Para sistemas lineales con costo cuadrático, la función de valor es cuadrática en x , por ende, la HJB se reduce a una ecuación diferencial ordinaria (ODE de su sigla en inglés), conocida como la “Riccati Differential Equation” RDE, para la cual es posible encontrar soluciones numéricas.

Del Hamiltoniano y control óptimo surgen las ecuaciones canónicas Hamiltonianas (HCEs de su sigla en inglés) (véase [117] para problemas generales y [134],

2. FORMALISMO DEL PROBLEMA DE CONTROL ÓPTIMO

p. 406 para el caso con estado final libre)

$$\dot{x} = \left(\frac{\partial \mathcal{H}^0}{\partial \lambda} \right)' (x, \lambda) \triangleq \mathcal{F}(x, \lambda); \quad x(0) = x_0, \quad (2.23)$$

$$\dot{\lambda} = - \left(\frac{\partial \mathcal{H}^0}{\partial x} \right)' (x, \lambda) \triangleq -\mathcal{G}(x, \lambda); \quad \lambda(t_f) = \frac{\partial \mathcal{K}}{\partial x}(x(t_f)), \quad (2.24)$$

las cuales definen un sistema de ecuaciones diferenciales ordinarias de dimensión $2n$ para el campo vectorial Hamiltoniano

$$\mathcal{X} \triangleq \begin{pmatrix} x \\ \lambda \end{pmatrix},$$

$$\begin{pmatrix} \dot{x} \\ \dot{\lambda} \end{pmatrix} = \begin{pmatrix} \mathcal{F}(x, \lambda) \\ -\mathcal{G}(x, \lambda) \end{pmatrix} = \dot{\mathcal{X}}(x, \lambda). \quad (2.25)$$

Las soluciones de las ecuaciones (2.23, 2.24) son equivalentes a resolver la HJB y además brindan las trayectorias óptimas tanto para el estado $x^*(t)$ como para el coestado óptimo $\lambda^*(t)$. Para las trayectorias óptimas el Hamiltoniano permanece constante en todo el intervalo $[0, t_f]$, o visto de otra manera a través de (véase [1]),

$$\frac{d\mathcal{H}^0}{dt}(x^*(t), \lambda^*(t)) = \left(\frac{\partial \mathcal{H}^0}{\partial x} \right) \cdot \mathcal{F} + \left(\frac{\partial \mathcal{H}^0}{\partial \lambda} \right) \cdot [-\mathcal{G}] = 0. \quad (2.26)$$

2.2.1 Formalismo Hamiltoniano en problemas lineales

La solución clásica al problema de control óptimo se da para sistemas lineales, variantes en el tiempo y sujeto a un funcional de costo cuadrático LQR, dado por la ecuación (2.3) de dimensión n , con condiciones finales libres, de horizonte finito y controles irrestrictos. El funcional de costo

$$\mathcal{J}(u) = \int_0^{t_f} [x'Q(t)x + u'R(t)u]dt + x'(t_f)Sx(t_f), \quad (2.27)$$

debe ser minimizado con respecto a todas las trayectorias de control admisibles (aquí las continuas a tramos) $u : [0, t_f] \rightarrow \mathbb{R}^m$, sujeto a la restricción dinámica lineal del tipo

$$\dot{x} = A(t)x + B(t)u; \quad x(0) = x_0, \quad (2.28)$$

2.2 La función de valor y el formalismo Hamiltoniano

donde x, u mantienen su definición de la Sección 2.1. Las matrices de las ecuaciones (2.27, 2.28) son asumidas con las siguientes propiedades: $Q(t), S \geq 0$ (definidas semi-positivas) de dimensión $n \times n$, $R(t) > 0$ (definida positiva) de dimensión $m \times m$, $A(t)$ es de $n \times n$, $B(t)$ de $n \times m$, y el par (A, B) controlable. Los objetos concernientes al funcional son del tipo:

$$L(t, x, u) = x'Q(t)x + u'R(t)u, \quad (2.29)$$

$$\mathcal{K}(x(t_f)) = x'(t_f)Sx(t_f), \quad t_f < \infty. \quad (2.30)$$

Bajo estas consideraciones el Hamiltoniano $\mathcal{H}(t, x, \lambda, u)$ del problema se expresa como

$$\mathcal{H}(t, x, \lambda, u) = L(t, x, u) + \lambda'f(t, x, u) = x'Q(t)x + u'R(t)u + \lambda'(A(t)x + B(t)u), \quad (2.31)$$

dado que se asumió que el Hamiltoniano $\mathcal{H}(t, x, \lambda, u)$ es diferenciable y de la condición $\frac{\partial \mathcal{H}}{\partial u} = 0$ (ecuación (2.18)), la acción de control óptima u^* para este problema toma la siguiente forma explicita

$$u^0(t, x^*(t), \lambda^*(t)) = u^*(t) = -\frac{1}{2}R^{-1}(t)B'(t)\lambda^*(t), \quad (2.32)$$

(independiente de x), también llamado control “ \mathcal{H} -mínimo”. La solución de la ecuación HJB para este caso tiene la forma (ver [8, 18, 134])

$$\mathcal{V}(t, x) = x'P(t)x, \quad (2.33)$$

con el coestado óptimo resultante λ^* y de acuerdo a la ecuación (2.20), la derivada de la función de valor (véase [99]),

$$\lambda^*(t) = \left(\frac{\partial \mathcal{V}}{\partial x} \right)' (t, x^*(t)) = 2P(t)x^*(t). \quad (2.34)$$

donde $x^*(t)$ denota a la trayectoria óptima de estado y $P(t)$ a la solución de la ecuación diferencial de Riccati (RDE de su sigla en inglés)

$$\dot{P} = PW(t)P - PA(t) - A(t)'P - Q(t), \quad (2.35)$$

2. FORMALISMO DEL PROBLEMA DE CONTROL ÓPTIMO

con condición de borde final $P(t_f) = S$ y $W(t) \triangleq B(t)R^{-1}(t)B'(t)$. La ecuación (2.35) permite obtener la solución del problema regular sin restricciones en modo “feedback”, ya que combinando las ecuaciones (2.32, 2.34), la acción de control en lazo abierto (que depende de los coestados del sistema) se convierte en una ecuación con dependencia de los estados

$$u^*(t) = -R^{-1}(t)B'(t)P(t)x^*(t). \quad (2.36)$$

Las trayectorias óptimas del estado y coestado son soluciones de las ecuaciones canónicas Hamiltonianas en el espacio de fase dadas por las ecuaciones (2.23, 2.24). Para este caso en particular, si se considera el Hamiltoniano óptimo el dado por la ecuación (2.31), resultan ser

$$\dot{x} = A(t)x + B(t)u^0(t, x, \lambda) = A(t)x - \frac{1}{2}W(t)\lambda, \quad (2.37)$$

$$\dot{\lambda} = 2 \left[\dot{P}(t)x + P(t)\dot{x} \right] = -2Q(t)x - A'(t)\lambda. \quad (2.38)$$

Las ecuaciones anteriores pueden agruparse en una ecuación diferencial matricial lineal, variante en el tiempo, de dimensión $2n$

$$\dot{\Psi} = \mathbf{H}(t) \Psi, \quad (2.39)$$

con

$$\Psi(t) \triangleq \begin{pmatrix} x(t) \\ \lambda(t) \end{pmatrix}, \mathbf{H}(t) \triangleq \begin{pmatrix} A(t) & -\frac{1}{2}W(t) \\ -2Q(t) & -A'(t) \end{pmatrix} \quad (2.40)$$

y condiciones de borde mixtas

$$x(0) = x_0, \lambda(t_f) = 2Sx(t_f). \quad (2.41)$$

En [18, 134] se pueden consultar los tratamientos detallados de este problema.

2.3 Controles acotados y el Principio de Pontriagyn

En la mayoría de aplicaciones practicas, la variable manipulada puede sólo asumir valores en un conjunto de valores acotados [67, 101, 131]. El termino ‘manipulado’ indica que una persona o un instrumento asigna un valor a una señal generada por medios físicos también llamada ‘entrada’. En algunos casos, para los valores de estos controles hay limites que no se pueden exceder; por ejemplo la cantidad de combustible suministrado a un motor, temperatura, corriente, voltaje, etc., no pueden tomar valores arbitrarios. La variable manipulada se restringe entonces a tomar valores en el interior y en los limites de un subconjunto acotado del espacio métrico, entonces es consecuente afirmar que los valores admisibles del control están en un subconjunto compacto de \mathbb{R}^m . La figura 2.3 muestra la saturación de la variable manipulada para $m = 1$.

$$u(t) \in \mathbb{U} := [u_{\text{mín}}, u_{\text{máx}}] . \quad (2.42)$$

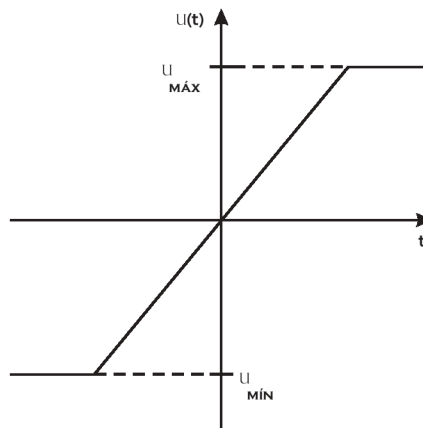


Figura 2.3: Control saturado para dimensión $m = 1$.

Las características cualitativas de la soluciones de control óptimo para problemas con restricciones difieren sustancialmente de la solución del respectivo problema irrestricto [8, 43, 46, 117]. En la teoría quedan aún preguntas abiertas como: cuánto realmente difieren dichas soluciones en el costo alcanzado, cuál clase de problemas generan soluciones “bang-bang”, o cuándo las soluciones restrictas son sólo saturaciones de trayectorias óptimas del mismo problema sin restricciones.

2. FORMALISMO DEL PROBLEMA DE CONTROL ÓPTIMO

La búsqueda de soluciones a los problemas restringidos frecuentemente conduce al Principio del Máximo de Pontryagin (PMP) [117]. Sin embargo, aún cuando sea computable, la solución dada por el PMP no es suficientemente flexible para tratar perturbaciones en los estados, es decir, el PMP genera sólo trayectorias de control en lazo abierto (no se garantizan soluciones del tipo feedback).

Si se asume que existe un instante de tiempo $\tau \in (0, t_f)$ para los que el problema de control óptimo restringido toma el valor: $u_{x_0}^*(\tau) \in (u_{\min}, u_{\max}) = \text{Int}(\mathbb{U})$, tal que exista una solución de control óptimo $u_{x_0}^*(\cdot)$ (para el problema de control con restricciones y todas las condiciones iniciales en x_0). A partir de resultados estándares [88, 134] se deduce que para cada t en alguna vecindad de τ , una trayectoria de control \tilde{u} definida por

$$\tilde{u}(t) \triangleq u^0(x_{x_0}^*(t), \lambda^*(t)), \quad (2.43)$$

permite construir la estrategia óptima correspondiente $u_{x_0}^*$, de la siguiente manera:

$$u_{x_0}^*(t) = \tilde{u}^{sat}(t) = \begin{cases} u_{\min} & \text{si } \tilde{u}(t) \leq u_{\min} \\ \tilde{u}(t) & \text{si } u_{\min} < \tilde{u}(t) < u_{\max} \\ u_{\max} & \text{si } \tilde{u}(t) \geq u_{\max} \end{cases} . \quad (2.44)$$

La saturación tiene efectos complicados sobre las prestaciones de un sistema de control: éstos pueden ser bajos rendimientos o inestabilidad a grandes perturbaciones. En general, la existencia de saturación propende a reducir la ganancia del dispositivo (por ejemplo el amplificador) según se incrementa la entrada. El resultado consecuente es, si un sistema es inestable en su rango lineal, este comportamiento divergente se puede convertir en una oscilación automantenida, debido a la inhibición creada por el componente de saturación sobre las señales del sistema. Por otro lado, si el sistema es estable en su rango lineal, la saturación tiende a bajar la respuesta del sistema, porque se reduce la ganancia efectiva [67, 131].

Las características de la inteligencia que suelen calificarse de analíticas son en sí mismas poco susceptibles de análisis. Sólo las apreciamos a través de sus resultados. Entre otras cosas sabemos que, para aquel que la posee en alto grado, son fuentes del mas vivo goce.

Edgar Allan Poe

CAPÍTULO

3

Tratamiento de Sistemas Lineales

La mayor parte de los problemas físicos, económicos, sociales, tecnológicos o biológicos variantes en el tiempo pueden ser abordados desde el estudio de sistemas dinámicos. Los sistemas físicos llamados “sistemas lineales” sólo lo son en rangos de operación limitados. Aunque los problemas o procesos sean básicamente no lineales, el estudio de sistemas lineales continúa y continuará demandando gran esfuerzo y tiempo de estudio a la comunidad científica en general, ya que éstos permiten realizar estudios analíticos y partir de soluciones exactas. Incluso cuando se trabaja con sistemas no lineales, son frecuentemente utilizados en algún grado los resultados de sistemas lineales.

En el libro [87] el autor escribe en el prefacio

“Linear systems have been under study for a long time, and from several different points of view, in physics, mathematics, engineering, and many other fields. But the subject is such a fundamental and deep one that there is no doubt that linear systems will continue to be an object of study for as long as one can foresee.”

3. TRATAMIENTO LINEAL

Este capítulo explica el tratamiento para sistemas lineales, empezando con un tratamiento numérico para problemas con un sólo período regular; en el que se plantea la obtención del control óptimo por medio de la actualización de los valores de fases finales, haciendo uso para ello de objetos matemáticos. También se analiza para sistemas que presentan varios períodos regulares y una forma de acometerlos. Las estrategias planteadas son exhibidas mediante aplicaciones numéricas.

3.1 Tratamiento Numérico del Problema LQR por Actualización de los Valores de Fase Finales

Un nuevo enfoque fue desarrollado por el Grupo de Sistemas No Lineales del INTEC del cual formo parte, para aproximar la solución del problema LQR, basado en la actualización de los estados finales y el coestado del problema regular, y moviendo ligeramente los tiempos de conmutación (los instantes donde el control alcanza los límites). La idea inicial fue la propuesta de un control sub-óptimo en forma de feedback usando la solución de alguna ecuación de Riccati compatible. El método del gradiente es aplicado para reducir el costo vía fórmula explícitas algebraicas para sus derivadas con respecto a los estados/coestados ocultos del problema regular y los tiempos de conmutación. Este método numérico es considerado eficiente porque no involucra integración de los estados o costos de trayectoria, además de reducir al mínimo la dimensión de los parámetros desconocidos en la condición final. Los objetos relevantes son calculados desde unas pocas matrices auxiliares, las cuales son utilizadas sólo una vez.

El LQR es probablemente, el problema más estudiado y utilizado en la literatura de control óptimo. Por otra parte el formalismo Hamiltoniano (visto en la sección 2.2) ha sido central en el desarrollo de la teoría moderna de control óptimo [3, 8, 44, 117, 141]. El problema del regulador cuadrático lineal de horizonte finito n -dimensional con controles no acotados conduce a $2n$ ecuaciones diferenciales ordinarias (2.23, 2.24). Existen métodos tradicionales para resolver este problema de condiciones de frontera, a veces transformándolo a un sistema de condiciones iniciales al introducir ciertos objetos matemáticos adicionales [40, 134]. En problemas de horizonte infinito, para algunas generalizaciones como el regulador cuadrático bilineal, y también en cambios de set-point, existen intentos por buscar las condiciones iniciales desconocidas para la variable del coestado a partir de los datos de cada problema en particular, lo que permite integrar la ecuaciones Hamiltonianas on-line con el proceso de control relacionado [38].

En el contexto de los controles acotados, el problema de control óptimo es en general no regular, y todavía no consta con un método estándar de resolución [8, 94, 118, 127, 134, 137]. Desde los años sesenta, el PMP ha sido el marco teórico necesario para tratar dichos problemas no regulares. Para resolver el LQR con

3. TRATAMIENTO LINEAL

restricciones se toma ventaja de ciertas relaciones matemáticas que existen entre el PMP y el formalismo Hamiltoniano. El resultado teórico subyacente [43], es base de esta sección. El tratamiento puede ser parafraseado de la siguiente manera: en general, la solución óptima de un problema LQR ‘simple’ con controles acotados es la saturación de la solución de otro LQR irrestricto, este último con la misma dinámica y la misma función de coste que el original, pero con diferentes condiciones iniciales \tilde{x}_0 , y sujeto a una penalización final cuadrática con matriz de penalización final \tilde{S} diferente a la original. En este contexto se denomina un problema LQR ‘simple’ a aquel que posee un único período regular.

En [45], se desarrollaron estrategias “off-line” y “on-line” (basadas en el método del gradiente) para detectar las “nuevas” condiciones iniciales \tilde{x}_0 y la matriz de penalización final \tilde{S} . Se daban allí fórmulas algebraicas para calcular las derivadas parciales del costo con respecto a \tilde{S} y \tilde{x}_0 , con el objetivo de evitar la integración numérica de trayectoria de estados, control y/o costo, y bajar el esfuerzo computacional.

La contribución principal del método viene de la reformulación del resultado teórico previo de [43]. Se muestra que en lugar de la penalización final \tilde{S} de $n \times n$ y las condiciones iniciales \tilde{x}_0 , es suficiente encontrar el vector de estados y coestados finales (ρ, μ) de dimensión $2n$. Entonces, se consiguen nuevas fórmulas algebraicas para calcular las derivadas parciales con respecto a los nuevos parámetros. Desde el punto de vista práctico, buscar estos parámetros en lugar de los originales implica una reducción significativa en la dimensión de las incógnitas y por ende una disminución del esfuerzo computacional, que para el tratamiento de sistemas de gran dimensión puede resultar determinante en cuanto a la viabilidad de la instrumentación.

Se pretende que la estrategia de control trabaje en forma de feedback cuando el control esté acotado, entonces el conocimiento preciso de la condición inicial para el problema irrestricto LQR relacionado no es sustancial. En este contexto es suficiente conocer la localización de los instantes de tiempo donde el control alcanza los límites. El esquema propuesto actualiza (ρ, μ) , y también los tiempos de “conmutación” τ_i donde se satura el control, mientras que el costo total es reducido vía el método del gradiente [19, 62, 114]. En general, se espera que la solución que

3.2 Problema LQR Regular para sistemas lineales

produce el método sea sub-óptima, aunque en algunos casos es posible alcanzar la solución óptima numéricamente.

El esquema numérico toma ventaja de la capacidad de actualizar online las matrices de Riccati que corresponden a un rango de valores de parámetros en la penalización final, a través de la solución de un par de ecuaciones diferenciales parciales de primer orden [33, 39]. Dada esta simplicidad y menor esfuerzo computacional, el algoritmo puede ser considerado como una potencial herramienta a ser usada en combinación con el convencional “receding” o “shrinking horizon” [23], en un contexto de MPC ampliado, y tal vez podría llegar a ser considerado como una alternativa a enfoques de programación sofisticados [121], los cuales dependen de la discretización adoptada en el tiempo y espacio.

3.2 Problema LQR Regular para sistemas lineales

El problema LQR invariante en el tiempo está sujeto a un funcional de costo de la forma (2.27), y a una restricción dinámica lineal del tipo (2.28), con matrices A, B, Q, R constantes y de dimensiones apropiadas y con (A, B) controlable. Este planteamiento se denotará como el problema $(A, B, Q, R, S, t_f, \mathbb{R}^m, x_0)$. En esta sección, se mantendrá la dimensión de la entrada siempre como $m = 1$. La generalización para mayores variables manipuladas $m > 1$, aunque no presenta obstáculos esenciales, es engorrosa y se deja para estudios posteriores.

Equivalentemente a la ecuación (2.36), la ley de control óptima por retroalimentación de estados (o feedback) en el caso regular es

$$u_f(t, x) = -R^{-1}B'P(t)x . \quad (3.1)$$

Cuando los valores de control son restringidos, la regularidad global del Hamiltoniano no se puede asegurar, por lo tanto es necesaria una búsqueda alternativa de la estrategia de control óptimo.

La matriz Hamiltoniana del problema original es de la forma (2.40), y para el problema invariante en el tiempo, con las matrices constantes queda entonces determinada por

$$\mathbf{H} := \begin{pmatrix} A & -W/2 \\ -2Q & -A' \end{pmatrix} , \quad (3.2)$$

3. TRATAMIENTO LINEAL

asociada a la matriz fundamental del sistema

$$\mathbf{U}(t) := e^{-\mathbf{H}t}, \quad (3.3)$$

que serán empleadas en la elaboración de las fórmulas algebraicas para la derivada parcial del costo. La matriz $\mathbf{U}(t)$ es de $2n \times 2n$, y por conveniencia será particionada en 4 sub-matrices de $n \times n$ denotadas como

$$\begin{pmatrix} \mathbf{U}_1(t) & \mathbf{U}_2(t) \\ \mathbf{U}_3(t) & \mathbf{U}_4(t) \end{pmatrix} := \mathbf{U}(t). \quad (3.4)$$

En resultados previos del problema LQR en la forma Hamiltoniana (ver [18, 44, 53, 134]) aparecen otros objetos matemáticos que serán útiles en la comprensión de la solución con restricciones, por ejemplo, las matrices $\alpha(t_f, S)$ y $\beta(t_f, S)$. Estas dos matrices auxiliares son definidas como:

$$\begin{pmatrix} \alpha(t_f - t, S) \\ \beta(t_f - t, S) \end{pmatrix} = \mathbf{U}(t_f - t) \begin{pmatrix} I \\ 2S \end{pmatrix}. \quad (3.5)$$

$$\begin{pmatrix} \alpha(t_f, S) \\ \beta(t_f, S) \end{pmatrix} \triangleq \mathbf{U}(t_f) \begin{pmatrix} I \\ 2S \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1(t_f) + 2\mathbf{U}_2(t_f)S \\ \mathbf{U}_3(t_f) + 2\mathbf{U}_4(t_f)S \end{pmatrix}. \quad (3.6)$$

Las matrices permiten calcular, para un problema LQR sin restricciones, la solución de la RDE $P(\cdot, t_f, S)$ a través de la fórmula algebraica [44]

$$P(t, t_f, S) = \frac{1}{2}\beta(t_f - t, S) [\alpha(t_f - t, S)]^{-1} \quad \forall t \in [0, t_f], \quad (3.7)$$

además las matrices α y β están también relacionadas a las condiciones de frontera de las HCEs a través de la siguiente relación

$$x(t_f) = \alpha(t_f, S)^{-1}x(0), \quad \lambda(0) = \beta(t_f, S)x(t_f), \quad \lambda(t_f) = 2Sx(t_f). \quad (3.8)$$

3.3 Control con Restricciones. Actualización de Estados y Coestados Finales

Problemas con cotas en la variable de control como los expuestos en el capítulo 2, sección 2.3, serán tratados en este capítulo con la intención de explotar el resultado teórico expuesto en [43]:

Si se asume que existe un instante de tiempo $t \in (0, t_f)$ para el cual se verifica: $u_{x_0}^*(t) \in \text{Int}(\mathbb{U}) = (u_{\text{mín}}, u_{\text{máx}})$, entonces, existirá un intervalo de tiempo $I \subset (0, t_f)$ que contiene a t tal que la trayectoria de fase óptima $\{x_{x_0}^*, \lambda_{x_0}^*\}$ del problema original $(A, B, Q, R, S, t_f, \mathbb{U}, x_0)$ coincidirá en I con la trayectoria de fase $\{\hat{x}, \hat{\lambda}\}$, quien a su vez es óptima para cierto (en general distinto) problema $(A, B, Q, R, \hat{S}, t_f, \mathbb{R}, \hat{x}_0)$.

El problema $(A, B, Q, R, \hat{S}, t_f, \mathbb{R}, \hat{x}_0)$ no tiene restricciones sobre los valores de control ($\mathbb{U} \equiv \mathbb{R}$), y lo llamaremos el problema ‘oculto’, ‘relacionado’, proceso ‘fantasma’, o el problema $\hat{\cdot}$. En lo que sigue, se asumirá que existe al menos un intervalo ‘regular’ $(\tau_1, \tau_2) \subset (0, t_f)$ donde el control toma valores en $(u_{\text{mín}}, u_{\text{máx}})$.

El resultado transforma el problema original con controles acotados (cuya solución debe buscarse en el espacio de dimensión infinita de las trayectorias admisibles del control) en una búsqueda de dimensión finita (para las condiciones finales ocultas (\hat{x}_0, \hat{S})). Un comportamiento típico de estos objetos es representado en la figura 3.1.

La generación de estrategias de control sub-óptimas basadas en la aproximación de los parámetros desconocidos $\{\hat{S}, \hat{x}_0\}$ ha sido publicada recientemente, también en [43]. En [45] se utilizan fórmulas algebraicas para calcular las derivadas parciales del costo total con respecto a los coeficientes de la matriz \hat{S} y las condiciones iniciales \hat{x}_0 , sin necesidad de generar las trayectorias de estado ni evaluar el costo. En dicho caso, el número de variables a actualizar fue $\frac{(n+3)n}{2}$, que para aplicaciones de gran dimensión puede resultar poco práctico. Por eso se buscó reducir el número de incógnitas, evitando todavía la integración numérica de ODEs, como en los resultados presentados de [45]. Una consecuencia directa es la reducción del esfuerzo computacional necesario para minimizar el costo total.

3. TRATAMIENTO LINEAL

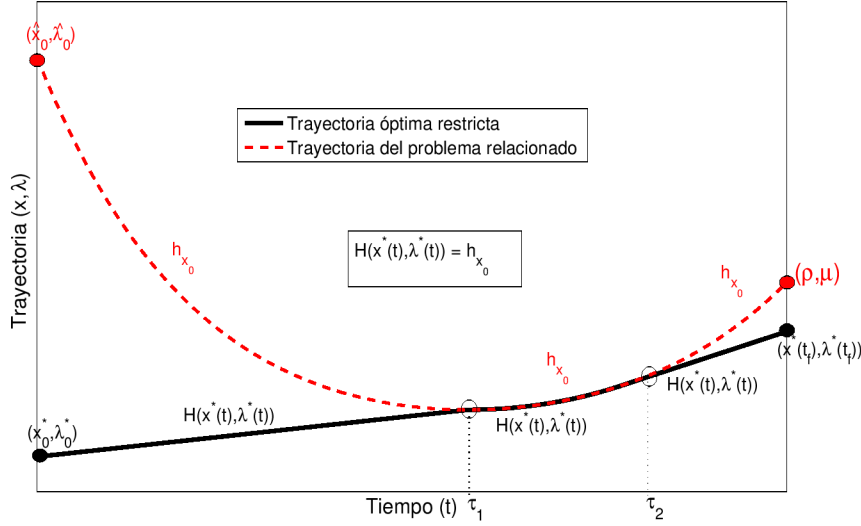


Figura 3.1: Comportamiento cualitativo típico de trayectorias de fase óptimas para el problema original restringido $\{x_{x_0}^*, \lambda_{x_0}^*\}$, y para el proceso relacionado irrestricto $\{\hat{x}, \hat{\lambda}\}$.

3.3.1 Un hecho teórico relevante para la actualización de los parámetros

Sea $(A, B, Q, R, \hat{S}, t_f, \mathbb{R}, \hat{x}_0)$ el problema sin restricciones relacionado, y sea $\{\hat{x}, \hat{\lambda}\}$ la trayectoria del estado/coestado óptima de dicho problema. Después de definir $\rho := \hat{x}(t_f)$ y $\mu := \hat{\lambda}(t_f)$,

$$\tilde{S} := \frac{1}{2} \frac{\mu \mu'}{\rho' \mu} \quad (3.9)$$

(siempre que $\rho' \hat{S} \rho > 0$, si no $\tilde{S} := \hat{S}$), se puede afirmar lo siguiente:

Teorema 3.1. *El problema $(A, B, Q, R, \tilde{S}, t_f, \mathbb{R}, \hat{x}_0)$ tiene las mismas trayectorias óptimas del estado y coestado que el problema $(A, B, Q, R, \hat{S}, t_f, \mathbb{R}, \hat{x}_0)$, o sea $\{\hat{x}, \hat{\lambda}\}$.*

Demostración. Denotaremos a la trayectoria $\{\tilde{x}, \tilde{\lambda}\}$ como la solución a las HCEs de cualquiera de los problemas (dichas ODEs no dependen de \tilde{S} o \hat{S}) con condi-

ciones de frontera $\tilde{x}(t_f) = \rho$, $\tilde{\lambda}(t_f) = 2\tilde{S}\rho = \mu$. Asimismo nótese que

$$\tilde{\lambda}(t_f) = 2\tilde{S}\rho = \mu = 2\hat{S}\rho, \quad (3.10)$$

y por lo tanto $\{\tilde{x}, \tilde{\lambda}\}$ es la trayectoria óptima del problema $(A, B, Q, R, \hat{S}, t_f, \mathbb{R}, \hat{x}_0)$, es decir

$$\{\tilde{x}, \tilde{\lambda}\} \equiv \{\hat{x}, \hat{\lambda}\}. \quad (3.11)$$

Pero ya ambos pares son soluciones de las ecuaciones Hamiltonianas, ellas también deben satisfacer $\tilde{\lambda} = 2\tilde{P}\tilde{x}$, $\hat{\lambda} = 2\hat{P}\hat{x}$, con \tilde{P} y \hat{P} dos soluciones de la RDE (2.35) con condiciones finales \tilde{S} y \hat{S} , respectivamente. Por otra parte la solución x de

$$\dot{x} = (A - W\tilde{P})x, \quad x(0) = \hat{x}_0 \quad (3.12)$$

es óptima para el problema $(A, B, Q, R, \tilde{S}, t_f, \mathbb{R}, \hat{x}_0)$. Ahora, definiendo $\delta := x - \hat{x}$, se obtiene

$$\begin{aligned} \dot{\delta} &= A\delta - W(\tilde{P}x - \hat{P}\hat{x}) = A\delta - W(\tilde{P}x - \tilde{P}\tilde{x}) \\ &= A\delta - W(\tilde{P}x - \tilde{P}\hat{x}) = (A - W\tilde{P})\delta. \end{aligned} \quad (3.13)$$

Dado que $\delta(0) = 0$, esto significa que $x \equiv \hat{x} \equiv \tilde{x}$. \square

\square

Corolario 3.1. $\tilde{P}\hat{x} = \tilde{P}\tilde{x} = \hat{P}\hat{x}$, y por lo tanto el feedback óptimo (3.1) es el mismo para los problemas $\hat{\cdot}$ y $\tilde{\cdot}$.

Como consecuencia útil de estos nuevos resultados, los parámetros desconocidos (\hat{S}, \hat{x}_0) pueden ser reemplazados por (ρ, μ) desde que: (i) la trayectoria $(\hat{x}, \hat{\lambda})$ caracterizada por (\hat{S}, \hat{x}_0) también se determina de forma única por (ρ, μ) , y (ii) las matrices de Riccati \tilde{P} , \hat{P} son obtenidas de sus valores finales \tilde{S} , \hat{S} , respectivamente. Estas tienen el mismo efecto en la construcción del feedback óptimo, y por lo tanto son igualmente útiles para los propósitos de control.

También nótese que \hat{S} tiene $\frac{(n+1)n}{2}$ entradas desconocidas (ya que es simétrica), y \hat{x}_0 tiene otras n desconocidas. En total $\frac{(n+3)n}{2}$ desconocidas. En el presente desarrollo el número es reducido a $2n$ (para ρ, μ). Por ejemplo para $n = 100$ la cantidad de parámetros desconocidos se reduce desde 5150 a 200. El procedimiento numérico intentara alcanzar los parámetros ocultos (ρ, μ) , comenzando desde unos apropiados $(\rho_{semilla}, \mu_{semilla})$, y por la aplicación repetida del método del gradiente, forzando el costo a decrecer hasta alcanzar una convergencia.

3. TRATAMIENTO LINEAL

3.3.1.1 Fórmulas algebraicas

3.3.1.2 Objetos auxiliares

Los siguientes controles de tipo (2.44), en este caso realimentado serán utilizados a menudo en las próximas subsecciones para obtener el control sub-óptimo del problema

$$\tilde{u}(t) := \begin{cases} u_{\text{mín}}, & \forall t \in [0, \tau_1) \\ -R^{-1}B'P(t, \rho, \mu)x(t), & \forall t \in [\tau_1, \tau_2) \\ u_{\text{máx}}, & \forall t \in [\tau_2, t_f] \end{cases}, \quad (3.14)$$

donde $\tilde{u}(t)$ es una notación simplificada para $\tilde{u}_{\rho, \mu, \tau_1, \tau_2}(t)$, la cual es usada para indicar que el “feedback” está asociado a los parámetros $(\rho, \mu, \tau_1, \tau_2)$. La nueva estrategia propone variaciones de las variables ρ y μ de tal forma que se puedan generar matrices $\tilde{S}_i \rightarrow \tilde{S}$. Más precisamente, el procedimiento deberá tener en cuenta las siguientes consideraciones:

(i) La matriz inicial \tilde{S}_0 será construida a partir de los valores finales de las trayectorias óptimas del estado y coestado del problema irrestricto (2.27, 2.28), aplicando el feedback (3.1). El coestado verifica en t_f que $\lambda(t_f) = 2P(t_f)x(t_f) = 2Sx(t_f)$, y entonces denotando $\rho_{\text{semilla}} := x(t_f)$, $\mu_{\text{semilla}} := \lambda(t_f)$, la matriz semilla \tilde{S}_0 se puede calcular como

$$\tilde{S}_0 := \frac{1}{2} \frac{\mu_{\text{semilla}} \mu'_{\text{semilla}}}{\rho'_{\text{semilla}} \mu_{\text{semilla}}}. \quad (3.15)$$

(ii) Se adopta la siguiente trayectoria de control “semilla” para iniciar el método iterativo,

$$u_{\text{semilla}}(t) := \begin{cases} u_{\text{mín}} & \text{if } -R^{-1}B'P(t, \tilde{S}_0)x(t) \leq u_{\text{mín}} \\ u_{\text{máx}} & \text{if } -R^{-1}B'P(t, \tilde{S}_0)x(t) \geq u_{\text{máx}} \\ -R^{-1}B'P(t, \tilde{S}_0)x(t) & \text{otro caso} \end{cases}. \quad (3.16)$$

La trayectoria de estados correspondiente a el control u_{semilla} empezando en x_0 , es decir $x_{u_{\text{semilla}}}$, se denotará como x_{semilla} . Nótese que a partir del control semilla y de las ecuaciones de estado, simuladas para la nueva matriz final \tilde{S}_0 , se pueden detectar, en caso de que existan, los primeros valores de los tiempos de conmutación denotados como $\tau_{1,0} \leq \tau_{2,0}$.

3.3 Control con Restricciones

(iii) En lo que sigue $P(t, \rho, \mu)$ denotará la solución de la RDE (2.35), con condición final $P(t_f) = \tilde{S}$. Cuando sea claro en el texto la dependencia de P con ρ, μ , se utilizará la notación simplificada $P(t)$. De la ecuación (3.7) se obtiene la siguiente identidad, que será de utilidad (para $\eta := \rho \circ \mu$)

$$\begin{aligned} \frac{\partial P(t, \eta)}{\partial \eta} &= \frac{\partial \left[\frac{1}{2} \beta(t_f - t, \eta) [\alpha(t_f - t, \eta)]^{-1} \right]}{\partial \eta} \\ &= \frac{1}{2} [\beta_\eta \alpha^{-1} - \beta \alpha^{-1} \alpha_\eta \alpha^{-1}] (t_f - t, \eta) \\ &= \frac{1}{2} [\beta_\eta - 2P(t, \eta) \alpha_\eta] \alpha^{-1}, \end{aligned} \quad (3.17)$$

donde $\alpha, \alpha_\eta := \frac{\partial \alpha}{\partial \eta}, \beta_\eta := \frac{\partial \beta}{\partial \eta}$ deben ser evaluadas en $(t_f - t, \rho, \mu)$, y a partir de las ecuaciones (3.4, 3.6), las derivadas parciales α_η, β_η resultan

$$\alpha_\eta = 2\mathbf{U}_2 \frac{\partial S}{\partial \eta}, \quad \beta_\eta = 2\mathbf{U}_4 \frac{\partial S}{\partial \eta}. \quad (3.18)$$

(iv) La matriz fundamental “saturada” es:

$$\Psi(t, \tau) := \int_\tau^t e^{A(t-\sigma)} d\sigma = e^{At} \int_\tau^t e^{-A\sigma} d\sigma = e^{A(t-\tau)} \int_0^{t-\tau} e^{-As} ds, \quad (3.19)$$

y sus matrices relacionadas

$$\hat{\Psi}(t, \tau) := \int_\tau^t \Psi'(\sigma, \tau) Q e^{A(\sigma-\tau)} d\sigma = \int_0^{t-\tau} \Psi'(s + \tau, \tau) Q e^{As} ds, \quad (3.20)$$

$$\check{\Psi}(t, \tau) := \int_\tau^t e^{A'(\sigma-\tau)} Q e^{A(\sigma-\tau)} d\sigma = \int_0^{t-\tau} e^{A's} Q e^{As} ds, \quad (3.21)$$

también serán necesarias en lo que sigue. Mediante la definición de las matrices $F_1(\Delta), F_2(\Delta), G(\Delta), H(\Delta), K(\Delta), L_2(\Delta), L_3(\Delta), M_1(\Delta)$, y $M_2(\Delta)$, dependientes del escalar Δ , de acuerdo con las siguientes particiones:

$$\begin{pmatrix} F_1(\Delta) & G(\Delta) \\ 0 & F_2(\Delta) \end{pmatrix} := \exp \left(\begin{bmatrix} -A' & Q \\ 0 & A \end{bmatrix} \Delta \right), \quad (3.22)$$

$$\begin{pmatrix} I & H(\Delta) \\ 0 & e^{-A\Delta} \end{pmatrix} := \exp \left(\begin{bmatrix} 0 & I \\ 0 & -A \end{bmatrix} \Delta \right), \quad (3.23)$$

3. TRATAMIENTO LINEAL

$$\begin{pmatrix} I & M_1(\Delta) & K(\Delta) \\ 0 & L_2(\Delta) & M_2(\Delta) \\ 0 & 0 & L_3(\Delta) \end{pmatrix} := \exp \left(\begin{bmatrix} 0 & I & 0 \\ 0 & -A' & Q \\ 0 & 0 & A \end{bmatrix} \Delta \right) ; \quad (3.24)$$

empleando luego algunas fórmulas apropiadas para integrales a través de exponenciales especiales de una matriz hamiltoniana [143], recurriendo luego a la integración por partes y manipulaciones algebraicas similares (apéndice A); se tiene por último que las integrales en las ecuaciones (3.19 - 3.21) pueden ser expresadas como sigue:

$$\check{\Psi}(t, \tau) = F_2'(t - \tau) G(t - \tau) , \quad (3.25)$$

$$\Psi(t, \tau) = e^{A(t-\tau)} H(t - \tau) , \quad (3.26)$$

$$\hat{\Psi}(t, \tau) = \Psi'(t, \tau) e^{-A'(t-\tau)} \check{\Psi}(t, \tau) - K(t - \tau) ; \quad (3.27)$$

mostrando que todas ellas pueden ser calculadas sin necesidad de recurrir a integraciones numéricas.

En lo que sigue, en conjunto con la matriz $\mathbf{U}(t)$, todas las matrices auxiliares serán calculadas, interpoladas off-line y guardadas en memoria para luego ser utilizadas durante el procedimiento de optimización. Entonces se asume que estarán disponibles como funciones de dos variables (t, τ) , en el rango $[0, t_f] \times [0, t_f]$.

Cuando $u(t) \equiv u_{\min}$ en $[0, \tau_1]$, el estado en τ_1 resulta

$$x(\tau_1) = e^{A\tau_1} x_0 + \Psi(\tau_1, 0) B u_{\min} . \quad (3.28)$$

(v) En este punto es importante notar que el estado en τ_2 podría ser calculado del flujo Hamiltoniano. Ya que para cada $t \in [\tau_1, \tau_2)$ el control es $-R^{-1}B'P(t, \rho, \mu)x(t)$ y el coestado (correspondiente a este subconjunto de trayectoria regular), denotado como $\tilde{\lambda}(t)$, es $\tilde{\lambda}(t) = 2P(t, \rho, \mu)x(t)$, y

$$\begin{pmatrix} x(t) \\ \tilde{\lambda}(t) \end{pmatrix} = \mathbf{U}(\tau_1 - t) \begin{pmatrix} x(\tau_1) \\ \tilde{\lambda}(\tau_1) \end{pmatrix} , \quad (3.29)$$

el estado finalmente resulta en

$$\begin{aligned} x(\tau_2) &= \mathbf{U}_1(\tau_1 - \tau_2)x(\tau_1) + \mathbf{U}_2(\tau_1 - \tau_2)\tilde{\lambda}(\tau_1) \\ &= (\mathbf{U}_1(\tau_1 - \tau_2) + 2\mathbf{U}_2(\tau_1 - \tau_2)P(\tau_1, \eta))x(\tau_1) . \end{aligned}$$

En lo que sigue el argumento t de $\mathbf{U}(t)$ será omitido cuando el contexto sea claro y lo permita. Cuando $u(t) \equiv u_{\max}$ en $[\tau_2, t_f]$, el estado en el tiempo final t_f resultará

$$x(t_f) = e^{A(t_f-\tau_2)}x(\tau_2) + \Psi(t_f, \tau_2)B u_{\max} . \quad (3.30)$$

3.3.1.3 Derivadas parciales del costo

Los resultados publicados en [55] indican que el control óptimo $u^*(\cdot)$ para el problema original $(A, B, Q, R, S, t_f, \mathbb{U}, x_0)$ puede ser obtenido por la saturación del control óptimo $\hat{u}(\cdot)$ correspondiente al problema irrestricto $\hat{\cdot}$. Pero en realidad no es necesario determinar toda la trayectoria de $\hat{u}(\cdot)$, mientras que se conozcan los puntos de saturación $\{\tau_1, \tau_2\}$, y también se conozcan los valores que $\hat{u}(\cdot)$ asume para $t \notin \mathcal{J} \triangleq [\tau_1, \tau_2]$. Entonces es suficiente con calcular τ_1, τ_2 y $\hat{P}(t)$, con $t \in \mathcal{J}$ para definir toda la estrategia $u^*(\cdot)$. En lo que sigue este método va a buscar $\rho, \mu, \tau_1, \tau_2$ con el fin de evitar el cálculo de lazo abierto $\hat{u}(\cdot)$ y saturar éste posteriormente. Esto es equivalente a determinar un número $2(n+1)$ de parámetros a ser actualizados vía el método del gradiente.

Se sabe de [55], que el costo total $\mathcal{J}(\tilde{u})$ es diferenciable con respecto a las variables $(\rho, \mu, \tau_1, \tau_2)$. Para obtener las derivadas, por conveniencia el costo total será dividido en 4 partes:

$$J(\rho, \mu, \tau_1, \tau_2) := \mathcal{J}(\tilde{u}) = J_1 + J_2 + J_3 + J_4, \quad (3.31)$$

donde J_1 representa el costo de trayectoria asociado con el período de saturación $[0, \tau_1)$, J_2 con el intervalo regular $[\tau_1, \tau_2]$, J_3 con el otro período de saturación $(\tau_2, t_f]$, y J_4 mide la penalización final. Los costos parciales pueden ser expresados de la siguiente manera:

$$J_1 = Ru_{\min}^2 \tau_1 + \int_0^{\tau_1} x'(t) Q x(t) dt \quad (3.32)$$

$$J_2 = x'(\tau_1) P(\tau_1) x(\tau_1) - x'(\tau_2) P(\tau_2) x(\tau_2), \quad (3.33)$$

$$J_3 = Ru_{\max}^2 (t_f - \tau_2) + \int_{\tau_2}^{t_f} x'(t) Q x(t) dt, \quad (3.34)$$

$$J_4 = x'(t_f) S x(t_f). \quad (3.35)$$

Las derivadas D_η con respecto a las variables ρ y μ (denotadas indistintamente por la variable η) resultan:

$$D_\eta J_1 = 0, \quad (3.36)$$

$$D_\eta J_2 = x'(\tau_1) \left[\frac{\partial P(\tau_1)}{\partial \eta} - 4(\mathbf{U}_1 + 2\mathbf{U}_2 P(\tau_1))' P(\tau_2) \mathbf{U}_2 \frac{\partial P(\tau_1)}{\partial \eta} \dots, \right. \\ \left. - (\mathbf{U}_1 + 2\mathbf{U}_2 P(\tau_1))' \frac{\partial P(\tau_2)}{\partial \eta} (\mathbf{U}_1 + 2\mathbf{U}_2 P(\tau_1)) \right] x(\tau_1), \quad (3.37)$$

3. TRATAMIENTO LINEAL

donde las derivadas parciales de P se calculan a partir de las ecuaciones (3.17-3.18), y

$$\frac{\partial \tilde{S}_{kl}}{\partial \rho_i} = \frac{1}{2} (\mu_k \mu_l) \left(-\frac{\mu_i}{(\rho' \mu)^2} \right), \quad (3.38)$$

$$\frac{\partial \tilde{S}_{kl}}{\partial \mu_j} = \frac{1}{2} \left[\frac{Z_{j(kl)}}{(\rho' \mu)} - \frac{(\mu_k \mu_l) \rho_j}{(\rho' \mu)^2} \right], \quad (3.39)$$

donde

$$Z_{j(kl)} := \frac{\partial \mu_k \mu_l}{\partial \mu_j} = \begin{cases} 2\mu_j & \text{si } k = l = j \\ \mu_k & \text{si } l = j \text{ y } k \neq j \\ \mu_l & \text{si } k = j \text{ y } l \neq j \\ 0 & \text{si } k \neq j \text{ y } l \neq j \end{cases}. \quad (3.40)$$

$$D_\eta J_3 = \int_{\tau_2}^{t_f} 2x'(t) Q \frac{\partial x(t)}{\partial \eta} dt = 4 \left[x'(\tau_2) \check{\Psi}(t_f, \tau_2) + \dots \right. \\ \left. u_{\text{máx}} B' \hat{\Psi}(t_f, \tau_2) \right] \mathbf{U}_2(\tau_1 - \tau_2) \frac{\partial P(\tau_1, \eta)}{\partial \eta} x(\tau_1), \quad (3.41)$$

después de reemplazar $x'(t) = x'(\tau_2) e^{A(t-\tau_2)} + B' \Psi'(t, \tau_2) u_{\text{máx}}$, y expandiendo

$$\frac{\partial x(t)}{\partial \eta} = e^{A(t-\tau_2)} \frac{\partial x(\tau_2)}{\partial \eta} = e^{A(t-\tau_2)} 2 \mathbf{U}_2(\tau_1 - \tau_2) \frac{\partial P(\tau_1, \rho, \mu)}{\partial \eta} x(\tau_1). \quad (3.42)$$

Finalmente,

$$D_\eta J_4 = 2x'(t_f) S \frac{\partial x(t_f)}{\partial \eta} = 4x'(t_f) S e^{A(t_f-\tau_2)} \mathbf{U}_2(\tau_1 - \tau_2) \frac{\partial P(\tau_1, \eta)}{\partial \eta} x(\tau_1). \quad (3.43)$$

Las derivadas con respecto a los tiempos de conmutación τ_1 y τ_2 son

$$D_{\tau_1} J_1 = Ru_{\text{mín}}^2 + x'(\tau_1) Q x(\tau_1), \quad (3.44)$$

$$D_{\tau_1} J_2 = 2x'(\tau_1) P(\tau_1) B u_{\text{mín}} + x'(\tau_1) P(\tau_1) W P(\tau_1) x(\tau_1) \dots \\ - x'(\tau_1) Q x(\tau_1) - 2x'(\tau_2) P(\tau_2) \frac{\partial x(\tau_2)}{\partial \tau_1}, \quad (3.45)$$

$$\begin{aligned}
D_{\tau_1} J_3 &= \int_{\tau_2}^{t_f} 2x'(t) Q e^{A(t-\tau_2)} dt \frac{\partial x(\tau_2)}{\partial \tau_1} \\
&= \int_{\tau_2}^{t_f} (e^{A(t-\tau_2)} x(\tau_2) + \Psi(t_f, \tau_2) B u_{\max})' Q e^{A(t-\tau_2)} dt \frac{\partial x(\tau_2)}{\partial \tau_1} \\
&= x'(\tau_2) \int_{\tau_2}^{t_f} e^{A'(t-\tau_2)} Q e^{A(t-\tau_2)} dt \frac{\partial x(\tau_2)}{\partial \tau_1} \dots \\
&\quad + B' u_{\max} \int_{\tau_2}^{t_f} \Psi'(t_f, \tau_2) Q e^{A(t-\tau_2)} dt \frac{\partial x(\tau_2)}{\partial \tau_1} \\
&= \left(x'(\tau_2) \check{\Psi}(t_f, \tau_2) + u_{\max} B' \hat{\Psi}(t_f, \tau_2) \right) \frac{\partial x(\tau_2)}{\partial \tau_1}. \tag{3.46}
\end{aligned}$$

Para la penalización final de la derivada respecto a τ_1 , de la ecuación (3.30),

$$D_{\tau_1} J_4 = 2x'(t_f) S e^{A(t_f-\tau_2)} \frac{\partial x(\tau_2)}{\partial \tau_1}, \tag{3.47}$$

donde el último termino puede ser expresado en términos de los datos y las matrices auxiliares como

$$\begin{aligned}
\frac{\partial x(\tau_2)}{\partial \tau_1} &= \mathbf{U}_1(\tau_1 - \tau_2)(Ax(\tau_1) + Bu_1) - (A - WP(\tau_2))x(\tau_2) + \dots \tag{3.48} \\
&\quad 2\mathbf{U}_2(\tau_1 - \tau_2)[P(\tau_1)(WP(\tau_1)x(\tau_1) + Bu_1) - (A'P + Q)x(\tau_1)].
\end{aligned}$$

Similarmente, las derivadas con respecto a τ_2 son

$$D_{\tau_2} J_1 = 0, \tag{3.49}$$

$$D_{\tau_2} J_2 = x'(\tau_2)(P(\tau_2)WP(\tau_2) + Q)x(\tau_2), \tag{3.50}$$

$$D_{\tau_2} J_3 = -(Ru_{\max}^2 + x'(\tau_2)Qx(\tau_2)), \tag{3.51}$$

$$D_{\tau_2} J_4 = -2x'(t_f) S e^{A(t_f-\tau_2)} (WP(\tau_2)x(\tau_2) + Bu_{\max}). \tag{3.52}$$

3. TRATAMIENTO LINEAL

3.3.1.4 Actualizando los parámetros

La primera aproximación $\tau_{1,0}, \tau_{2,0}$ a los puntos de saturación óptimos, son calculados después de simular la trayectoria de estados $x_{semilla}$ como se describe en 3.3.1.2. Para $t \in [0, t_f]$ el control es establecido como

$$\tilde{u}_0(t) \equiv u_{semilla}(t). \quad (3.53)$$

Los parámetros $(\rho, \mu, \tau_1, \tau_2)$ son actualizados para construir sucesivas estrategias de control \tilde{u}_j ; $j = 1, 2, \dots$ que decrementsen el valor del costo total,

$$\mathcal{J}(\tilde{u}_{j+1}) \leq \mathcal{J}(\tilde{u}_j) \leq \dots \mathcal{J}(u_{semilla}); \quad j = 1, 2, \dots \quad (3.54)$$

de acuerdo con las prescripciones del método del gradiente:

$$\begin{cases} \rho_j = \rho_{j-1} - \gamma_j \frac{\partial J}{\partial \rho}(\rho_{j-1}, \mu_j, \tau_{1,j}, \tau_{2,j}) \\ \mu_j = \mu_{j-1} - \gamma_j \frac{\partial J}{\partial \mu}(\rho_j, \mu_{j-1}, \tau_{1,j}, \tau_{2,j}) \end{cases}, \quad (3.55)$$

$$\begin{cases} \tau_{1,j} = \tau_{1,j-1} - \gamma_j \frac{\partial J}{\partial \tau_1}(\rho_j, \mu_j, \tau_{1,j-1}, \tau_{2,j}) \\ \tau_{2,j} = \tau_{2,j-1} - \gamma_j \frac{\partial J}{\partial \tau_2}(\rho_j, \mu_j, \tau_{1,j}, \tau_{2,j-1}) \end{cases}, \quad (3.56)$$

hasta que se alcance la convergencia o una decisión práctica de parada, donde γ_j es un pequeño número real que mide la porción del vector gradiente a ser aplicado en cada iteración, elegido y modificado por el usuario.

Algunas versiones mas sofisticadas del método del gradiente pueden ser exploradas en este contexto, por ejemplo, aplicando el método de Fletcher y Reeves [19, 62]. Las direcciones del gradiente conjugado s^j son calculadas secuencialmente comenzando en

$$s^0 := -\nabla J(p^0), \quad (3.57)$$

donde p denota los parámetros involucrados $(\rho, \mu, \tau_1, \tau_2)$, y p^0 contiene sus valores semillas. El valor s^0 es la dirección de búsqueda inicial y ∇ es el operador gradiente (fila). Las ecuaciones actualizadas por la dirección de búsqueda (prescrito por esta técnica) es

$$s^{j+1} = -\nabla J(p^{j+1}) + s^j \frac{\nabla J(p^{j+1}) \nabla' J(p^{j+1})}{\nabla J(p^j) \nabla' J(p^j)}, \quad j = 1, 2, \dots, \quad (3.58)$$

y la actualización de los parámetros es realizada de la siguiente manera:

$$(\rho, \mu, \tau_1, \tau_2)_{j+1} = (\rho, \mu, \tau_1, \tau_2)_j + \gamma_j s^j. \quad (3.59)$$

3.3.1.5 Descripción esquemática para el procedimiento numérico

Una instrumentación práctica de los resultados discutidos en las secciones previas incluye los siguientes pasos básicos:

(i) Calcular las matrices auxiliares como se describe en 3.3.1.2, después guardarlas en memoria. Obtener las primeras aproximaciones de los estados y de la trayectoria de control, $x_{semilla}$ y $u_{semilla}$ respectivamente (por ahora, éstas corresponden a la saturación del control óptimo del problema irrestricto). Estas trayectorias proveen valores iniciales tentativos para los parámetros $(\rho, \mu, \tau_1, \tau_2)$.

(ii) Actualizar los parámetros a través de la versión elegida del método del gradiente y del escalar γ , como se propone en 4.2.1.1.1.

(iii) Confirmar por integración numérica si el costo está realmente decreciendo, y en tal caso retornar al paso (ii). Este paso puede ser ejecutado después de varias iteraciones del método del gradiente, de acuerdo a la intuición del programador y de las particularidades de la aplicación.

(iv) Si el costo no decrece, reducir la velocidad del método por intermedio del parámetro γ hasta que lo haga, o

(v) Adoptar el último resultado como mejor aproximación al control óptimo alcanzado por este método, y parar.

3.4 Aplicaciones y Resultados Numéricos de la Estrategia de Actualización de los Valores de Fase Finales

3.4.1 Caso de estudio: Frenado óptimo de un tren

El primer caso de estudio de la estrategia de actualización de parámetros por intermedio de los valores finales hace referencia a la ‘detención óptima de un tren’ [3, 43, 44, 45, 52], su problema LQR definido por los parámetros:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t); & x_1(0) &= 1, \\ \dot{x}_2(t) &= u(t); & x_2(0) &= -1, \\ u(t) &\in [0, 3]; & 0 \leq t \leq t_f &= 1, \end{aligned} \quad (3.60)$$

$Q = 10I$, $R = 0,5$, y $S = 100I$, donde I es la matriz identidad de 2×2 . La solución óptima ya fue hallada analíticamente en [43]. El resultado de actualizar los parámetros a través de aplicar el método del gradiente a $\tau_1, \tau_2, \rho, \mu$, es mostrado en las figuras (3.3 - 3.6). En la figura 3.3, la semilla y la trayectoria de control óptima son ilustradas. La figura 3.4 muestra la reducción del costo total contra el número de iteraciones hasta obtener un mínimo valor de costo (valor de convergencia), el valor óptimo $J^* = 15,1632$.

La actualización de los tiempos de saturación puede llevar a diferentes situaciones, como las que son representadas en la figura 3.2. Los valores actualizados, después de la aplicación del método del gradiente como en la ecuación (3.56), son denotados en la figura 3.2 por τ_1, τ_2 . La actualización de ρ, μ puede llevar a diferentes tipos de intervalos regulares, por ejemplo $[\tau_1^{sat,1}, \tau_2^{sat,1}]$ o $[\tau_1^{sat,2}, \tau_2^{sat,2}]$. En el caso 1, el control es construido como

$$u^1(t) := \begin{cases} u_{\min} & \text{si } t \in [0, \tau_1] \\ \tilde{u}(t) & \text{si } t \in [\tau_1, \tau_2] \\ u_{\max} & \text{si } t \in [\tau_2, t_f] \end{cases}, \quad (3.61)$$

esto es, los valores saturados τ_1, τ_2 son respetados, forzando al control a presentar saltos debido a los valores de $\tilde{u}(t)$ en el intervalo regular. En el caso 2,

$$u^2(t) := \begin{cases} u_{\min} & \text{si } t \in [0, \tau_1^{sat,2}] \\ \tilde{u}(t) & \text{si } t \in [\tau_1^{sat,2}, \tau_2^{sat,2}] \\ u_{\max} & \text{si } t \in [\tau_2^{sat,2}, t_f] \end{cases}, \quad (3.62)$$

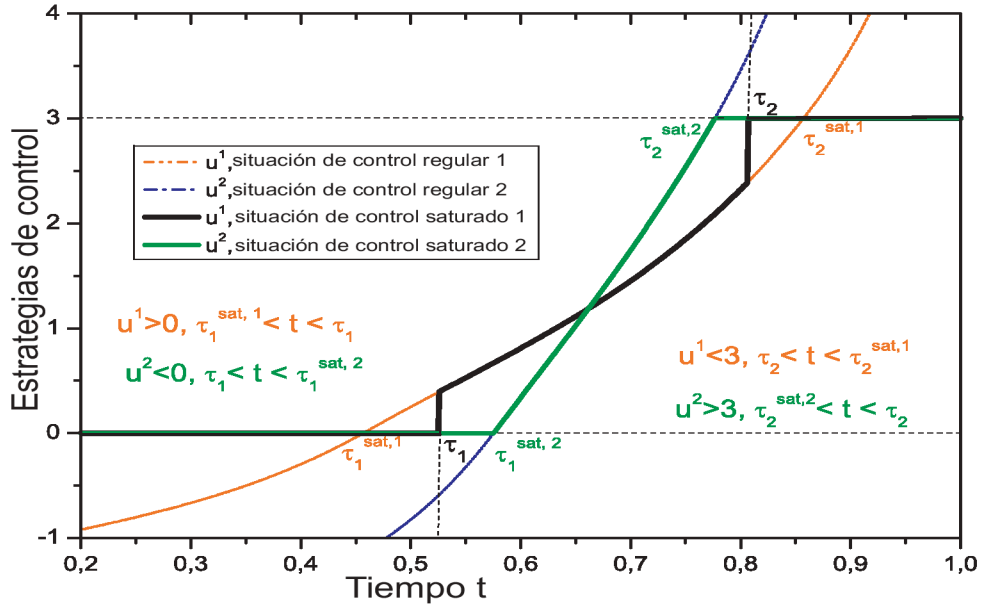


Figura 3.2: Como actualizar los tiempos de conmutación bajo diferentes situaciones en las que el método del gradiente es aplicado.

en los cuales a τ_1 y τ_2 les correspondería tomar valores del control regular $\tilde{u}(t)$ por fuera del dominio permitido $\mathbb{U} = [u_{\min}, u_{\max}]$. Todas las demás situaciones son manejadas adecuadamente, esto es: τ_1 y τ_2 son respetados en cuanto sea posible, pero los $\tau_i^{sat,j}$ son usados cuando los valores de $\tilde{u}(t)$ caen en el complemento de \mathbb{U} .

Se realizaron dos simulaciones para estudiar el comportamiento del parámetro γ del método del gradiente: en primer lugar, el valor del parámetro γ fue mantenido constante, $\gamma = 0,005$ (línea negra a rayas en la figura 3.4), y en la segunda simulación se fue actualizando cuando resultó necesario que \mathcal{J} siguiera decreciendo. Es claro que al principio del método γ puede tener un valor grande, siempre y cuando el costo siga disminuyendo. Sin embargo, en alguna iteración este valor grande de γ genera grandes gradientes y por lo tanto podría ‘saltar’ el valor óptimo y caer sobre un eventual mínimo local, interrumpiendo la minimización del costo. En este momento γ tiene que ser reducido. Para este caso de estudio, el primer valor fue puesto en $\gamma = 0,01$, y el último valor alcanzado fue de $\gamma = 0,005$ (línea azul solida en la figura 3.4). De esta manera, cambiando el parámetro γ , se obtiene un mejor desempeño del método que manteniendo el parámetro constante. En las figuras 3.5 y 3.6 se muestra la evolución de τ_1, τ_2 ; y de ρ, μ , respectivamente. Lue-

3. TRATAMIENTO LINEAL

go de 700 iteraciones los parámetros convergen a sus valores óptimos. Los valores finales fueron : $\tau_1 = 0,5753$, $\tau_2 = 0,7764$, y $\tilde{S}_{11} = 456,613$, $\tilde{S}_{21} = -134,475$, $\tilde{S}_{22} = 39,6015$, calculados utilizando la ecuación (3.9) y los valores de ρ, μ que se muestran en la figura 3.6.

Finalmente, se realizó una comparación del tiempo de cómputo necesario entre: (i) el viejo método basado en la actualización de la matriz \tilde{S} , descrito en [45], y (ii) el nuevo método propuesto, basado en la actualización de ρ y μ . El tiempo necesario para actualizar los coeficientes de la matriz \tilde{S} fue 2,4480 segundos por iteración, y la actualización de ρ, μ tomo 3,1822 por iteración, los tiempos de cómputo son aproximadamente del mismo orden para los dos métodos, lo cual no es sorprendente ya que el sistema es de dimensión 2. En este caso la actualización de ρ, μ fue alrededor de un 30% mas lento que actualizar \tilde{S} . Sin embargo, se debe destacar que en este caso de estudio el método que actualiza ρ y μ llega a la solución óptima, lo que no se consigue con el método de S, x_0 . En este caso de estudio, trabajar con las variables ρ y μ no constituye mejora de tiempo, ya que el sistema es de dimensiones bajas. Las variables actualizadas en el método S, x_0 fueron 3, mientras que en el método con ρ, μ son 4. Para aplicaciones mas complejas como las introducidas en la siguiente subsección, es posible observar mejoras del método.

3.4.2 Una típica situación de un modelo linealizado: un proceso de laminación.

El segundo caso de estudio modela un proceso de laminado como se describe en [45, 73], cuya dinámica (dimensión infinita), viniendo de un balance de energía estándar, obedece a la siguiente PDE de primer orden

$$\frac{\partial \theta}{\partial t} = -V \frac{\partial \theta}{\partial z} + a(\theta_a - \theta) + b(\theta_a^4 - \theta^4), \quad (3.63)$$

donde $\theta(t, z)$ es la temperatura de la banda metálica al tiempo t y localización en la dirección z , $V(t)$ es la velocidad lineal de la lámina, y θ_a es la temperatura ambiente (asumida constante en esta configuración). Los coeficientes a, b pesan la velocidad de calentamiento debido a la conducción y radiación, respectivamente. El sistema es simplificado al despreciar la radiación (para un b pequeño), y

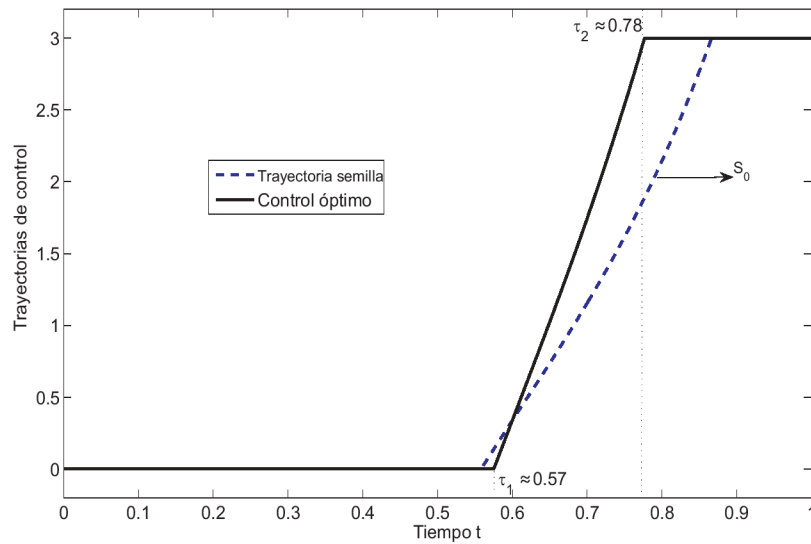


Figura 3.3: Estrategias de control resultantes de la aplicación del método del gradiente al ejemplo del tren cuando se realizan modificaciones en los valores de fase finales, problema bidimensional.

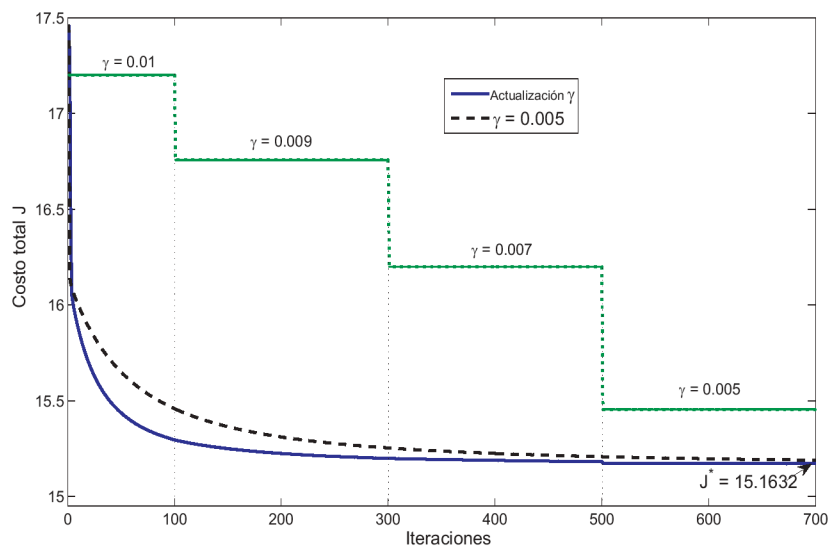


Figura 3.4: Evolución del costo total con modificaciones en los valores de fase finales resultante del método del gradiente, problema bidimensional.

3. TRATAMIENTO LINEAL

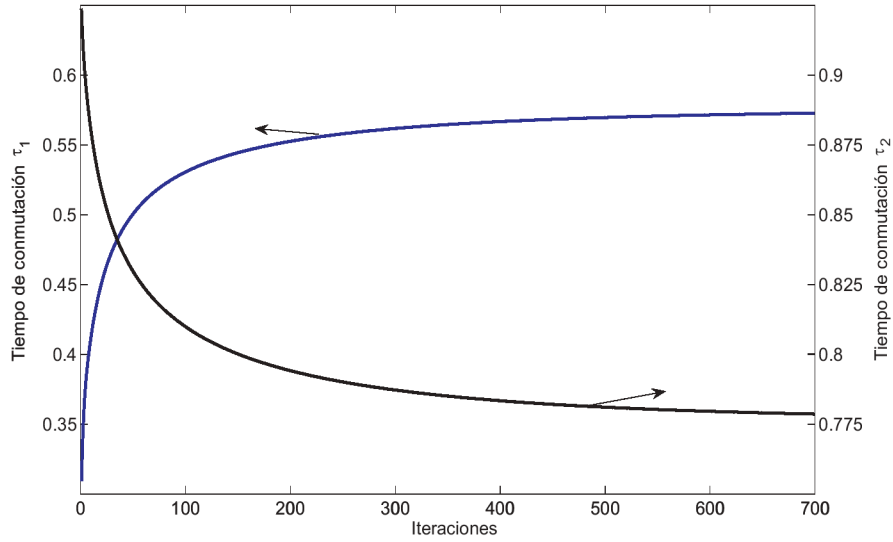


Figura 3.5: Evolución de los τ_s con modificaciones en los valores de fase finales resultante del método del gradiente, problema bidimensional.

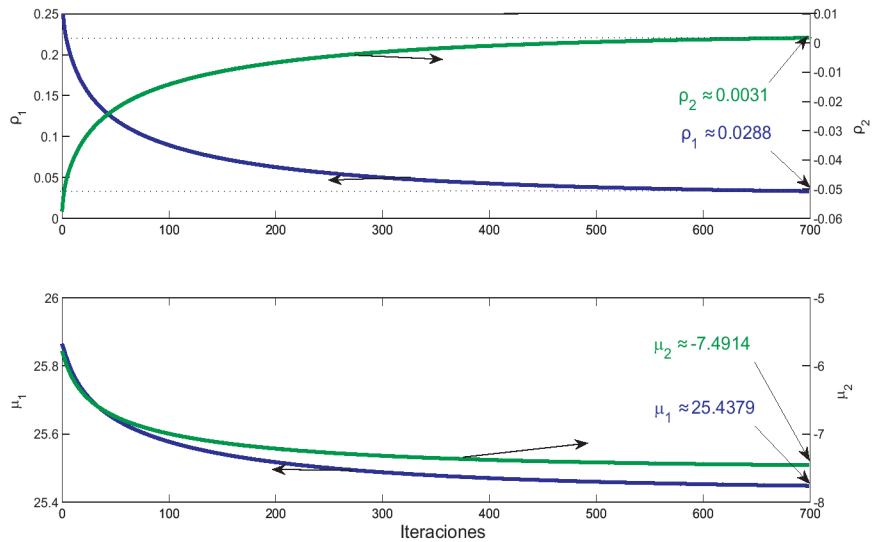


Figura 3.6: Evolución de los ρ_s y μ_s con modificaciones en los valores de fase finales resultante del método del gradiente, problema bidimensional.

suponiendo que la temperatura se mantendrá alrededor de un perfil de equilibrio

$$\theta_{SS}(z) = \theta_a + (\theta_0 - \theta_a) \exp(-az/V_0) , \quad (3.64)$$

el cual es la solución de la ecuación (3.63) con $b = 0$, $\partial\theta/\partial t = 0$, $V(t) \equiv V_0$, y $\theta_0 := \theta_{SS}(0)$ (algunas constantes apropiadas son caracterizadas para cada configuración física). Las siguientes definiciones

$$\Delta\theta(t, z) := \theta(t, z) - \theta_{SS}(z) ; u(t) := V(t) - V_0 , \quad (3.65)$$

permiten expresar de manera aproximada la dinámica de las fluctuaciones a través de una versión ‘linealizada’ de la ecuación (3.63),

$$\frac{\partial\Delta\theta}{\partial t} = -V_0 \frac{\partial\Delta\theta}{\partial z} - a\Delta\theta + \left[\frac{a}{V_0} (\theta_0 - \theta_a) \exp(-az/V_0) \right] u , \quad (3.66)$$

despreciando el termino $u\partial\Delta\theta/\partial z$, al ser el producto de dos ‘pequeñas’ cantidades.

Desde la perspectiva de la teoría de control, el estado x en la ecuación (3.66) puede identificarse, en cada tiempo t , con la totalidad del perfil de temperatura

$$x(t) := \{ \Delta\theta(t, z), z \in [0, L] \} . \quad (3.67)$$

Esto, en principio, hace a la dinámica de dimensión infinita, de la forma

$$\dot{x} = F(x, u) , \quad (3.68)$$

F denota el operador asociado al (RHS) de la ecuación (3.66) (un tratamiento riguroso esta fuera del alcance de este análisis). Una aproximación de n dimensiones ha sido construida al discretizar la variable z en la forma

$$h := L/n ; z_i := (i - 1) h ; i = 1, \dots, n ; \quad (3.69)$$

definiendo ahora n variables de estado x_i (o equivalente a un vector de variables de estados $x(\cdot)$ con valores $x(t)$ en \mathbb{R}^n),

$$x_i(t) := \Delta\theta(t, z_i) ; i = 1, \dots, n ; \quad (3.70)$$

$$x(t) := (x_1(t), x_2(t), \dots, x_n(t))' ; \quad (3.71)$$

3. TRATAMIENTO LINEAL

y finalmente aproximando la derivada parcial z por alguna combinación lineal apropiada de la función $\Delta\theta(t, \cdot)$ evaluada en los valores discretizados z_i , por ejemplo,

$$\frac{\partial\Delta\theta}{\partial z}(t, z_i) \approx \frac{x_{i+1}(t) - x_i(t)}{h}; \quad i = 1, \dots, n-1; \quad (3.72)$$

$$\frac{\partial\Delta\theta}{\partial z}(t, z_n) \approx \frac{x_n(t) - x_{n-1}(t)}{h}. \quad (3.73)$$

Después de ciertas manipulaciones la siguiente estructura de un sistema de control lineal es obtenida

$$\dot{x} = Ax + Bu, \quad (3.74)$$

donde la matriz A de $n \times n$ y el vector columna B de n , toman la forma

$$A = (a_{ij}) : \begin{cases} a_{ii} = V_0/h - a; & a_{i,i+1} = -V_0/h; & i = 1, \dots, n-1 \\ a_{n,n-1} = V_0/h; & a_{nn} = -(a + V_0/h) \\ \text{todos los elementos restantes iguales a } 0 \end{cases}, \quad (3.75)$$

$$B = (b_i) = \left[\frac{a}{V_0}(\theta_0 - \theta_a) \exp(-az_i/V_0); \quad i = 1, \dots, n \right]. \quad (3.76)$$

Los autovalores de la matriz A son dominados por la relación entre el calor ganado en cada posición por convección contra el calor extraído en ese punto por el ambiente, implícita en el término $V_0/h - a$, el cual aparece en la diagonal principal, excepto en su último elemento. Ahora, de un lado, la evolución libre tiene que ser estable para mantener cualquier significado físico en las ecuaciones (la temperatura no puede crecer para siempre). Pero, si el control tiene que ser adecuado para aumentar la estabilidad, es apropiado explorar esas situaciones cerca de donde el sistema podría perder la estabilidad (por ejemplo, debido a las perturbaciones ambientales). Con estos objetivos contradictorios en mente, se investigaron los siguientes valores para los parámetros

$$V_0 = h = 1; \quad a = 1,001. \quad (3.77)$$

La versión discretizada de la ODE (3.74) de la ecuación (3.66) confirmó ser numéricamente una aproximación aceptable [45]. Los estados iniciales $x_0 = x(0)$ utilizados para la simulación del sistema definido por las ecuaciones (3.74 - 3.76) fueron

$$x_i(0) = 100; \quad i = 1, \dots, n, \quad (3.78)$$

con los siguientes valores para las temperaturas de referencia (en °C):

$$\theta_a = 20, \theta_0 = 700. \quad (3.79)$$

Al costo del problema LQR se le asignaron los siguientes parámetros

$$t_f = 0,5, Q = 0,05I_n, R = 100, S = 15I_n, \quad (3.80)$$

y $n = 10$. Los límites impuestos a los valores de control se eligieron aquí como

$$[u_{\text{mín}}, u_{\text{máx}}] = [-1,8, -0,8], \quad (3.81)$$

sólo para ilustrar un caso en el que los costos del problema sin restricciones y la primera aproximación, son lo suficientemente diferentes como para justificar buscar mejores aproximaciones y obtener una importante reducción de costos mediante la implementación del método.

La ‘semilla’ utilizada corresponde al control acotado en forma de ‘feedback’, éste tiene una mayor variación con respecto a la tendencia de los controles y presenta un costo superior. Los valores numéricos para dos semillas candidatas son presentados:

Semilla open loop: $\tau_{1,0} = 0,03182, \tau_{2,0} = 0,07705, \mathcal{J}_{\text{sopenloop}} = 1,0951 \times 10^6$.

Semilla closed loop: $\tau_{1,0} = 0,0413, \tau_{2,0} = 0,0865, \mathcal{J}_{\text{sclosedloop}} = 1,1316 \times 10^6$.

Los costos de las semillas candidatas indican que es posible obtener mayores variaciones con la estrategia en forma de ‘feedback’ o de closed loop, esto debido a que el costo obtenido a través de la optimización debe encontrarse entre el valor de acuerdo con la estrategia elegida y el valor del costo para el control sin restricciones, el cual es de $\mathcal{J}_{\text{irrestringido}} = 4,3338 \times 10^5$.

En la figura 3.7 la trayectoria de control semilla es ilustrada (línea punteada), y también el control irrestringido óptimo (línea discontinua) y su evolución hasta alcanzar el control óptimo (línea solida). En este caso de estudio el control óptimo corresponde al valor máximo aplicado en todo el horizonte de tiempo, sobre todo debido a los altos valores asignados a R y S en la formulación original del costo. Después de simular las trayectorias de control y las de los estados, se encontró que $\tau_{1,0} = 0,0413, \tau_{2,0} = 0,0865, \mathcal{J}_{\text{semilla}} = 1,1316 \times 10^6$. El costo óptimo fue obtenido en 25 iteraciones y su valor fue $\mathcal{J}^* = 9,1010 \times 10^5$. La reducción total de costo

3. TRATAMIENTO LINEAL

Tabla 3.1: Valores semilla y finales para los parámetros

<i>Componente</i>	$\rho_{semilla}$	ρ^*	$\mu_{semilla}$	μ^*
1	-23, 4223	24, 5168	-1402, 842	-631, 500
2	31, 7068	-12, 7917	634, 734	940, 896
3	51, 967	-26, 5032	1383, 567	1518, 771
4	59, 4135	-31, 5422	1658, 772	1731, 146
5	62, 1500	-33, 3940	1759, 913	1809, 198
6	63, 1557	-34, 0743	1797, 081	1837, 874
7	63, 5253	-34, 3274	1810, 772	1848, 498
8	63, 6609	-34, 3959	1815, 561	1851, 761
9	63, 7122	-34, 545	1818, 557	1856, 167
10	63, 7254	-34, 3012	1816, 606	1850, 087

relativa con respecto al costo de la trayectoria semilla $\mathcal{J}_{semilla}$ fue alrededor de 20 %.

Los valores semilla y los finales de ρ, μ son mostrados en la Tabla 3.1. La verificación del control óptimo $u^*(t) \equiv -0,8 = u_{m\acute{a}x} \forall t$ se comprobó mediante el cálculo de: (i) La solución de las ecuaciones Hamiltonianas (2.23, 2.24), hacia atrás desde (ρ^*, μ^*) , la cual reproduce las condiciones iniciales para el estado y (ii) El Hamiltoniano (\mathcal{H}) a lo largo de la trayectoria, que resultó $\mathcal{H}(x^*(t), \lambda^*(t), u^*(t)) = 2,2191 \times 10^6$, constante como se esperaba.

El parámetro γ opera como la tolerancia o el método de criterio de parada. El criterio de parada en este caso de estudio se presenta cuando $\gamma > 1e - 15$.

Los valores finales para los tiempos de conmutación son: $\tau_1 = -0, 0198, \tau_2 = -0, 0028$, estos valores negativos comprueban la pérdida del período regular.

Para este problema de moderadamente alta dimensión, se realizó una comparación de tiempo de cómputo: el primer intento mediante la actualización de los 55 coeficientes de la matriz S , y el segundo de ellos mediante la actualización de los 20 coeficientes de ρ, μ . Como se esperaba, el primer experimento tomó 458,95 segundos por iteración mientras el segundo, solo necesitó 151,79 segundos por iteración. Esto significa que el segundo método es 200 % más rápido que el primero.

La actualización de los parámetros utilizando el método del gradiente conju-

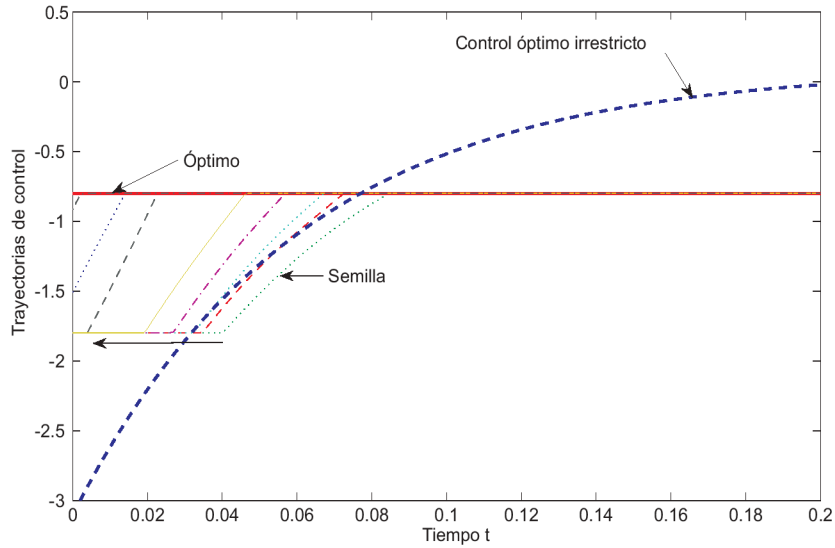


Figura 3.7: Evolución de las trayectorias de control con las iteraciones hasta alcanzar el control óptimo.

gado, en este caso de estudio, produce una pequeña mejora del tiempo total de cómputo requerido respecto al método básico del gradiente. La evolución de las variables $\tau_1, \tau_2, \rho, \mu$ se representan en las figuras 3.9 y 3.10. Todos los componentes de ρ, μ convergen a sus valores óptimos.

3. TRATAMIENTO LINEAL

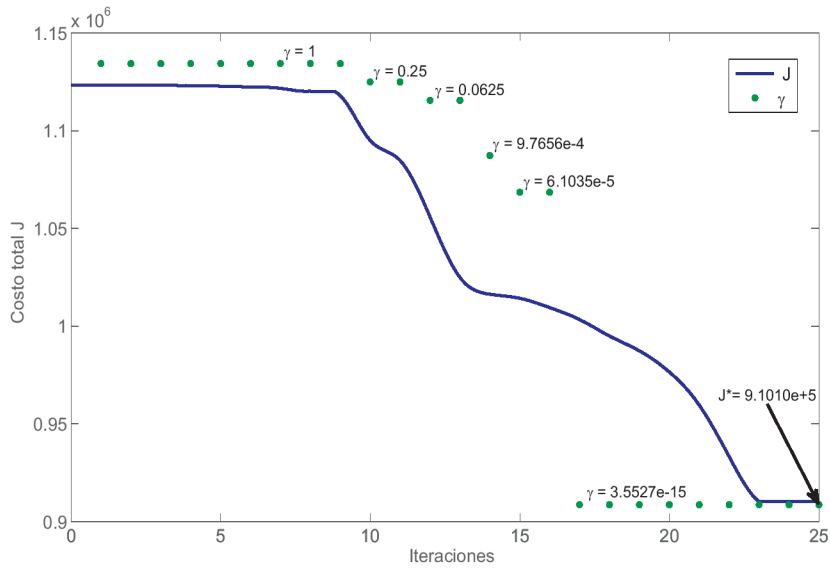


Figura 3.8: Evolución del costo total cuando se aplica el método del gradiente conjugado.

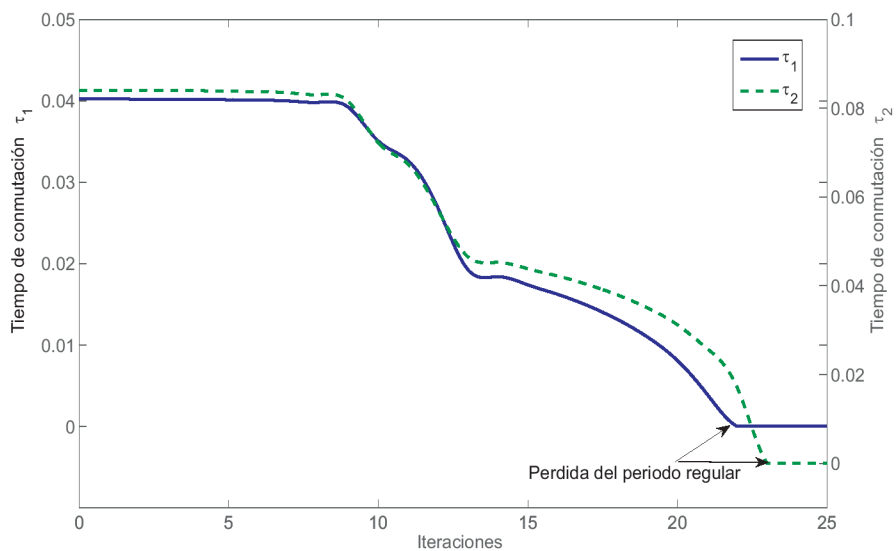


Figura 3.9: Evolución de los τ_s cuando se aplica el método del gradiente conjugado.

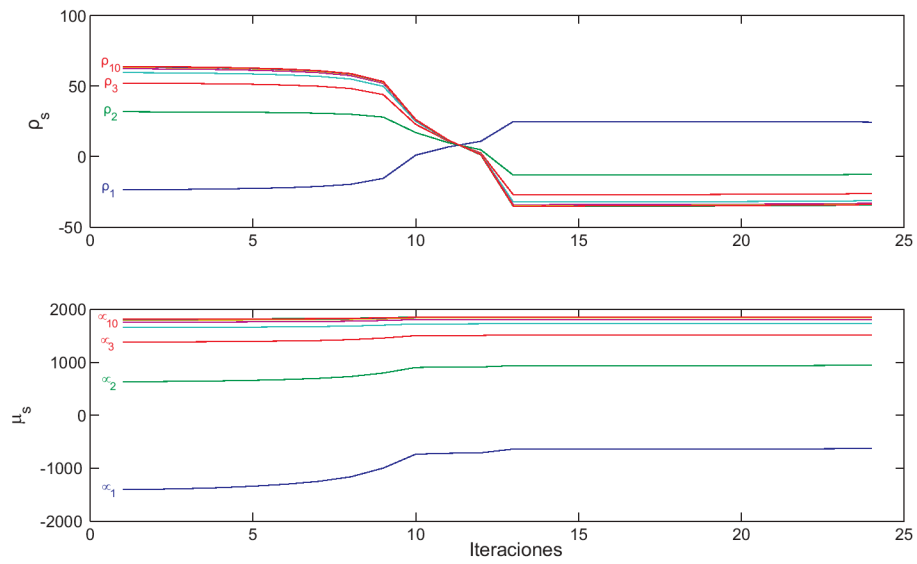


Figura 3.10: Evolución de los ρ y μ cuando se aplica el método del gradiente conjugado.

3. TRATAMIENTO LINEAL

3.5 Varios períodos regulares

En análisis para sistemas en los cuales se ven involucrados varios arcos regulares, como los tratados en esta tesis, buscan encontrar una solución de la RDE (ecuación 2.35), por cada uno de los períodos regulares contenidos, lo cual conduce a realizar una optimización por partes (se realiza como mínimo un tratamiento para cada período regular que tenga el problema), en las que se pretende arribar a soluciones óptimas o sub-óptimas para cada una de las particiones, pero siempre teniendo en cuenta que el objetivo es la minimización del costo total. Sin embargo, puede resultar difícil expresar todos los objetivos o contribuciones de cada uno de los arcos regulares dentro de un simple funcional de costo: el problema puede llegar a entenderse o tratarse como alguna clase de formulación ‘multiobjetivo’. Esta formulación puede ser aplicada en situaciones en donde es imposible satisfacer todos los objetivos simultáneamente sin perjudicar el costo total. La estrategia propuesta modifica los valores finales de los arcos regulares, los cuales a su vez representan los valores iniciales para el arco regular siguiente, es imprescindible entonces encontrar un equilibrio de modo que no se vea reflejado en el costo total. En la programación multiobjetivo es recurrente que se presenten conflictos, y por lo general no existe una solución global única. Algunas técnicas para obtener soluciones eficientes (óptimos de Pareto) para problemas multiobjetivos pueden encontrarse en [24].

3.5.1 Dos períodos regulares

El desarrollo trabajado en esta sección pretende incorporar al ejemplo ‘detención óptima de un tren’ estudiado en la sección 3.4, una modificación de parámetros donde se generen dos arcos regulares, con los cuales surgen un nuevo par de tiempos de conmutación τ_s dentro de todo el horizonte de tiempo del problema.

El problema es afrontado resolviendo los períodos de optimización en forma secuencial, de manera que cuando se esté optimizando el primer período regular se tenga en cuenta la contribución que es generada para el segundo período regular (para problemas con mayor cantidad de períodos regulares es posible utilizar una estrategia de control similar). Se pretende minimizar un funcional de costo de un

problema LQR, ecuación 2.27, con los valores de los parámetros definidos de la siguiente manera: $Q = 250I_n$, $R = 1$ y $S = 100I_n$. Sujeto a las restricciones en la variable manipulada, como se describe en la ecuación 2.42, con $u_{\min} = 0,7$ y $u_{\max} = 2$. El problema de control óptimo no puede encontrarse con las soluciones tradicionales, y hay que recurrir a métodos numéricos para encontrar o acercarse al nuevo óptimo.

3.5.1.1 Semilla para Dos Períodos regulares

Se calcula primero una semilla con la cual se partirá para la confección de la optimización, ésta se puede apreciar en la figura 3.11. La gráfica muestra los puntos donde se presentan los tiempos de conmutación candidatos, correspondientes a los diferentes τ_s , delimitando dos períodos regulares. Si el sistema está sujeto a una dinámica lineal como en la ecuación 2.28, y siendo el control óptimo en el caso regular de acuerdo a la ecuación 2.32, al encontrarse con restricciones en las variables de entrada (saturación en los controles) el problema deja de ser óptimo y no se verifica el principio del mínimo de Pontryagin.

El control llamado ‘semilla’ para el problema es construido en este caso, en la forma

$$\tilde{u}^{sat}(t) = \begin{cases} u_{\min} & \text{si } -R^{-1}B'P(t)x < u_{\min} \\ u^*(t) & \text{si } u_{\min} \leq -R^{-1}B'P(t)x \leq u_{\max} \\ u_{\max} & \text{si } -R^{-1}B'P(t)x > u_{\max} \end{cases} \quad (3.82)$$

A partir de la semilla se utilizan métodos que permitan encontrar el mejor control posible ante la presencia de estas restricciones, verificándose su optimalidad posteriormente a través del principio de Pontryagin.

La semilla tiene un costo total asociado de

$$\mathcal{J}_{semilla} = 224,3048844 , \quad (3.83)$$

correspondiente a todo el horizonte de tiempo; esto es, la suma de las soluciones desde t_0 hasta τ_2 , y desde τ_2 hasta t_f . De la figura 3.11 se pueden identificar los tiempos de conmutación $\tau_1 = 0,045$, $\tau_2 = 0,426$, $\tau_3 = 0,892$, $\tau_4 = 0,968$, indispensables para realizar los cálculos de la optimización por partes, ya que se toman

3. TRATAMIENTO LINEAL

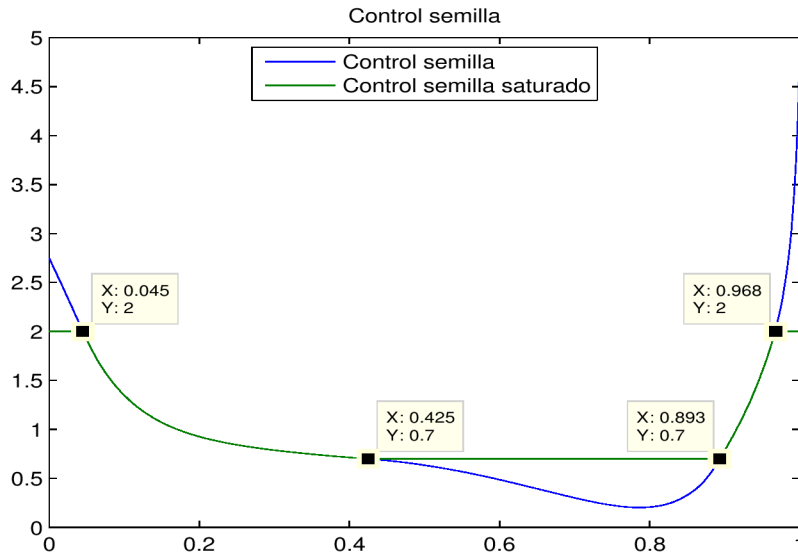


Figura 3.11: Control semilla

Tabla 3.2: Costo para el problema regular inicial

Intervalo	Costo
$[0, \tau_2)$	132,6455
$[\tau_2, t_f)$	66,2853
t_f	25,3699

éstos como semilla para indicar los puntos donde se empieza a realizar la optimización. Es importante tener presente que el valor de la S cambiará en los diferentes períodos regulares, además de una optimización adicional realizada en el último período de saturación, elaborándose de una forma diferente para buscar garantizar que se mantenga $\lambda = 2Sx(t_f)$. En la Tabla 3.2 muestra el costo asociado a cada intervalo.

3.5.1.2 Optimización por medio del proceso “oculto”, “relacionado” o “fantasma”

Se pretende buscar matrices \tilde{S} diferentes para cada uno de los períodos regulares, mas una adicional correspondiente al último período de saturación por medio de la

ecuación 3.9. Este método es válido tanto para sistemas lineales como no lineales. Con métodos como el propuesto, en problemas no lineales se obtiene el control del proceso fantasma como una función que depende del coestado λ (ecuación 2.22), luego éste se satura y se aplica en el proceso real. De esta manera no se tienen en cuenta las perturbaciones que se puedan presentar en los estados reales.

Para iniciar la optimización se resuelve la ecuación diferencial de Riccati hacia atrás, desde el tiempo horizonte t_f , y se almacena para su uso posterior (esta solución fue requerida también para calcular la solución del semilla, ya que se hace con los valores nominales).

Para la actualización de los valores de fase finales, así como para los tiempos de conmutación encontrados en la optimización del modelo relacionado, se implementan las ecuaciones 3.55 y 3.56 de manera similar, teniendo en cuenta que se realiza para cada par de tiempos de conmutación τ_s que se ven involucrados por cada arco de período regular, obtenidos de la implementación del control en el modelo real.

3.5.1.2.1 Optimización desde t_0 hasta τ_2

Se resuelve la dinámica hasta τ_2 y se obtiene el valor de los estados en τ_2 .

$$x_{semilla}(\tau_2) = \begin{pmatrix} 0,6937 \\ -0,5330 \end{pmatrix}, \quad (3.84)$$

siendo posible calcular un valor semilla $\lambda_{semilla}(\tau_2)$

$$\lambda_{semilla}(\tau_2) = 2 * P(\tau_2) * x_{semilla}(\tau_2) = \begin{pmatrix} 265,6639 \\ -1,2995 \end{pmatrix}. \quad (3.85)$$

Los valores de fase finales en esta etapa son definidos como $\rho_{\tau_2} = x_{semilla}(\tau_2)$ y $\mu_{\tau_2} = \lambda_{semilla}(\tau_2)$, modificados en las diferentes iteraciones a través de pequeñas variaciones y del método del gradiente aplicado al valor del tiempo de conmutación τ_2 .

Estos valores en τ_2 se someten a perturbaciones por medio de las ecuaciones:

$$\begin{aligned} \rho_{\tau_2(j+1)} &= \rho_{\tau_2(j)} \pm h_{\tau_2}, \\ \mu_{\tau_2(j+1)} &= \mu_{\tau_2(j)} \pm k_{\tau_2}, \quad j = 1, 2, \dots, n. \\ \tau_2(j+1) &= \tau_2(j) \pm g_{\tau_2}, \end{aligned} \quad (3.86)$$

con $h_{\tau_2} = 0,001$, $k_{\tau_2} = g_{\tau_2} = 0,01$.

3. TRATAMIENTO LINEAL

Tabla 3.3: Valores utilizados en la primera optimización

Parámetro	Valor
n	100
h_{τ_2}	0,001
k_{τ_2}	0,01
g_{τ_2}	0,01
γ	$1e^{-2}$
γ_{τ_2}	$1e^{-3}$

A partir de estas variaciones se calcula una \tilde{S} correspondiente al problema relacionado o fantasma, para este caso

$$\tilde{S}(\tau_2) = \frac{1}{2} * \frac{\mu_{\tau_2} * \mu'_{\tau_2}}{\rho'_{\tau_2} * \mu_{\tau_2}}, \quad (3.87)$$

con ρ_{τ_2} y μ_{τ_2} como los nuevos estados y coestados en τ_2 respectivamente .

Con los nuevos valores de la matriz de penalización en τ_2 se resuelve la RDE hacia atrás, y se obtienen nuevos valores de P en el intervalo de $[t_0, \tau_2)$, denominada, en la primera optimización para diferenciarla de la solución semilla \tilde{P}_1 . La solución es almacenada para ser utilizada en el cálculo de la trayectoria de control del problema real (en este momento hasta τ_2 , y el semilla en el resto del tiempo horizonte). El control saturado en este intervalo es dado por la ecuación

$$\tilde{u}_1^{sat}(t) = (-R^{-1}B'\tilde{P}_1(t)x(t))^{sat}, \quad \forall t \in [t_0, \tau_2). \quad (3.88)$$

La trayectoria de control obtenida permite calcular el costo que se genera desde t_0 hasta τ_2 . Para evitar que se presenten modificaciones en $x(\tau_2)$ que provoquen que en la optimización del costo total no se logre una disminución con respecto al costo semilla (ecuación 3.83), se calcula el valor del costo para el problema completo, por lo que es necesario hallar la contribución del costo desde τ_2 hasta t_f . Como aún no se ha optimizado ese intervalo, el costo se obtiene resolviendo la ecuación de estados con los valores de la ecuación de Riccati semilla $P(t)$, sumando además la contribución debida la penalización final. En la Tabla 3.3 se muestran algunos valores utilizados para la optimización del primer período regular.

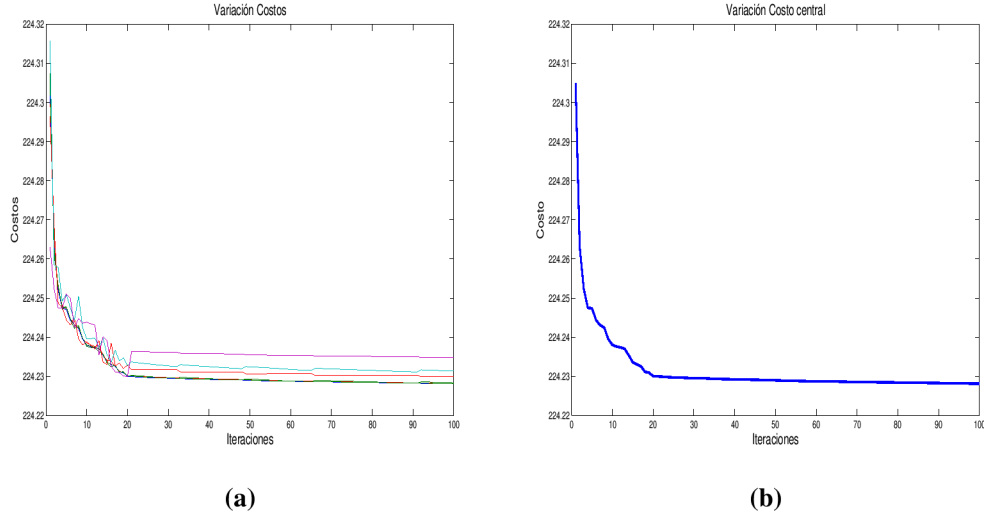


Figura 3.12: Optimización del primer período regular. (a) Perturbaciones del costo, (b) gradiente.

En la figura 3.12 se muestra la reducción del costo total para n iteraciones realizadas al optimizar el primer intervalo que contiene un período regular, en la parte 3.12a todos los costos correspondientes a las perturbaciones generadas con la ecuación 3.86 y en la parte 3.12b las correspondientes a emplear el método del gradiente con la ecuación 3.55.

La figura 3.13 muestra una comparación entre el control semilla y el iterado, en la parte 3.13a se muestran los controles sin restricciones, y en la parte 3.13b cuando se aplica la restricción en la variable manipulada. El control iterado sin restricciones está determinado en esta etapa por la siguiente fórmula

$$\tilde{u}_{ite_1}(t) = \begin{cases} -R^{-1}B'\tilde{P}_1(t)x(t) & \text{si } t \leq \tau_2 \\ -R^{-1}B'P(t)x(t) & \text{si } \tau_2 < t < t_f \end{cases}, \quad (3.89)$$

en 3.13b se muestra el control semilla y el iterado para el problema con restricciones, con la siguiente estructura

$$\tilde{u}_{ite_1}^{sat}(t) = \begin{cases} u_{\text{mín}} & \text{si } -R^{-1}B'\tilde{P}_1(t)x \text{ ó } -R^{-1}B'P(t)x < u_{\text{mín}} \\ \tilde{u}_{ite_1}(t) & \text{si } u_{\text{mín}} \leq -R^{-1}B'\tilde{P}_1(t)x \leq u_{\text{máx}} \\ u_{\text{máx}} & \text{si } -R^{-1}B'\tilde{P}_1(t)x \text{ ó } -R^{-1}B'P(t)x > u_{\text{máx}} \end{cases}. \quad (3.90)$$

3. TRATAMIENTO LINEAL

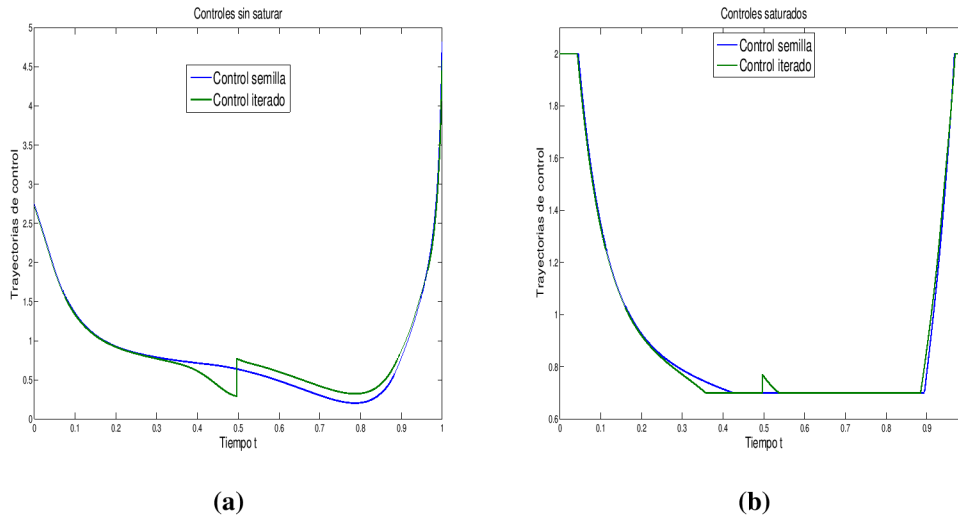


Figura 3.13: Comparación de controles en el primer período regular optimizado. (a) Sin restricciones, (b) con restricciones.

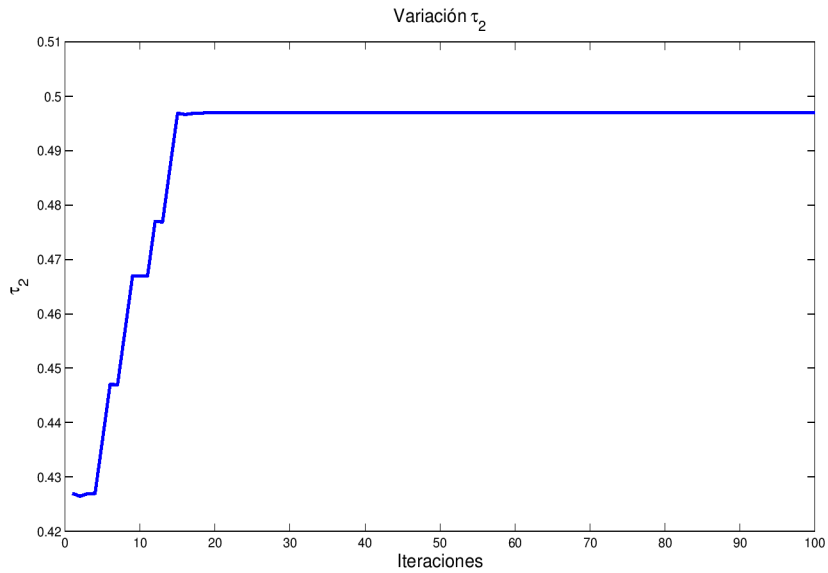


Figura 3.14: Gradiente de τ_2

En la figura 3.14 se muestran las variaciones en τ_2 durante las n iteraciones. La figura 3.15 muestra las variaciones para cada iteración de \tilde{S} a través de la actualización de los valores de fase en τ_2 para el proceso relacionado. De la figura 3.13

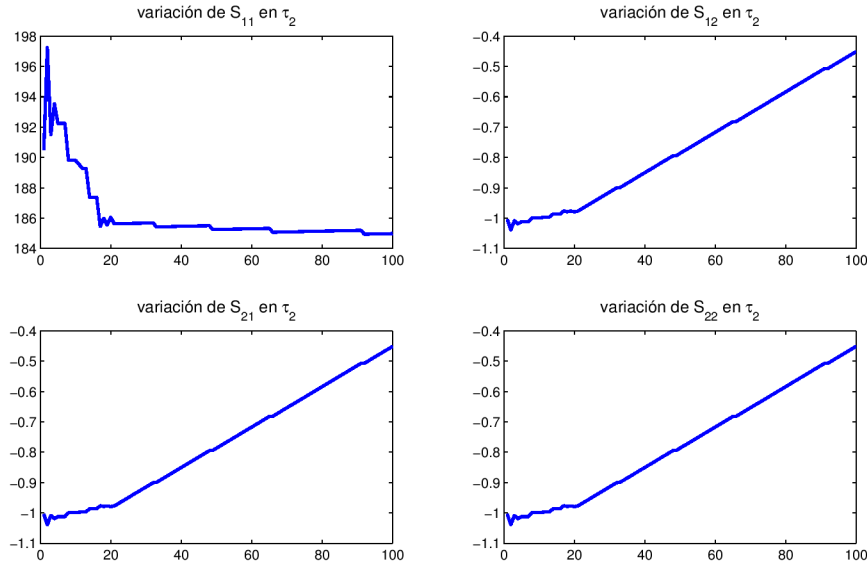


Figura 3.15: Variación de \tilde{S} con las iteraciones

el tiempo donde se presenta un pico o discontinuidad en la trayectoria de control señala hasta dónde se resuelve con el modelo relacionado para la primera etapa de la optimización, correspondiente al valor τ_2 obtenido a través del método del gradiente. El punto de discontinuidad ocurre porque existe un cambio en la solución de Riccati utilizada, pasando de la solución por medio de \tilde{P}_1 en el intervalo $[0, \tau_2]$ a P en el intervalo $[\tau_2, t_f]$, lo que suscita un cambio en la trayectoria de control. Para ilustración se resuelve la ecuación de estados completa y se muestran algunos resultados obtenidos en la Tabla 3.4.

3.5.1.2.2 Optimización desde τ_2 hasta τ_4

Se resuelve la dinámica hasta τ_4 y se obtiene el valor de los estados en τ_4 .

$$x_{semilla}(\tau_4) = \begin{pmatrix} 0,5048 \\ -0,1105 \end{pmatrix}, \quad (3.91)$$

3. TRATAMIENTO LINEAL

Tabla 3.4: Valores obtenidos con la primera optimización

Variable	Valor
τ_2	0,4970
$x_1(\tau_2)$	0,6565
$x_2(\tau_2)$	-0,4909
$\rho_1(\tau_2)$	0,7159
$\rho_2(\tau_2)$	-0,5330
$x_1(t_f)$	0,5019
$x_2(t_f)$	-0,0483
\mathcal{J}_{op1}	224,2832

con el cual se obtiene un valor semilla del coestado $\lambda_{semilla}(\tau_4)$

$$\lambda_{semilla}(\tau_4) = 2 * P(\tau_4) * x_{semilla}(\tau_4) = \begin{pmatrix} 108,3046 \\ -4,15 \end{pmatrix}. \quad (3.92)$$

Los valores de fase finales de esta etapa son definidos de manera similar al presentado en la etapa previa, en este caso con $\rho_{\tau_4} = x_{semilla}(\tau_4)$ y $\mu_{\tau_4} = \lambda_{semilla}(\tau_4)$, modificados junto a el valor del tiempo de conmutación τ_4 .

Los valores de fase final en τ_4 y el tiempo de conmutación τ_4 se someten a perturbaciones por medio de las ecuaciones:

$$\begin{aligned} \rho_{\tau_4(j+1)} &= \rho_{\tau_4(j)} \pm h_{\tau_4}, \\ \mu_{\tau_4(j+1)} &= \mu_{\tau_4(j)} \pm k_{\tau_4}, \quad j = 1, 2, \dots, n. \\ \tau_{4(j+1)} &= \tau_{4(j)} \pm g_{\tau_4}, \end{aligned} \quad (3.93)$$

con $h_{\tau_4} = 0,001$, $k_{\tau_4} = g_{\tau_4} = 0,01$.

Con estas variaciones se calcula una \tilde{S} correspondiente al problema relacionado para esta segunda etapa de optimización, en este caso $\tilde{S}(\tau_4)$ es

$$\tilde{S}(\tau_4) = \frac{1}{2} * \frac{\mu_{\tau_4} * \mu'_{\tau_4}}{\rho'_{\tau_4} * \mu_{\tau_4}}, \quad (3.94)$$

con ρ_{τ_4} y μ_{τ_4} como los nuevos estados y coestados en τ_4 respectivamente .

Para los nuevos valores de la matriz de penalización final $\tilde{S}(\tau_4)$ se resuelve la RDE hacia atrás hasta τ_2 , obteniendo nuevos valores de la P en el intervalo de $[\tau_2, \tau_4)$, denominada de forma consecuente a la primera optimización como \tilde{P}_2 . La

Tabla 3.5: Valores utilizados en el segundo intervalo de optimización

Parámetro	Valor
n	30
h_{τ_4}	0,001
k_{τ_4}	0,01
g_{τ_4}	0,01
γ	$1e^{-2}$
γ_{τ_4}	$1e^{-3}$

solución se almacena y luego se obtiene la trayectoria de control del problema real (hasta τ_4 , mas el control obtenido previamente a partir de la primera optimización y el uso del semilla para el resto del tiempo horizonte). El control saturado en este intervalo de tiempo analizado es dado según la ecuación

$$\tilde{u}_2^{sat}(t) = (-R^{-1}B'\tilde{P}_2(t)x(t))^{sat}, \quad \forall t \in [\tau_2, \tau_4]. \quad (3.95)$$

Con la trayectoria de control calculada es posible obtener el costo que se genera desde τ_2 hasta τ_4 . En esta etapa de optimización, al igual que en la primera, se tiene en cuenta el valor del costo total desde t_0 hasta t_f , por lo que es necesario además hallar la contribución del costo desde τ_4 hasta t_f . En este punto se suma el costo de las dos etapas optimizadas y la de intervalos o etapas posteriores que son encontradas con el valor semilla de $P(t)$, sumando además la contribución debida a la penalización final con los nuevos estados finales que se van obteniendo. En la Tabla 3.5 se muestran algunos de los valores utilizados en la optimización de la segunda etapa.

Para esta segunda etapa de optimización se muestran los gráficos considerando solamente desde τ_2 hasta t_f . La figura 3.16 muestra la disminución total del costo cuando se optimizó el intervalo que contenía al segundo arco regular, esto teniendo en cuenta las contribuciones que se generan en las etapas posteriores correspondientes. En 3.16a se aprecian los costos debidos a cada perturbación y en 3.16b el costo debido al método del gradiente con las ecuaciones 3.93 y 3.55 respectivamente. La figura 3.17 muestra una comparación entre los controles semilla y el iterado obtenido con el gradiente, en 3.17a sin saturar el control, y en 3.17b con el

3. TRATAMIENTO LINEAL

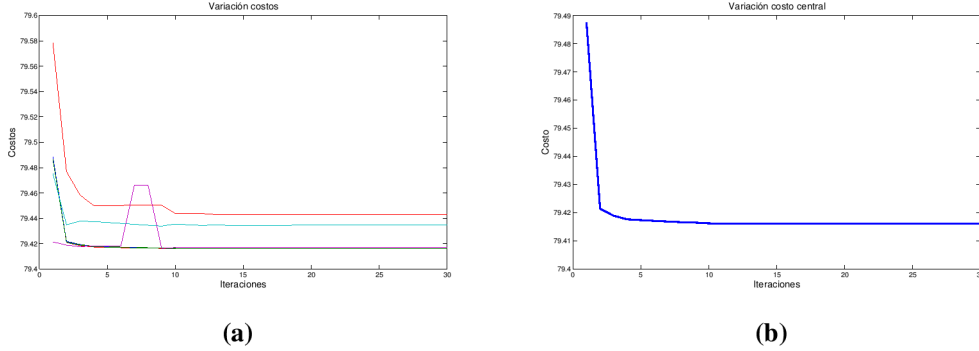


Figura 3.16: Gradientes del costo, optimización del segundo período regular

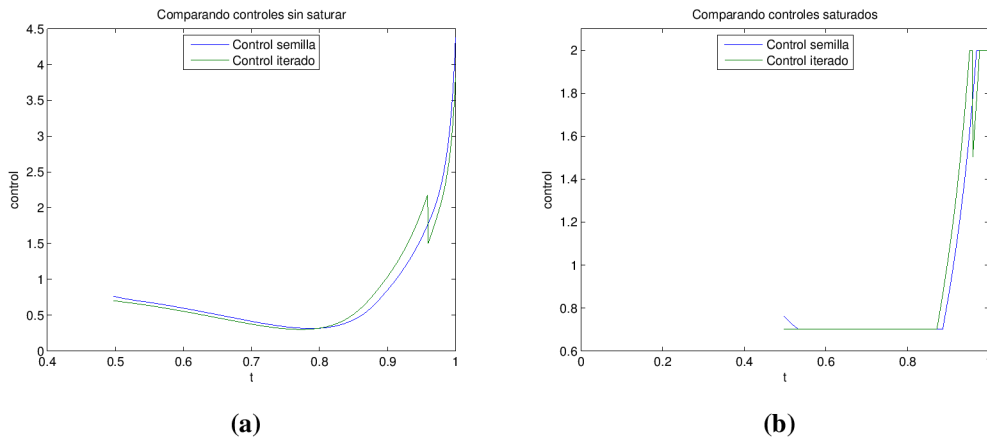


Figura 3.17: Comparación de controles, optimización segundo período regular

control saturado. El control 3.17a iterado sin restricciones para esta segunda etapa esta determinado por

$$\tilde{u}_{ite_2}(t) = \begin{cases} -R^{-1}B'\tilde{P}_2(t)x(t) & \text{si } t \leq \tau_4 \\ -R^{-1}B'P(t)x(t) & \text{si } \tau_4 < t < t_f \end{cases} \quad (3.96)$$

En 3.17b se muestra el control iterado para el problema con restricciones, con la siguiente estructura

$$\tilde{u}_{ite_2}^{sat}(t) = \begin{cases} u_{\min} & \text{si } -R^{-1}B'\tilde{P}_2(t)x \text{ ó } -R^{-1}B'P(t)x < u_{\min} \\ \tilde{u}_{ite_2}(t) & \text{si } u_{\min} \leq -R^{-1}B'\tilde{P}_2(t)x \leq u_{\max} \\ u_{\max} & \text{si } -R^{-1}B'\tilde{P}_2(t)x \text{ ó } -R^{-1}B'P(t)x > u_{\max} \end{cases} \quad (3.97)$$

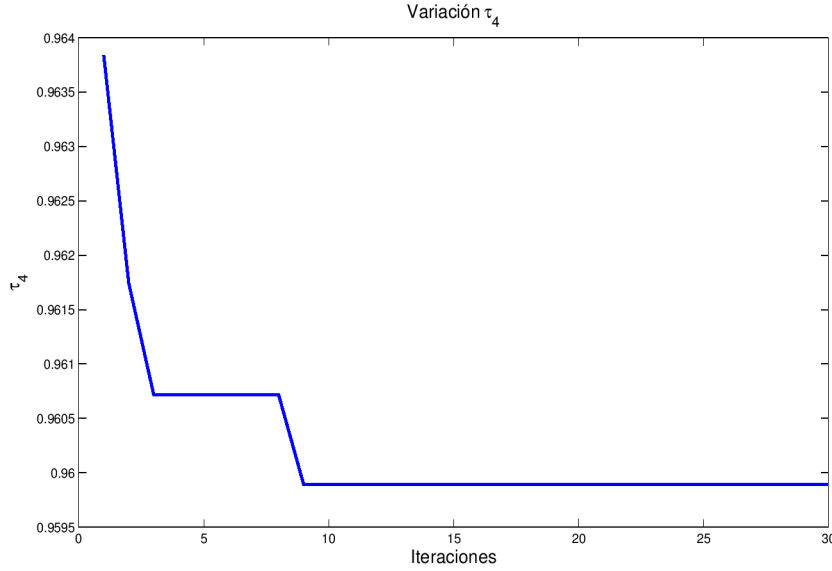


Figura 3.18: Gradiente de τ_4

La figura 3.18 muestra las variaciones en τ_4 durante las n iteraciones y la figura 3.19 muestra las variaciones para cada iteración de \tilde{S} (ecuación 3.94) a través de la actualización de los valores de fase en τ_4 para el proceso relacionado.

La figura 3.17 muestra el control iterado para la optimización en el segundo período regular. Como en el caso de optimización del primer período regular se presenta un pico en el valor de τ_4 que arroja el método del gradiente, es allí hasta donde se resuelve con el modelo real con la técnica planteada, esto ocurre porque al igual que en la primera etapa de optimización, existe un punto donde hay un cambio en la solución de Riccati utilizada, en este caso pasa de \tilde{P}_2 en $[\tau_2, \tau_4]$ a P en $[\tau_4, t_f]$, provocando el correspondiente cambio en el control. Después de esta nueva optimización realizada, se observa que el pico que sobrepasaba el valor de saturación en la primera optimización desciende por debajo de la cota mínima u_{\min} , quedando sólo el pico al final del período de optimización actual τ_4 . Para observar los nuevos valores que se alcanzan en t_f se resuelve la ecuación de estados completa, los resultados se ilustran en la Tabla 3.6.

3. TRATAMIENTO LINEAL

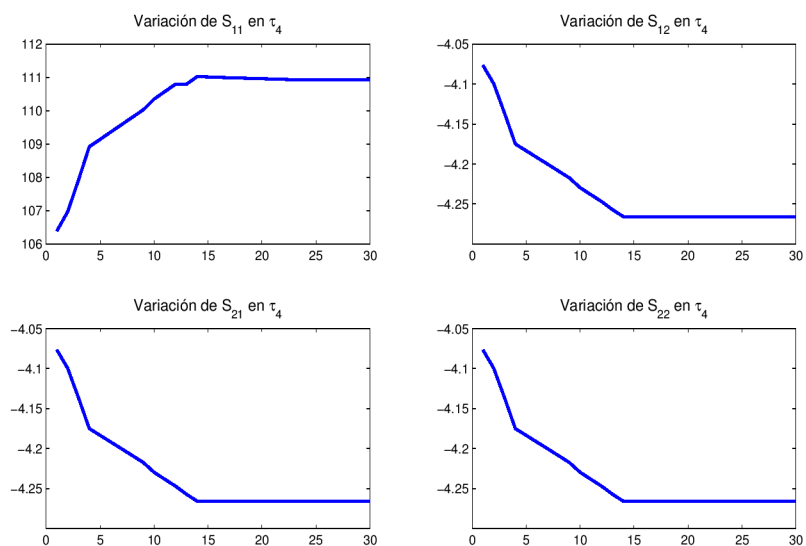


Figura 3.19: Variación de S con las iteraciones

Tabla 3.6: Valores obtenidos de la segunda optimización

Variable	Valor
τ_4	0,9599
$x_1(\tau_4)$	0,5060
$x_2(\tau_4)$	-0,11368
$\rho_1(\tau_4)$	0,4835
$\rho_2(\tau_4)$	-0,1089
$x_1(t_f)$	0,5024
$x_2(t_f)$	-0,0323
\mathcal{J}_{op2}	224,2515791

3.5.1.2.3 Optimización desde τ_4 hasta t_f

Para la última etapa de actualización del problema completo (adicional, para eliminar los picos presentados en la actualización previa), se hacen sólo variaciones de $x(t_f)$, siendo $\rho(t_f) = x(t_f)$. Las semillas son actualizadas por medio de $\rho(t_f)$ para las diferentes iteraciones que utilizan el método del gradiente, los valores iniciales son los obtenidos después de resolver la segunda etapa de optimización en t_f .

Los valores en t_f se someten a perturbaciones por medio de las ecuaciones:

$$\begin{aligned} \rho_{t_f(j+1)} &= \rho_{t_f(j)} \pm h_{t_f}, \\ \mu_{t_f(j+1)} &= 2S\rho_{t_f(j+1)}, \quad j = 1, 2, \dots, n. \end{aligned} \quad (3.98)$$

con $h_{t_f} = 0,07$. Las variaciones permiten calcular una \tilde{S} correspondiente al problema fantasma, para este caso $\tilde{S}(t_f)$ de la siguiente manera

$$\tilde{S}(t_f) = \frac{1}{2} * \frac{\mu_{t_f} * \mu'_{t_f}}{\rho'_{t_f} * \mu_{t_f}}, \quad (3.99)$$

con ρ_{t_f} y μ_{t_f} como los nuevos estados y coestados en t_f respectivamente, concierne a un nuevo proceso relacionado a esta etapa.

con $\tilde{S}(t_f)$ se resuelve la Riccati hacia atrás hasta τ_4 , con esto se obtiene un nuevo valor para la P en el intervalo de $[\tau_4, t_f]$, denominada según la etapa de optimización correspondiente, en este caso como \tilde{P}_3 . El valor de la solución de la RDE es guardado para calcular la trayectoria de control del problema real, calculado como

$$\tilde{u}_3^{sat}(t) = (-R^{-1}B'\tilde{P}_3(t)x(t))^{sat}, \quad \forall t \in [\tau_4, t_f]. \quad (3.100)$$

Se tiene en cuenta el valor del costo total desde t_0 hasta t_f . Al costo de la tercera etapa de optimización se suman los costos de las dos etapas de optimización previas, además de la contribución debido la penalización final. En la Tabla 3.7 se muestran algunos de los valores utilizados en la optimización de esta etapa.

En esta tercera etapa de optimización se muestran los gráficos desde τ_2 hasta t_f . La figura 3.20 muestra la disminución total del costo al optimizar el último intervalo, aunque dicho intervalo no contenga un período regular.

La figura 3.21 muestra la optimización de la tercera etapa. El pico que se presentaba durante la optimización de la segunda etapa en τ_4 ya no aparece en el control, esto debido a que se le agregó una optimización para la tercera etapa; en

3. TRATAMIENTO LINEAL

Tabla 3.7: Valores usados en el tercer intervalo de optimización

Variable	Valor
n	40
h_{t_f}	0,07
γ	$3e^{-3}$

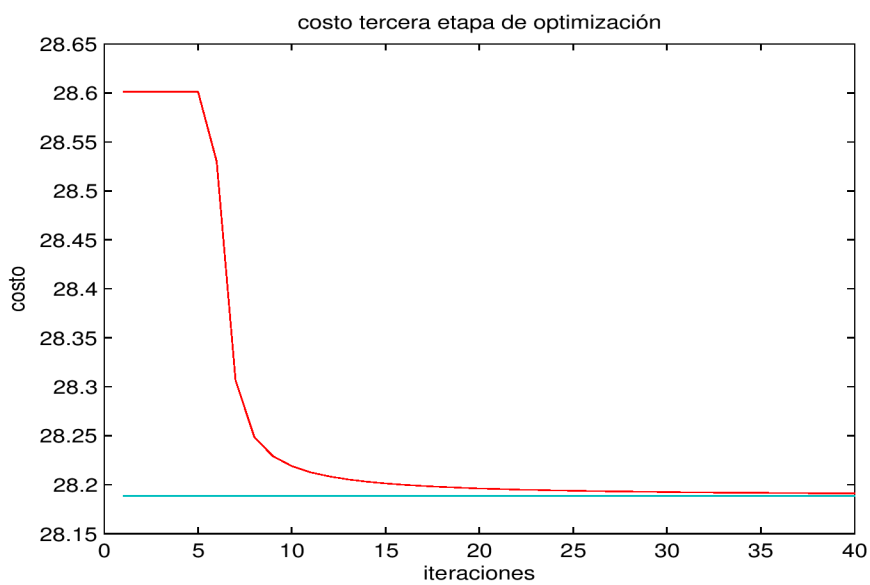


Figura 3.20: Gradientes del costo en la tercera etapa de optimización

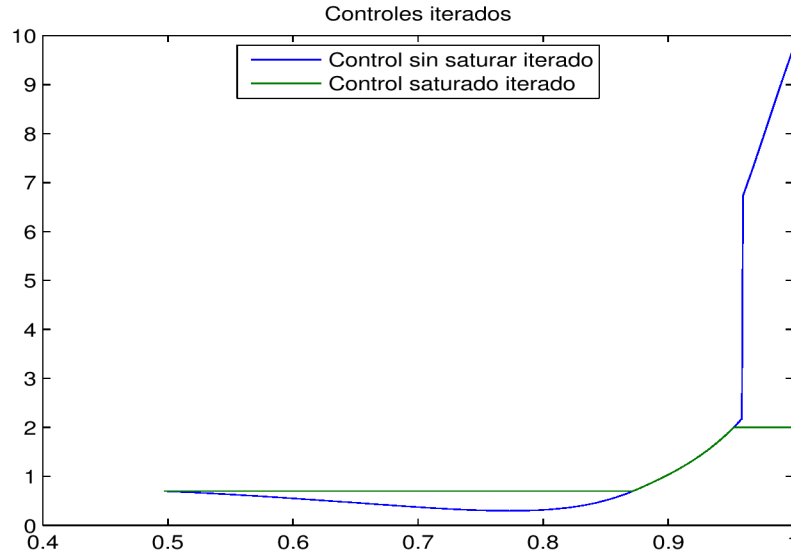


Figura 3.21: Tercera etapa optimizada

Tabla 3.8: Valores obtenidos de la optimización completa

Variable	Valor
$\rho_1(t_f)$	0,50295
$\rho_2(t_f)$	-0,09968
$x_1(t_f)$	0,5024
$x_2(t_f)$	-0,0323
\mathcal{J}_{op_3}	224,15775

su lugar se observa que en el valor de τ_4 encontrado en la última etapa de optimización se presenta un crecimiento más acelerado de la acción de control, aunque al ser mayor a la cota de saturación $u_{m\acute{a}x}$ no presenta relevancia en la disminución total del costo.

La trayectoria de control final obtenido se muestra en la figura 3.22, los resultados obtenidos para la optimización total del problema con dos períodos regulares se pueden observar en la Tabla 3.8.

3. TRATAMIENTO LINEAL

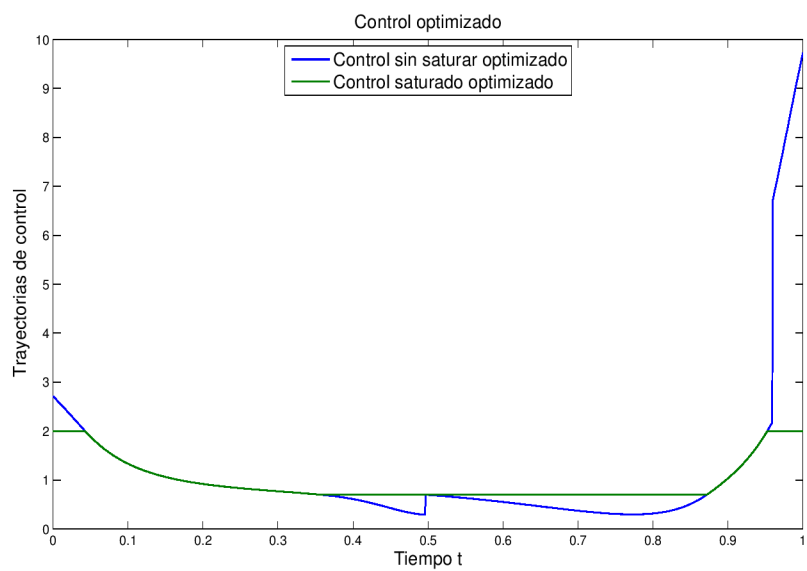


Figura 3.22: Control final obtenido

3.6 Conclusiones, perspectivas y discusión

Aunque una ley en forma de feedback puede ser preferido en la práctica, cuando se esperan que aparezcan perturbaciones, un control en lazo cerrado es en general sub-óptimo cuando existen restricciones en la variable manipulada. Tomando estas limitaciones en cuenta, se desarrolló un algoritmo eficiente para aproximar el control óptimo de lazo abierto vía feedback, basado en resultados teóricos recientes. Las estrategias resultantes difieren bastante de la forma del control óptimo saturado correspondiente al problema irrestricto con los mismos parámetros y condiciones iniciales, la cual es usada por el algoritmo como la primera aproximación, o trayectoria “semilla”.

3.6.1 Un período regular

En este caso se asume que existe un único intervalo regular $\mathcal{J} = (\tau_1, \tau_2) \subset (t_0, t_f)$. Un tratamiento sistemático de casos con un número contable de arcos regulares está en desarrollo a lo largo de líneas similares (como se explicó en la subsección 3.5).

La idea principal consiste en obtener una reducción total del costo utilizando fórmulas algebraicas a través de la actualización de los parámetros y los tiempos de conmutación. La estabilidad del método está garantizada ya que (i) al costo no se le permite aumentar, y (ii) está acotado previamente.

Algunas características positivas del nuevo método iterativo propuesto son:

- El método está basado en resultados teóricos que aseguran que ρ , μ y los tiempos de conmutación τ_1, τ_2 , del problema “oculto” son los parámetros críticos a ser determinados.

- Calcular la matriz de penalización final S en términos de ρ y μ reduce la cantidad de parámetros a actualizar. Con el método propuesto en [45], la cantidad mínima de variables a actualizar es $\frac{n(n+3)}{2} + 2$, ya que \tilde{S} es una matriz simétrica. Mediante el uso de la nueva estrategia el número total de variables es reducido a $2n + 2$, una reducción significativa para los sistemas de alta dimensión.

- El método toma ventaja de la disponibilidad de las matrices α, β como funciones de $(t_f - t, \tilde{S})$, y consecuentemente de la posibilidad de generar las matrices

3. TRATAMIENTO LINEAL

de Riccati $P(t, t_f, \rho, \mu)$ online sólo por operaciones algebraicas mientras se actualizan ρ, μ ; esto es, la RDE no tiene que ser resuelta cada vez que cambian los valores de ρ, μ .

- La actualización de parámetros $(\rho, \mu, \tau_1, \tau_2)$ se hace por el método del gradiente, y gracias a las fórmulas algebraicas desarrolladas, no se utiliza la integración numérica de ODEs para la predicción de trayectorias de estado, coestado, y costo.

3.6.2 Dos períodos regulares

En este caso se asume que existen dos intervalos regulares $J = (\tau_1, \tau_2) \cap (\tau_3, \tau_4) \subset (t_0, t_f)$, pero se prevé que la técnica desarrollada se pueda usar con un número finito de arcos regulares entre pares de tiempos de conmutación. Las estrategias utilizadas para este nuevo problema son basadas en el cálculo de un proceso fantasma u oculto, el cual debe coincidir con el óptimo en el período regular del problema real. Lo que se pretende lograr en la optimización es una estrategia de control construida a partir de la ecuación de Riccati, la cual es válida para el modelo lineal. Esta estrategia funciona bien ante la corrección por perturbaciones, porque trabaja con los estados del sistema real, usando la Riccati del modelo fantasma. Los efectos de las restricciones en la variable de control han sido tratadas en varias publicaciones del Grupo, y la búsqueda de soluciones a través del método del gradiente se puede ver en [43]. Cuando se pretende trabajar un problema no lineal, la trayectoria de control necesita ser construida por medio de λ (coestados) de acuerdo a la ecuación (2.22), por lo que no es posible utilizar una estrategia como la de Riccati, que tenga en cuenta la corrección de perturbaciones que se presenten en el sistema real [45]. Para este problema se observa que la reducción total del costo es insignificante, porque la trayectoria semilla y el resultado de la optimización están cerca del óptimo del problema. Estos resultados se obtuvieron con MATLAB y pueden ser llevados a una distribución en paralelo usando MATLAB [108] para lograr una reducción en el tiempo de cálculo. Se hicieron unas pruebas con *mathematica*, que reforzaron la creencia de que la semilla correspondía casi a la óptima.

3.6 Conclusiones, perspectivas y discusión

La velocidad de cómputo es el asunto central cuando se piensa en aumentar la complejidad de las estrategias de control, sobre todo debido a los requerimientos de tiempo real. El incremento de variables a actualizar en sistemas de gran dimensión genera la necesidad de cambiar los paradigmas actuales de computación. Es por eso que se analiza la instrumentación del algoritmo aquí propuesto mediante herramientas de computación en paralelo, como una alternativa, para alcanzar el control óptimo en tiempo real.

*En el corazón de los más temerarios
hay cuerdas que no pueden tocarse
sin emoción. Aun el más relajado
de los seres, para quien la vida y la
muerte son igualmente un juego, sa-
be que hay cosas con las cuales no
se puede jugar.*

Edgar Allan Poe

CAPÍTULO

4

Tratamiento de Sistemas no Lineales

Aunque muchas relaciones físicas se aproximan a menudo mediante ecuaciones lineales, en algunos casos el tratamiento no lineal se hace inevitable, sobre todo si los sistemas evolucionan lejos de sus puntos de equilibrio. Se dice que un sistema es no lineal si no cumple con el principio de superposición en algún caso [91, 97, 110]. Esto constituye una de las causas de que no se hayan desarrollado técnicas de análisis de sistemas no lineales tanto como aquellas para sistemas lineales, ya que las herramientas de análisis implican una matemática más avanzada. A diferencia de los sistemas lineales, donde puede ser posible buscar la solución explícita o analítica para cualquier IVP, son escasos los casos en los que éstas pueden obtenerse para los sistemas no lineales. Las propiedades básicas para los sistemas lineales tales como la existencia y la unicidad de soluciones no se mantienen, incluso algunos sistemas no lineales no poseen solución para determinados IVP dados, u otros tienen un número infinito de soluciones diferentes; además no se cuenta con la certeza de los sistemas lineales, ya que los sistemas no lineales pueden tender hacia puntos de equilibrios diferentes al variar levemente las IC, o en el caso práctico debido a perturbaciones o ruidos que se puedan presentar de acuerdo a la conformación de un problema determinado a través de un sistema dinámico. En general

4. TRATAMIENTO NO LINEAL

los sistemas dinámicos no lineales presentan comportamientos complejos que impiden obtener soluciones analíticas. Debido a la dificultad matemática asociada a los sistemas no lineales, resulta necesario introducir los sistemas lineales “equivalentes” como aproximación. Tales sistemas lineales equivalentes sólo son válidos para un rango limitado de operación. Una vez que se aproxima un sistema no lineal mediante un modelo matemático lineal, pueden aplicarse varias herramientas lineales para análisis y diseño.

Este capítulo pretende exponer herramientas diferentes para analizar sistemas de control no lineales, el estudio de estas técnicas no lineales son importantes porque permiten conservar las características de la dinámica, y porque pueden ser examinadas por intermedio de simulaciones que ratifiquen los resultados teóricos. Los sistemas de control para dinámicas no lineales constituyen en sí una gran dificultad. Si además se consideran restricciones al diseño de los controles esta dificultad aumenta, por eso la necesidad de aguzar el estudio teórico y la puesta a punto a través de las simulaciones en la que se evalúen los diseños del control, comparando en caso de rendimientos inapropiados y sugiriendo directivas para la modificación en los diseños. Para el tratamiento de sistemas no lineales se propone en primera medida abordarlo a través de una estrategia simple, en el cual se hacen variaciones de los controles por intermedio de un único método, y a través de una estrategia mixta, donde se empieza a obtener las correcciones de control desde una formulación para problemas regulares, siendo luego necesario considerar variaciones de otro tipo (incluso la propuesta para la estrategia simple). Cada una de las estrategias es analizada mediante aplicaciones numéricas.

4.1 Estrategia simple para sistemas no lineales

El diseño y la implementación de controladores no lineales es una tarea difícil comparado con el diseño de controladores lineales. Por lo tanto, hay pocas estrategias de control que tratan con las restricciones no lineales del control óptimo reportados en la literatura. Técnicas de aproximación para la resolución de sistemas no lineales se implementan en [11], donde la solución de un sistema no lineal se aproxima descomponiéndolo en una secuencia de ecuaciones lineales variantes en el tiempo (LTV), cuyas soluciones se encuentran arbitrariamente cerca de la solución del sistema no lineal original. El método mencionado es utilizado en [82], para estabilizar un sistema caótico no-homogéneo a través de control óptimo lineal con la solución de cada una de las LTV convergiendo hacia la solución no lineal de un control sin restricciones óptimo LQ, donde el control diseñado crea una trayectoria óptima que sigue una señal de referencia.

Una representación de la técnica de aproximaciones LTV se exhibe en [84] para controles con restricciones, cuya respuesta también converge a la respuesta de sistemas no lineales en el límite (en [10, 27] se resuelve explícitamente un problema de control óptimo en el que se demuestra la convergencia del método). Las técnicas mencionadas anteriormente hacen uso de secuencias aproximadas para las ecuaciones de Riccati, las cuales hacen posible utilizar métodos de control lineal en sistemas no lineales permitiendo la implementación de la teoría de control del LQR, convirtiendo un problema de control óptimo no lineal en un problema de control óptimo cuadrático LTV para cada iteración de la secuencia aproximada.

Recientemente [46] ha presentado un resultado novedoso sobre el problema con controles acotados para sistemas no lineales. El resultado teórico principal busca una relación con el tratamiento propuesto para sistemas lineales (ver sección 3.1), vinculando con la existencia del proceso fantasma (ver subsección 3.3) y la obtención de los distintos estados/coestados (x, λ) iniciales y/o finales. En [46] el Grupo de Sistemas No Lineales presenta un ejemplo monodimensional en el que, a través de una de las versiones del método del gradiente para la reducción del costo a partir de una trayectoria semilla, conduce a las condiciones faltantes para describir la solución óptima. Sin embargo, el método requiere la integración de las HCE

4. TRATAMIENTO NO LINEAL

del sistema no lineal (2.23, 2.24), que debido a su inestabilidad inherente resultan imprevisibles cuando se aplica a estos sistemas.

Debido a esto y pensando en lo adecuado las herramientas de cálculo que son utilizadas en los sistemas lineales (por ejemplo estrategias del control del tipo feedback), se plantea en este capítulo la posibilidad de trabajar con la RDE (2.35) para la linealización de un sistema por trayectoria [38, 42]. Este sistema aproximado resultante es lineal pero con matrices $A(t)$, $B(t)$ variantes en el tiempo.

La linealización se puede calcular a partir de un control semilla $u_{semilla}$, obtenido (por ejemplo) de la integración de la ecuación de HJB (2.19) para el sistema no lineal. Entonces se aplica el método del gradiente para generar unas nuevas condiciones ρ_Δ , μ_Δ que provoquen las mayores disminuciones posibles del costo, y calcular la nueva trayectoria de control U para el sistema lineal. El control no lineal resultante es entonces de la forma $\tilde{u} = u_{semilla} + U$. Luego se aplica este control, saturado, al proceso y las trayectorias de estado obtenidas de esta aplicación se convierten en la nuevas trayectorias semillas. Las variaciones U del control se van generando utilizando las soluciones $P(t)$ de la RDE para los sucesivos problemas lineales variantes en el tiempo. Se propone este procedimiento hasta que el método converge a la solución óptima o hasta que se da una orden de parada cuando sea conveniente. Se procura en este capítulo minimizar un funcional de costo cuadrático para un sistema no lineal Σ (ecuación (2.1)).

4.1.1 Búsqueda del proceso “oculto”, “fantasma” o irrestricto subyacente para sistemas no lineales

En [46], se muestra que existe un control óptimo $u_{x_0}^*$, continuo, para el problema restringido (con valores dentro del conjunto de valores admisibles), que puede ser (localmente) construido a partir de la saturación de la solución óptima de un problema regular irrestricto con la misma función de costo y dinámica pero con distintas condiciones iniciales y/o finales, en los estados y coestados. Para sistemas lineales dicha solución se construye utilizando la RDE (como se expuso en el capítulo 3) con una condición final adecuada. Sin embargo, en los casos no lineales, no está disponible la RDE, por lo que recurriremos a linealizaciones especiales.

Utilizando un planteamiento parecido al expuesto para sistemas lineales (sección 3.3), que buscan las trayectorias de control óptima para un proceso fantasma $(\hat{x}, \hat{\lambda})$, se procura encontrar una relación para sistemas no lineales, la cual quedaría totalmente determinada si se conocieran sus condiciones iniciales $(\hat{x}_0, \hat{\lambda}_0)$, o finales $(\hat{x}(t_f), \hat{\lambda}(t_f)) := (\rho, \mu)$. La búsqueda del problema fantasma requerirá en principio la integración de las HCE (2.23 - 2.24). En el capítulo 3 y en [43], se mostró que para sistemas lineales se puede evitar el uso de las HCE a través de una RDE que termina dependiendo de (ρ, μ) , y que además tiene la ventaja de generar controles del tipo feedback. Siguiendo la misma idea, en las siguientes secciones se planteará un método numérico para la búsqueda de las condiciones finales (ρ, μ) del proceso fantasma, sin recurrir a las HCE, con base en linealizaciones del modelo.

4.1.2 Linealización del modelo

Si se conoce una solución $x_{semilla}(t)$ de la ecuación (2.1) correspondiente a un cierto control sub-óptimo $u_{semilla}(t)$, se puede generar una linealización válida en las vecindades de $x_{semilla}(t)$ y $u_{semilla}(t)$. La dinámica que describe las evoluciones de las variables

$$\begin{aligned} X(t) & : = x(t) - x_{semilla}(t) \\ U(t) & : = u(t) - u_{semilla}(t), \end{aligned} \tag{4.1}$$

donde el vector de estados $x(t)$ se desviara del vector $x_{semilla}(t)$ debido a las pequeñas correcciones de control $U(t)$ que permiten al error $X(t)$ mantenerse también pequeño, verifica:

$$\begin{aligned} \dot{X}(t) & = f(x(t), u(t)) - f(x_{semilla}(t), u_{semilla}(t)) \\ & \approx \frac{df}{dx}(x_{semilla}(t), u_{semilla}(t))X(t) + \\ & \quad \frac{df}{du}(x_{semilla}(t), u_{semilla}(t))U(t). \end{aligned} \tag{4.2}$$

4. TRATAMIENTO NO LINEAL

Denotaremos las matrices dependientes del tiempo como

$$\begin{aligned} A(t) & : = \frac{df}{dx}(x_{sem}(t), u_{sem}(t)) \\ B(t) & : = \frac{df}{du}(x_{sem}(t), u_{sem}(t)), \end{aligned} \quad (4.3)$$

con lo que aparece un modelo lineal con las matrices de la dinámica variantes en el tiempo. Al ser un sistema lineal, se puede calcular un feedback $U(t) = k(t)X(t)$ a través de la RDE aún si es un sistema variante en el tiempo. La RDE no presenta la inestabilidad propia de las HCE, y además brinda una relación muy útil, $\lambda = 2Px$ como veremos. En [43] se ha mostrado que para sistemas lineales, la solución al problema con restricciones queda caracterizado si se encuentra los estados/coestados finales (denotados ρ_Δ, μ_Δ para las variaciones) del proceso fantasma descrito anteriormente. Queda entonces planteado el problema de cómo generar variaciones $U(t)$ que conduzcan al control óptimo a través de actualizaciones del $u_{semilla}(t)$.

4.1.3 Variaciones basadas en la ecuación diferencial de Riccati

En este caso las variaciones de control $U(t)$ serán soluciones al problema LQR consistente de la dinámica linealizada (4.2), y un funcional de costo que se puede reescribir de la siguiente manera:

$$\mathcal{J}_{lin}(u) \triangleq \int_0^{t_f} [X'(\tau)QX(\tau) + U'(\tau)RU(\tau)]d\tau + X'(t_f)SX(t_f), \quad (4.4)$$

donde la matriz Q conserva las dimensiones de $n \times n$, pero en este caso pesa las desviaciones de los estados con respecto a la trayectoria de referencia, la matriz R de $m \times m$ hace lo mismo pero para las desviaciones del control.

En esta subsección la notación Δ indicará objetos matemáticos con relaciones al sistema lineal de las variaciones. La solución de un problema lineal variante en el tiempo irrestricto, se basa en la solución $P_\Delta(\cdot)$ de la RDE

$$\dot{P}_\Delta(t) = P_\Delta(t)W(t)P_\Delta(t) - P_\Delta(t)A(t) - A'(t)P_\Delta(t) - Q, \quad (4.5)$$

con $P_\Delta(t_f) = S_\Delta$, donde

$$W(t) \triangleq B(t)R^{-1}B'(t), \quad (4.6)$$

que permite construir una desviación del control

$$U(t) = -R^{-1}B'(t)P_\Delta(t)X(t). \quad (4.7)$$

Reemplazando la ecuación (4.7) en (4.2) se obtiene

$$\dot{X}(t) = [A(t) - W(t)P_\Delta(t)]X(t), \quad (4.8)$$

donde $X(t)$ se resuelve bajo la condición final $X(t_f) = \rho_\Delta$. La matriz de penalización final para resolver la ecuación (4.5) se obtendrá por medio de

$$S_\Delta = \frac{1}{2} \frac{\mu_\Delta \mu'_\Delta}{\rho'_\Delta \mu_\Delta}, \quad (4.9)$$

donde ρ_Δ corresponde a las variaciones de los estados, y μ_Δ las variaciones de los coestados desde la trayectoria semilla, donde queda claro que se usa S_Δ para lograr $\mu_\Delta = 2S_\Delta\rho_\Delta$ en cada caso. Este tipo de variaciones generan pequeñas variaciones desde las trayectorias semillas (debido a las penalizaciones involucradas en el funcional del costo). Todo esto garantiza que las linealizaciones involucradas en las definiciones de X, U son también buenas aproximaciones y que las trayectorias de $\{x, u\}$ se mantengan cerca de la trayectoria semilla, buscando la reducción del costo por el método del gradiente.

4.1.4 Actualización de parámetros

Las primeras aproximaciones $\tau_{i,0}$ de los instantes de saturación óptimos τ_i pueden ser estimados, por el momento, a partir de la solución del problema de control irrestricto disponible de la simulación (offline) vía (2.19), esto es por intermedio de la solución de la variable adjunta de la ecuación (2.20) reemplazada en (2.22). Los puntos de saturación $\tau_{i,0}$ pueden ser entonces adoptados como los puntos donde \tilde{u} alcanza las fronteras del conjunto de valores de control admisibles \mathbb{U} . Para

4. TRATAMIENTO NO LINEAL

$t \in [0, t_f]$ el control adoptado como el candidato inicial u_0 (llamado la trayectoria ‘semilla’ y denotado $u_{semilla}$), de forma similar que la ecuación (3.53), es,

$$u_{semilla}(t) = u_0(t) \equiv [u^0(x_{semilla}(t), \lambda_{semilla}(t))]^{sat}, \quad (4.10)$$

donde $\lambda_{semilla}(t)$ puede ser obtenida de forma similar a la ecuación (2.20), en este caso,

$$\lambda_{semilla}(t) = \left(\frac{\partial \mathcal{V}}{\partial x}(t, x_{semilla}(t)) \right)', \quad t \in [0, t_f]. \quad (4.11)$$

Los valores de los parámetros en este caso ($\rho_\Delta, \mu_\Delta, \tau_i$) son actualizados para construir estrategias de control sucesivas de forma que decrementen el valor total de costo según la ecuación (3.54) y por intermedio del método del gradiente similar al usado en (3.55). El control es actualizado mediante

$$\tilde{u}_{j+1}(t) = [\tilde{u}_j(t) + U(t)]^{sat}, \quad j = 1, 2, \dots \quad (4.12)$$

donde U_j y X_j se calculan a partir de las ecuaciones (4.6 - 4.8), y P_Δ resolviendo la ecuación (4.5) con condición final

$$P_\Delta(t_f) = S_\Delta = \frac{1}{2} \frac{\mu_{\Delta j} \mu'_{\Delta j}}{\rho'_{\Delta j} \mu_{\Delta j}}. \quad (4.13)$$

4.1.5 Aplicación numérica para la estrategia simple

Se describe a continuación la aplicación de la metodología expuesta en las subsecciones anteriores (4.1.3, 4.1.4) en un modelo no lineal de dos dimensiones, el cual es una modificación al problema tratado en el capítulo 3 que simula el proceso de frenado de un tren con rozamiento [80, 81].

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \alpha u(t) + \beta u(t)x_2(t) \\ u(t) &\in [0, 3] \subset \mathbb{R}, \end{aligned} \quad (4.14)$$

donde la variable manipulada o de control $u(t)$ es una función con valor escalar y se interpreta como la acción de frenado en un tren que tiene por estados la posición x_1 y la velocidad x_2 . El parámetro α es una constante que normaliza las unidades

de la acción de control y β es una constante asociada a la fricción del tren sobre los rieles. Las condiciones iniciales son elegidas como $x_1(0) = 1$ y $x_2(0) = -1$, los valores de los parámetros como $\alpha = 1$ y $\beta = 0,15$. El objetivo es minimizar un funcional de costo cuadrático, donde los valores adoptados para los parámetros del costo (2.27) son $Q = 10I_n$, $R = 0,5$, tanto para el problema no lineal como para su linealización, la penalización final del problema no lineal es $\mathcal{K}(x(t_f)) = x(t_f)'Sx(t_f)$ con $S = 100I_n$. Las matrices variantes en el tiempo resultan,

$$A(t) \triangleq \begin{pmatrix} 0 & 1 \\ 0 & 0,15\tilde{u}^{sat}(t) \end{pmatrix}, \quad B(t) \triangleq \begin{pmatrix} 0 \\ 1 + 0,15\tilde{x}_2(t) \end{pmatrix}. \quad (4.15)$$

4.1.5.1 Solución óptima del problema irrestricto

El modelo (4.14) representa un sistema bilineal de control. En [41] se describe el problema de control óptimo para esta clase de sistemas, donde el control óptimo resulta:

$$u^*(x, t) = -\frac{1}{2}R^{-1}(B + Nx)'\lambda(t). \quad (4.16)$$

y las HCE:

$$\dot{x} = Ax + (B + Nx)u^*; \quad x(0) = x_0 \quad (4.17)$$

$$\dot{\lambda} = -2Q(x - \bar{x}) - A'\lambda - N'\lambda u^*, \quad (4.18)$$

con condición final $\lambda(t_f) = 2Sx(t_f)$ desconocida. De la aplicación del principio de Pontryagin se sabe que el control óptimo del problema con controles acotados tomará la forma de la ecuación (2.44). Utilizando un método numérico (off-line y similar al “shooting”) se obtuvo la solución óptima, para comparar los resultados de la nueva técnica descrita. Los objetos relevantes de la solución están en la Tabla 4.1 y además: $\lambda_1(0) = 43,717479$, $\lambda_2(0) = 8,677516$.

En [41] se demuestra que la relación $\lambda = 2P(t, x)x$, sigue siendo válida, pero con una matriz $P(t, x)$ que verifica la siguiente ecuación en derivadas parciales:

$$\begin{aligned} \frac{\partial P}{\partial t} + \frac{\partial P}{\partial x} \cdot (Ax - W_x P x) = \\ -Q - A'P - PA + P W_x P + R^{-1}(N P x)(B + N x)' \end{aligned} \quad (4.19)$$

4. TRATAMIENTO NO LINEAL

donde $W_x = (B + Nx)R^{-1}(B + Nx)'$ y $P(t_f, x) = S$. Notar que si $N = 0$, la ecuación en derivadas parciales se convierte en la RDE para sistemas lineales. El resultado teórico para el problema con controles acotados dice que la solución óptima para este problema puede encontrarse a través de la saturación de otro problema irrestricto pero con condiciones distintas, es decir, x_0 y S diferentes al del problema original. Se comprobará, para sistemas bilineales, que existe un proceso irrestricto con \tilde{x}_0 y \tilde{S} tales que $\tilde{u}(t)$ toma la forma de la ecuación (2.44) con $u_{x_0}^* = -R^{-1}(B + Nx)'P(t, x)x$. Para recuperar el proceso fantasma se hizo lo siguiente: (i) debido a que en el intervalo \mathcal{J} los estados y coestados de la solución óptima coinciden con el proceso fantasma, entonces se integraron desde τ_2 hacia el final del horizonte las HCE (4.17 - 4.18), con lo que se aproximaron los estados/coestados finales (ρ, μ) fantasma (ver figura 4.1), (ii) partiendo de estos valores se integró hacia atrás las HCE para obtener los estados iniciales $\tilde{x}_1(0) = 0,484511$; $\tilde{x}_2(0) = 3,09403$ buscados (la matriz de penalización del fantasma \tilde{S} calculada con la ecuación (4.13) y los (ρ, μ) resulta tener los coeficientes $\tilde{S}_{11} = 1214,89$; $\tilde{S}_{12} = -342,526$; $\tilde{S}_{22} = 96,5718$), y (iii) se comprobó que la saturación del control de ese fantasma coincide con el control de la solución óptima en el intervalo para $t \in [\tau_1, \tau_2]$ (ver figura 4.1).

4.1.5.2 Método del gradiente para el problema restringido

En esta subsección se aplica el nuevo algoritmo mediante los siguientes pasos:

(i) Se calcula una trayectoria semilla a partir de la solución del problema de control óptimo irrestricto no lineal, que se satura, con lo que se genera un $\tilde{u}_0(t)$, y se aplica al proceso para obtener $\tilde{x}_0(t)$, y por consiguiente ρ_0, μ_0, τ_{10} , y τ_{20} .

(ii) Se obtienen la matrices de la dinámica variantes en el tiempo para el problema linealizado, correspondientes a $A(t), B(t)$ con los valores semilla \tilde{x}_0, \tilde{u}_0 .

(iii) Se aplican variaciones a la trayectoria de control semilla por intermedio de las pequeñas variaciones “controladas” del control, al resolver la RDE para la linealización con condición final (4.13), y se calculan los diferentes costos de las variaciones.

(iv) Se calcula el gradiente del costo y se actualiza la trayectoria de control semilla.

(v) Se envía este control al proceso y se calcula el nuevo estado semilla.

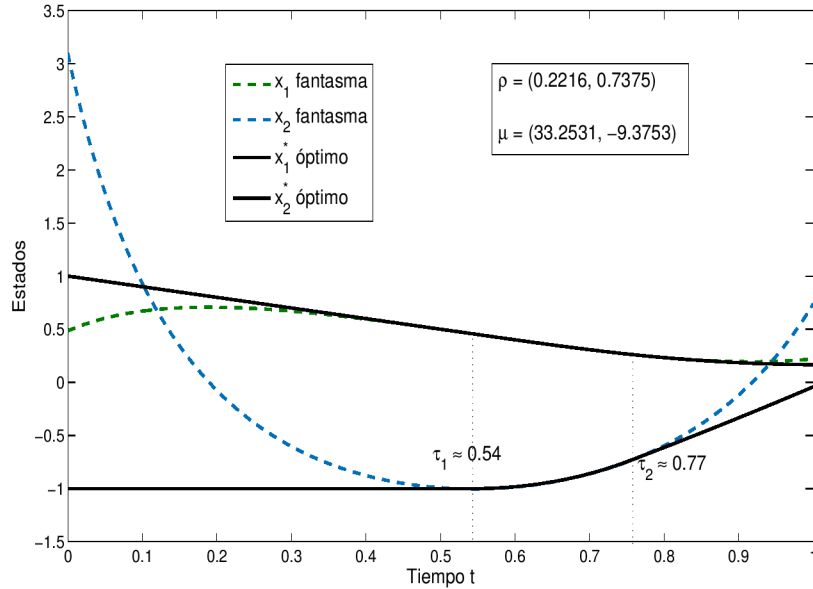


Figura 4.1: Modelo relacionado.

Este procedimiento se repite hasta que se alcanza una tolerancia dada en el valor del gradiente. La trayectoria semilla fue hallada a través de la integración de la HJB (2.19) para la función de valor. De la solución se aproximó numéricamente el valor inicial de los coestados como la derivada parcial de V con respecto a x , y evaluada en $t = 0$, resultando

$$\lambda(0) = \begin{pmatrix} 31,535778618 \\ 1,78205946 \end{pmatrix}. \quad (4.20)$$

La solución de la ecuación (2.19), además de permitir calcular los coestados también predice el costo óptimo total del problema irrestricto, que para la trayectoria semilla es $V(0, X(0)) = 14,7514$ (ver figura 4.2). La HJB no siempre se puede integrar a causa de la no linealidad del sistema y su dimensión. Debido a esto y para futuras aplicaciones, una semilla alternativa podría provenir de las bilinealizaciones del sistema, utilizando las ecuaciones de la subsección anterior.

Con la linealización se busca construir variaciones “controladas” a los estados/coestados y al control por medio de la solución de la RDE garantizando que las variaciones del problema original con restricciones se mantenga cerca de la trayectoria semilla, que se va actualizando.

4. TRATAMIENTO NO LINEAL

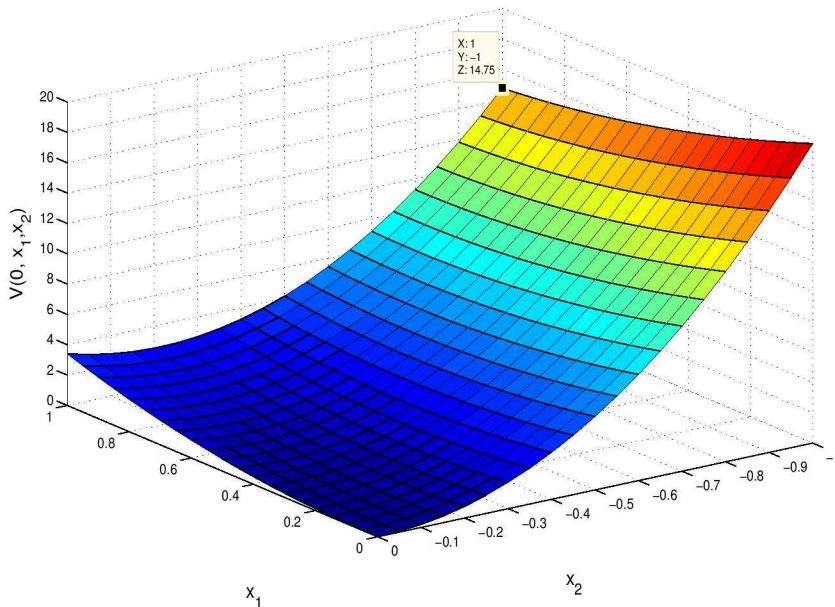


Figura 4.2: Solución de la función de valor en $t = 0$.

Los valores obtenidos con la estrategia numérica después de 700 iteraciones se muestran en la Tabla 4.1. Además se compara con la solución óptima del problema y la trayectoria semilla. El método numérico prácticamente llega a la solución óptima. La ventaja de la metodología presentada reside en que permitiría construir estrategias con leyes de control del tipo feedback, y evitando la integración numérica de las HCE.

En la figura 4.3 se muestra la evolución de las trayectorias de control óptimo y trayectoria final del método numérico. Se gráfica también la semilla y una semilla alternativa proveniente de la bilinealización del sistema, generada por intermedio de la saturación del control obtenido para una matriz de Riccati $P(t, x)$ como se propone en [41]. Nótese que la solución semilla difiere considerablemente de la solución óptima y de la aproximada por el método numérico, aún cuando la diferencia relativa de los costos es solo del 9%. Para la semilla alternativa se puede apreciar una mejor aproximación, aunque de la misma manera que para la primera semilla la mejora obtenida por el método propuesto muestra una aproximación más cercana al óptimo. La solución numérica aproxima bastante bien a la solución óptima por lo que se puede concluir que esquemas basados en linealizaciones por trayectoria

Tabla 4.1: Valores numéricos finales estrategia simple

	Sol. óptima	Sol. semilla	Método numérico
$x_1(t_f)$	0,16657	0,2187	0,1664
$x_2(t_f)$	-0,0423	-0,0137	-0,0456
τ_1	0,5442	0,23	0,565
τ_2	0,7688	0,88	0,837
\mathcal{J}	15,441	16,779	15,448

del sistema no lineal serían útiles en la generación de controles subóptimos que cumplen con las restricciones.

La figura 4.4 ilustra la reducción del costo con las iteraciones por medio del método del gradiente, y las variaciones de los estados $X(t)$ para el problema linealizado, que a su vez permite construir una variación de control en forma de “feedback” de manera sistemática. Después de 700 iteraciones el costo converge casi al valor óptimo.

La figura 4.5 muestra las variaciones de los instantes de saturación con las iteraciones, los cuales se mueven buscando valores críticos para que el intervalo regular (donde el control no se satura) tienda al intervalo del problema no lineal original.

4. TRATAMIENTO NO LINEAL

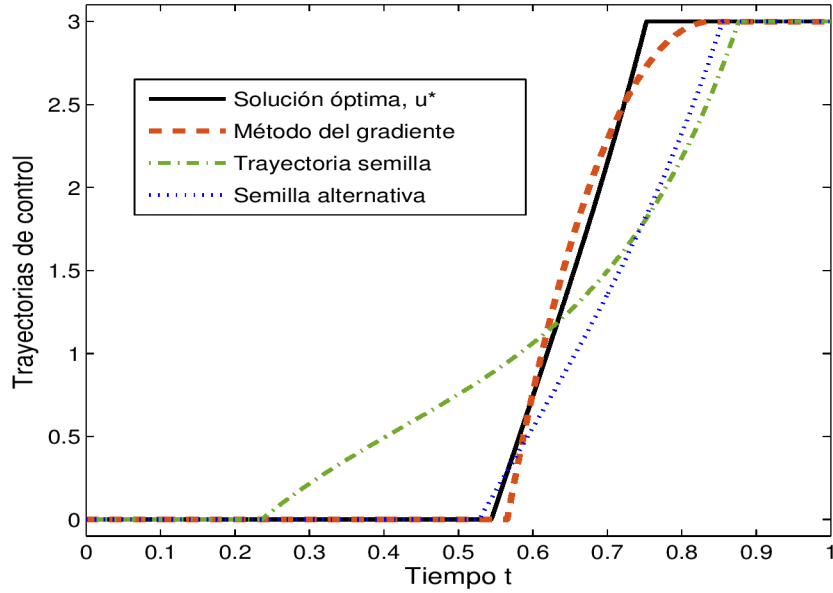


Figura 4.3: Trayectorias de control.

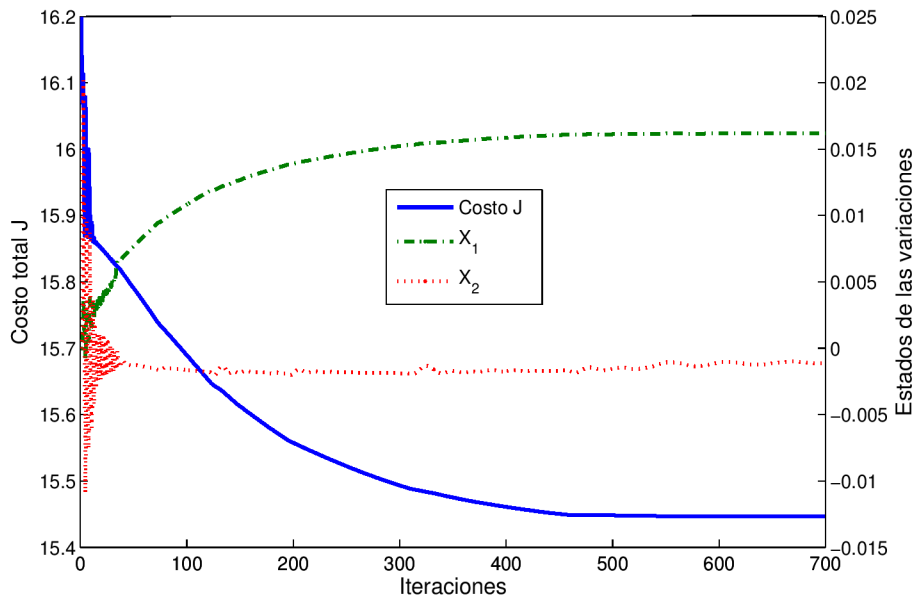


Figura 4.4: Variación del costo y de los estados $X(t)$ para el sistema linealizado.

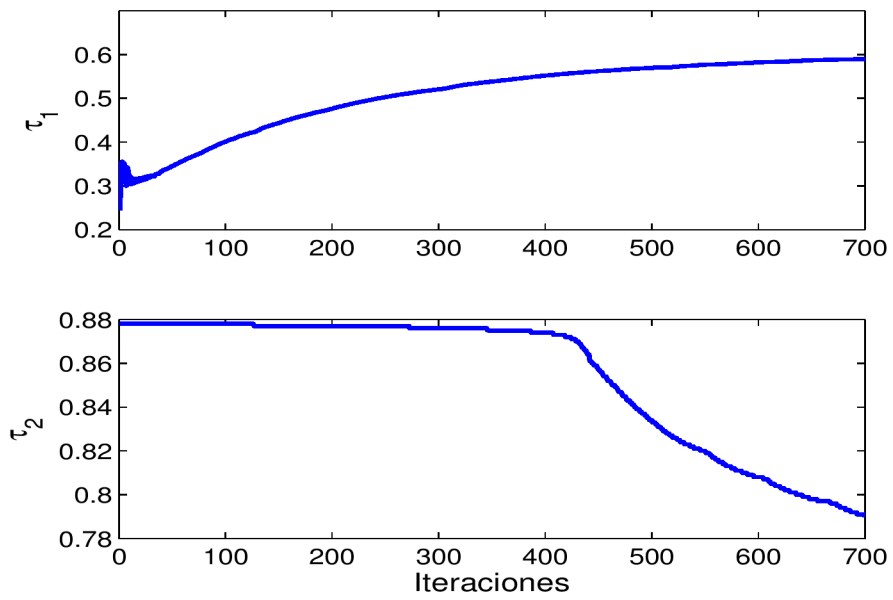


Figura 4.5: Evolución de los taus con las iteraciones.

4.2 Estrategia mixta para sistemas no lineales

La estrategia mixta para un sistema de control no lineal restringido busca reducir sistemáticamente el costo total asociado con una secuencia particular de estrategias de control, después de proponer una trayectoria inicial $u_{semilla}$, (tratadas de igual manera que en la sección 4.1), utilizando una combinación de enfoques del problema: (i) a través de la formulación de Sontag, la cual se basa en un método numérico donde se incluye el costo total como una variable de estado para un sistema extendido, intentando recursivamente modificaciones de los controles tal que la diferencia de la nueva variable con la modificación y la pasada se mantengan negativas (< 0), (ver [134]), (ii) prueba de distintos tipos de variaciones, basadas en diferentes estrategias que formulan pequeñas acciones “controladas” para reducir sistemáticamente el costo total. La nueva estrategia permite inicialmente explotar una expresión teórica para el diferencial del funcional del costo. Una de las ventajas de esta fórmula es que el decrecimiento del costo puede ser obtenido sin necesidad de integrar la dinámica no lineal numéricamente. Esta configuración es totalmente válida para situaciones regulares, y algunas de sus características son también usadas en muchos casos cuando la saturación ocurre. Alrededor de la semilla el costo y la dinámica son diferenciados y un método iterativo es diseñado para mejorar las semillas. Las derivadas involucradas implican controles irrestrictos y regularidad del problema, las cuales no son garantizadas en este caso (debido a las restricciones de la variable manipulada). Sin embargo, ellas tienen las siguientes características principales positivas. (i) la fórmula resultante produce en general resultados aceptables durante un cierto número de iteraciones, y (ii) la expresión del diferencial del costo debido a las variaciones de control es válida en cuanto éstas permanezcan pequeñas, independientemente si los controles están saturados o no, lo cual permite activar una condición de alerta cuando el diferencial se vuelve negativo (una clara indicación que la validez de la configuración general ha colapsado). Si se produce una de estas descripciones, un enfoque alternativo es activado, entonces un número de diferentes perturbaciones de la estrategia de control actual son generadas y testeadas para forzar una disminución de costo adicional. Algunas de las perturbaciones corresponden a las soluciones de la RDE trabajadas en la sección 4.1, pero en este caso con pequeñas perturbaciones en el estado

final (no hace uso del método del gradiente, éstas son explicadas mas adelante). Cuando estas variaciones de control son insuficientes, otras como (i) ‘variaciones espaciales’ de Pontryagin, similares a las utilizadas en la generación de los estados finales inscritos en el cono terminal [141, 142], y (ii) oscilaciones en el interior de los períodos regulares, por la modificación de los tiempos de saturación τ_i , o por añadir cambios admisibles en los valores de control durante pequeños subintervalos del tiempo horizonte.

4.2.1 Métodos de reducción del costo

4.2.1.1 Reducción del costo en problemas regulares

En esta sección el problema original (ecuación 4.2), será ligeramente transformado en uno equivalente. La nueva formulación agrega dos nuevas variables de estado x_e y z asociado con la función original de Lagrange acumulada y la penalización final. El nuevo estado

$$x^\# \triangleq \begin{pmatrix} x' & z & x_e \end{pmatrix}', \quad (4.21)$$

evolucionará en \mathbb{R}^{n+2} con la dinámica

$$\dot{x}^\# = f^\#(x^\#, u) = \begin{pmatrix} f(x, u) \\ 0 \\ L_b(x, u) \end{pmatrix}, \quad x^\#(0) = \begin{pmatrix} x_0 \\ \mathcal{K}(x(t_f)) \\ 0 \end{pmatrix}, \quad (4.22)$$

con la función del Lagrangiano redefinido como

$$L_b(x, u) \triangleq L(x, u) + \frac{z}{t_f}, \quad (4.23)$$

por lo que la ecuación de costo original (2.3) puede ser expresada de la forma de una penalización final pura (problema de control óptimo en la forma de [134]):

$$\mathcal{J}(u) = x_e(t_f) \quad (4.24)$$

Las expresiones de los objetos relevantes para el sistema aumentado luego de su linealización alrededor de la trayectoria x, \tilde{u} resultan en [61, 141]

$$A^\#(t) = \begin{pmatrix} \frac{\partial f(x, \tilde{u})}{\partial x} & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\partial L(x, \tilde{u})}{\partial x} & \frac{1}{t_f} & 0 \end{pmatrix}, \quad B^\#(t) = \begin{pmatrix} \frac{\partial f(x, \tilde{u})}{\partial u} \\ 0 \\ \frac{\partial L(x, \tilde{u})}{\partial u} \end{pmatrix}, \quad (4.25)$$

4. TRATAMIENTO NO LINEAL

$$[\lambda^\#(t_f)]' = (\nabla \mathcal{K}(x(t_f)), -1, 1), \quad (4.26)$$

donde $\lambda^\#(t_f)$ es el valor final de la variable del coestado para el sistema aumentado. Los dos nuevos valores de los coestados son obtenidos de acuerdo a la formulación de Pontriagyn, en el cual existirá un cono $\mathfrak{K} \subset \mathbb{R}^{n+2}$, conteniendo estados $x^\#(t_f)$ obtenidos como resultado de la aplicación de perturbaciones a la trayectoria de control óptima, esto implica la existencia de un hiperplano tangente \mathcal{T} de dimensión n que, aplicando en el vértice 0_{n+2} , divide a \mathbb{R}^{n+2} en dos partes, dejando los puntos del cono en uno de los semiespacios. Existirá un vector $\lambda^\#(t_f) \perp \mathcal{T}$ (con el vector $\lambda^\#(t_f)$ apuntando hacia el semiespacio donde está el cono), tal que

$$\lambda^\#(t_f) \cdot x^\#(t_f) \geq 0 \quad \forall x^\#(t_f) \in \mathfrak{K}, \quad (4.27)$$

en este caso la variable $\lambda_{n+2}^\#(t_f) \geq 0$ de forma que apunte hacia el semiespacio donde está el cono de estados de Pontriagyn, si se supone que los estados finales $x(t_f)$ son libres, se deberá adoptar un valor de $\lambda_{n+2}^\#(t_f) = 1$ para evitar que el vector que apunta hacia el semiespacio sea nulo, ya que se considera $x(t_f) = 0_n$. Para la variable $\lambda_{n+1}^\#(t_f)$, al estar afectados los estados por la penalización final, se considera de forma que su estado final esta restringido a una superficie dada \mathcal{S} de dimensión l en \mathbb{R}^n , definida por una función suave $\Phi : \mathcal{O} \rightarrow \mathbb{R}^l$, a su vez definida en una abierto $\mathcal{O} \subset \mathbb{R}^n$. Para arribar estrictamente al punto determinado por el estado en $n + 1$, es decir $x_{n+1}^\#(t_f) = z$, correspondiente a la penalización final $\mathcal{K}(x(t_f)) \in \mathbb{R}$, el enunciado del PMP verifica para el arribo a un punto $l = 1$, entonces

$$\begin{aligned} \Phi(x^\#) &= \mathcal{K}(x) - z \\ \Phi(x^\#) &= \mathcal{K}(x^\#) - x_{n+1}^\# \\ D\Phi(x_1^\#, x_2^\#, \dots, x_n^\#, x_{n+1}^\#) &= (\nabla \mathcal{K}(x_1, x_2, \dots, x_n), -1), \end{aligned} \quad (4.28)$$

de la expresión se desprende que $\lambda_{n+1}^\#(t_f) = -1$. En lo que sigue vamos a obviar el signo $\#$ por simplicidad, asumiendo que el significado del nuevo estado aumentado y la expresión del costo son claros como en la ecuación (4.22, 4.24). El enfoque numérico para la minimización de $\mathcal{J}(u)$ en el contexto irrestricto [134] procura calcular una perturbación $\mu : [0, t_f] \subset \mathbb{R}^m$ de $u_{semilla}$ tal que

$$\mathcal{J}(u_{semilla} + h\mu) < \mathcal{J}(\tilde{u}) \quad (4.29)$$

para un h lo suficientemente pequeño. Una variación adecuada se puede elegir como [134]

$$\mu(t) \triangleq -B(t)' \Phi(\tau, t)' \lambda(t_f), \quad (4.30)$$

donde $\Phi(\tau, t)$ es la matriz fundamental asociada con $A(\cdot)$, esto es la solución a la ecuación lineal (véase [29]):

$$\frac{\partial \Phi(\tau, t)}{\partial \tau} = A(\tau) \Phi(\tau, t), \quad \Phi(\tau, \tau) = I. \quad (4.31)$$

El diferencial del costo puede ser estimado por la fórmula [134]

$$\Delta \mathcal{J}_S = \lambda(t_f)' \int_0^{t_f} \Phi(t_f, s) B(s) \Delta u(s) ds, \quad (4.32)$$

donde el incremento de control es, para problemas regulares,

$$\Delta u \triangleq h \mu. \quad (4.33)$$

4.2.1.1.1 Actualización de las trayectorias de control

Las primera aproximación en la estrategia mixta puede obtenerse de la misma forma que en 4.1.4 vía HJB. Esta estrategia permite elegir otras trayectorias de control semillas, de acuerdo por un lado a la intuición o el conocimiento de un problema en particular, o cuando la solución de la HJB se vuelve difícil de resolver. La trayectoria semilla es actualizada como en (3.54), la cual permite estimar las disminuciones del costo a través de la ecuación (4.32), donde ahora

$$\Delta \tilde{u}_{j+1} \triangleq \tilde{u}_{j+1} - \tilde{u}_j. \quad (4.34)$$

Para tener en cuenta las restricciones del problema que nos ocupa, se hace un primer intento de disminuir el costo a través de una modificación natural del procedimiento descrito anteriormente para problemas regulares. Esta modificación consiste en:

- (i) Para $j = 0, 1, 2, \dots$ estimar μ_j de la ecuación (4.30), donde $B(t)$, $\Phi(\tau, t)$, $\lambda(t_f)$ son evaluados de trayectorias $x_{semilla}$, $u_{semilla}$, entonces
- (ii) Saturar el control regular actualizado, esto es definir un nuevo control candidato pero usando la formula general

$$\tilde{u}_{j+1}(t) \triangleq [\tilde{u}_j(t) + h \mu_j]^{sat}, \quad (4.35)$$

4. TRATAMIENTO NO LINEAL

para algún $h > 0$ pequeño. De esta manera, todos los u_j siempre tomarán valores admisibles. El valor de h es elegido para tener un valor negativo $\Delta\mathcal{J}_S$, calculado de la ecuación (4.32). Si, después de reducir h hasta cierta tolerancia, el $\Delta\mathcal{J}_S$ no puede hacerse negativo, entonces el método falla y deberá ser abandonado. Si el método mantiene la reducción del costo, todavía es conveniente estar preocupados por una condición de alerta

$$\Delta\mathcal{J}_S \approx \Delta\mathcal{J}_A, \quad (4.36)$$

donde $\Delta\mathcal{J}_A$ es alguna medida del error generada por la saturación de la trayectoria de control, y donde el simbolo \approx significa que ambos términos son del mismo orden de magnitud (estas dos afirmaciones serán discutidas en la subsección 4.2.1.2). Por el momento, si la condición (4.36) es alcanzada, el método deberá también ser abandonado. Si no,

(iii) Renombrar $u_{semilla} \leftarrow \tilde{u}_{j+1}$, $j \leftarrow j + 1$, y repetir los pasos (i) a (iii) durante un número fijo de modificaciones de $u_{semilla}$. En este momento, es apropiado actualizar la trayectoria de estados, y entonces

(iv) Evaluar x_{j+1} a través de la integración numérica de

$$\dot{x} = f(x, \tilde{u}_{j+1}), x(0) = x_0, \quad (4.37)$$

(v) Actualizar $x_{semilla} \leftarrow x_{j+1}$ y reanudar los pasos (i – v) hasta que $\Delta\mathcal{J}_S$ sea considerado lo suficientemente pequeño.

4.2.1.2 Criterios de alerta de falla durante el procedimiento regular

Cuando las restricciones en la variable de control aparecen, las derivadas con respecto a u del Hamiltoniano o Lagrangiano pueden no estar definidas en las fronteras del conjunto de valores admisibles de control, y la fórmula para el diferencial del costo podría volverse no válida. Es entonces de esperar que, cuando se satura la solución óptima, el procedimiento descrito en la sección 4.2.1.1 puede eventualmente dejar de reducir el costo. Se analizan dos criterios para evaluar un eventual fallo del método regular:

(i) El teorema principal de Pontryagin para control óptimo es basado en el análisis de los efectos de perturbaciones espaciales en el control sobre los estados finales alcanzados [142]. El truncamiento de las estrategias de control debido a la

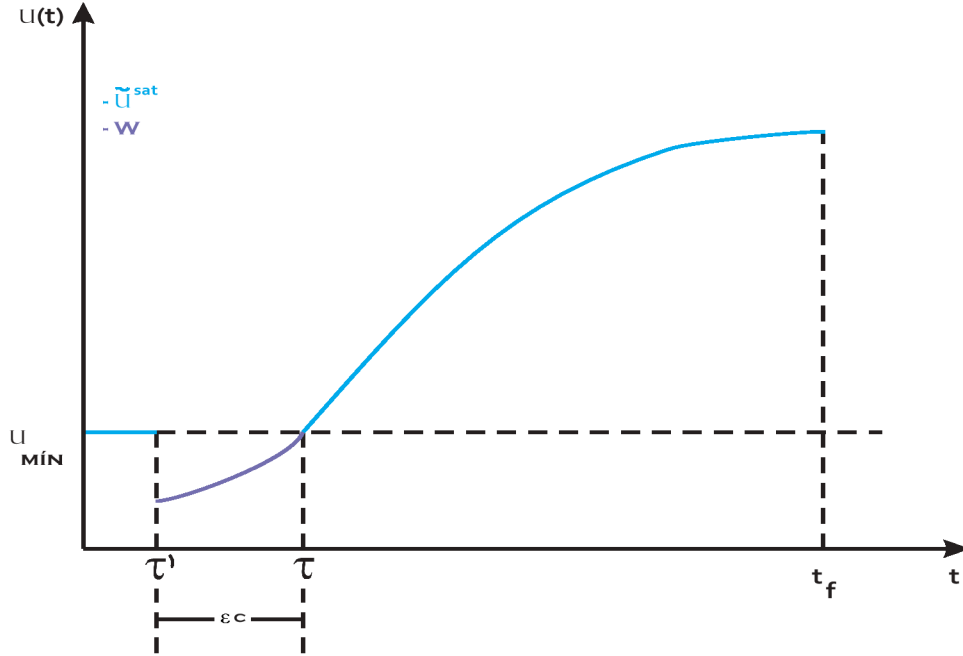


Figura 4.6: Truncamiento cuando la restricción es inferior.

existencia de los límites (restricciones) puede ser considerado como una perturbación de control, y sus efectos evaluados vía las matrices de sensibilidad descritas en [117, 141].

Para estrategias de control $\tilde{u}^{sat} \in \mathbb{U}$ en $[0, t_f]$ y considerando que las saturaciones (truncaciones) ocurren en los tiempos de saturación $\tau_i \in (0, t_f)$, las variaciones de control $u_s(\cdot)$ con respecto a los controles permisibles $\tilde{u}^{sat}(\cdot)$ pueden ser creados de la siguiente manera:

$$u_s(t) = \begin{cases} w & \forall t \in [\tau', \tau) \\ \tilde{u}^{sat}(t) & \forall t \in [0, \tau') \cup [\tau, t_f) \end{cases}, \quad (4.38)$$

representadas en las figuras 4.6 y 4.7, donde w es el valor de control generado por el procedimiento regular al instante de tiempo τ' cerca de τ . Llamando $x_{\tilde{u}^{sat}}(\cdot)$ a la trayectoria de estado correspondiente a $\tilde{u}^{sat}(\cdot)$, y $x_e(\cdot)$ la correspondiente a la variación $u_s(\cdot)$, el valor final del estado será (ver [141]):

$$x_e(t_f) = x_{\tilde{u}^{sat}}(t_f) + \varepsilon c Z_\tau(t_f) \Delta f(\tau, w) + \varepsilon l(\varepsilon), \quad (4.39)$$

4. TRATAMIENTO NO LINEAL

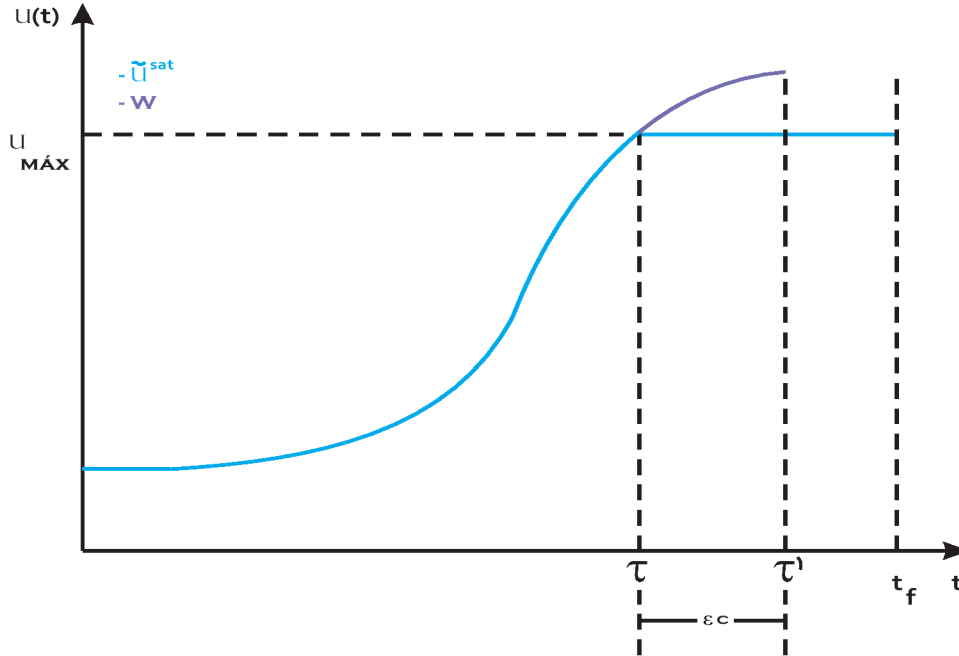


Figura 4.7: Truncamiento cuando la restricción es superior.

donde ε es un pequeño número, tal que $\varepsilon c = \tau - \tau'$, y $l(\varepsilon)$ es un vector (infinitesimal) cuya norma tiende a 0 cuando $\varepsilon \rightarrow 0$. El cambio en el valor final es entonces, aproximadamente,

$$\Delta x(t_f) = \varepsilon c Z_\tau(t_f) \Delta f(\tau, w) . \quad (4.40)$$

De las ecuaciones (4.22, 4.24), las variaciones del costo son reflejadas en el último componente $(n + 2)$, cuya expresión es

$$\Delta \mathcal{J}_Z = [\Delta t^j Z_\tau(t_f) \Delta f(\tau, w)]_{n+2} , \quad (4.41)$$

donde $\Delta t^j \triangleq \tau^{j+1} - \tau^j$; $j = 1, 2, \dots$ sustituye a la perturbación de tiempo previamente explicada para εc .

La matriz de sensibilidad Z_τ surge de las variaciones espaciales del control, esto es, de la diferencia del control propuesto en la ecuación (4.30) y el control que es verdaderamente aplicado después de la saturación. $Z_\tau(\cdot)$ es una matriz de $(n + 2) \times (n + 2)$, solución del siguiente problema de valor inicial, definido en $[\tau, t_f]$:

$$\dot{Z} = A(t) \cdot Z ; Z(\tau) = I ,$$

y $\Delta f(\tau, w)$ es una expresión que representa la diferencia de la función f evaluada con y sin perturbación en el tiempo de saturación τ , es decir,

$$\Delta f(\tau, w) \triangleq f(x_{\tilde{u}^{sat}}(\tau), w) - f(x_{\tilde{u}^{sat}}(\tau), \tilde{u}^{sat}(\cdot)). \quad (4.42)$$

Considerando ahora la suma de los efectos de las variaciones para un número finito de tiempos de saturación (k), la ecuación para los resultados de las variaciones del costo es

$$\Delta \mathcal{J}_Z = \sum_{i=1}^k [\Delta t_i^j Z_{\tau_i}(t_f) \Delta f(\tau_i, w_i)]_{n+2}. \quad (4.43)$$

(ii) El segundo criterio para la detección de fallo reside en la evaluación de las variaciones del costo debidas sólo a la saturación del control, y luego comparar con la variación total del costo. La variación de la saturación es obtenida por la aproximación de la fórmula (4.32) cuando se aplican a estas perturbaciones espaciales de control en intervalos $[\tau', \tau)$,

$$\Delta \mathcal{J}_T = \lambda(t_f)' \tilde{\Phi} \tilde{B} \Delta u \Delta t, \quad (4.44)$$

donde $\tilde{\Phi}$ y \tilde{B} son valores promedios para la matriz fundamental y para la matriz de control entre $[\tau', \tau)$, con $\Delta u = w$ y $\Delta t = \Delta t^j$ tomando el mismo significado que en el tratamiento previo.

Entonces, considerando la suma de los efectos de las variaciones para un número finito de tiempos de saturación (k), la reducción total del costo será

$$\Delta \mathcal{J}_T = \lambda(t_f)' \sum_{i=1}^k \tilde{\Phi}(\tau'_i, \tau_i) \tilde{B}(\tau'_i, \tau_i) \Delta u_i \Delta t_i^j. \quad (4.45)$$

Cualquiera o ambos criterios (i) y (ii) pueden ser calculados y comparados con la fórmula (4.32), y entonces decidir si es necesario abandonar el procedimiento regular. Cuando el orden de magnitud de cualquiera de los dos criterios es el mismo que la reducción total del costo (4.32), entonces nuestra estrategia de calculo decide abandonar la reducción de costo regular (a través del método de costo diferencial (4.30)), ya que probablemente se producirá un error en una de las siguientes iteraciones. En general para pequeñas variaciones del control los dos criterios arrojan resultados similares, entonces se puede decir en adelante

$$\Delta \mathcal{J}_A \triangleq \Delta \mathcal{J}_Z \approx \Delta \mathcal{J}_T, \quad (4.46)$$

4. TRATAMIENTO NO LINEAL

donde $\Delta\mathcal{J}_A$ permite verificar la condición de alerta (4.36), como se había anunciado.

4.2.2 Cómo proceder cuando el tratamiento regular está próximo a fallar

Varios tipos de control pueden ser usados para forzar reducciones de costo adicionales después que el procedimiento regular es abandonado. Las variaciones del tipo de una estrategia simple (subsección 4.1.3) pueden ser utilizadas para complementar las del procedimiento regular. Los valores del estado final para resolver la ecuación del error (4.8) también pueden adoptarse como $X(t_f) = \pm v$, para algún pequeño valor $v > 0$, no siendo necesario en este caso realizar la búsqueda del problema fantasma, sino sólo garantizando de acuerdo al valor de v que las correcciones o variaciones de control realizadas mantengan las nuevas trayectorias de control y estados cerca de las trayectorias semillas.

4.2.2.1 Otros tipos de variaciones de control

En la iteración j un número M_j de puntos de saturación pueden aparecer en la estrategia de control. En esta subsección, el intervalo entre cada par k ($k = 1, 2, \dots, M_j/2$) de subsecuentes puntos de saturación puede ser denotado como (τ_{1k}, τ_{2k}) . Las siguientes curvas de tercer orden fueron propuestas como alternativas de variaciones de control:

$$\mu_{jk}(t) \triangleq \begin{cases} 0 & \text{si } t \leq \tau_{1k} - \delta_1 \\ 0 & \text{si } t \geq \tau_{2k} + \delta_2 \\ (t - \tau_{1k} + \delta_1)(\tau^* - t)(t - \tau_{2k} - \delta_2)\varepsilon_\tau & \text{si no,} \end{cases}, k = 1, 2, \dots, M_j \quad (4.47)$$

donde M_j es el número de tiempos de saturación $\{\tau_{1k}, \tau_{2k}\}$ detectado en la etapa j , y

$$\tau^* = \frac{\tau_{1k} + \tau_{2k} + \delta_2 - \delta_1}{2}, \quad (4.48)$$

con δ_1, δ_2 perturbaciones de los tiempos de saturación τ_{1k} and τ_{2k} respectivamente, y ε_τ es la amplitud de la desviación (ver figuras 4.8, 4.9). El control es actualizado mediante

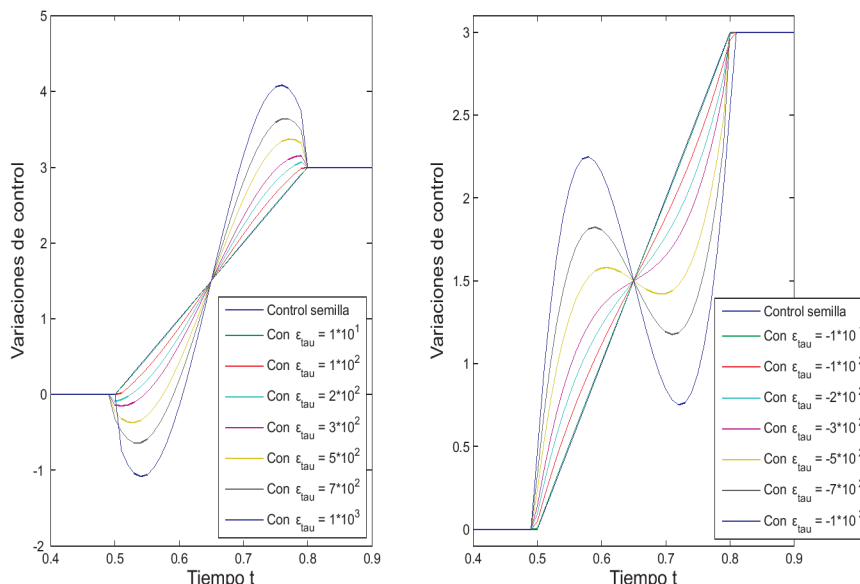


Figura 4.8: Variaciones ϵ_{τ} con los δ_i constantes para otro tipo de variaciones de control. $\delta_1 = \delta_2 = 0,01$

$$\tilde{u}_{j+1}(t) = [\tilde{u}_j(t) + \mu_{jk}]^{sat}, \quad j = 1, 2, \dots \quad (4.49)$$

4.2.3 Aplicaciones numéricas en la estrategia mixta

El caso de estudio analizado en la aplicación mixta es la continuación del estudiado para 4.1.5 con las mismas condiciones iniciales y valores de los parámetros, y después de la transformación en su equivalente, y linealizado aumentado. El tratamiento ‘regular’ para el problema, como se describe en 4.2.1.1, genera:

$$A(t) \triangleq \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0,15\tilde{u}^{sat}(t) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 20\tilde{x}_1(t) & 20\tilde{x}_2(t) & 1 & 0 \end{pmatrix}, \quad B(t) \triangleq \begin{pmatrix} 0 \\ 1 + 0,15\tilde{x}_2(t) \\ 0 \\ \tilde{u}^{sat}(t) \end{pmatrix}, \quad (4.50)$$

$$\lambda(t_f)' = (200\tilde{x}_1(t_f), 200\tilde{x}_2(t_f), -1, 1). \quad (4.51)$$

4. TRATAMIENTO NO LINEAL

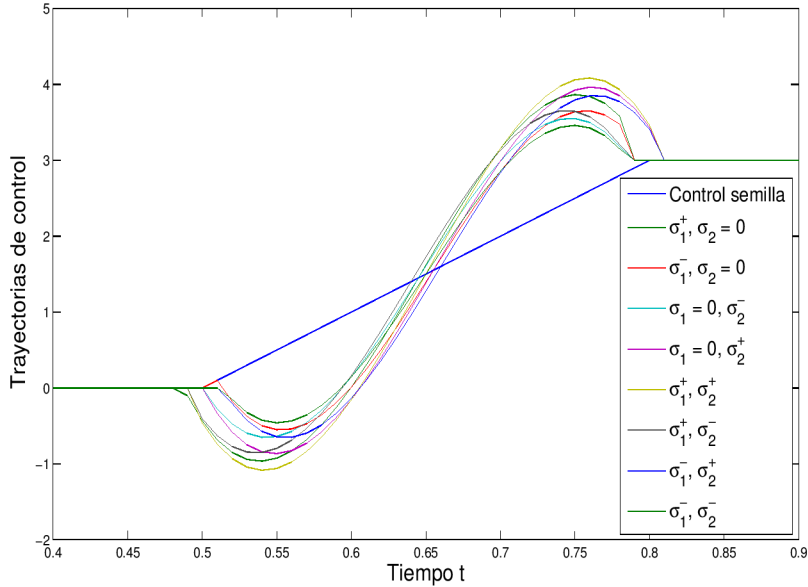


Figura 4.9: Variaciones δ_i con el ε_τ constante para otros tipos de variaciones de control. $\varepsilon_\tau = 1 \times 10^3$

El primer valor de h adoptado en la simulación es $h = 0,01$. Una alerta de fallo se detecta cuando (4.32) alcanza el orden de (4.45). El costo obtenido cuando esto ocurre es $\mathcal{J}_s = 15,4947$. La figura 4.10 muestra una reducción del costo total cuando se utiliza el método ‘regular’ y posteriormente al utilizar el procedimiento numérico.

De la ecuación (4.38), las variaciones de control $u_s(\cdot)$ son definidas como

$$u_s(t) = \begin{cases} w_1 & \forall t \in [\tau'_1, \tau_1) \\ w_2 & \forall t \in [\tau'_2, \tau_2) \\ \tilde{u}^{sat}(t) & \forall t \in [0, \tau'_1) \cup [\tau_1, \tau'_2) \cup [\tau_2, t_f) \end{cases}, \quad (4.52)$$

donde τ'_1, τ'_2 son pequeñas perturbaciones de los tiempos de saturación. La figura 4.11 muestra la evolución de $\Delta\mathcal{J}$ obtenido por el procedimiento formulado en la subsección 4.2.1.2. El resultado de la ecuación (4.32) es comparado con las ecuaciones (4.43, 4.45) para decidir cuándo la condición de alerta (4.36) se activa (esto ocurre alrededor de la iteración 30).

La figura 4.12 muestra variaciones alternativas después de aplicar (4.30). Las

4.2 Estrategia mixta

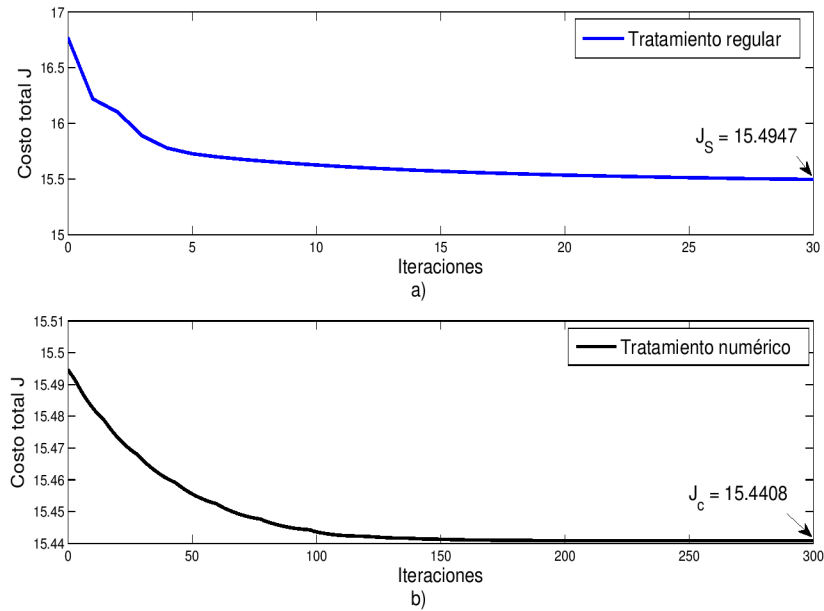


Figura 4.10: a) Evolución del costo total resultante del método regular, b) evolución del costo total resultante del método numérico.

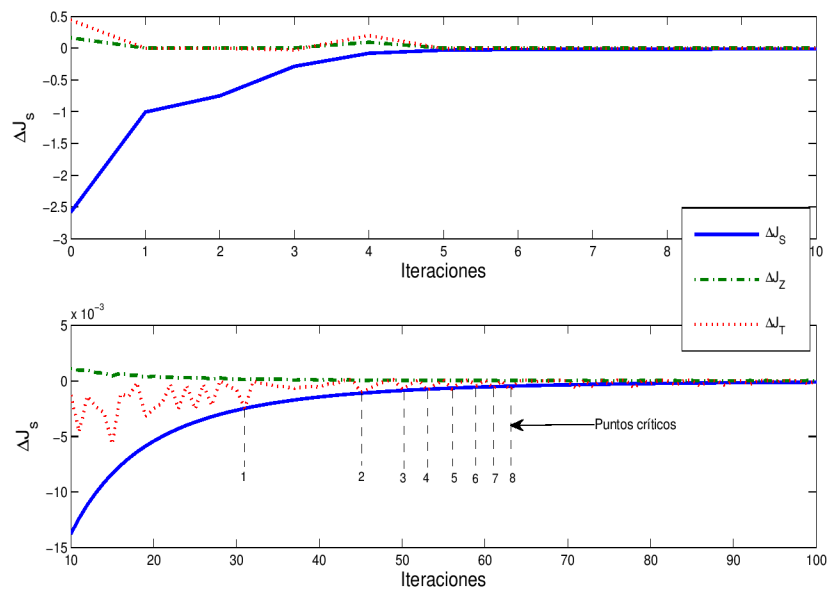


Figura 4.11: ΔJ obtenido de la formulación 4.2.1.2.

4. TRATAMIENTO NO LINEAL

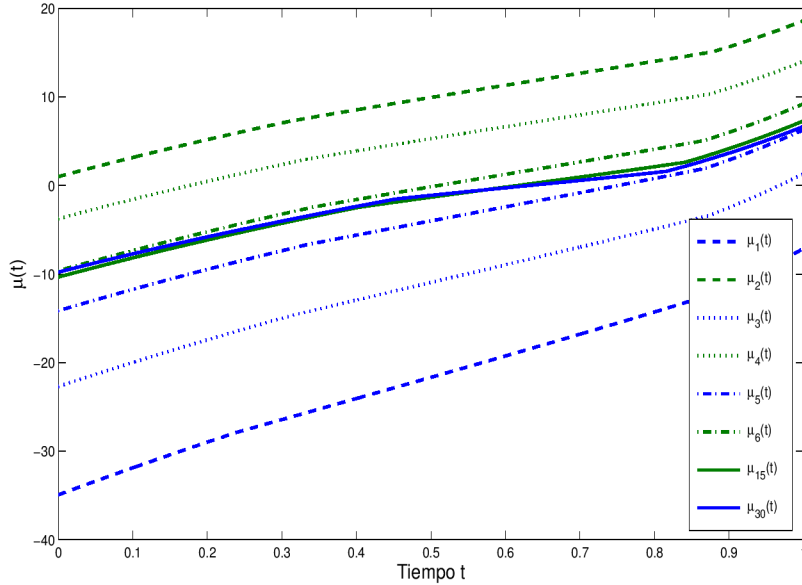


Figura 4.12: Variaciones de control desde la estrategia regular (4.30).

correcciones $\mu(t)$ en el procedimiento regular pueden resultar altas, pero éstas son modificadas con h para mantener $\Delta\mathcal{J}_S < 0$.

Cuando la alerta es activada, las variaciones de control alternativas son utilizadas. En las variaciones de tipo (4.7), la perturbación impuesta al final del estado final es $X_i(t_f) = \pm 0,01$ en todas las direcciones de las componentes i , y para las variaciones del tipo 4.2.2.1 $\varepsilon_\tau = 1$. La figura 4.10 muestra la reducción del costo total hasta obtener un mínimo valor a través de variaciones alternativas. La reducción relativa del costo total después de aplicar las variaciones regulares y alternativas resulta en $\frac{J_{semilla} - J_c}{J_c} * 100\% = 8,67\%$, con J_c como la reducción de costo combinada.

Las figuras 4.13- 4.14 muestran algunas variaciones de control utilizadas en el tratamiento numérico, generadas a partir de la ecuación (4.7). La figura 4.13 representa variaciones obtenidas perturbando los estados finales, y la figura 4.14 las variaciones relacionadas a la ecuación de Ricatti de la linealización. La figura 4.15 muestra los diferentes controles obtenidos a lo largo de los diferentes métodos usados. El hamiltoniano es constante en todo el intervalo de optimización $[0, t_f]$,

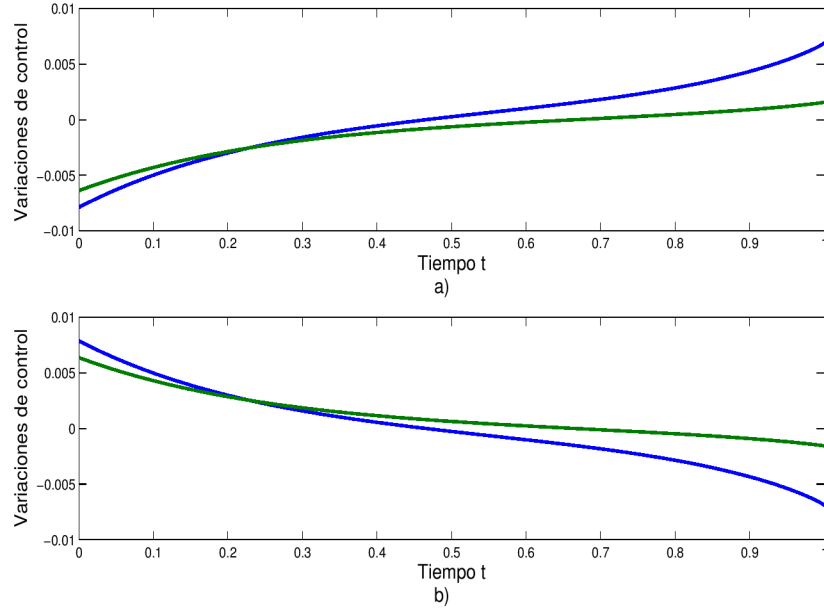


Figura 4.13: Variaciones de control para todas las variaciones de los estados, a) variaciones positivas, b) variaciones negativas.

$\bar{h}_{x_0} \equiv -23,56$, confirmando la optimalidad de la trayectoria de control. En la Tabla 4.2 se exponen otros resultados numéricos relevantes.

El segundo ejemplo en el cual se aplica la estrategia mixta es una modificación del problema previo, donde otro tipo de variaciones se hicieron necesarias. El modelo no lineal bidimensional es modificado agregando un autovalor inestable en la linealización, y diferentes restricciones en los valores de control:

$$\begin{aligned} \dot{x}_1(t) &= 1,5x_1(t) + x_2(t) \\ \dot{x}_2(t) &= \alpha u(t) + \beta u(t)x_2(t) \\ u(t) &\in [-1, 1] \subset \mathbb{R}, \end{aligned} \quad (4.53)$$

Las matrices variantes en el tiempo generadas en este caso son:

$$A(t) \triangleq \begin{pmatrix} 1,5 & 1 & 0 & 0 \\ 0 & 0,15\tilde{u}^{sat}(t) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 20\tilde{x}_1(t) & 20\tilde{x}_2(t) & 1 & 0 \end{pmatrix}, \quad B(t) \triangleq \begin{pmatrix} 0 \\ 1 + 0,15\tilde{x}_2(t) \\ 0 \\ \tilde{u}^{sat}(t) \end{pmatrix}, \quad (4.54)$$

4. TRATAMIENTO NO LINEAL

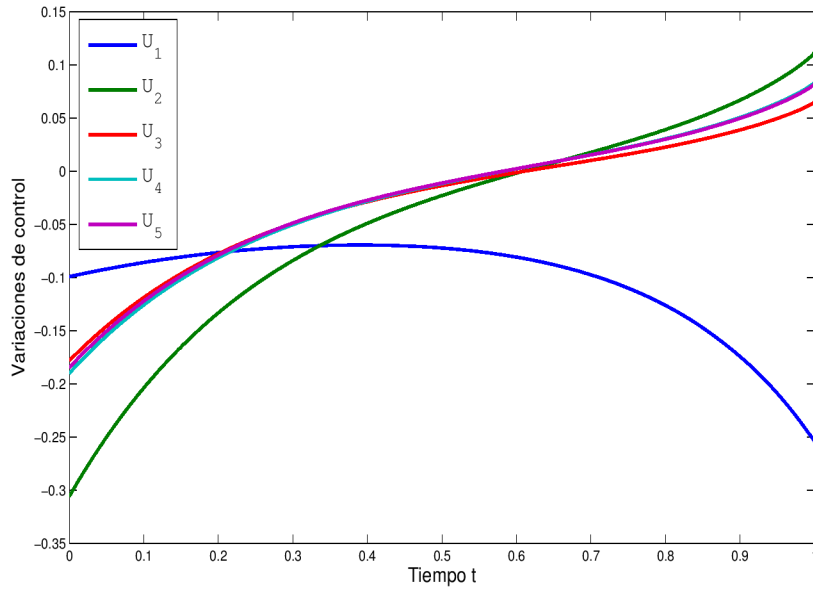


Figura 4.14: Variaciones de control generadas solucionando la RDE

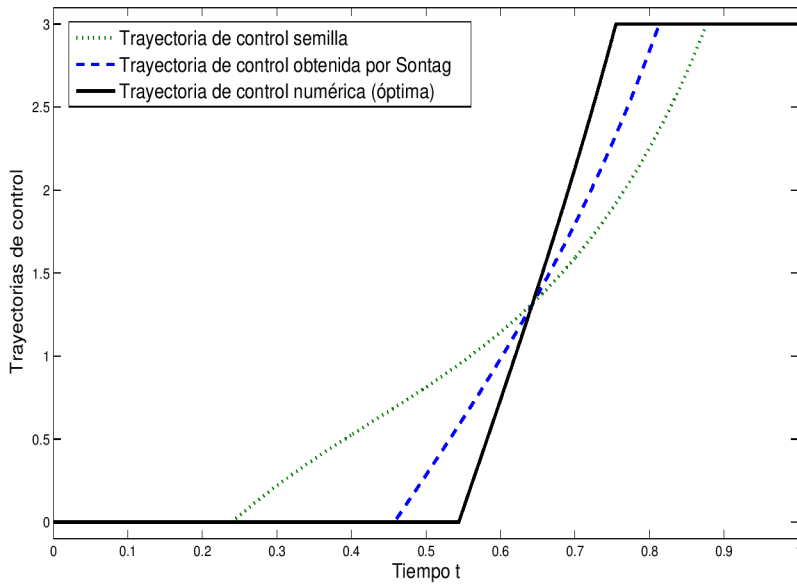


Figura 4.15: Estrategia de control mixta para un ejemplo bidimensional.

Tabla 4.2: Valores numéricos finales estrategia mixta ejemplo 1

	Trayectoria semilla	Mejor trayectoria 'regular'	Mejor trayectoria obtenida de las variaciones alternativas (solución óptima)
$x_1(t_f)$	0,2187	0,17370	0,1657952
$x_2(t_f)$	-0,0137	-0,0485	-0,0450726
τ_1	0,23	0,4570	0,5440
τ_2	0,88	0,8130	0,7549
\mathcal{J}	16,779	15,4947	15,4408

Las condiciones iniciales en este caso son elegidas como $x_1(0) = 1$ y $x_2(0) = -0,5$. Los otros parámetros permanecen iguales. Cuando la alerta de falla es detectada por primera vez el costo es $\mathcal{J}_s = 803,8446$. La figura 4.16 muestra la reducción del costo total para el procedimiento regular. El comportamiento del procedimiento regular, si se mantiene después de que alguna condición de alerta (4.36) se active, es mostrado en la figura 4.17. Variaciones de control inaceptables producirán aumento de costo total después de algunas iteraciones, confirmando que el método regular debe ser abandonado. Entonces, variaciones de control alternativas fueron utilizadas. En variaciones del tipo (4.7), la perturbación impuesta en el estado final es $X_i(t_f) = \pm 0,01$ en todas las direcciones de las componentes i , y para las variaciones descritas en la subsección 4.2.2.1, $\varepsilon_\tau = 1 \times 10^6$ y $\delta_1 = \delta_2 = \pm 0,01$. La parte inferior de la figura 4.17 muestra reducciones de costo total hasta alcanzar un mínimo valor a través de variaciones alternativas. La reducción relativa del costo total después de aplicar las variaciones para el procedimiento regular y las alternativas resulta en $\frac{J_{semilla} - J_c}{J_c} * 100 \% = 80,7 \%$.

Los resultados de este ejemplo preservan algunas características de los problemas previos: el mismo número de tiempos de saturación y un solo período regular. Las variaciones de control asociadas con la ecuación (4.52) generan reducciones de costo total $\Delta \mathcal{J}_S$ (4.32) significativamente mayores que los errores $\Delta \mathcal{J}_Z$ (4.43); esto es aceptable durante las primeras iteraciones. Alrededor de la iteración 40 se consideró que el valor absoluto de $\Delta \mathcal{J}_S$ y $\Delta \mathcal{J}_Z$ estaban demasiado cerca (ver figura 4.18) como para activar la condición de alerta de fallo. La figura 4.19 muestra

4. TRATAMIENTO NO LINEAL

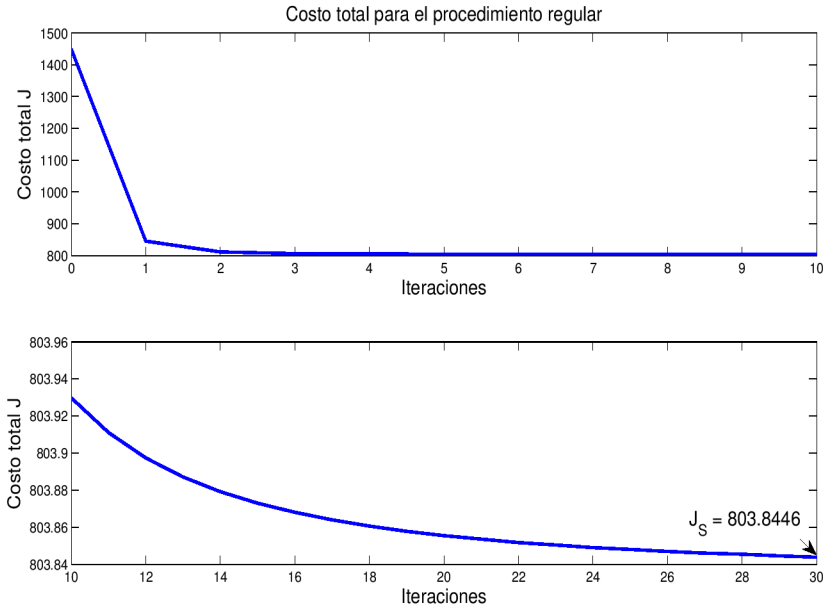


Figura 4.16: Evolución del costo total resultante del método regular.

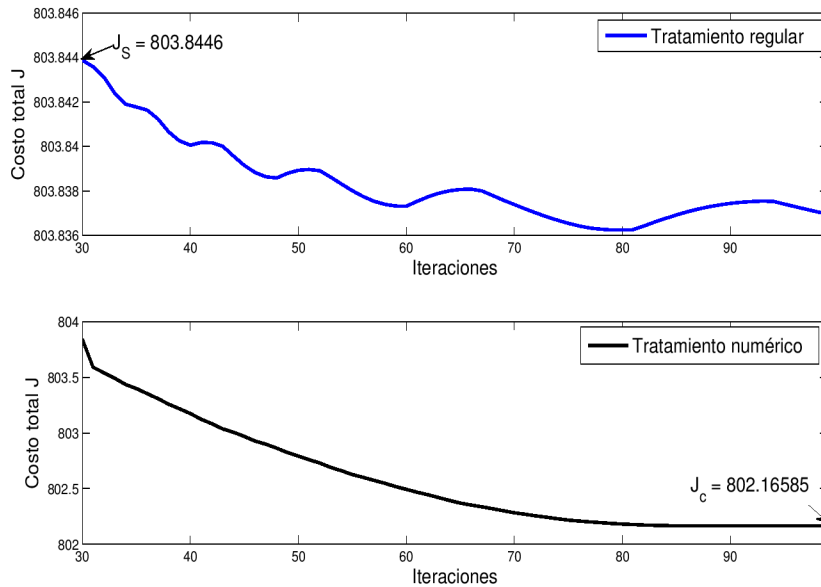


Figura 4.17: Falla del método, y evolución del costo total resultante del método numérico.

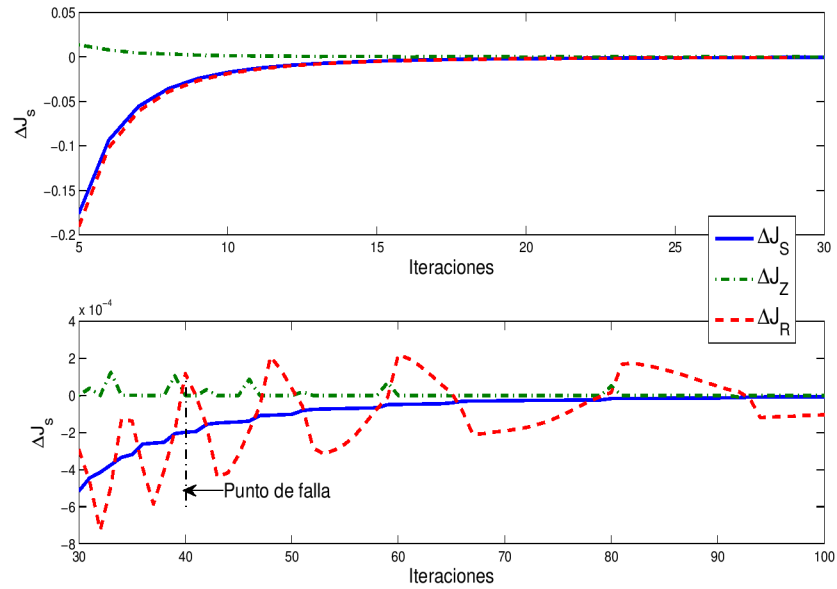


Figura 4.18: ΔJ obtenidos de la formulación de las ecuaciones (4.32, 4.43)

los diferentes controles obtenidos al usar los diferentes métodos. El hamiltoniano es constante en todo el intervalo de optimización $[0, t_f]$, $\bar{h}_{x_0} \equiv -1438$, confirmando la optimalidad de la trayectoria de control. En la Tabla 4.3 se exponen otros resultados numéricos relevantes.

4. TRATAMIENTO NO LINEAL

Tabla 4.3: Valores numéricos finales estrategia mixta ejemplo 2

	Trayectoria semilla	Mejor trayectoria 'regular'	Mejor trayectoria obtenida de las variaciones alternativas (solución óptima)
$x_1(t_f)$	3,7361	2,6442	2,617883
$x_2(t_f)$	0,2661	-0,8175	-0,887606
τ_1	0,035	0,676	0,715
τ_2	0,204	0,6850	0,718
\mathcal{J}	1449,228	803,8445	802,1658

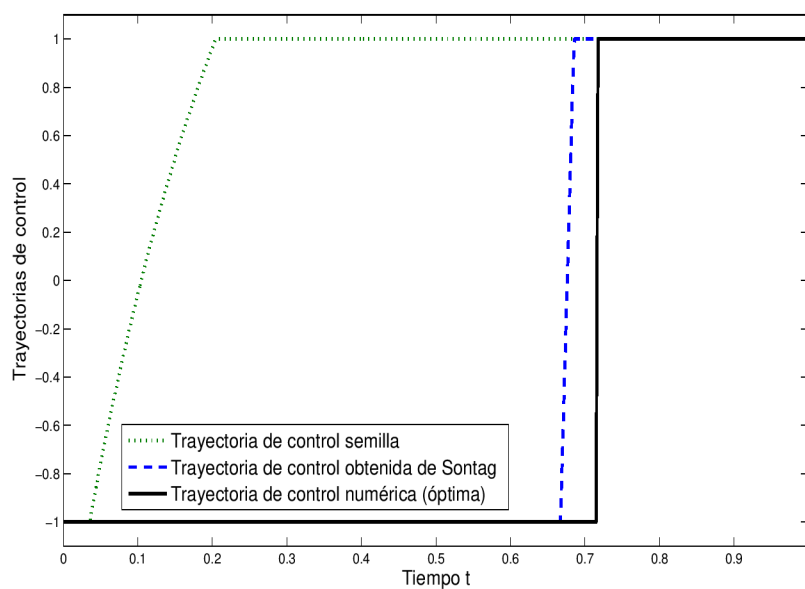


Figura 4.19: Estrategia de control mixta para el ejemplo bidimensional modificado.

4.3 Discusión, conclusiones y perspectivas

Las leyes de control de tipo feedback son las deseadas en las aplicaciones industriales cuando se espera que aparezcan perturbaciones, pero para sistemas no lineales con criterios de rendimiento (índice de desempeño) y restricciones en las variable manipulada, las realimentaciones son difícilmente instrumentables. Con estas limitaciones en mente, un par de algoritmos eficientes se han ideado para aproximar el control óptimo en lazo abierto, basado en resultados teóricos. Las estrategias resultantes son bastante diferentes de la saturación del control óptimo correspondiente al problema irrestricto con los mismos parámetros y condición inicial. La saturación del control óptimo irrestricto es inicialmente usada como una primera aproximación, la cual es denominada como el ‘semilla’.

4.3.1 Estrategia simple

Para la estrategia simple, tomando como base un resultado teórico reciente, se desarrolló un método computacional para calcular una aproximación numérica de la solución óptima del problema de control con restricciones para sistemas no lineales, más eficiente que lo presentado en [46] y tendiendo a que la solución pueda ser sintetizable en tiempo real.

Para el problema linealizado se encontró una solución sub-óptima cercana a la solución óptima encontrada por medio del método offline para el sistema no lineal. Esto nos indica que trabajar con la linealización del problema original e ir haciendo pequeñas modificaciones con respecto al semilla resulta apropiado. El tratamiento realizado al nuevo problema permite también trabajar con estrategias de control en lazo cerrado, por lo que se puede tener una mejor respuesta ante posibles perturbaciones del sistema; esto debido a que la ley de control se construye con la ecuación diferencial de Riccati. La linealización permitió trabajar también con ODEs para la solución de manera iterativa del problema, evitando así la solución de la PDE de HJB.

En esta estrategia simple la condición final para la ecuación de Riccati depende de los ρ , μ que se van generando, y es independiente de la penalización final $\mathcal{K}(x)$ del problema original. Esta última sólo interviene en el cómputo del costo J y de su gradiente.

4. TRATAMIENTO NO LINEAL

Para problemas de mayor dimensión, es necesario tener en cuenta aspectos relacionados a la velocidad de cómputo para el método propuesto, observar cambios en la constante de tiempo de respuesta del sistema, y finalmente analizar los efectos de la estabilidad interna respecto al método. La velocidad de cálculo es probablemente uno de los principales temas de preocupación en el mundo de control computacional, principalmente cuando se pretende implementarlos en tiempo real. La respuesta para obtener mayor velocidad de cálculo en problemas de grandes dimensiones también puede caer en un ámbito de computación en paralelo [64], siendo ésta una herramienta que permite distribuir el cálculo del gradiente en varios procesadores y hace más competitiva la técnica presentada a través de la estrategia descrita.

4.3.2 Estrategia mixta o combinada

La estrategia de control mixta permite trabajar con una trayectoria de control semilla diferente a la restricción del control irrestricto, ya que el método no depende estrictamente de la aproximación de la trayectoria inicial (semilla); entonces es posible en general evitar resolver el problema no lineal irrestricto. El método procede entonces recursivamente, basado en la linealización de las dinámicas alrededor de las últimas trayectorias aproximadas y una fórmula ideada para la reducción del costo en el contexto regular [134] resulta aquí también útil. Dado que el problema de estudio es generalmente no regular, el procedimiento iterativo puede eventualmente fallar, esto es que el costo cesa de decrecer, o éste entra en una zona peligrosa donde el error de saturación es del mismo orden que la reducción del costo alcanzada. Entonces una segunda serie de iteraciones son utilizadas, mediante la introducción de otras clases de variaciones de control: soluciones de las RDE para la dinámica linealizada sometida a un nuevo objetivo de costo (tal como en la estrategia simple), variaciones cúbicas de arcos regulares, y variaciones espaciales del tipo de Pontryagin con perturbaciones en los tiempos de saturación.

En general, los controles resultantes después de estas iteraciones resultan subóptimos, aunque en los casos estudiados con la estrategia mixta los valores óptimos son alcanzados, en el primero de los ejemplos numéricos tratados, la reducción del costo relativa al costo de la trayectoria semilla fue $\frac{J_{semilla} - J_c}{J_n} * 100\% \approx 9\%$, y en el

4.3 Discusión, conclusiones y perspectivas

segundo de ellos fue mayor al 80 %, mostrando un comportamiento impredecible de las trayectorias provenientes de las soluciones regulares cuando éstos se saturan (las correcciones son del mismo orden del error). En ambos casos las soluciones óptimas fueron alcanzadas.

La estabilidad del método es garantizada puesto que (i) el procedimiento regular se detiene después de detectar una condición de alerta, (ii) a la aproximación del costo diferencial ΔJ_S no se le permite que cambie de signo, y (iii) el costo total proveniente del control previamente restringido es verificado mediante la integración de la dinámica cuando se actualiza las trayectorias semilla.

Nuestro conocimiento es necesariamente finito, mientras que nuestra ignorancia es necesariamente infinita.

Karl Popper

CAPÍTULO

5

Computación en paralelo para control óptimo basado en modelos

El aumento considerable de cálculo en problemas de control óptimo ha desbordado la capacidad de cómputo disponible para manejar sistemas complejos en tiempo real. Por tal razón, alternativas como la de computación en paralelo son estudiadas en este capítulo, donde se trabaja el problema distribuyendo las tareas entre varios procesadores, en algunos casos sin la necesidad de realizar comunicaciones intermedias. Se pretende explorar el uso de estos métodos en algunos casos de estudio, analizar e investigar si existe reducción de tiempo total de cálculo, y también proponer cómo implementar soluciones en tiempo real, donde se necesitan tomar decisiones en intervalos de tiempo de muestreo establecidos. Se procura ir aumentando paulatinamente el orden de los problemas estudiados para comparar las mejoras alcanzadas. Se estudian alternativas de cálculo en paralelo tales como el paquete especializado de MATLAB, programación de un cluster Beowulf, y también a través de múltiples elementos de procesamiento dentro de una sola máquina, con lenguajes de nivel medio y con rutinas apropiadas.

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

5.1 Computación en paralelo

La computación en paralelo involucra múltiples procesadores (núcleos) o computadoras que trabajan como grupo para la realización de tareas. Cada núcleo puede trabajar en una parte del problema y compartir información con los demás procesadores a intervalos variables. En general es aplicada para resolver problemas que involucran gran cantidad de tiempo para obtener su solución, algunos relacionados con CFD (sigla del inglés ‘Computational Fluid Dynamics’), predicción del clima, modelado de materiales sólidos, dinámicas y mecanismos de estructuras, fundición de metal, etc.. En [135] se estudian aplicaciones de FEM (siglas del inglés ‘Finite Element Method’) orientado a problemas CFD corriendo en un cluster Beowulf. Las arquitecturas paralelas han sido clasificadas [74] en microcomputadoras, minicomputadores, mainframe y supercomputador. Por otro lado [139] exponen la clasificación de Flynn’s, u otras basadas en arreglo de memorias y comunicación entre PE (del inglés ‘Processing Elements’), o en las conexiones de PE y módulos de memoria, o en características naturales de PE, y por último, algunos tipos específicos de arquitectura paralela. Aquellas basadas en arreglo de memorias pueden clasificarse en: (i) procesadores con memoria compartida (ver figura 5.1a), o (ii) procesadores con memoria distribuida o local (ver figura 5.1b).

En procesadores de memoria compartida en los que se dispone de varios procesadores individuales y una memoria común, cada uno de ellos tiene acceso a cualquier variable almacenada en la memoria común, es decir, todos los procesadores utilizan el mismo espacio de direccionamiento: éste es el caso de computadoras de granulado grueso como Cray YMP, Cray C90, etc. En los procesadores distribuidos o de memoria local, en cambio, cada procesador dispone de su propia memoria privada, y en este caso el espacio de direcciones no es común: todas las direcciones son locales y hacen referencias al espacio propio del procesador. Este caso es característico de memorias de granulado fino como los hipercubos, mallas de 2D y 3D, los árboles y los clusters [138].

La manera de programar en computadoras paralelas depende altamente de la manera en que los procesadores acceden a la memoria. El paso de mensajes es utilizado ampliamente en computadoras paralelas escalables con memoria distribuida,

a través de la red de comunicación. Aunque existen algunas variantes, el concepto de comunicación de procesos mediante mensajes es la base de todas ellas.

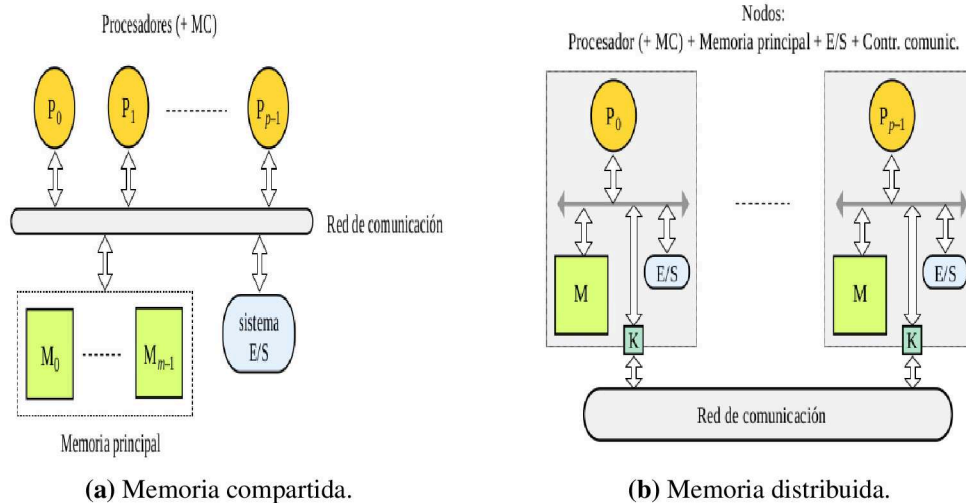


Figura 5.1: Clasificación basada en arreglo de memorias y comunicación entre PE [7].

5.1.1 Aspectos prácticos

En el uso de algoritmos en paralelo juegan un rol importante no sólo la arquitectura de hardware y la capacidad de procesamiento, sino también la habilidad del programador, que será el encargado de realizar el análisis de dependencia y la distribución de la carga de cómputo en los elementos de procesamiento, utilizando para esto distribución de carga estática o dinámica de acuerdo a la necesidad de implementación en tiempo real, así como de la capacidad de comunicación entre los procesos. La posibilidad de paralelismo en una aplicación aparece al detectar cálculos que se puedan ejecutar simultáneamente. La existencia de interdependencia entre dos cálculos a realizar simultáneamente implica que éstos no son muy apropiados para desarrollarse de manera paralela: en general, a menor dependencia mayor posibilidad de paralelismo. La presencia de ecuaciones acopladas indica una mayor complejidad del algoritmo a paralelizar [78], es por eso que la mayoría de las aplicaciones para sistemas computacionales no han sido usualmente diseñadas para procesamiento en paralelo.

Los lenguajes de programación disponibles para el cálculo en paralelo se basan en lenguajes secuenciales estándares tales como Fortran, C, o C++. Para memo-

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

ria compartida, se utiliza OpenMP, POSIX, X3H5, PCF, threads o SysV [25], que permiten mayor control en las zonas de ejecución en paralelo para el programa. En el caso de memoria distribuida, para acceder a un dato de otra memoria es necesario el modelo de paso de mensajes, y para ello se utiliza Express, nCUBE's Vertex, p4, PARMACS, Zipcode, Chimp, PVM, Chameleon, PICL, MPI, que permiten más escalabilidad [60, 133]. A continuación discutimos algunos aspectos de estos lenguajes y programas.

5.1.1.1 MATLAB

MATLAB se ha constituido en una herramienta ampliamente utilizada para el desarrollo y simulación de algoritmos en cálculo científico debido a su entorno interactivo y a la gran cantidad de rutinas de que dispone, como aquéllas utilizadas para realizar operaciones matriciales, las cuales han sido optimizadas de tal manera que igualan en prestaciones a otras cuidadosamente elaboradas en lenguajes de nivel medio (Fortran, C++ y C). MATLAB es un lenguaje de programación de alto nivel, cuya posibilidad de visualización y mayor facilidad de programación, en comparación con los lenguajes de nivel medio, lo hacen muy atractivo y una excelente plataforma para el desarrollo de computación en paralelo.

Dentro de MATLAB se han incluido diversos toolbox (paquetes especializados), en los cuales varían los mecanismos de comunicación (ver Tabla 1.1 de [60]). El Lincoln Laboratory del MIT desarrolló las librerías `MatlabMPI` y `pMatlab`, que permiten programación en paralelo en MATLAB con la lógica de las librerías hechas para los lenguajes de nivel medio pero con las prestaciones de MATLAB.

Para las versiones de MATLAB 2007b y posteriores, es posible trabajar con 'Parallel Computing Toolbox (PCT)', extensión de MATLAB que aprovecha las máquinas multinúcleo y los clusters. Éste puede correr de manera local en un computador multinúcleo, hasta con 12 núcleos en MATLAB2011b o versiones más reciente (8 núcleos en versiones más viejas). El aumento en velocidad dependerá del número de núcleos físicos. En MATLAB se toma el término 'worker' como equivalente a un procesador. Para poder ser usado en un cluster es necesario contar con el paquete 'MATLAB Distributed Computing Server (MDCS)' el cual se encarga de controlar la ejecución en paralelo.

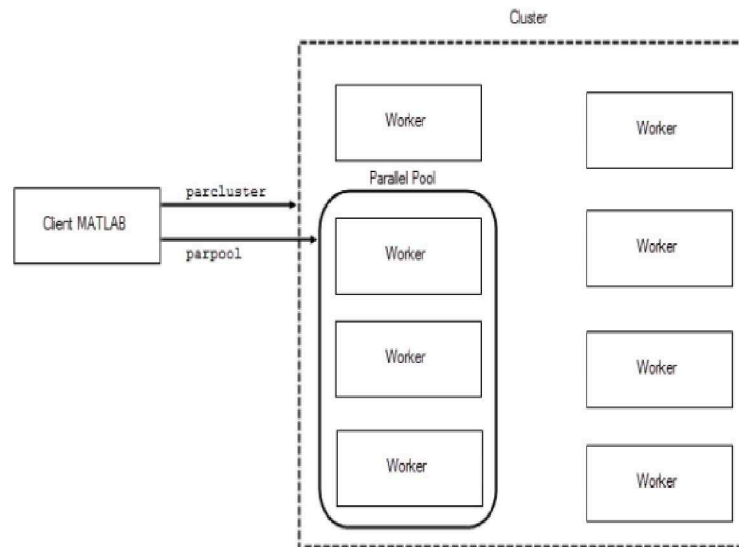


Figura 5.2: Sesión interactiva `parpool` o `parcluster` [108].

Se utilizan también funciones simplificadas del modelo MPI ('Message Passing Interface', Interfaz de Paso de Mensajes) en PCT, cuando se requiere un alto grado de control sobre el algoritmo paralelo, ya que permiten comunicación entre los workers [149]. Cada worker trabaja con su propia memoria y se le asigna un ID (identificador) o número de referencia.

Las aplicaciones se pueden ejecutar de forma interactiva o de manera offline, en un entorno de procesos por lotes. Para trabajar con un entorno de procesos por lotes, es necesario cargar el programa completo en un código fuente haciendo uso de los comando necesarios, siendo indispensable crear un archivo de ejecución por lotes (.cmd o .bat), en el que se especifica el número de procesadores a utilizar; por el contrario en las sesiones interactivas ofrecidas por MATLAB y utilizadas también para realizar la programación paralela, se puede proceder a través de `parpool` o `parcluster` (ver figura 5.2) y `pmode start` (ver figura 5.3). Las aplicaciones después de ser programadas pueden ser ejecutadas en procesadores multinúcleos (localmente) o en un cluster de computadoras según la configuración sin necesidad de hacer cambios en los códigos.

Algunas de las herramientas utilizadas para la programación en paralelo con MATLAB se pueden observar en [108]:

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

- `Parfor`: Es una versión paralela del `for`. Para poder paralelizar el `for` se debe garantizar en el código que se realicen iteraciones independientes, de tal manera que se distribuye automáticamente el trabajo sobre los workers configurados, aunque las iteraciones no son evaluados en ningún orden en particular.
- `SPMD` (del inglés ‘Simple Program Multiple Date’): Es un tipo de arquitectura específica que combina la facilidad de la programación SIMD (‘Single Instruction Multiple Data’) con la flexibilidad de MIMD (‘Multiple Instruction Multiple Data’). `SPMD` permite intercambio continuo de programación en serie y en paralelo. El sistema es controlado por un programa y se ejecuta un código idéntico en los diferentes workers, aunque cada uno puede tener diferentes datos.
- `Pmode`: Se ejecuta en caso de sesiones interactivas y en contraste con `spmd` provee una interfaz gráfica. Esta herramienta es utilizada con sesiones locales, funciona de la misma manera que `spmd` con instrucciones que se ejecutan paralelamente en distintos workers; la sesión inicia con el comando `pmode`. Cuando se cierra la sesión `pmode` se eliminan todos los datos y las variables almacenadas en los workers.

Para algunas pruebas realizadas durante el trabajo de tesis se ha utilizado una versión de `MATLAB 2014a` con disponibilidad de 8 workers locales. Aunque las simulaciones ejecutadas utilizaron los 8 workers, es necesario tener en cuenta que la velocidad está limitada por los núcleos físicos. En este caso fueron programadas en un procesador multinúcleo con 4 núcleos físicos.

En la figura 5.3 se observa una sección `pmode` con tres workers. El comando `pmodebidimensional` corresponde al nombre del código realizado para trabajar el primer caso de estudio (ver B.1.1).

La versatilidad aportada por `MATLAB` para trabajar problemas de carácter numérico lo convierten en una herramienta ampliamente utilizada en el mundo académico. Sin embargo hasta la inclusión del `PCT` para las versiones posteriores a 2007b no era posible el uso de cálculo paralelo, a menos que se le agregaran las librerías. La estandarización en el `MPI Forum` (1993 – 1994) no consideraba el uso de `MATLAB`

5.1 Computación en paralelo

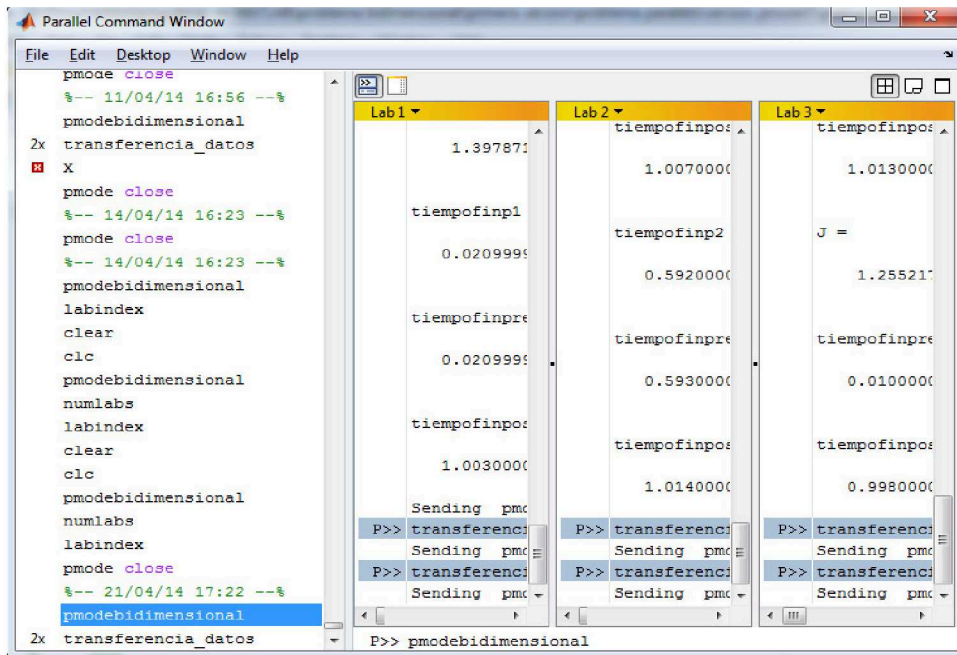


Figura 5.3: pmode con 3 workers.

para aplicaciones paralelas por lo que se generó un desarrollo y adaptación de gran cantidad de librerías usadas en programación tradicional.

5.1.1.2 Lenguaje de nivel medio: Uso de un cluster

Buscando alternativas para aumentar de una manera significativa el cálculo dentro de cada ‘Sampling Time’, se llega de manera racional a usar un cluster. Entendiendo el cluster como un conjunto de computadoras independientes (cada una con procesador y memoria) que operan mediante la interconexión de una red de área local de alta velocidad. La configuración más popular de los cluster consiste en un conjunto de nodos del tipo GNU/Linux OS y un software de código abierto para la implementación por medio de paso de mensajes. La configuración descrita es conocida como Beowulf y fue diseñada para la ejecución de programas que explotan explícitamente las ventajas del paralelismo. Se obtienen ventajas debido a que los nodos no necesitan un disco duro y por la facilidad de adherir nuevos nodos al cluster [135].

Aprovechando la existencia de un cluster en las instalaciones del “CIMEC”, en

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

el parque tecnológico CONICET de la ciudad de Santa Fe, se ha optado por abordar el problema con un lenguaje de nivel medio compatible. El cluster utilizado inicialmente es conocido como Aquiles [26]. También es posible realizar programas con memoria compartida a través de OpenMP, y correrlos por medio de otros procesadores disponibles también en dicho instituto.

Para las primeras pruebas realizadas, la programación se desarrolló en C++ con funciones de MPI, una configuración estándar (MPI Forum) creada debido al crecimiento de aplicaciones paralelas y a la diversidad de soluciones comerciales, que comprometían portabilidad, performance y prestaciones. Su mayor ventaja es la portabilidad en procesadores de cualquier tipo, lo cual representa la posibilidad de ejecución de manera transparente en sistemas heterogéneos, es decir, en conjuntos de procesadores de diferentes arquitecturas. MPI es entonces una librería que contiene funciones que definen la sintaxis y semántica de paso de mensajes para ser usado en múltiples procesadores [133].

Los programas que se usan con MPI son del tipo SPMD, diseñado especialmente para aplicaciones de este patrón, que realiza automáticamente cualquier conversión de datos necesaria y utiliza el protocolo de comunicación correcto. Al arrancar una aplicación se lanzan en paralelo N copias (procesos) del mismo programa. Éstas no están sincronizadas instrucción a instrucción, por lo que la sincronización se debe hacer por parte del programador, ya que los procesos tienen un espacio en memoria completamente separado. El mecanismo básico de comunicación MPI gobierna la transmisión de datos entre dos pares: un procesador envía y el otro recibe los datos. Existen diversas formas de intercambiar mensajes entre dos procesos, en función del modelo y del modo de comunicación [133].

La solución numérica de las ecuaciones diferenciales involucradas en el modelo trabajado en la programación del cluster con C++ fue llevada a cabo por medio de `Odeint`, es una librería moderna optimizada para dicha tarea [4].

5.2 Computación en paralelo aplicada a problemas de control

La aplicación práctica del control óptimo en los últimos años se ha visto restringida por la capacidad computacional, debido en gran parte a los límites físicos presentes, tales como el ciclo de reloj interno de las computadoras, el cual afecta la velocidad de transmisión entre procesador y memoria. Para ciclos de reloj altos, se producen también problemas de disipación de calor. Por tal razón se han buscado alternativas que permitan atacar problemas de control con gran carga de cómputo en tiempo real [13, 113, 138].

En [130], se muestra una configuración para la evaluación paralela del problema de control óptimo usando un método iterativo de programación dinámica, donde el problema total es reducido a la optimización de subproblemas. Aunque el método no menciona directamente el uso de computación en paralelo, sí realiza una separación que permite realizar cálculos en paralelo de los subsistemas para conformar el problema total. Mas recientemente en [75, 119, 120] se utiliza computación en paralelo para el óptimo de sistemas dinámicos lineales, basado en funciones de Lyapunov y en la ecuación matricial de Riccati. Se reportan también soluciones obtenidas por medio del uso de GPU ('Graphics Processor Unit') como en [16, 17], o través de la reducción de modelos (un modelo considerablemente menor que presenta un comportamiento similar al del modelo original) (ver [59]).

En los problemas de optimización en tiempo real se utilizan modelos 'on-line', a partir de los que se calculan los estados del sistema real y se realizan los cálculos de predicción. Los modelos on-line están relacionados con los observadores, que utilizan la salida medida para predecir los estados. No discutiremos observadores aquí.

Los sistemas de tiempo real deben responder y tomar decisiones frente a estímulos externos dentro de períodos de tiempo acotados. Los tiempos de muestreo se hacen más cortos cuanto mayores son las exigencias de rendimiento impuestas sobre el sistema, y los algoritmos se hacen más complejos con el desarrollo de la teoría de control [140].

En este capítulo aparece como ejemplo inicial un problema LQR. El objetivo principal es explorar si existen ventajas en distribuir la simulación del modelo del

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

sistema y el cálculo del control en procesadores independientes. En el caso-estudio relacionado a esta primera exploración se utilizaron inicialmente dos procesadores, uno que emula el modelo y otro que se encarga de calcular el control en tiempo real, para lo que se emplearon un cluster y MATLAB. En MATLAB se simuló un tercer procesador para la obtención del control óptimo con actualización de parámetros en tiempo real. Posteriormente se graficaron los controles obtenidos por los dos sistemas encargados de calcularlos (el cluster y MATLAB) para observar las diferencias.

5.2.1 Primer acercamiento a una aplicación de control óptimo

Se analiza un problema con una dinámica lineal y un funcional de costo cuadrático (LQR) con un tiempo final t_f (coincidente en este caso con el horizonte, o duración del período de optimización). Para este primer ejemplo se hace uso de dos prácticas herramientas mediante las cuales es posible realizar cómputo en paralelo. Una de ellas corresponde a MATLAB, configurado para trabajar de manera local en una sola máquina multinúcleo. La otra corresponde a un cluster de computadoras programado a través de C++ mediante rutinas MPI en el que se utilizan dos procesadores (ver apéndice B.1.2).

5.2.1.1 Caso de estudio en el frenado óptimo de un tren de la sección 3.4.1

El caso de estudio corresponde al estudiado en la sección 3.4.1, con las mismas condiciones iniciales y parámetros para el funcional del costo. Se calcula la solución de la ecuación (2.35) correspondiente al problema de Riccati para todo el tiempo horizonte tanto en MATLAB como en el cluster. La solución de la ecuación de Riccati se calcula de manera off-line, de tal modo que se encuentre guardada para que pueda ser utilizada en el cálculo de los estados, control, etc.

El tiempo final será en este caso $t_f = 10$. Se realiza una división en intervalos de muestra de duración $\Delta t = 1$, en cada uno de los cuales se calcula la solución de los estados utilizando para esto la solución de Riccati. Además se usa computación en paralelo, con la emulación del sistema en un procesador y la obtención del control en otro.

5.2 Computación en paralelo aplicada a problemas de control

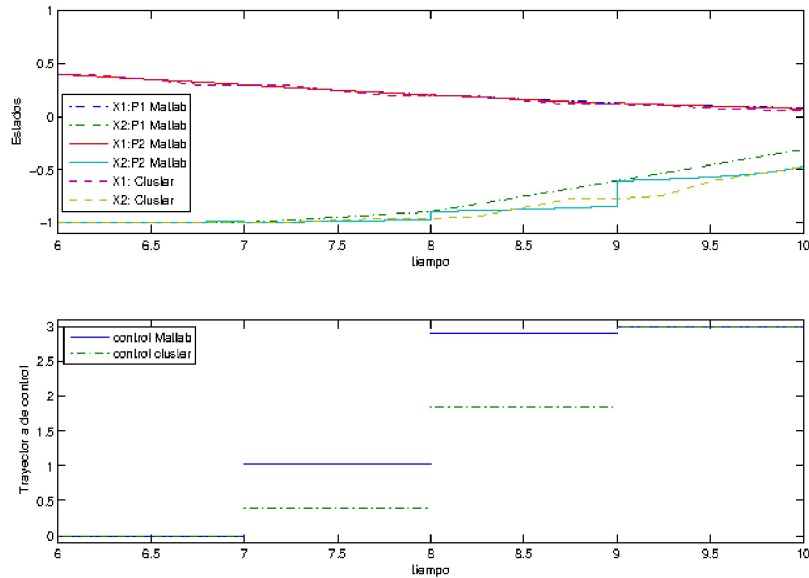


Figura 5.4: En la figura superior se encuentran los estados y en la figura inferior las trayectorias de control. Salidas de MATLAB y el cluster.

La figura 5.4 muestra los estados (velocidad y posición) calculados en MATLAB y en el cluster. Las figuras se muestran desde el intervalo de tiempo 6, donde se pueden observar los mayores cambios que presenta la solución. Adicionalmente en MATLAB se resuelve también la ecuación de la dinámica de estados en un segundo procesador (funcionando de forma similar a un observador), donde se calcula el valor del control para cada uno de los puntos generados por la solución numérica dentro del intervalo de muestra (depende del paso dado para la solución de la ODE), y se decide aplicar el promedio de éstos durante el próximo período en el procesador 1 con el fin de alcanzar mejores soluciones de la dinámica pero conservando aún el poder aplicar controles constantes. El tamaño del vector solución en la ecuación de estados se va aumentando de manera de no sobrepasar el límite de tiempo prefijado para enviar el control hacia el primer procesador. Para esto se define un contador de tiempo (timer) que permita manejar el incremento de la solución.

En la figura 5.4 se observa que para la posición (primer estado) la solución es casi igual en el cluster que en las dos soluciones de MATLAB; para la velocidad (se-

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

gundo estado) se pueden observar algunas diferencias. La solución de la velocidad correspondiente a la trayectoria verde oscura (líneas y puntos) es la que se realiza en MATLAB, la cual emula el modelo. La correspondiente a las predicciones en MATLAB es la que se encuentra de azul claro (línea continua), que coincide en los intervalos de muestra con los estados iniciales del procesador 1, ya que éstos son los utilizados para calcular las predicciones. La solución observada en verde claro (líneas discontinuas) es la calculada a través del cluster. La figura 5.4 muestra en su parte inferior los controles obtenidos en MATLAB (trayectoria azul línea continua) y en el cluster (trayectoria verde líneas y puntos). Los controles tienen, por diseño, la forma de una función constante a trozos que corresponden a el control aplicado en cada intervalo de muestra. Para los estados es imperceptible el salto de los valores de control, ya que la solución de éstos debe permanecer continua a lo largo de toda la trayectoria hasta el tiempo horizonte. Las condiciones iniciales del intervalo de muestra próximo corresponden con las condiciones finales del intervalo de muestra anterior.

En el programa implementado para el cluster fue necesario utilizar funciones de comunicación punto a punto para comunicar los dos procesadores reservados, uno para la emulación del modelo y otro para el cálculo del control. Tanto la emulación del sistema como el cálculo del control se llevaron a cabo en el mismo instante de tiempo. Las diferencias observadas en los controles se deben a que en el segundo procesador del cluster se reciben los valores finales de los estados en el tiempo de muestra y con éstos se calcula el control para el próximo intervalo de tiempo, mientras que en MATLAB se aplica un control refinado, ya que corresponde al promedio obtenido con la predicción del procesador 2. Como se resuelve para todo el intervalo de muestra con las ecuaciones que representan la dinámica del sistema, se obtienen diferentes valores de control a lo largo del intervalo. Al ser un control que crece en el transcurso del tiempo, el promedio obtenido es mayor que el correspondiente al control obtenido con el cluster (se calcula a partir de los estados al inicio del intervalo de tiempo).

Para MATLAB se ha reservado un tercer procesador con el fin de comparar la estrategia de tiempo real con la obtención del control óptimo desde la búsqueda del problema fantasma (ver secciones 3.1 y 3.5.1.2), encargado de actualizar el parámetro S y a su vez de calcular el control mediante el método propuesto en [43]

5.2 Computación en paralelo aplicada a problemas de control

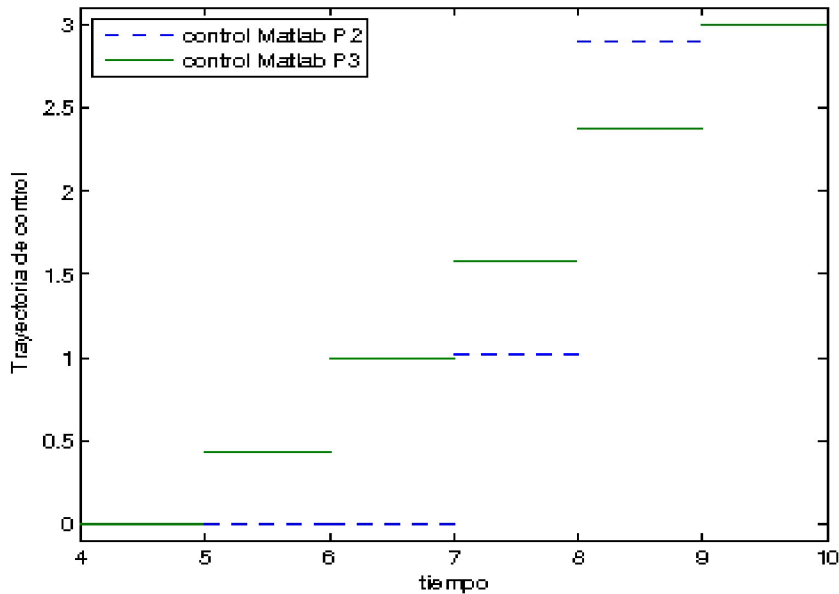


Figura 5.5: Comparación de controles procesador 2:3 con MATLAB.

y en el capítulo 3. Para los dos primeros procesadores se realiza una comunicación recíproca al final de cada intervalo, en el cual se le comunica al procesador 2 los estados finales desde el procesador 1. A su vez desde el procesador 2 se comunica al procesador 1 el valor del control calculado. Para el procesador que está emulando el modelo es imperceptible el proceso de cálculo del control: este procesador sólo necesita al final del intervalo de tiempo contar con el valor del control a utilizar durante el siguiente intervalo.

En la figura 5.5 se pueden ver los dos controles utilizados para calcular las soluciones de la dinámica en los procesadores 1 y 3. El control utilizado en el procesador 1 para calcular su dinámica es el correspondiente al color azul (línea discontinua, mostrado anteriormente). El control del procesador 3 es el correspondiente al color verde y éste cambia como consecuencia de la actualización del parámetro S . La figura 5.5 muestra la visualización de los cálculos de los procesadores 2 y 3. En el procesador 2 se realiza el cálculo por medio de la saturación del control óptimo para el problema irrestricto, y se obtuvo el costo a partir de la ecuación (2.27), resultando en un valor de $\mathcal{J}_2 = 139,7872$. En el procesador 3 se calcula el control luego de actualizar el parámetro S , obteniendo un nuevo valor del

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

costo, menor al anterior $\mathcal{J}_3 = 125,5217$. La reducción del costo fue de 10,2%. El parámetro actualizado corresponde a la penalización final para la solución óptima en el caso de estudio con el control saturado [8, 117].

Se demuestra que para encontrar el problema oculto es esencial el cálculo en paralelo que permita instrumentar estrategias de control óptimo en las cuales están involucradas las restricciones. Para esta parte del análisis del problema completo (sin separarlo en intervalos de tiempo) se busca obtener una reducción del costo hasta el final del tiempo horizonte a través del método del gradiente aplicado a los valores finales del estado ρ y coestado μ del problema fantasma (véase el código en el apéndice B.1.3). La dinámica del problema con restricciones es resuelta inicialmente para conseguir un valor de costo semilla que permita verificar si se presenta disminución de éste a través del método: en este caso el valor obtenido es $\mathcal{J}_{semilla} = 130,1$

Las semillas serán actualizadas por medio de $\tilde{x}(t_f)$ y $\tilde{\lambda}(t_f)$. Estos valores semilla se someten a perturbaciones por medio de las ecuaciones:

$$\begin{aligned}\rho &= \tilde{x}(t_f) \pm h, \\ \mu &= \tilde{\lambda}(t_f) \pm k,\end{aligned}\tag{5.1}$$

con h y k a calibrar.

Se adopta una nueva condición final para el problema fantasma de acuerdo a lo expuesto con la ecuación (3.9) con los valores de estado y coestado actualizados por (5.1).

Se resuelve de nuevo la ecuación (2.35) con esta nueva condición final (3.9) junto a la ecuación de la dinámica, y los valores de control utilizados se almacenan (serían aproximaciones del control fantasma). Los controles almacenados se saturan (serían aproximaciones del control óptimo deseado, en concordancia con el resultado teórico) y se usa como una estrategia en lazo abierto para resolver nuevamente la ecuación de la dinámica y el costo dado por la ecuación (2.27). Se continúan las aproximaciones de manera iterativa hasta alcanzar convergencia en el costo. Los valores $\tilde{x}(t_f)$ y $\tilde{\lambda}(t_f)$ son actualizados para la siguiente iteración a través del método del gradiente de la siguiente manera:

$$\begin{pmatrix} \tilde{x}(t_f) \\ \tilde{\lambda}(t_f) \end{pmatrix}_{j+1} = \begin{pmatrix} \tilde{x}(t_f) \\ \tilde{\lambda}(t_f) \end{pmatrix}_j - \gamma (\nabla \mathcal{J})', j = 1, 2, \dots, 30.\tag{5.2}$$

5.2 Computación en paralelo aplicada a problemas de control

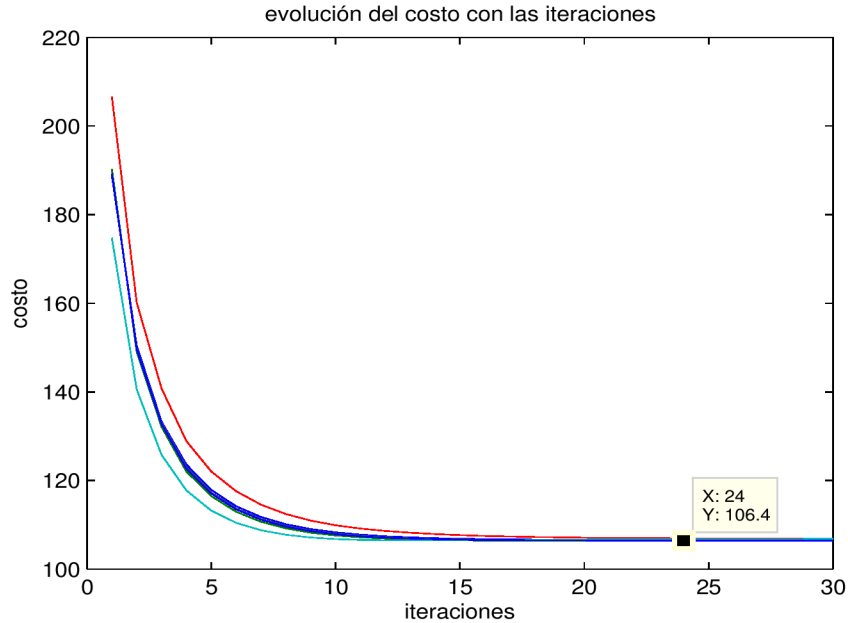


Figura 5.6: Gradiente del costo. Los parámetros de la iteración usados fueron $h = 0,1$, $k = 0,5$, $\gamma = 2 * 10^{-3}$. Corrida con 30 iteraciones.

$$\nabla \mathcal{J} = \left(\frac{\partial \mathcal{J}}{\partial \tilde{x}_1}, \frac{\partial \mathcal{J}}{\partial \tilde{x}_2}, \frac{\partial \mathcal{J}}{\partial \tilde{\lambda}_1}, \frac{\partial \mathcal{J}}{\partial \tilde{\lambda}_2} \right).$$

En la figura 5.6 son ilustradas las variaciones de los costos y el valor de convergencia de éstos, con lo cual se llega a un valor de $\mathcal{J}_{iterado} = 106,4$.

La reducción relativa del costo semilla producida por el costo de la última iteración es de 18,2168 %. En la figura (5.7) se ilustra el control semilla y el control obtenido en la última iteración, con sus respectivos tiempos de saturación. Para el control semilla, el comportamiento del control y los tiempos de saturación en la cota inferior y superior tienen un comportamiento aproximado a lo expuesto cuando el problema fue abordado con las divisiones en los intervalos de muestra. En el control iterado se puede observar que no existe saturación en la cota inferior, y el punto de saturación en la cota superior se desplazó cerca del tiempo final, lo cual ilustra que la estrategia conduce a resultados no triviales.

Las corridas del problema en esta nueva etapa fueron hechas en forma secuencial y en paralelo a través de MATLAB, con la intención de observar mejoras en el tiempo total de ejecución. Los resultados obtenidos se muestran a continuación:

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

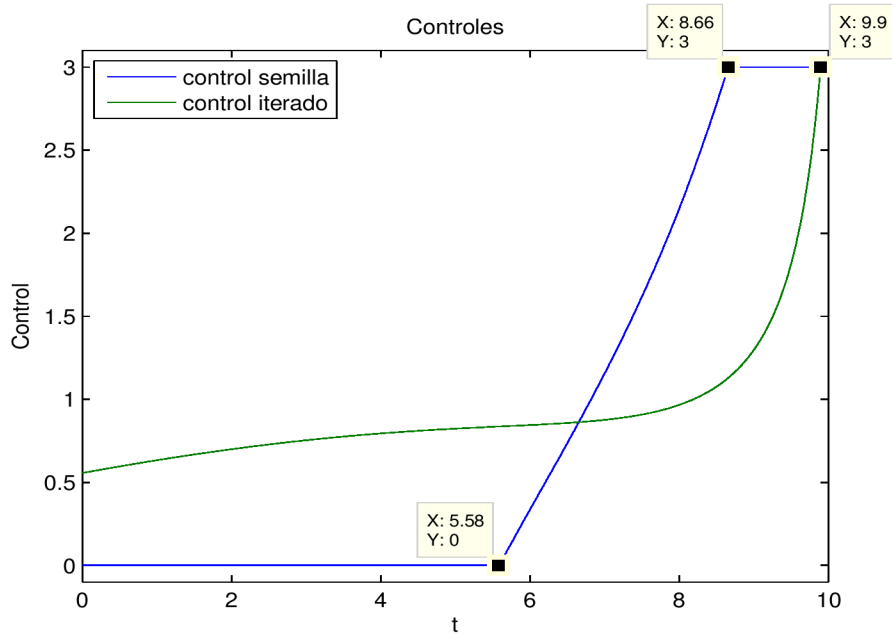


Figura 5.7: Control semilla e iterado

(i) Tiempo serial: $t_s = 44,112971$ segundos.

(ii) Tiempo en paralelo: $t_p = 11,319160$ segundos, con $p = 4$, con p como el número de procesadores utilizados.

La medida del rendimiento que puede obtenerse al agregar más nodos de cálculo se define a través de la aceleración o speedup y de la eficiencia [139]. Con computación en paralelo, la aceleración del tiempo de cómputo idealmente para arquitecturas homogéneas es lineal, esto es:

$$S_p = \frac{t_1}{t_p}, \quad (5.3)$$

con t_1 tiempo de cálculo de un solo procesador y t_p tiempo para p procesadores. Según la ecuación 5.3 el speedup debe ser igual al número de procesadores utilizados en el cálculo, aunque el speedup real es utópico ya que todas las fracciones del código en general no puede paralelizarse y además puede presentarse tiempos de retardo introducidos por las comunicaciones, sincronizaciones, etc. Si el programa no se puede paralelizar completamente (no se puede alcanzar una aceleración óptima), el speedup es menor o igual al número de procesadores. Esto depende

5.2 Computación en paralelo aplicada a problemas de control

del porcentaje de código implementado en serie, de acuerdo a la ley de Amdahl y Gustafson [74].

De la ecuación (5.3) el speedup obtenido en el ejemplo sería:

$$S_4 = 3,896952 , \quad (5.4)$$

de acuerdo a la ley de Amdahl [5] el speedup es:

$$S_{pA} = \frac{1}{\frac{1-f_s}{p} + f_s} , \quad (5.5)$$

y de acuerdo a la ley de Gustafson [69] el speedup vendría dado por:

$$S_{pG} = p + f_s(1 - p) , \quad (5.6)$$

con f_s como la fracción serial del programa (en este caso es $f_s = 0,0322$), de lo cual resultarían otros valores para el speedup

$$S_{4A} = 3,6496 , \quad (5.7)$$

$$S_{4G} = 3,9034 . \quad (5.8)$$

El valor de la fracción serial es correspondiente al gasto en el cálculo de la semilla. Los valores obtenidos son cercanos al encontrado según (5.3).

La eficiencia del programa es medida como:

$$E_p = \frac{S_p}{p} * 100 \% . \quad (5.9)$$

De (5.9), la eficiencia obtenida es:

$$E_4 = 97,4238 \% . \quad (5.10)$$

De la ecuaciones (5.4, 5.7, 5.8) y (5.10), se desprende que el cálculo en paralelo se convierte en una opción eficiente para instrumentar este tipo de estrategias, sobre todo cuando la dimensión del sistema aumenta.

5.3 Otro acercamiento al control óptimo

5.3.1 Computación en paralelo para el tratamiento numérico de la sección 3.1

Al problema del proceso de laminación proveniente de un balance de energía estándar y gobernado por una PDE de primer orden estudiado para sistemas lineales vía la actualización de la matriz de penalización final por intermedio de la actualización de los valores de los vectores de fase final, se le realizaron simulaciones que permitieron observar la reducción de tiempo cuando se aplica computación en paralelo para un problema con gran cantidad de carga computacional, observando las particularidades que presente el modelo al ir aumentando el orden del mismo.

En esta parte de la simulación, para realizar operaciones involucradas en el método propuesto se hizo uso de la librería Armadillo [128], es una librería de alta calidad para operaciones de álgebra lineal en lenguaje C++.

El problema se trabajó en memoria compartida, y se realizaron pruebas tanto paralelizando secciones como paralelizando bucles.

5.3.1.1 Paralelización del programa a través de secciones

Para trabajar en mejorar la aproximación lograda con el proceso de laminación 3.4.2 se utilizaron diferentes dimensiones del modelo, donde se realizaron distintas discretizaciones espaciales aumentando de manera paulatina las dimensiones del mismo. Usando la ecuación 3.69 y manteniendo el valor de $L = 10$, se modificaron los valores de las dimensiones n y el paso de discretización h para obtener los distintos x_i , a_{ij} y b_i correspondientes.

Las instrucciones de paralelización son realizadas por medio de OpenMP. Las fracciones de cálculo que puede ser paralelizado trabajando de esta manera se ven limitadas por las 4 derivadas de costo correspondiente (J_1, J_2, J_3, J_4) de acuerdo a cada una de las variables involucradas, en la variable η por las ecuaciones (3.36, 3.37, 3.41, 3.43), para los tiempos de conmutación τ_1 en las ecuaciones (3.44- 3.47) y τ_2 en (3.49- 3.52), respectivamente. La porción del programa paralelizada por ende corresponde a una distribución por secciones de cada una de ellas B.2.4, las que implican mayor carga computacional son las correspondientes a las que calculan

5.3 Otro acercamiento al control óptimo

Tabla 5.1: Medidas por iteración del programa paralelizado con omp sections

n	Sp	Ef	$t_1(seg)$	$t_2(seg)$	$t_1 - t_2(seg)$
20	1,996	99,8 %	80,5296	40,3436	40,186
50	1,979	98,95 %	206,086	104,112	101,974
100	1,878	93,89 %	482,959	257,176	225,783
150	1,66	83 %	1062,97	639,83	423,14
200	1,52	76 %	1845,22	1213,28	631,94

los valores de fase finales. Por lo tanto, se obtiene el mayor ahorro de cálculo a partir de su paralelización, en este caso con un máximo de 3 secciones debido a el valor 0 de la ecuación (3.49). Los resultados trabajando con $p = 2$ procesadores locales pueden apreciarse en la Tabla 5.1.

La figura 5.8 muestra el speedup y la ganancia total de tiempo obtenidos al paralelizar con $p = 2$ procesadores locales, aunque se evidencia una disminución del speedup con el aumento en la dimensión n del problema que conlleva consecuentemente a una disminución en la eficiencia del sistema cuando se habla de una arquitectura paralela, el tiempo total de cálculo se reduce de manera considerable precisamente ante el incremento de la dimensión. Se puede afirmar entonces que se obtienen resultados más rápidos en grandes dimensiones resignando eficiencia del sistema paralelo.

5.3.1.2 Paralelización del programa a través de los bucles

La paralelización del programa por medio de los bucles es realizada sobre las derivadas parciales de la matriz de penalización final con respecto a los valores de fase finales involucrados en la ecuación (3.18), haciendo una distribución de los elementos del vector que conforman los valores de fase finales ρ_i, μ_j de las ecuaciones (3.38, 3.39) (La fracción del programa paralelizada a través de los bucles puede verse en B.2.5).

Los resultados de la paralelización trabajando con distinta cantidad de procesadores en el cluster (actualmente el cluster disponible en el CIMEC es denominado ‘coyote’) y diferentes dimensiones del problema de $n = 10, 20, 50$ y 100 . Para correr en el cluster es necesario crear un job (ver apéndice B.2.5.2).

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

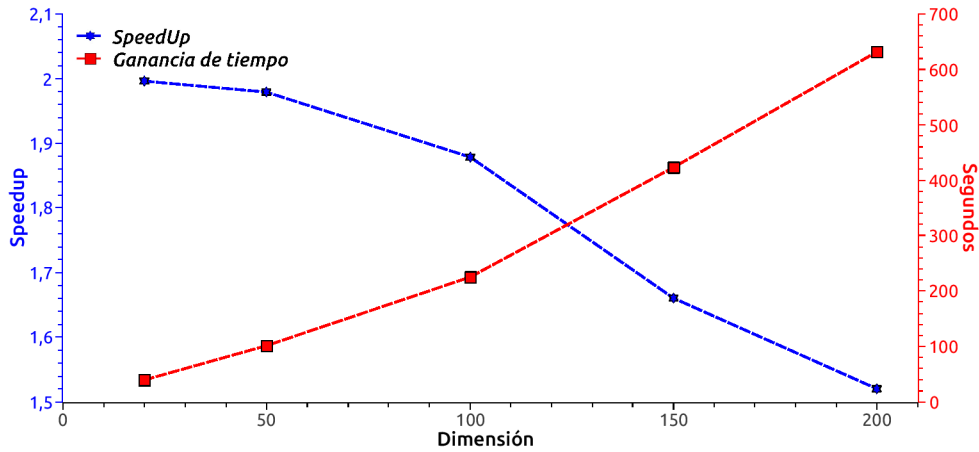


Figura 5.8: Speedup y ganancia total de tiempo para el problema de actualización de fase finales paralelizado con secciones y $p = 2$ procesadores locales.

Tabla 5.2: Medidas por iteración del programa paralelizado con omp for en el cluster con $n = 10$ y medidas de tiempo en segundos

p	t_1	t_2	t_4	t_6	t_8	Sp	Ef
1	80,846	—	—	—	—	—	—
2	80,846	40,738	—	—	—	1,9845	99,22 %
4	80,846	—	24,7467	—	—	3,2669	81,67 %
6	80,846	—	—	16,4298	—	4,921	82,01 %
8	80,846	—	—	—	16,6895	4,844	60,55 %

En las tablas se presentan los resultados obtenidos para las diferentes dimensiones, en la Tabla 5.2 para $n = 10$, en 5.3 para $n = 20$, en 5.4 para $n = 50$ y en la Tabla 5.5 para $n = 100$.

La figura 5.9 muestra el entorno del cluster con una descripción del mismo, exhibiendo la distribución de los nodos con las características de los procesadores que lo componen. Para la ejecución del programa es necesario además cargar el modulo de armadillo, en el recuadro rojo se marca el identificador del job asignado (JOBID) después de enviar el programa, para ver el estado de la corrida se utiliza el siguiente comando:

```
$ squeue -l
```

5.3 Otro acercamiento al control óptimo

Tabla 5.3: Medidas por iteración del programa paralelizado con omp for en el cluster con $n = 20$ y medidas de tiempos en segundos

p	t_1	t_2	t_4	t_6	t_8	Sp	Ef
1	164,599	—	—	—	—	—	—
2	164,599	84,5346	—	—	—	1,94712	97,35 %
4	164,599	—	43,6946	—	—	3,767	94,18 %
6	164,599	—	—	35,7894	—	4,599	76,65 %
8	164,599	—	—	—	27,5946	5,965	74,56 %

Tabla 5.4: Medidas por iteración del programa paralelizado con omp for en el cluster con $n = 50$ y medidas de tiempo en segundos

p	t_1	t_2	t_4	t_6	t_8	Sp	Ef
1	493,606	—	—	—	—	—	—
2	493,606	293,933	—	—	—	1,6793	83,96 %
4	493,606	—	197,504	—	—	2,499	62,47 %
6	493,606	—	—	165,663	—	2,9796	49,66 %
8	493,606	—	—	—	148,489	3,302	41,28 %

Tabla 5.5: Medidas por iteración del programa paralelizado con omp for en el cluster con $n = 100$ y medidas de tiempo en segundos

p	t_1	t_2	t_4	t_6	t_8	Sp	Ef
1	2011,13	—	—	—	—	—	—
2	2011,13	1610,05	—	—	—	1,249	62,46 %
4	2011,13	—	1409,59	—	—	1,4268	35,69 %
6	2011,13	—	—	1347,16	—	1,4928	24,88 %
8	2011,13	—	—	—	1309,11	1,536	19,2 %

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

El recuadro en amarillo indica el job correspondiente, pudiéndose observar además que se encuentra trabajando en el nodo2 del cluster con un solo nodo (lo cual se debe a la configuración del job B.2.5.2), este nodo según la descripción del cluster cuenta con 8 procesadores.

```
quest01@coyote:~/ejemplo/Anderson
quest01@172.16.254.215's password:
Last login: Wed Feb 22 18:24:31 2017 from 172.16.250.19
* Welcome to the COYOTE HPC cluster (32 active nodes) *
Server       : E5335   2.80GHz 2x4 cores
Nodes[2-7]   : E5420   2.50GHz 2x4 cores 8GB RAM, Feature=E5420_RAM8
Nodes[8-24]  : W3690   3.47GHz 1x6 cores 16GB RAM, Feature=W3690_RAM16
Nodes[1,25-32]: E5-1660 3.30GHz 1x6 cores 16GB RAM, Feature=1660_RAM16
Tesla 2050 (node [25]) , Feature=1660_RAM16,gpu,T2050
Tesla 1060 (nodes[26-27]), Feature=1660_RAM16,gpu,T1060
GTx       580 (nodes[31-32]), Feature=1660_RAM16,gpu, G580
--> Booking of nodes for runs: use 'sbatch' from SLURM job scheduler
--> Nodes's load (lnstde CCT-Santa Fe): http://172.16.254.215/ganglia/
2012/10/26: upgrade to Fedora 17, NFS4 (TCP).
2015/07/16: besides the system MPI releases, also there are:
  /usr/local/mpich      : stable 3.1.4 (Feb 20, 2015)
  /usr/local/mvapich    : stable 2.1 (Apr 03, 2015)
  /usr/local/openmpi-1.8.7 : stable 1.8.7 (Jul 15, 2015)
2015/09/25: latest Open MPI releases:
  /usr/local/openmpi    : stable 1.10.0 (Aug 25, 2015)
however, please, ask how to use it.
Disk quota: .6 GB used, 1.0 GB allowed (66 %)
[quest01@coyote ~]$ cd ejemplo/Anderson
[quest01@coyote Anderson]$ module load armadillo
[quest01@coyote Anderson]$ make main100par
g++ -I/usr/local/armadillo/7.800/include -L/usr/local/armadillo/7.800/lib/ -fopenmp -pg rho_mu_100_par.cpp funciones.cpp -o main100par.bin -O3 -funroll-loops -larmadillo
[quest01@coyote Anderson]$ sbatch runjob
Submitted batch job 12743
[quest01@coyote Anderson]$ squeue -l
Wed Feb 22 20:12:19 2017

```

JOBID	PARTITION	NAME	USER	STATE	TIME	TIME	LIMI	NODES	NODELIST(REASON)
12743	test	AGM	quest01	RUNNING	1:44	30:00		1	node2
12685	work	w170-ext	dbustos	RUNNING	6-02:42:25	UNLIMITED		1	node11
12684	work	w160-ext	dbustos	RUNNING	6-02:42:35	UNLIMITED		1	node10
12683	work	w150-ext	dbustos	RUNNING	6-02:42:48	UNLIMITED		1	node9
12682	work	w140-ext	dbustos	RUNNING	6-02:42:55	365-00:00:00		1	node8
12455	work	CNT1Fe1	pqualno	RUNNING	6-18:15:32	365-00:00:00		2	node[16-17]

```
[quest01@coyote Anderson]$
```

Figura 5.9: Corrida en el cluster 'coyote'.

Las figuras 5.10 y 5.11 representan el speedup y la eficiencia alcanzados en el cluster, respectivamente. Las medidas de tiempo tomadas en el cluster fueron superiores a las tomadas de manera local. Esto se debe al uso de la librería armadillo, ya que las bibliotecas en el cluster no suelen estar instaladas a nivel de sistema, porque a veces se utilizan varias versiones simultáneamente. Por eso, al compilar es necesario indicar directorios adicionales para que el compilador encuentre los .h, .a y/o .so.

Los resultados obtenidos al aumentar a varios procesadores presentan mejoras significativas cuando las dimensiones n son relativamente bajas. Al aumentar las dimensiones del modelo (en busca de mayor precisión), aunque se siguen presentando estas mejoras en el ahorro total de tiempo de cálculo, la eficiencia de la arquitectura paralela disminuye. Estos resultados se deben principalmente a la porción de código que es paralelizada dentro del programa: al no constituir dicha

5.3 Otro acercamiento al control óptimo

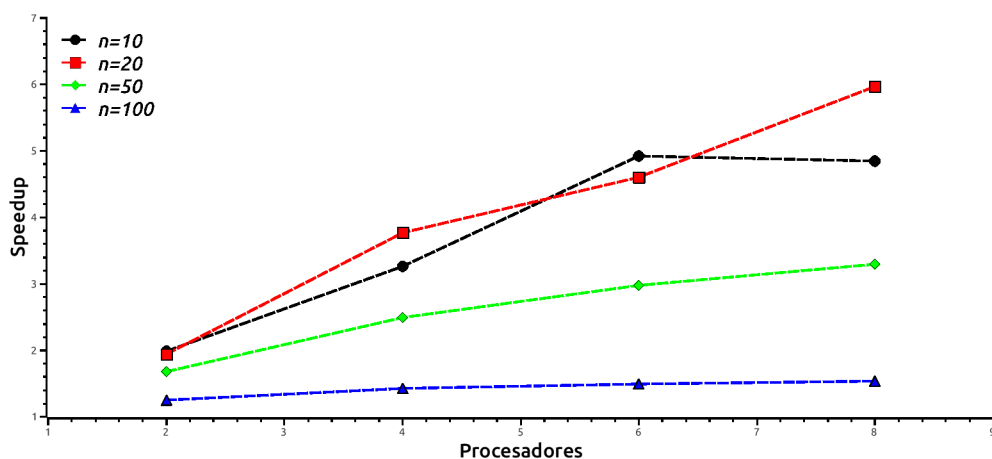


Figura 5.10: Speedup para el problema de actualización de fase finales paralelizado con bucles for hasta con $p = 8$ procesadores.

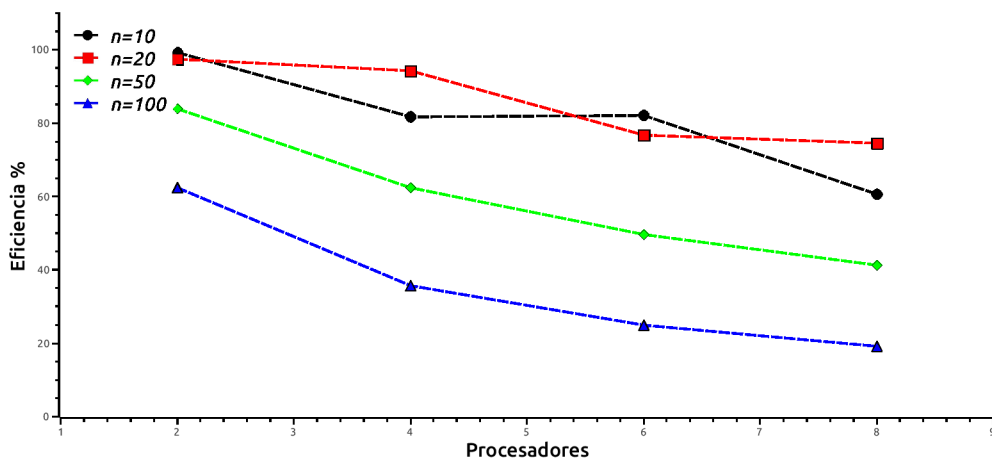


Figura 5.11: Eficiencia de la arquitectura paralela para el problema de actualización de fase finales paralelizado con bucles for hasta con $p = 8$ procesadores.

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

porción una gran cantidad de código ni acaparar la mayor cantidad de cálculos, las mejoras en tiempos se deberán sólo a lo que pueda aportar la parte paralelizada.

5.4 Discusión de los resultados

Se exploró la posibilidad de tratar problemas en tiempo real por intermedio del uso de computación en paralelo, para soportar una gran carga de cálculo en intervalos de muestra pequeños en problemas de control óptimo. Se encontró que al agregar procesadores se puede reducir el costo total del problema de control en más del 10%. En estrategias de ‘MPC’ con restricciones y optimización ‘on-line’, el costo computacional puede llevar a que la entrada de control no sea actualizada en cada instante del intervalo de muestra. Si estos intervalos de muestra son muy pequeños y se trabaja con modelos con características no lineales, el costo computacional requerido aumentará significativamente.

El uso del cluster brinda una mayor escalabilidad, gracias a que se cuenta con muchos nodos. Esto permite pensar en un futuro desarrollo para utilizar técnicas que impliquen mayor complejidad computacional. Esta complejidad permitirá pensar en una distribución en la cual no sólo sea necesario comunicación punto a punto (en el caso de memoria distribuida), sino también comunicación colectiva, en la que se pueda obtener una solución más cercana a la óptima, basándose en comparaciones de las soluciones de los procesadores de manera independiente, permitiendo esto resolver problemas en tiempo real.

Para el problema de actualización de los valores de fase finales 3.1 se exploró y utilizó OpenMP con una configuración de memoria compartida, mostrándose dos estrategias para paralelizar el problema: (i) a través de la distribución por medio de secciones de las derivadas del costo con respecto a los valores finales ρ , μ y los tiempos de conmutación τ_s , y (ii) a través de la paralelización de los bucles involucrados en los barridos de los valores de fase finales ρ_i , μ_j (3.38, 3.39).

Se evidenció en 5.3.1.2 que para modelos en los cuales la fracción de programa que puede ser paralelizada no constituyen mayor parte del problema total, las mejoras que pueden obtenerse debido a trabajar con computación en paralelo se ven limitadas, no siendo posible obtener mejores resultados aunque se incremente y utilicen paulatinamente más procesadores del cluster.

La escalabilidad que se puede alcanzar mediante el uso del cluster es importante para la solución del problema LQR con penalización final en el que se presenten varios períodos de saturación, lo que implicara una mayor carga computacional.

5. COMPUTACIÓN EN PARALELO PARA CONTROL ÓPTIMO BASADO EN MODELOS

Además para el tratamiento de modelos no lineales (ver [46]), se hace más apropiado el uso de la computación en paralelo, puesto que ésta puede disminuir los tiempos de procesamiento cuando se deben simular trayectorias de estados y coestados.

*La práctica debería ser producto de
la reflexión, no al contrario.*

Hermann Hesse

CAPÍTULO

6

Otros Problemas adicionales de control óptimo

En este capítulo son analizados problemas de control abordados desde métodos alternativos que buscan estrategias óptimas, pero que no se insertan estrictamente dentro de las líneas principales de la tesis. El primero de ellos es un método para prescribir el tratamiento terapéutico de pacientes diabéticos tipo 1, que optimiza un objetivo médico. El segundo optimiza una realimentación dinámica determinada por la asignación de trabajo individual según la demanda global de una planta de proceso tradicional, para un conjunto de equipos operando en paralelo.

Para la regulación de pacientes diabéticos se necesita que el manejo sea cuidadoso, debido a las restricciones inherentes y a que la insulina exógena sólo reduce el nivel de glucemia. Teniendo en cuenta las características discretas introducidas por la medición de los estados involucrados, y en la administración discreta de la insulina exógena para el tratamiento de la enfermedad, se propone una estrategia híbrida de programación, la cual admite de forma natural la incorporación de restricciones y límites seguros de regulación, tanto en las variables de estado como en la acción de control. La estrategia propuesta se analiza a través de simulaciones con pacientes virtuales, obteniéndose resultados promisorios (aunque preliminares)

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

que sugieren explorar la continuación del estudio de esta estrategia en personas con diabetes mellitus tipo 1.

Para la distribución de trabajo al conjunto de equipos operando en paralelo, a las referencias asignadas a cada dispositivo controlado convencionalmente se les permite cambiar continuamente mientras: *(i)* se minimiza un costo combinado, el cual es acumulativo en el tiempo y toma en cuenta la dinámica de todas las calderas individualmente, y *(ii)* la estrategia puede hacer frente a perturbaciones, al ruido de mediciones y a los cambios en la composición del combustible.

6.1 Programación híbrida para el manejo de la glucemia en pacientes diabéticos tipo I

La Diabetes Mellitus es una alteración del metabolismo, las cuales producen un conjunto de enfermedades crónicas caracterizadas por eventos de hiperglucemia asociados con acciones impropias de la insulina. La gravedad depende de la patogenia correspondiente, de la participación de los factores ambientales, del daño progresivo de los diversos órganos y tejidos, y del grado de deficiencia de insulina adquirido [63]. Para los pacientes que presentan deficiencia absoluta en la secreción de insulina (diabetes mellitus tipo 1 o pacientes insulino dependiente) se requiere administración regular de dosis de insulina exógena para mantener los niveles de glucosa en la zona de normoglucemia. La diabetes mellitus tipo 1 (T1DM) es una enfermedad que afecta a aproximadamente 25 millones de personas en el mundo, en [31, 32, 112, 150] se evidencia como el tratamiento intensivo de la glucosa en la sangre aumenta la esperanza de vida en un estudio para pacientes T1DM, conducido por la Diabetes Control and Complications Trial (DCCT) en el cual se demostró que mantener los niveles de glucosa en sangre lo mas cercano posible a una persona sana disminuye en gran medida la aparición o progresión del decaimiento de los órganos debido a la enfermedad y las complicaciones subyacentes.

La diabetes mellitus es una de las enfermedades más comunes en casi todos los países y está presentando un aumento significativo en número y en importancia, esto debido en gran parte a los estilos de vida, a la disminución de la actividad física y al aumento considerable de personas con problemas de obesidad. En América Latina el número de personas estimado con diabétes tipo 1 y tipo 2 en el año 2000 era de 377500 y 14863700, respectivamente [12], y se valora que la prevalencia de diabetes mellitus aumentará en 250 % en los próximos 20 años y pueden representar un costo alrededor del 6 % [144].

Los niveles hiperglucemicos (como condición inicial) puede ser comparable con un episodio postprandrial de un paciente diabético mal controlado. La terapia tradicional para contrarrestar y controlar los efectos de la enfermedad de pacientes insulino dependiente que presenten altos niveles de glucosa en la sangre consiste en la administración de dosis de insulina exógena junto con la medición de los valores de glucemia durante el día. Con la aparición de las nuevas tecnologías, emerge

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

como una modalidad importante para la implementación de una terapia intensiva, la inyección de insulina por medio de bombas de infusión continua subcutánea CSII (Continuous Subcutaneous Insulin Infusion), pudiendo resultar ventajoso en pacientes para los que se tenga un control de glucemia deficiente y que a su vez le permita funcionar como una alternativa a la terapia por medio de múltiples inyecciones diarias MDII (Multiple Daily Insulin Injection) y los monitores continuos de glucosa, además en [116, 122] se ha demostrado pequeñas mejoras en el control de la glucemia usando CSII en comparación con la terapia MDII. El esquema de dosificación MDII no garantiza la permanencia en la región segura del nivel de glucosa, y el riesgo de sobre-dosificación es elevado y potencialmente peligroso, sin embargo es importante su consideración en períodos postprandiales donde los niveles de glucosa son altos. Por tal motivo ha surgido un gran interés en cerrar el lazo entre las bombas y los glucómetros de medición continua (CGM) que posibiliten aplicar una infusión automática de insulina exógena y permita mejorar la calidad de vida de pacientes T1DM, evitando o retardando las complicaciones intrínsecas de la enfermedad, debido a esto hace más de 50 años ya fue visionada la idea de un dispositivo (páncreas artificial) que cierre el lazo de control, en [28] se puede apreciar una descripción de la evolución del AP (abreviatura del inglés “Artificial Pancreas”) junto con los elementos y configuraciones necesarios para realizarlo. Uno de los primeros trabajos en esta dirección fue el de [86] aplicando control “on-off”.

La terapia intensiva con insulina aumenta el riesgo de hipoglucemia, siendo una complicación común en personas con T1DM que se encuentran bajo tratamiento, especialmente durante períodos nocturnos. Esta complicación sigue siendo uno de los problemas abiertos en el diseño del páncreas artificial. El CGM a permitido reducir los episodios de hipoglucemia y su integración con las bombas de insulina dio inicio al desarrollo del AP al ser los datos del CGM las entradas del algoritmo de control del AP. Aún así, la precisión del CGM se ve afectada por diferentes factores, principalmente por el largo tiempo de difusión de la glucosa en plasma (glucemia) al fluido intersticial considerándose que existe un retardo entre estos que oscila entre los 5-15 min dependiendo de la velocidad de cambio de la glucosa, o por errores de pérdida transitoria de sensibilidad o ruido aleatorio [28, 129]. Por ende los CGM modernos vienen integrados de algoritmos en tiempo real y modelos

predictivos para intentar predecir tendencias y generar alertas al pacientes sobre la proximidad de eventos de hipo- o hiperglucemia al reconstruir la medición de glucosa en sangre.

Muchos trabajos han considerado como su primer objetivo la reducción de eventos hipoglucémicos en períodos nocturnos, con la restricción principal de que el paciente no consuma alimentos en dicho lapso. El mantenimiento nocturno de la normoglucemia es uno de los grandes desafíos, ya que la mayoría de casos severos de hipoglucemia ocurren en la noche, los cuales representan un 75 % del total de convulsiones hipoglucémicas en niños y pueden estar asociadas con el 6 % de las muertes en pacientes menores de 40 años con T1DM [115]. Algunos estudios de regulación nocturna de la glucosa se presentan en [58, 79, 95, 146], en [95] los bolos de insulina se administran cada 15 min, lo cual produce una concentración de insulina en sangre que es indistinguible de aquellas generadas por administración de insulina continua, además posee la ventaja práctica del suministro discreto de insulina que se incluye dosificación más precisa de insulina y optimización de la duración de la batería de la bomba. Aparte de los algoritmos de control, también se ha diseñado un sistema de alarmas con detección y/o predicción de hipoglucemia para manejar este problema [2]. Un estudio comparativo de las tecnologías actuales y los desafíos para lograr el uso masivo del páncreas artificial es presentado en [56].

En los últimos años, MPC ha sido una de las estrategias que más se ha desarrollado en el proyecto de páncreas artificial [57, 66, 68, 106]. Esta estrategia usa acciones de control discretas de forma escalonada, y para la optimización una versión discreta del modelo de referencia. Asegurar factibilidad y un buen dominio de atracción del controlador aumenta considerablemente el costo computacional. El uso natural de la restricciones, tanto en estados como en acciones de control es la característica más significativa de estas estrategias.

Estrategias más simples de control también se han desarrollado. Por ejemplo, en [107, 145] se proponen controladores PID, y en [109] una versión “inteligente” es desarrollada utilizando la técnica de “Model free control”, cuya principal ventaja es la facilidad en la elección de los parámetros del controlador, asimismo estrategias basadas en lógica difusa [9] o combinaciones de controladores PID con lógica difusa [136]. En [100, 123] se investiga un algoritmo que trabaja como una capa

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

de seguridad alrededor de un controlador principal de glucosa en lazo cerrado (de cualquier naturaleza), encargado de evitar o reducir eventos de hipoglucemia especialmente en períodos posteriores a la ingesta de alimentos, es basado en control de modo deslizante. El algoritmo es robusto contra fallas en los sensores o sobreestimación de la dosis de insulina a administrar. Sin embargo, ninguno de estos métodos asegura la eliminación por completo de episodios hipoglucémicos y/o la sobredosis de insulina.

En esta sección se ilustra la utilización de una versión sofisticada de programación dinámica para minimizar episodios hiperglucémicos y a la vez evitar que el sistema entre en la zona de hipoglucemia, es decir, se busca mantener al paciente T1DM en una zona segura. Esta estrategia tiene las ventajas del MPC, como el manejo natural de restricciones de estados y control, pero el esfuerzo computacional es menor, debido a que la mayor parte de los cálculos sólo se realizan una vez y se guardan en memoria para aplicarlos cuando es necesario. Un punto importante es elegir el modelo que se utiliza para describir el comportamiento dinámico del paciente. En este respecto, un nuevo modelo es utilizado [105], el cual a pesar de ser lineal captura bien las dinámicas principales y corrige algunos defectos de los modelos tradicionales. Otra característica adicional del modelo es que representa adecuadamente a largo plazo el comportamiento del paciente T1DM (alrededor de 3 o 4 días), en contraposición con los modelos tradicionales que utilizan las otras estrategias, donde su rango de validez es sólo de algunas horas. Una desventaja del modelo elegido es que uno de los estados no tiene mayor significado físico (la función derivada de la insulina) por lo que se introduce un estado asociado con la insulina subcutánea y se plantea un cambio de variables para expresar el modelo en los nuevos estados fisiológicos.

6.1.1 Modelo de pacientes con diabetes mellitus tipo 1

El modelo utilizado es el descrito por [105], el cual representa adecuadamente la interacción glucosa-insulina en pacientes T1DM. Debido a que se consideran sólo escenarios nocturnos donde el paciente no ingiere alimentos, se plantean 3 ecua-

ciones diferenciales:

$$\dot{\tilde{x}} = \begin{pmatrix} 0 & -k_{si} & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{1}{T_u^2} & -\frac{2}{T_u} \end{pmatrix} \tilde{x} + \begin{pmatrix} 0 \\ 0 \\ \frac{k_u}{V_i} \frac{1}{T_u^2} \end{pmatrix} \tilde{u} + \begin{pmatrix} k_l - k_b \\ 0 \\ 0 \end{pmatrix}. \quad (6.1)$$

La dinámica de la glucosa obedece a la primera ecuación diferencial del sistema (6.1), y representa la acumulación de glucosa que responde a: (i) la concentración de insulina en sangre (insulinemia), y (ii) la diferencia entre la producción endógena de glucosa en el hígado (EGP) y el consumo de glucosa. El término EGP k_l es considerado constante por simplicidad. La tasa de absorción de glucosa incluye el consumo del cerebro k_b sin necesidad de insulina (también tomado constante), el almacenamiento que realiza el hígado y el consumo de glucosa por los músculos bajo la acción de la insulina. La diferencia entre la producción de glucosa EGP y el consumo del cerebro es considerado bajo un solo parámetro $\vartheta_1 = k_l - k_b$. El ensayo clínico realizado en [77] avala la suposición de que el término insulino-dependiente del consumo de glucosa no depende de la concentración de glucosa en sangre (ver [105] para más detalles). Por lo tanto ese término resulta $-k_{si}x_2$, donde el parámetro k_{si} está relacionado al factor de sensibilidad a la insulina (ISF) también conocido como el factor de corrección (CF).

La incorporación de insulina exógena a través de la variable de control u (en Unidades de Insulina ‘UI’) se compone de la inyección de insulina y la insulina basal u_b . La relación entre esta variable, y la insulinemia, x_2 (UI/l), es obtenida de las características farmacocinéticas de la insulina. Tal como se describe en [105], la dinámica de la insulina se modela con un integrador generalizado de segundo orden con una sola constante de tiempo T_u .

Notar que este modelo tiene un estado igual a la derivada de la insulina, lo cual no tiene mayor significado físico. Debido a esto un nuevo estado es introducido, la concentración de insulina subcutánea igual a $x_{sc} = \frac{V_i}{k_u} T_u \tilde{x}_3 + \frac{V_i}{k_u} \tilde{x}_2$. Al incorporar este estado, se propone el siguiente cambio de variables basados en [104]:

$$x_1 = \tilde{x}_1, \quad x_2 = \tilde{x}_2 - u_b, \quad (6.2)$$

$$x_3 = x_{sc} - u_b, \quad u = \tilde{u} - u_b \quad (6.3)$$

donde x_2 , y x_3 representan la concentración de insulina en sangre y subcutánea respectivamente. Además u_b es el nivel basal de insulina. Simplificando la notación

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

se define $\vartheta_2 = \frac{k_u}{V_i} k_{si}$, y $\vartheta_3 = T_u$. El nivel basal entonces es $u_b \triangleq \frac{V_i}{k_u} \frac{\vartheta_1}{k_{si}} = \frac{\vartheta_1}{\vartheta_2}$. Luego de algunas manipulaciones algebraicas, el modelo resultante es

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ &= \begin{pmatrix} 0 & -\vartheta_2 & 0 \\ 0 & -\frac{1}{\vartheta_3} & \frac{1}{\vartheta_3} \\ 0 & 0 & -\frac{1}{\vartheta_3} \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\vartheta_3} \end{pmatrix} u(t). \end{aligned} \quad (6.4)$$

Notar que este modelo, cuando se analiza en lazo abierto ($u \equiv 0$), es críticamente estable con autovalores $\lambda_1 = \lambda_2 = -\frac{1}{\vartheta_3}$ y $\lambda_3 = 0$.

6.1.2 Costo asociado a la acción terapéutica

El funcional de costo, representando el “costo médico total”, a ser minimizado entre todas las terapias aceptables (de insulina inyectada) se puede diseñar como sigue:

$$\begin{aligned} \mathcal{J}(u) = \int_{t_0}^{t_f} \{ & (q_1(x_1 - x_{ref})^2 + q_2x_2 + q_3x_3 + ru^2) \} dt \\ & + s(x_1(t_f) - x_{ref})^2, \end{aligned} \quad (6.5)$$

y consiste en un “costo de trayectoria” $Q(t_0; t_f; x_0; u)$ (usualmente expresado como la integral del Lagrangiano $L(t; x; u)$ o diferencial de costo que ocurre durante el tratamiento), y un costo final llamado “penalización final” $K(x(t_f))$ asociado a la diferencia entre el estado deseado (referencia) y el estado realmente obtenido. En este caso el Lagrangiano incluye varios términos no triviales, como $(x_1 - x_{ref})^2$, cuadrado de la diferencia entre la glucosa en sangre y un valor de glucosa deseada x_{ref} ; q_2x_2 , q_3x_3 penalizan el exceso de la cantidad de insulina que el paciente actualmente tiene en el cuerpo debido a la dosificación, que por ser variables positivas (el subsistema de insulina es una matriz Metzler) se escriben sin la forma cuadrática, y el último término ru^2 es para minimizar la cantidad de insulina a inyectar.

En este punto cabe mencionar que una versión sofisticada del costo podría incluir costos asimétricos para penalizar las desviaciones con respecto a los límites seguros, tanto de hiper como de hipoglucemia (ver [66]).

6.1.2.1 La necesidad de discretizar

Plantear un problema de control óptimo en tiempo continuo no es práctico para el caso de pacientes diabéticos T1DM. Sin embargo, la salud del paciente está sometida a un deterioro continuo, posiblemente siguiendo el modelo dinámico (6.4).

El tratamiento actual de la aplicación de insulina se realiza de manera discreta y en ciertos tiempos, dependiendo de la medición de glucosa, sea con el medidor CGM (mediciones cada 5 min) o mediciones discretas. Esto significa que si bien las dinámicas y el costo de trayectoria evolucionan en tiempo continuo, es necesario considerar un método híbrido discreto/continuo para resolver el problema de optimización.

La naturaleza híbrida del problema se potencia por (i) la disponibilidad de medidas de algunos de los estados descritos, (ii) la administración discreta de la insulina exógena, (iii) restricciones en la variable de control, (iv) retardos tanto en la evaluación del paciente como en el tiempo que necesita el médico para tomar decisiones terapéuticas. A continuación se muestra cómo se discretizarán los estados y el control para ser utilizados en esta estrategia. Definamos los espacios de valores de estados y control como:

$$\begin{aligned}
 \mathcal{X}_1 &:= \{x_{1L}, x_{1L} + \Delta x_1, x_{1L}, x_{1L} + 2\Delta x_1, \dots, x_{1U}\} \\
 \mathcal{X}_2 &:= \{x_{2L}, x_{2L} + \Delta x_2, x_{2L}, x_{2L} + 2\Delta x_2, \dots, x_{2U}\} \\
 \mathcal{X}_3 &:= \{x_{3L}, x_{3L} + \Delta x_3, x_{3L}, x_{3L} + 2\Delta x_3, \dots, x_{3U}\} \\
 \mathcal{U} &:= \{u_L, u_L + \Delta u, u_L, u_L + 2\Delta u, \dots, u_U\}
 \end{aligned}
 \tag{6.6}$$

donde los valores inferiores (L) y superiores (U) limitan el tamaño de paso (Δ) para cada variable. Estos límites deben elegirse de acuerdo a restricciones reales, dependiendo en algunas variables de las características de los instrumentos de medición, la dosis mínima de aplicación de insulina, y recomendaciones dadas por los especialistas médicos. El costo (6.5) debe ser reemplazado, debido a la discretización, por otro con la siguiente forma:

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

$$\begin{aligned} \mathcal{J}(u) &= \sum_{k=0}^{t_f/h} \int_{t_k}^{t_{k+1}} \{ (q_1(x_1 - x_{ref})^2 + q_2 x_2 + q_3 x_3 \\ &\quad + r u_k^2) \} dt + s(x_1(t_f) - x_{ref})^2, \\ t_{k+1} &= t_0 + hk. \end{aligned} \quad (6.7)$$

donde $x(t)$ deben ser entendidos, en cada intervalo $[t_k, t_{k+1}]$, como el resultado redondeado proveniente de la función de transición $\varphi(t; t_0, x, u(\cdot))$ asociada al modelo continuo (6.4) (la estrategia híbrida proviene entonces del uso de φ continua en combinación con la discretización de los estados, el costo y el control), siendo

$$x(t)' = \varphi(t; t_k, x_k, \tilde{u}_k), \quad (6.8)$$

$$x(t_{k+1})' = \text{round}(\varphi(t; t_k, x_k, \tilde{u}_k)), \quad (6.9)$$

$$\tilde{u}_k \equiv u_k, \quad (6.10)$$

donde “round” actúa sobre x_{k+1} de manera “segura”, es decir

$$x(t_{k+1}) \triangleq \text{valor más grande cercano a } x(t_{k+1}) \text{ en } \mathcal{X}. \quad (6.11)$$

El Hamiltoniano discreto del problema se define de la siguiente manera:

$$\mathcal{H}_k \triangleq \mathcal{L} + l'_{k+1} f_k, \quad (6.12)$$

donde el coestado $l_k \in \mathbb{R}^3$ juega el papel del gradiente de la función de valor o (función de Bellman) tal como en la formulación de tiempo continuo (ver sección 2.2 y [21, 134]).

Luego de tomar variaciones en el funcional de costo, la dinámica del Hamiltoniano puede describirse como

$$l'_k = \left(\frac{\partial f_k}{\partial x} \right)' l_{k+1} + \left(\frac{\partial \mathcal{L}_k}{\partial x} \right)', \quad (6.13)$$

$$\frac{\partial \mathcal{H}_k}{\partial u} = \frac{\partial \mathcal{L}_k}{\partial u} + l'_{k+1} \left(\frac{\partial f_k}{\partial u} \right), \quad (6.14)$$

sujeto a la condición final

$$l_N = \left(\frac{\partial K}{\partial x} \right)' \Big|_N = 2s(x_1(t_f) - x_{ref})', \quad (6.15)$$

donde se usa la siguiente interpretación de la función de transición discreta

$$x_{k+1} = f_k(x_k, u_k) \triangleq \varphi(t_{k+1}, t_k, x_k, u_k) . \quad (6.16)$$

Ahora ya es posible realizar la integración hacia atrás iniciando en la condición final (6.15). El valor del control óptimo local u_{N-1} puede en principio recuperarse de la solución de la ecuación (6.14). Luego este valor u_{N-1} es reemplazado en (6.13) con $k = N - 1$, y entonces se calcula l_{N-1} . Después el proceso continúa para $k = N - 2, \dots, 0$. Sin embargo, las restricciones en u no se pueden agregar fácilmente en esta formulación. Este inconveniente hace que la “Programación Dinámica” sea elegida como el método numérico para resolver esta estrategia continua/discreta en lugar del enfoque Hamiltoniano asociado con (6.12- 6.16).

6.1.3 Aplicación numérica al problema de diabétes tipo I

El sistema de glucosa-insulina debe evolucionar en una zona segura de actuación. Habitualmente una concentración normal de glucosa en sangre es de aproximadamente (70 – 109 mg/dl o 3,09 – 6,04 mmol/l). Si se encuentra fuera del rango normal, se dice que la persona tiene problemas de glucosa: hiperglucemia (≥ 140 mg/dl o 7,8 mmol/l) o hipoglucemia (< 40 mg/dl o 2,2 mmol/l) (ver [103]).

Se asume que las mediciones se realizan cada 5 min a través de un sensor de glucosa subcutánea, y existe un observador de glucosa en sangre que estima esta variable. Además se supone que inicialmente el paciente se encuentra en la zona de hiperglucemia, explícitamente:

$$\begin{aligned} x_1(0) &= 300 \text{ mg/dl}; & x_2(0) &= u_b \mu\text{UI/ml} \\ x_3(0) &= u_b \mu\text{UI/ml}. \end{aligned} \quad (6.17)$$

Para la implementación de programación dinámica es necesario definir un rango de posibles valores donde los estados se mueven ante la aplicación de diferentes valores de la insulina exógena. Es evidente que los valores de las variables involucradas deben estar por encima de 0 por el significado fisiológico de cada una de ellas.

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

Los valores máximos de la insulina en sangre y subcutánea se tomaron en relación con la aplicación de la máxima dosis de insulina. Las cotas menores (L) y superiores (U) se definen como:

$$x_{1L} = 50, x_{2L} = 0, x_{3L} = 0, u_L = 0, \quad (6.18)$$

$$x_{1U} = 300, x_{2U} = 0,2, x_{3U} = 0,5, u_U = 4. \quad (6.19)$$

La discretización (Δ) de la variable manipulada debería responder a la cantidad mínima que la bomba de insulina pueda administrar, o al valor de la mínima inyección que un paciente pueda hacer de forma manual. Para esta aplicación preliminar de la estrategia de programación dinámica se adoptó la siguiente discretización:

$$\Delta x_1 = 10, \Delta x_2 = 0,01, \Delta x_3 = 0,050, \Delta u = 0,2. \quad (6.20)$$

Las constantes del modelo se tomaron del artículo [105] con los siguientes valores: $\vartheta_1 = 0,1385$ mg/dl/min, $\vartheta_2 = 11,6230$ mg/dl/UI, $\vartheta_3 = 20$ min.

Los coeficientes adoptados para el funcional del costo fueron $r = 0,5$, $q_1 = q_3 = 10$, $q_2 = 50$, y $s_3 = 10$.

Las figuras 6.1 - 6.3 muestran las diferentes trayectorias (en este caso proyecciones sobre cada plano coordenado) óptimas obtenidas como solución del algoritmo de programación dinámica para diferentes condiciones iniciales en un rango de análisis de $[0, 600]$ minutos. Sólo se ilustran hasta los 120 min ya que después de ese valor las estrategias de control se mantienen en cero.

La figura 6.4 muestra las diferentes trayectorias de control óptimas discretas para las diferentes condiciones iniciales, variando los estados. Como es de esperarse, para el final del intervalo de optimización, al acercarse el nivel de glucosa al rango aceptable la necesidad de inyección de insulina exógena se va desestimando.

En las figuras 6.5 - 6.6 se muestran las salidas del modelo a partir de las condiciones iniciales descritas, tanto para el nivel de glucosa en sangre como para el comportamiento de la insulina en sangre cuando se aplica la estrategia de control óptimo discreta presentada en la figura 6.7, es decir al inyectar la insulina exógena. Se observa que rápidamente la glucosa es llevada a la zona normal de glucemia (después de una hora y media de iniciado el tratamiento la glucosa cruza el umbral de 140 mg/dl), para posteriormente estabilizarse en 80 mg/dl, muy por encima del umbral crítico de 40 mg/dl.

6.1 Programación dinámica

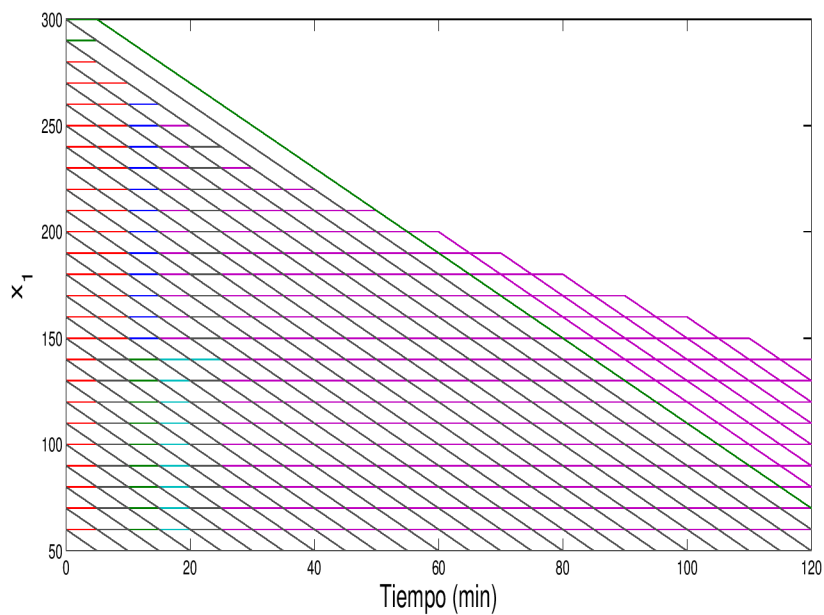


Figura 6.1: Trayectorias discretas de nivel de glucosa en sangre x_1 obtenidas por programación dinámica.

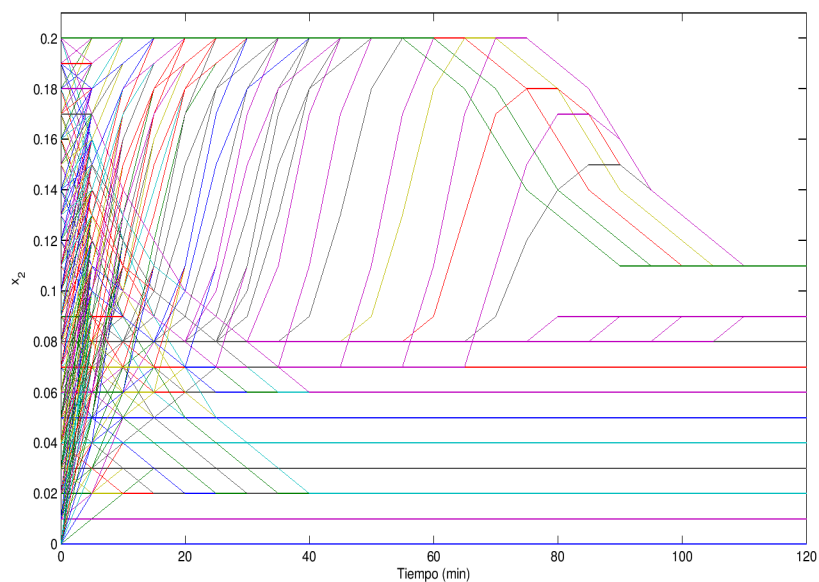


Figura 6.2: Trayectorias discretas de insulina en sangre x_2 obtenidas por programación dinámica.

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

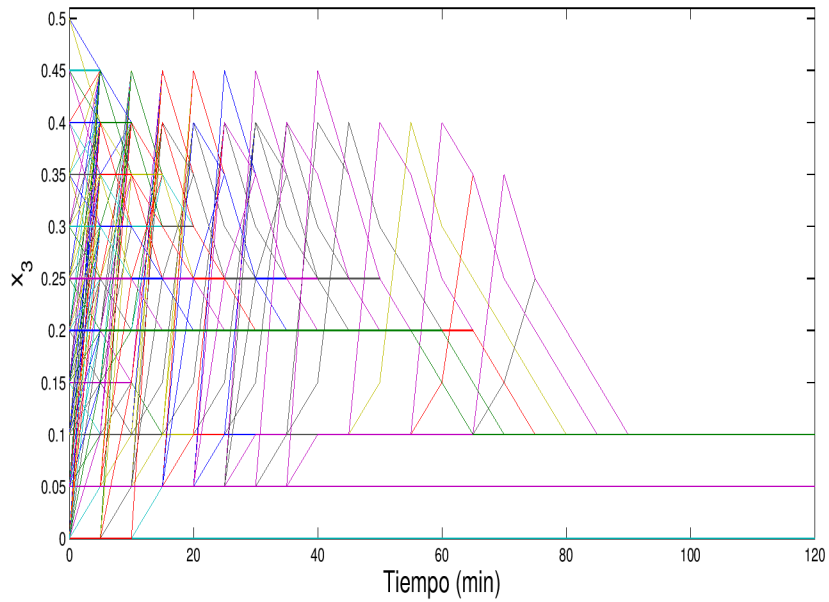


Figura 6.3: Trayectorias discretas de insulina subcutánea x_3 obtenidas por programación dinámica.

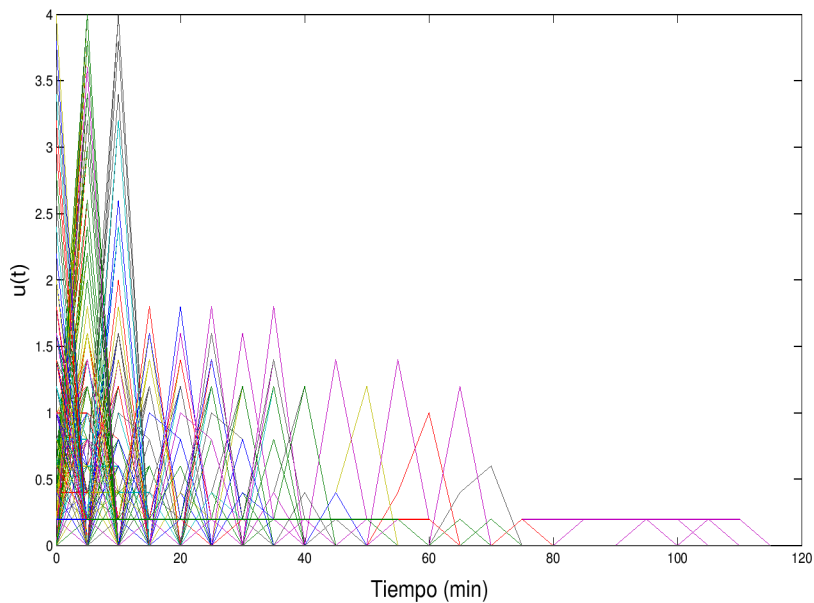


Figura 6.4: Trayectorias discretas de control obtenidas por programación dinámica.

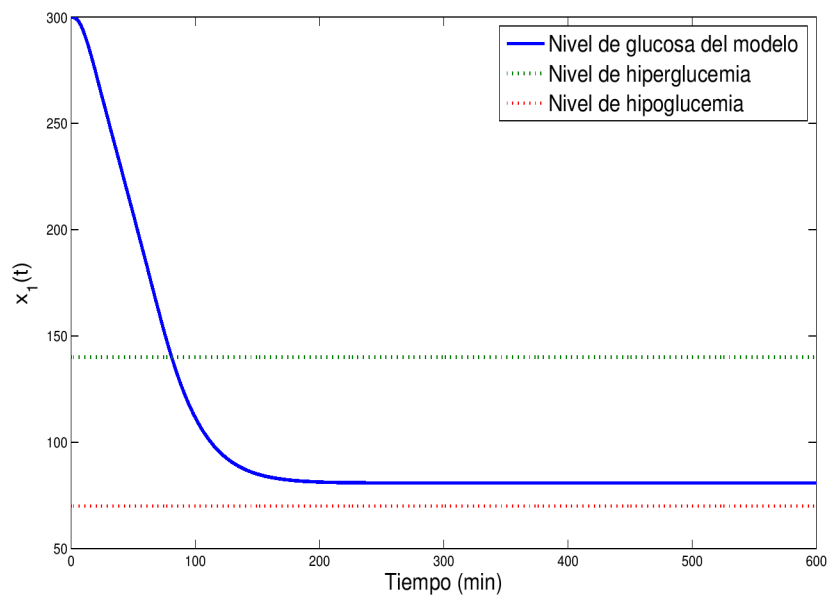


Figura 6.5: Trayectoria continua de la glucosa en sangre x_1 después de la aplicación de la secuencia de control obtenida por programación dinámica.

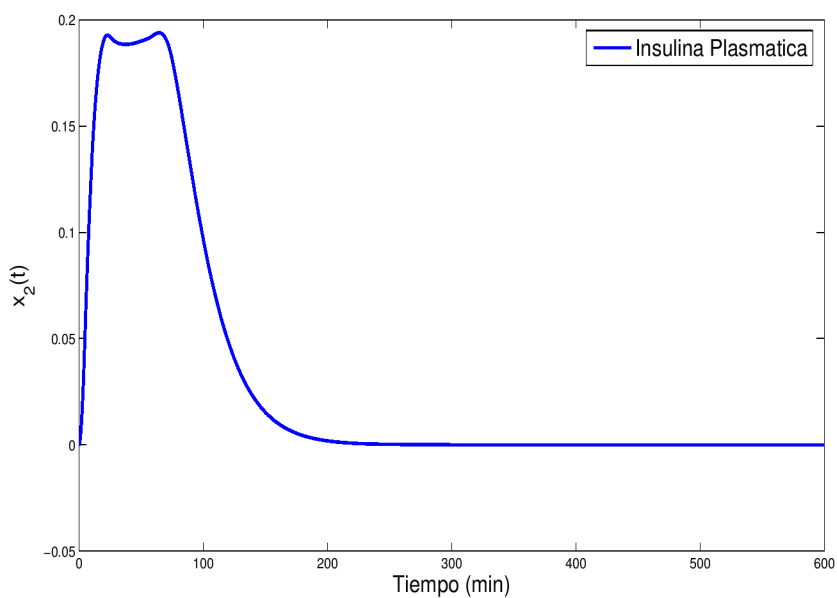


Figura 6.6: Trayectoria continua de insulina en sangre x_2 después de la aplicación de la secuencia de control obtenida por programación dinámica.

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

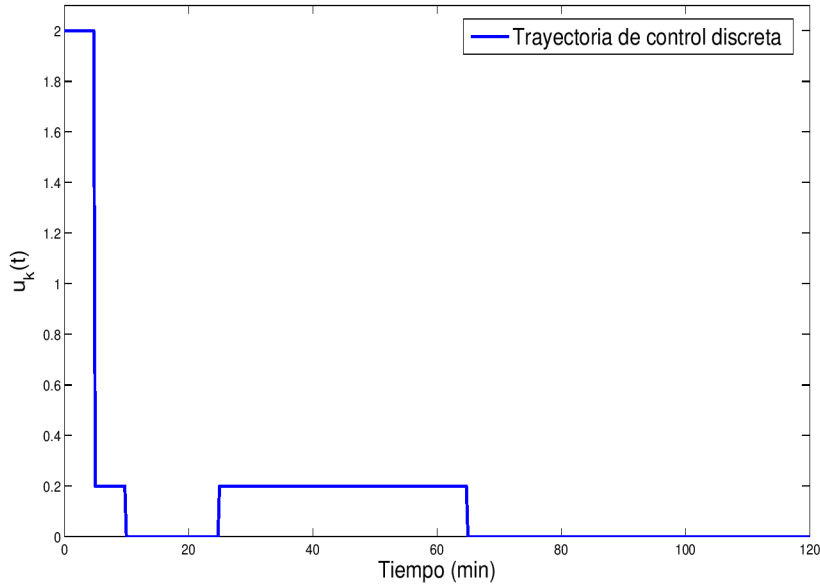


Figura 6.7: Estrategia de control u para un horizonte de 600 min obtenida por programación dinámica.

6.1.4 Conclusiones

El uso preliminar de un algoritmo de programación dinámica híbrida para la regulación del nivel de glucosa en sangre para pacientes con diabetes mellitus tipo I, en escenarios nocturnos, sin la ingesta de alimentos, ha sido ilustrado. Al estar el suministro de insulina exógena (variable de control) sólo afectando a x_3 directamente, y a los demás estados a través de sus integraciones (la acción de control no se manifiesta en el modelo de forma inmediata), se debe tener precaución de no sobredosificar al paciente, ya que esto puede causar episodios de hipoglucemia.

Una de las ventajas de la programación dinámica es que admite la formulación de funciones de costo complicadas (que por ejemplo admita distintos términos competitivos de optimización) y el diseño de zonas seguras. En esta sección se ha diseñado un costo simple pero que tiene en cuenta lo principal del problema de regulación de pacientes diabéticos. Admitiendo que la variable manipulada se aplica de forma discreta en intervalos regulares, y que los estados evolucionan de forma continua, se adaptó el algoritmo de programación dinámica para incluir

estos comportamientos llegando a una estrategia híbrida, que evalúa el costo y la dinámica de forma continua pero admite la entrada discreta y además restricciones tanto en los estados como en el control de forma “natural”, manteniéndose la forma de feedback implícito de las soluciones. La estrategia de control obtenida regula satisfactoriamente la glucosa en el período de tiempo elegido.

Así también se modificó el modelo en el cual está basada la estrategia para representar de manera más fisiológica a la interacción de glucosa-insulina en pacientes T1DM. Se propuso un nuevo estado, la insulina subcutánea x_3 , y un cambio de variables para adecuar el modelo.

Los resultados obtenidos son preliminares pero muestran un potencial para la regulación de la glucosa en sangre en pacientes diabéticos. Sin embargo, se debe continuar con el estudio y mejoramiento de esta estrategia que considere los siguientes aspectos: *(i)* un costo asimétrico que permita la definición de una zona segura de regulación, *(ii)* aumentar el número de pacientes virtuales, *(iii)* permitir la inclusión de escenarios diurnos, es decir, que contemple la regulación de perturbaciones por la ingesta de comida, *(iv)* contemplar variaciones intra e inter pacientes además de la robustez de la estrategia de control, y *(v)* su generalización a escenarios más cercanos a la realidad práctica de los personas que conviven con la enfermedad.

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

6.2 Asignación dinámica de consignas a equipos de Servicios Industriales como un problema de seguimiento óptimo estocástico

Los problemas de asignación óptima tienen una larga tradición en la práctica de ingeniería. En procesos químicos, la optimización dinámica frecuentemente se ocupa de la distribución de la demanda de servicios globales de la planta hacia objetivos individuales requeridos para cada miembro del grupo de un equipo de servicio, mientras se minimiza algún costo generalizado predeterminado. Equipos de servicio típicos incluyen conjuntos de calderas/generadores de vapor, intercambiadores de calor, bombas, compresores de aire, y similares [22, 30, 148]. En lo que sigue, los componentes individuales del ‘grupo’ bajo consideración serán concebidos como una ‘unidad’. Las unidades operan en paralelo para alcanzar la demanda total requerida por el grupo. Usualmente las demandas individuales son referencias (consignas) comunicadas a los controladores PID, los cuales son sintonizados para que funcionen adecuadamente. La suma de las demandas asignadas a las unidades es siempre asumida igual a la demanda total requerida para el grupo.

Las estrictas políticas ambientales, el aumento en los costos de energía, la crisis de la economía global, y las amenazas del cambio climático, han provocado un resurgimiento del tema de optimización y se está haciendo foco en la mejora de la eficiencia en los procesos industriales. La energía es suministrada (o sustraída de) a una planta principalmente a través de los servicios públicos. Una reducción en el consumo de estas utilidades resultan directamente en un ahorro de energía. En la figura 6.8 se presenta un diagrama esquemático de la utilidad de una planta. En negro se ilustra la estructura de una planta estándar, donde la potencia de la energía es generada por un sistema de turbinas en paralelo [92]. Estos equipos son alimentados por generadores de vapor. La demanda de potencia es producida por una disputa entre la obtenida del mercado y la actual producida por un optimizador (representada en verde), el cual impone la demanda total de vapor en los generadores de vapor. En la práctica de ingeniería tradicional el total de demanda de vapor entra al sistema de calderas como una referencia (constante durante toda la producción). Un nuevo esquema es introducido en este punto (representado en

6.2 Asignación dinámica en calderas

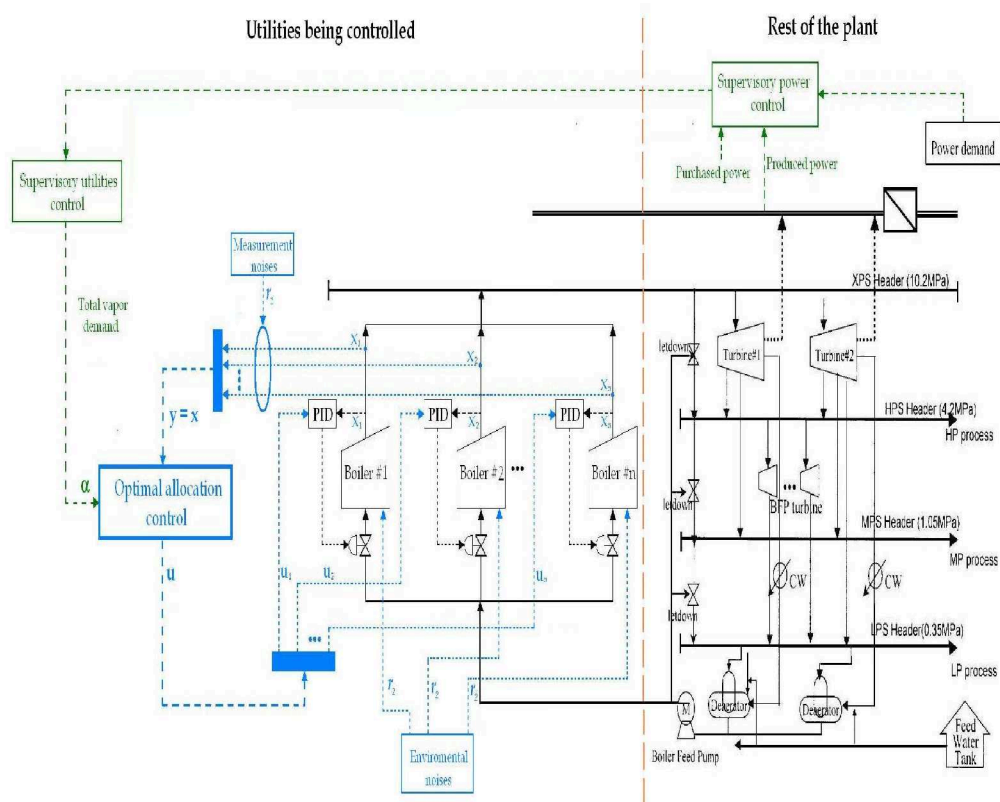


Figura 6.8: Esquema de control para la supervisión con la asignación de la demanda óptima de una planta [92].

azul), donde las referencias (u_1, u_2, \dots, u_n) de cada caldera son autorizados a cambiar en el tiempo por un controlador supervisor llamado control de asignación óptima.

Dos aspectos son revisados en forma novedosa: (i) la metodología para la decisión de las referencias individuales después de que una nueva demanda total es requerida por el grupo, y (ii) la conveniencia de cambiar los órdenes continuamente en el tiempo, para la optimización de algún costo acumulativo durante un tiempo de horizonte finito fijo.

Este tipo de enfoque (aunque a través de órdenes constantes en el tiempo) se ha aplicado a la generación de vapor [22] mientras se reducen la energía perdida al medio ambiente, o de forma equivalente, mientras se maximiza la eficiencia de un conjunto de unidades, definida ésta a través de ciertas relaciones teóricas entre las variables físicas involucradas. Esto da lugar a un problema de programación

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

estática que admite ser gestionado por programación lineal y no lineal; por ejemplo ver [92], la programación dinámica es aplicada para resolver el problema de asignación.

Otros tratamientos estándar siguen líneas de optimización estática, comunes en la investigación operativa de ingeniería (ver por ejemplo [71], [147]). Un punto de vista dinámico sólo ha sido aplicado para situaciones particulares, como en el caso de control redundante y problemas relacionados [70].

La estrategia de realimentación dinámica permite que los valores establecidos para las unidades puedan cambiar de manera continua. Se identifica la dinámica de las respuestas de cada unidad a un nuevo punto de referencia, directamente a partir de datos experimentales. El conjunto de n unidades adoptan un modelo general con un vector de control de $(n - 1)$ -dimensiones asociado con las demandas de vapor, con el control restante determinado por el residuo con respecto a la demanda total, conocido para cada horizonte de tiempo. Este nuevo sistema ampliado, junto con un funcional de costo cuadrático típico como el de la ecuación (2.27) conforman un problema de control óptimo para un sistema afín (6.25) y conduce a una ley ‘feedback’ con los coeficientes variantes en el tiempo. Tanto el coeficiente proporcional y el término de alimentación directa en la acción de control necesitan ser calculados una sola vez para una demanda global unitaria. La naturaleza del modelado también admite un manejo óptimo estocástico de ruido y perturbaciones sistémicas.

6.2.1 Configuración teórica

6.2.1.1 Modelado en espacio de estados para las unidades

Se asume que un grupo de equipos de servicio está en funcionamiento como una parte de una planta industrial. Estas unidades trabajan en paralelo, evolucionan en un rango admisible de sus principales variables, y cada miembro es controlado eficientemente, de acuerdo a prácticas de ingeniería convencionales para satisfacer la demanda asignada.

Es común aceptar que la demanda de cada unidad es en general no lineal [22, 72], de la forma de la ecuación (2.1). Denotando la variable manipulada como v ,

cuando una nueva consigna (demanda) u para la producción x_1 es recibida, entonces la variable manipulada sigue alguna estrategia de control finamente sintonizada

$$v = k(t, u) , \quad (6.21)$$

que maneja ‘eficientemente’ x_1 hacia u en el debido tiempo.

La ‘producción’ (estado x_1) esta relacionado con el ‘servicio’ requerido del equipo. Por ejemplo, si la unidad fuera una caldera, entonces el valor $x_1(t)$ deberá reflejar la cantidad de vapor producido por la caldera al tiempo t . De esto se deduce que también habrá un ‘gasto’ o ‘trabajo’ (estado x_2), necesario para que la unidad realice el servicio; otra vez para una caldera, $x_2(t)$ deberá describir la cantidad de combustible que la unidad consume al tiempo t .

Como consecuencia de aplicar las estrategias de control eficientes k , se puede asumir razonablemente que la dinámica de las variables x_1, x_2, u resulta aproximadamente lineal, es decir, que la unidad controlada deberá reaccionar casi proporcionalmente a las estrategias de control bajo diferentes puntos de ajuste admisibles. Esta hipótesis ha sido corroborada por datos experimentales en [22, 47, 72, 147], con lo cual se justifica adoptar un modelo lineal de la forma (2.28). Allí se redefine $x := (x_1, x_2)'$, por lo que A es una matriz de 2×2 , B es un vector columna de 2 dimensiones y u obra como la nueva variable de control. Los coeficientes son identificados a partir de los datos registrados de la planta (ver [47]).

Las funciones de demanda son entendidas como una consigna comunicada por el supervisor a cada unidad. En la literatura de ingeniería se mantiene constante (al prescribir un valor fijo de u , igual a la porción de la producción requerida del total del conjunto de las unidades de servicio) durante un período de tiempo, hasta que una nueva demanda sea decidida, es decir, u ha tomado la forma de una función constante a trozos. Dado que la capacidad para seguir las demandas individuales es preciso y rápido, no hay ningún impedimento técnico para admitir una función continua por trozos $u(\cdot)$ para las demandas individuales, buscando mejorar el rendimiento de todo el grupo. Se propone aquí, entonces, cambiar continuamente la referencia $u(t)$ para la operación de cada unidad, con el fin de alcanzar la demanda total asignada al grupo mientras es optimizado un criterio de costo económico.

Debe quedar claro que, para tratar de lograr este objetivo, la instrumentación de control convencional ya se ha logrado en cada unidad y debe ser preservado y

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

sujeto a las normas de mantenimiento estándar. Estos instrumentos por lo general son ampliamente testeados e implementados con controladores PID que siguen las estrategias (6.21) mencionadas anteriormente; la sintonización no es objeto de análisis. Sin embargo, en lo que sigue los objetos a optimizar serán los costos de producción de las unidades, cuyas nuevas entradas serán en cada caso una demanda variante en el tiempo $u(\cdot)$, que a su vez afecta la evolución de la nueva variable de estados $x(\cdot)$ como en un sistema de control lineal.

En el caso general, n unidades en paralelo son optimizadas. La identificación del modelo lineal para cada una se expresa como

$$\dot{x}_i = A_i x_i + B_i u_i, \quad y_i = x_i, \quad i = 1, \dots, n; \quad (6.22)$$

donde $x_i := (x_{i1}, x_{i2})'$ es el vector de estados para la unidad i , x_{i1} : variable producida, x_{i2} : variable consumida, u_i : valor demandado, y y_i : la salida, equivalente en lo que sigue al estado producido. Para cada caldera las matrices A_i, B_i son 2×2 y 2×1 , respectivamente.

Cuando una demanda total α es requerida para ser suministrada por las n calderas, entonces necesariamente

$$u_n = \alpha - u_1 - \dots - u_{n-1}. \quad (6.23)$$

Por lo tanto, a pesar del hecho de que n son los pedidos a las n unidades, sólo existe $n - 1$ grados de libertad para tratar el conjunto total. Esto conduce a una nueva configuración para el problema de asignación óptima a n unidades trabajando en paralelo. Redefiniendo

$$x := \left(x'_1 \vdots x'_2 \vdots \dots \vdots x'_n \right)' ; \quad u := (u_1, \dots, u_{n-1})' ; \quad (6.24)$$

la dinámica para el conjunto de calderas, bajo las restricciones impuestas por la ecuación (6.22), se convierte en un sistema lineal afín de $2n$ dimensiones con una variable de control con $(n - 1)$ dimensiones, esto es

$$\dot{x} = \hat{A}x + \hat{B}u + \varphi. \quad (6.25)$$

con los coeficientes de las matrices

$$\hat{A} := \begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & \vdots \\ \vdots & \cdots & A_{n-1} & 0 \\ 0 & \cdots & 0 & A_n \end{pmatrix}, \hat{B} := \begin{pmatrix} B_1 & \cdots & 0 \\ 0 & \ddots & \vdots \\ \vdots & \cdots & B_{n-1} \\ -B_n & \cdots & -B_n \end{pmatrix}, \varphi := \alpha \begin{pmatrix} 0 \\ \vdots \\ 0 \\ B_n \end{pmatrix}, \quad (6.26)$$

Asumiendo que cada modelo en la ecuación (6.22) está en su forma canónica (controlable y observable), se deduce de la forma de la matriz \hat{A} que el sistema global será bounded-input bounded-output (UBIBO) si y sólo si lo mismo es cierto para cada subsistema (unidad). Por lo tanto bajo estas suposiciones, ya que la propiedad UBIBO es equivalente a la estabilidad interna para cada caldera, el sistema global será internamente estable cuando cada matriz identificada A_i sea estable (sus autovalores tengan parte real negativa). Se puede entonces concluir que las propiedades de estabilidad del sistema de dimension $2n$, serán dependientes de las de cada una de las unidades controladas.

6.2.1.2 Asignación óptima como un problema LQR

El problema de control óptimo para el grupo de unidades de servicio expuesto en 6.2.1.1 será tratado con el fin de minimizar un funcional de costo cuadrático (2.27) modificado para tener en cuenta el seguimiento de trayectoria. El funcional reescrito queda de la siguiente forma:

$$\mathcal{J}(u) = \int_0^{t_f} [(x(t) - \bar{x})' Q (x(t) - \bar{x}) + u'(t) R u(t)] dt + (x(t_f) - \bar{x})' S (x(t_f) - \bar{x}), \quad (6.27)$$

sujeto al sistema de ecuaciones afín (6.25- 6.26). El valor de referencia

$$\bar{x} := \left(\bar{x}'_1 : \bar{x}'_2 : \cdots : \bar{x}'_n \right)' \quad (6.28)$$

es un vector de parámetros de diseño de dimension $2n$, cuyos componentes pueden ser interpretados como el promedio de los valores deseados de la producción y los gastos respectivamente, para las unidades $i = 1, 2, \dots, n$.

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

De manera usual Q y S son matrices simétricas no negativas de $2n \times 2n$, y R de $(n-1) \times (n-1)$ definida positiva. Los coeficientes de Q, R, S se adoptan usualmente de acuerdo a las características de la aplicación (ver la deducción más adelante). Las matrices Q y S pueden ser construidas a partir de submatrices simétricas de 2×2 desplegadas en diagonal, tal que cada submatriz este relacionada a cada unidad. Del mismo modo, $R = \text{diag}(r_1, \dots, r_{n-1})$, con todos los $r_i > 0$, deberá ser una elección práctica para ponderar el esfuerzo de la energía de control asociada al problema. Nótese que existen n unidades y sólo $n-1$ coeficientes disponibles r_i , una posible opción es adoptar

$$r_i = \tilde{r}_i + \frac{\tilde{r}_n}{n-1}, \quad i = 1, \dots, n-1, \quad (6.29)$$

donde \tilde{r}_i es el peso de control correspondiente a la demanda asignada a cada unidad i , para $i = 1, \dots, n$. La fórmula (6.29) deberá entonces satisfacer

$$E_u := \sum_{i=1}^{n-1} r_i u_i^2 = \sum_{i=1}^n \tilde{r}_i u_i^2 := \tilde{E}_u \quad (6.30)$$

cuando $u_i \equiv \alpha/n$ (una elección típica para \bar{x}_{i1}), $i = 1, \dots, n$, y E_u deberá ser una buena aproximación de \tilde{E}_u para los valores generales u_i bajo las siguientes restricciones: (i) $u_i \in [0, \alpha]$, y (ii) $u_1 + \dots + u_n = \alpha$, en este caso. Como es de esperar, esta elección de r_i siempre verifica: $\lim_{n \rightarrow \infty} r_i = \tilde{r}_i$, para $i = 1, \dots, n$.

Una pregunta interesante concerniente al problema de control óptimo es cómo elegir los parámetros Q, R y S involucrados en el funcional del costo. La elección de estas matrices deberá llevar a niveles ‘aceptables’ de $x(t)$, $u(t)$, y $x(t_f)$. Un enfoque clásico [21] inicializa Q_{ii} , R_{ii} y S_{ii} tentativamente, y luego modifica estos valores por ensayo y error, hasta alcanzar un compromiso entre el tiempo de respuesta y el esfuerzo de control. En literatura mas reciente hay varios artículos que cubren este tema, por ejemplo [54, 126]. Aquí, los pesos son adoptados por una diagonal para Q y S , de modo que pesan la cantidad de desviación de cada estado y de la entrada contribuye al costo total, esto es

$$Q = \text{diag}(q_1, q_2, \dots, q_1, q_2), \quad S = \text{diag}(s_1, s_2, \dots, s_1, s_2). \quad (6.31)$$

El siguiente índice

$$\eta = \frac{\sum_{i=1}^n (x_{i1}(t_f) - x_{i1}(0))}{\int_0^{t_f} \sum_{i=1}^n x_{i2}(t) dt} \quad (6.32)$$

será usado para sintonizar los parámetros Q , S , y R de tal forma que η es maximizado. Este índice puede ser visto como una medida de la eficiencia durante un período t_f , puesto que se refiere a la relación entre la variable producción neta de vapor y el consumo total de la variable de combustible.

Con el fin de aprovechar todas las herramientas existentes en sistemas lineales, el sistema lineal afín (6.25) que describe la dinámica lineal del grupo se transformará en uno lineal mediante la introducción del siguiente vector z .

$$z(t) := \int_0^t e^{\hat{A}(t-\tau)} \varphi(\tau) d\tau, \quad (6.33)$$

que satisface el problema de valor inicial

$$\dot{z}(t) = \hat{A}z(t) + \varphi(t), \quad z(0) = 0, \quad (6.34)$$

Después de hacer un cambio de variables $x(t) \rightarrow x(t) - z(t)$, y $r(t) \rightarrow \bar{x} - z(t)$, la dinámica de control se convierte, para la ‘nueva’ variable de estado x , en

$$\dot{x} = \hat{A}x + \hat{B}u, \quad x(0) = x_0, \quad (6.35)$$

y la función de costo objetivo (6.27) puede ser escrita en la forma:

$$\mathcal{J}(u) = \int_0^{t_f} [(x - r)'Q(x - r) + u'Ru] dt + (x(t_f) - r(t_f))' S (x(t_f) - r(t_f)), \quad (6.36)$$

Ahora, el problema definido por la ecuaciones (6.35, 6.36) es equivalente al original expresado en (6.25, 6.27). El problema de tracking óptimo tiene como objetivo conducir el nuevo estado x , ahora gobernado por una dinámica lineal, hacia la trayectoria de referencia recientemente definida $r(t)$. El rendimiento del control puede ser evaluado por un costo cuadrático con los mismos coeficientes que el problema original.

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

El problema de tracking lineal cuadrático tiene algunas características del problema LQR clásico. Por ejemplo, su Hamiltoniano \mathcal{H}

$$\mathcal{H}(t, x, \lambda, u) = (x - r)'Q(x - r) + u'Ru + \lambda'(\hat{A}x + \hat{B}u) \quad (6.37)$$

se minimiza por la misma expresión que en el caso LQR (ecuación (2.32)); con el u mínimo el Hamiltoniano \mathcal{H}^0 resultante es

$$\mathcal{H}^0(t, x, \lambda) = (x - r)'Q(x - r) - \frac{1}{4}\lambda'\hat{W}\lambda + \lambda'\hat{A}x. \quad (6.38)$$

El término afín induce una función de valor cuadrática \mathcal{V}

$$\mathcal{V}(t, x) := x'P(t)x + 2\xi'(t)x + \sigma(t), \quad (6.39)$$

con coeficientes variantes en el tiempo: una matriz P (en principio simétrica) de $2n \times 2n$, un vector columna ξ de dimensión n , y un factor escalar σ . La ecuación de HJB (2.19) tiene en este caso la siguiente condición final

$$\mathcal{V}(t_f, x) = (x(t_f) - r(t_f))'S(x(t_f) - r(t_f)) \quad (6.40)$$

y debe ser satisfecha por la función de valor propuesta y sus derivadas

$$\frac{\partial \mathcal{V}}{\partial t}(t, x) = x'\dot{P}(t)x + 2x'\dot{\xi}(t) + \dot{\sigma}(t), \quad (6.41)$$

$$\left(\frac{\partial \mathcal{V}}{\partial x}\right)'(t, x) = 2[P(t)x + \xi(t)], \quad (6.42)$$

lo cual requiere para todos los (t, x) admisibles,

$$\begin{aligned} & x'\dot{P}(t)x + 2x'\dot{\xi}(t) + \dot{\sigma}(t) = \\ & -(x - r)'Q(x - r) + [P(t)x + \xi(t)]'\hat{W}[P(t)x + \xi(t)] - 2[P(t)x + \xi(t)]'\hat{A}x, \end{aligned} \quad (6.43)$$

y al final del tiempo horizonte

$$x'Sx - 2Sr(t_f) + r(t_f)'Sr(t_f) = x'P(t_f)x + 2\xi'(t_f)x + \sigma(t_f). \quad (6.44)$$

Dado que estas igualdades involucran polinomios de segundo orden en x , sus coeficientes deberán ser iguales, resultando el siguiente sistemas de ODE's:

$$\dot{P} = P\hat{W}P - \hat{A}'P - P\hat{A} - Q, \quad P(t_f) = S, \quad (6.45)$$

$$\dot{\xi} = -(\hat{A} - \hat{W}P)'\xi + Qr, \quad \xi(t_f) = -Sr(t_f), \quad (6.46)$$

$$\dot{\sigma} = \xi'\hat{W}\xi - r'Qr, \quad \sigma(t_f) = r(t_f)'Sr(t_f). \quad (6.47)$$

Sus soluciones permiten expresar el control óptimo u^* en la forma

$$u^*(t) = -R^{-1}\hat{B}' [P(t)x^*(t) + \xi(t)] , \quad (6.48)$$

donde x^* denota el estado óptimo. El costo óptimo es calculado a partir de la función de valor (o sea de los coeficientes P, ξ, σ) de la siguiente manera

$$\mathcal{V}(0, x_0) = x_0'P(0)x_0 + 2x_0'\xi(0) + \sigma(0) . \quad (6.49)$$

La ecuación (6.45) coincide con la RDE para el problema LQR con coeficientes $(\hat{A}, \hat{B}, Q, R, S)$, y se encuentra desacoplada de las dos ODE's restantes. Es importante remarcar que a pesar de que el problema tratado no posee la estructura LQR, la ecuación (6.48) puede ser interpretada como una ley de feedback u_f para el sistema lineal afín, es decir

$$u_f(t, x) := -R^{-1}\hat{B}' [P(t)x + \xi(t)] . \quad (6.50)$$

Por el principio de Bellman es posible afirmar que en algún momento t el estado actual x puede diferir del estado óptimo esperado $x^*(t)$, sin embargo, el control óptimo en ese momento, denotado como u_x^* puede ser calculado con $u_x^*(t) = u_f(t, x)$. Esta propiedad hace que los resultados anteriores sean robustos contra errores de estado esporádicos.

6.2.1.2.1 Manejo de los cambios en la demanda total α

Las ecuaciones (6.45, 6.46) tienen condiciones finales (en lugar de iniciales), por lo que no pueden ser resueltas numéricamente en línea con el proceso. Tienen que ser resueltas offline y almacenadas en la memoria del controlador. Esto es un inconveniente común en a los problemas LQR, servo y tracking, para los cuales deben ser actualizados todos los objetos matemáticos relacionados, en todo el horizonte temporal, en caso de que se modifique la señal de referencia. Afortunadamente en este caso, el cálculo de los coeficientes variantes en el tiempo $P(\cdot), \xi(\cdot)$ de la ley de control feedback debe calcularse sólo una vez, para una demanda total unitaria ($\alpha = 1$), del mismo modo que el coeficiente del costo $\sigma(\cdot)$. Estas afirmaciones son cubiertas en términos precisos por las siguientes dos ecuaciones:

$$u^*(t) = -R^{-1}\hat{B}' [P(t)x(t) + \alpha\tilde{\xi}(t)] , \quad (6.51)$$

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

$$\mathcal{J}^* = \mathcal{V}(0, x_0) = x_0' P(0) x_0 + 2\alpha x_0' \tilde{\xi}(0) + \alpha^2 \tilde{\sigma}(0), \quad (6.52)$$

donde $\tilde{\xi}, \tilde{\sigma}$ denotan los coeficientes calculados para $\alpha = 1$, o equivalentemente para $\tilde{z}(t) := z(t)/\alpha, \tilde{r}(t) := r(t)/\alpha$, y $\tilde{x} := \bar{x}/\alpha = (\tilde{x}_1, 0, \tilde{x}_2, 0, \dots, \tilde{x}_n, 0)'$, $\sum_{i=1}^n \tilde{x}_i = 1$. Las ecuaciones (6.51, 6.52) pueden justificarse de la siguiente manera:

(i) la solución de la ecuación (6.46) con condición final $\xi(t_f) = -S r(t_f) = -\alpha S \tilde{r}$ es

$$\xi(t) = \alpha \Psi(t, t_f) \left\{ -S \tilde{r}(t_f) + \int_{t_f}^t \Psi(t, \tau) Q \tilde{r}(\tau) d\tau \right\} = \alpha \tilde{\xi}(t), \quad (6.53)$$

donde $\Psi(t, t_f)$ es la matriz fundamental asociada con el sistema lineal (variante en el tiempo) $\dot{\psi} = -(\hat{A} - \hat{W}P(t))' \psi$;

(ii) de forma similar, para la ecuación (6.47),

$$\sigma(t) = \alpha^2 \left\{ \tilde{r}(t_f)' S \tilde{r}(t_f) + \int_{t_f}^t [\tilde{\xi}'(\tau) \hat{W} \tilde{\xi}(\tau) - \tilde{r}' Q \tilde{r}] d\tau \right\} = \alpha^2 \tilde{\sigma}(t). \quad (6.54)$$

Ahora se debe decidir cómo manejar los cambios en la demanda total cuando ésta se produce en algún momento t del interior de un período $[t_0, t_0 + t_f]$. Se asume que, en tal caso, la optimización del sistema debe ser continuada, al menos para otro intervalo de duración t_f . Algunas decisiones de ‘receding-horizon’ se pueden encontrar en [47].

6.2.1.3 Filtrado óptimo

Se producen perturbaciones o ruidos en los parámetros de cada dispositivo, y en las mediciones y transmisiones de señales durante el funcionamiento de un proceso real. La configuración común para modelar estas influencias sobre los objetos determinísticos es de la siguiente manera

$$\dot{x}(t) = \hat{A}x(t) + \hat{B}\tilde{u}(t) + r_1, \quad (6.55)$$

$$y = \hat{C}x + r_2, \quad (6.56)$$

donde la notación \dot{x} debe entenderse como el diferencial de un proceso Browniano asociado con los estados del grupo, que resulta de la existencia de ruido blanco

6.2 Asignación dinámica en calderas

r_1 de media cero debido a las fluctuaciones en las condiciones del ambiente; \tilde{u} denota la variable de entrada $\tilde{u} = (\tilde{u}_1, \dots, \tilde{u}_{n-1})$, $\tilde{u}_n := \alpha - \sum_{i=1}^{n-1} \tilde{u}_i$; y donde r_2 son los ruidos blancos de media cero medidos en la salida y , que en este caso son conceptualmente lo mismos estados, es decir, se supone que cada subsistema i es observable (lo cual es claramente verdad en las plantas modernas). Obsérvese que y es el vector de salida de todo el grupo.

En este contexto, el problema de mínimos cuadrados del filtrado óptimo para el grupo de unidades es conocido por su solución [61] a través del siguiente par de ecuaciones

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}\tilde{u} + \hat{L}(y - \hat{C}\hat{x}(t)), \quad \hat{x}(0) = \mathbb{E}[x_0] \quad (6.57)$$

$$\dot{\Pi} = \hat{A}\Pi + \Pi\hat{A}' + \hat{Q} - \Pi\hat{D}\Pi; \quad \Pi(0) = Cov(x_0), \quad (6.58)$$

$$\hat{D} := \hat{C}'\hat{R}^{-1}\hat{C}, \quad \hat{L} := \Pi\hat{C}'\hat{R}^{-1}, \quad (6.59)$$

donde \hat{x} es la mejor estimación del estado x ; $\mathbb{E}\{r_1\} = 0$, $\mathbb{E}\{r_1(t).r_1'(\tau)\} = \hat{Q}.\delta(t-\tau)$, con \hat{Q} la matriz de covarianza de r_1 , y $\mathbb{E}\{r_2\} = 0$, $\mathbb{E}\{r_2(t).r_2'(\tau)\} = \hat{R}.\delta(t-\tau)$, con \hat{R} la matriz de covarianza (invertible) de r_2 , \hat{L} denota la ‘ganancia’ del filtro, el cual trabaja análogamente a un observador; x_0 es ahora un vector aleatorio con media $\mathbb{E}\{x_0\} = \hat{x}(0)$ y covarianza inicial $\mathbb{E}\{(x_0 - \hat{x}(0))(x_0 - \hat{x}(0))'\} = \Pi(0)$, y finalmente Π es la covarianza dinámica, solución a la ODE del tipo RDE (6.45). Las matrices \hat{C} , \hat{Q} , \hat{R} , \hat{L} y Π son de $2n \times 2n$; y pueden ser convencionalmente particionadas en la forma

$$G = \begin{pmatrix} \tilde{G}_{11} & \cdots & \tilde{G}_{1n} \\ \vdots & \ddots & \vdots \\ \tilde{G}_{n1} & \cdots & \tilde{G}_{nn} \end{pmatrix}, \quad (6.60)$$

con \tilde{G}_{ij} un bloque de 2×2 . Se adopta por simplicidad, la notación $\tilde{G}_{ii} := \tilde{G}_i$ $i = 1, \dots, n$.

Las siguientes características son naturalmente asumidas por estas matrices:

i) La matriz de observación \hat{C} es

$$\hat{C} := diag(\tilde{C}_1, \dots, \tilde{C}_n) \quad (6.61)$$

y $\tilde{C}_i = I_2$, $i = 1, \dots, n$.

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

ii) Los ruidos que afectan a la salida y los estados son independientes de aquéllos que afectan las otras unidades, por lo que la covarianza de ruidos entre las unidades es cero (cada unidad trabaja en paralelo). Esto se expresa a través de la elección

$$\hat{Q} := \text{diag}(\tilde{Q}_1, \dots, \tilde{Q}_n), \quad \hat{R} := \text{diag}(\tilde{R}_1, \dots, \tilde{R}_n), \quad (6.62)$$

las matrices \hat{Q}_i , y \hat{R}_i son las matrices de covarianza de r_{1i} y r_{2i} en cada unidad respectivamente. Además, como \hat{R} es ahora una matriz diagonal, su inversa es $\hat{R}^{-1} = \text{diag}(\tilde{R}_1^{-1}, \dots, \tilde{R}_n^{-1})$.

iii) Las fluctuaciones en la condición inicial dependen sólo del entorno de cada unidad, entonces la covarianza inicial de la estimación es:

$$\Pi(0) := \text{diag}(\tilde{\Pi}_1(0), \dots, \tilde{\Pi}_n(0)) = \text{diag}(\text{Cov}(x_{0_1}), \dots, \text{Cov}(x_{0_n})). \quad (6.63)$$

Notar que la ecuación (6.58), en la forma particionada, resulta

$$\dot{\tilde{\Pi}}_{ij} = \tilde{A}_{ik}\tilde{\Pi}_{kj} + \tilde{\Pi}_{ik}\tilde{A}'_{kj} - \tilde{\Pi}_{ik}\tilde{D}_{kl}\tilde{\Pi}_{lj} + \tilde{Q}_{ij}, \quad \tilde{\Pi}_{ij}(0) = \text{Cov}(x_{0_i})\delta_{ij}, \quad (6.64)$$

donde $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{otro caso} \end{cases}$, y $i, j, k, l = 1, \dots, n$, y los índices repetidos se suman de 1 a n (notación de Einstein).

Los términos que no están en la diagonal de Π son irrelevantes, esto se muestra a continuación. Considere el caso $i \neq j$, y recordando que \hat{A} , \hat{D} , y \hat{Q} son matrices ‘diagonales’ (en el sentido de la ecuación (6.60)), entonces la ecuación (6.64) se convierte en

$$\dot{\tilde{\Pi}}_{ij} = \tilde{A}_i\tilde{\Pi}_{ij} + \tilde{\Pi}_{ij}\tilde{A}'_j - \tilde{\Pi}_{ik}\tilde{D}_k\tilde{\Pi}_{kj}, \quad \tilde{\Pi}_{ij}(0) = 0. \quad (6.65)$$

Vamos a mostrar que las funciones $\tilde{\Pi}_{ij}(t) \equiv 0$ satisfacen la ecuación (6.65), lo cual nos permitirá descartar los términos no diagonales de (6.64). Notar que los términos $\tilde{A}_i\tilde{\Pi}_{ij}$, y $\tilde{\Pi}_{ij}\tilde{A}'_j$ desaparecen cuando se aplica esta solución. El término restante $Z_{ij} = \tilde{\Pi}_{ik}\tilde{D}_k\tilde{\Pi}_{kj}$ es también cero, porque haciendo $k = i$, $\tilde{\Pi}_i\tilde{D}_i\tilde{\Pi}_{ij} = 0$ debido al último $\tilde{\Pi}_{ij}$, y para $k = j$, también $\tilde{\Pi}_{ij}\tilde{D}_j\tilde{\Pi}_j = 0$ debido al primer $\tilde{\Pi}_{ij}$. Además para $k \neq i, j$ el resultado es evidente. Por lo tanto, para $i \neq j$, la conclusión es que $Z_{ij} = 0$.

Ahora, considerando el caso $i = j$, la ecuación (6.64) se puede escribir

$$\dot{\tilde{\Pi}}_i = \tilde{A}_i \tilde{\Pi}_i + \tilde{\Pi}_i \tilde{A}'_i + \tilde{Q}_i - \tilde{\Pi}_i \tilde{D}_i \tilde{\Pi}_i, \quad \tilde{\Pi}_i(0) = Cov(x_{0_i}), \quad (6.66)$$

que sólo depende de i , por lo tanto sólo los objetos de la unidad i están involucrados. Como consecuencia, $\Pi(t) = diag(\tilde{\Pi}_1(t), \dots, \tilde{\Pi}_n(t))$ es solución de la ecuación (6.58) para todo el grupo.

Dado que la ganancia del filtro se puede escribir como $\hat{L} = diag(L_1, \dots, L_n)$, donde $L_i = \Pi_i(t) C'_i \hat{R}_i^{-1}$ para $i = 1, \dots, n$, entonces cada vector de estado $\hat{x}_i(t)$ en la ecuación (6.57) verifica

$$\dot{\hat{x}}_i(t) = A_i \hat{x}_i(t) + B_i \tilde{u}_i(t) + L_i (y_i(t) - C_i \hat{x}_i(t)); \quad \hat{x}_i(0) = \mathbb{E}[x_{0_i}], \quad (6.67)$$

independientemente de las otras unidades.

La consecuencia práctica de esta configuración es que los problemas de filtrado y estimación pueden ser resueltos para cada unidad del grupo independientemente, evitando así la integración de ecuaciones diferenciales acopladas de $2n \times 2n$.

6.2.1.4 Control estocástico

En esta subsección se mostrará cómo la ley de control derivada del contexto determinístico es también 'óptima' bajo perturbaciones aleatorias en aparatos de medida y en los parámetros del modelo. Esta optimalidad del control se entiende en el sentido estocástico, es decir, teniendo en cuenta que las perturbaciones aleatorias fuerzan al modelo, especialmente a los estados $x(\cdot)$, a ser considerados como un proceso estocástico.

El problema de control óptimo estocástico intenta minimizar el funcional

$$\mathcal{J}_{est}(u) = \mathbb{E} \left\{ \int_0^{t_f} \{(x_u - r)' Q (x_u - r) + u' R u\} dt + (x_u(t_f) - r(t_f))' S (x_u(t_f) - r(t_f)) \right\} \quad (6.68)$$

con respecto al control determinístico $u(\cdot)$, donde \mathbb{E} denota el valor usual esperado, $r(t)$ es una trayectoria de referencia determinista dada, y sujeto a la restricción dinámica

$$\dot{x} = \hat{A}x + \hat{B}u + r_1, \quad x(0) = x_0, \quad (6.69)$$

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

donde la variable de entrada $u = (u_1, \dots, u_{n-1})$, $u_n := \alpha - \sum_{i=1}^{n-1} u_i$, y x_0 es un vector aleatorio con media $\mathbb{E}\{x_0\} = \hat{x}(0)$ y covarianza inicial $\mathbb{E}\{(x_0 - \hat{x}(0))(x_0 - \hat{x}(0))'\} = \Lambda_0 = \Pi(0)$. Las variables \hat{x} , y Π son la solución a los problemas de filtrado y de estimación discutidos en la subsección 6.2.1.3.

Para el caso determinístico (en la subsección 6.2.1.2), se demostró que el control óptimo es como en la ecuación (6.48), donde P, ξ son los coeficientes óptimos deterministas descritos por (6.45) y (6.46). Por analogía con el caso del regulador, la hipótesis de separación propone que el control óptimo para la configuración estocástica del tracking sea

$$\hat{u}^* := -R^{-1}\hat{B}' [P(t)\hat{x}(t) + \xi(t)]. \quad (6.70)$$

A continuación, la hipótesis será probada, y, además, se muestra que el costo óptimo es

$$\mathcal{J}_{est}^*(\hat{u}^*) = \mathcal{J}_{det}^*(x_0) + \int_0^{t_f} Tr(PQ)dt + Tr(P(0)\Lambda_0) + \int_0^{t_f} Tr(P\hat{W}P\Pi)dt. \quad (6.71)$$

donde $\mathcal{J}_{det}^*(x_0) = \mathcal{V}(0, x_0)$ (ver ecuación (6.49)). La prueba requiere el siguiente cambio de variable

$$u := \hat{u} + \tilde{u}, \quad (6.72)$$

y re-expresar la dinámica y el funcional de costo. La dinámica resulta

$$\dot{x} = (A - \hat{W}P)x + \hat{W}P(x - \hat{x}) - \hat{W}\xi + B\tilde{u} + r_1. \quad (6.73)$$

si $\tilde{A} := (A - \hat{W}P)$, y $\tilde{x} := (x - \hat{x})$, entonces la ecuación (6.73) se convierte en

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \hat{W}P\tilde{x} - \hat{W}\xi + B\tilde{u} + r_1.$$

De la ecuación (6.45) y de las transformaciones posteriores se obtienen las siguientes igualdades:

$$Q + P\hat{W}P = -\dot{P} - P\tilde{A} - \tilde{A}P, \quad (6.74)$$

$$x'Qx = -x'(\dot{P} + P\tilde{A} + \tilde{A}P)x - x'(P\hat{W}P)x. \quad (6.75)$$

Los siguientes cálculos involucrados en los términos del funcional del costo son necesarios:

$$(\hat{u} + \tilde{u})'R(\hat{u} + \tilde{u}) = \hat{u}'R\hat{u} + 2\hat{u}'R\tilde{u} + \tilde{u}'R\tilde{u}, \quad (6.76)$$

$$\hat{u}'R\hat{u} = \hat{x}'P\hat{W}P\hat{x} + 2\hat{x}'P\hat{W}\xi + \xi'\hat{W}\xi, \quad (6.77)$$

$$\hat{x}'P\hat{W}P\hat{x} = x'P\hat{W}Px + \tilde{x}'P\hat{W}P\tilde{x} - 2x'P\hat{W}P\tilde{x}, \quad (6.78)$$

$$2\hat{x}'P\hat{W}\xi = 2x'P\hat{W}\xi - 2\tilde{x}'P\hat{W}\xi, \quad (6.79)$$

$$\begin{aligned} 2\hat{u}'R\tilde{u} &= -2\hat{x}'PB\tilde{u} - 2\xi'B\tilde{u}, \\ &= -2x'PB\tilde{u} - 2\tilde{x}'PB\tilde{u} - 2\xi'B\tilde{u}. \end{aligned} \quad (6.80)$$

De la ecuación (6.46), la expresión $\dot{\xi} + \tilde{A}\xi = Qr$ es obtenida, y entonces

$$(x - r)'Q(x - r) = x'Qx - 2x'\dot{\xi} - 2x'\tilde{A}\xi + r'Qr. \quad (6.81)$$

El funcional de costo (6.68), después de realizar algunas manipulaciones algebraicas, se convierte en $\mathcal{J}_{est}(\tilde{u}) = \mathbb{E}\{\mathcal{J}_{est}^1 + \mathcal{J}_{est}^2 + \mathcal{J}_{est}^3\}$, donde

$$\begin{aligned} \mathcal{J}_{est}^1 &= \int_0^{t_f} \left(-x'\dot{P}x - 2x'Px + 2x'Pr_1 + \tilde{x}'P\hat{W}P\tilde{x} - 2\tilde{x}'PB\tilde{u} \right) dt \\ &\quad + x(t_f)'Sx(t_f), \end{aligned} \quad (6.82)$$

$$\mathcal{J}_{est}^2 = - \int_0^{t_f} \left(2\tilde{x}'P\hat{W}\xi + 2\xi'B\tilde{u} + 2\xi'\tilde{A}x + 2x'\dot{\xi} - \xi'\hat{W}\xi - r'\hat{Q}r \right) dt, \quad (6.83)$$

$$\mathcal{J}_{est}^3 = \int_0^{t_f} \tilde{u}'R\tilde{u} dt + r(t_f)'Sr(t_f) - 2x(t_f)'Sr(t_f). \quad (6.84)$$

En \mathcal{J}_{est}^1 , teniendo en cuenta que el error \tilde{x} es ortogonal a las mediciones y , y a la estimación \hat{x} , resulta que (ver [111]),

$$\mathbb{E} \left\{ \int_0^{t_f} \tilde{x}'PB\tilde{u} dt \right\} = 0, \quad \mathbb{E} \left\{ \int_0^{t_f} x'Pr_1 dt \right\} = 0, \quad (6.85)$$

y por las propiedades integrales de Ito [111]

$$d(x'Px) = x'\dot{P}xdt + 2x'Pdx + Tr(P\hat{Q}) dt. \quad (6.86)$$

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

y

$$\mathbb{E} \left\{ \int_0^{t_f} \tilde{x}' P \hat{W} P \tilde{x} dt \right\} = \int_0^{t_f} Tr(P \hat{W} P \Pi) dt. \quad (6.87)$$

El primer termino \mathcal{J}_{sto}^1 se puede operar para obtener

$$\begin{aligned} \mathbb{E} \{ \mathcal{J}_{est}^1 \} &= \mathbb{E} \left\{ \int_0^{t_f} -\frac{d}{dt} (x' P x) dt + x(t_f)' S x(t_f) + \int_0^{t_f} Tr(P \hat{Q}) dt + \int_0^{t_f} \tilde{x}' P \hat{W} P \tilde{x} dt \right\} \\ &= \mathbb{E} \left\{ x(0)' P(0) x(0) + \int_0^{t_f} Tr(P \hat{Q}) dt + \int_0^{t_f} \tilde{x}' P \hat{W} P \tilde{x} dt \right\} \\ &= x(0)' P(0) x(0) + Tr(P(0) \Lambda_0) + \int_0^{t_f} Tr(P \hat{Q}) dt + \int_0^{t_f} Tr(P \hat{W} P \Pi) dt. \end{aligned} \quad (6.88)$$

Ahora, el término $\tilde{A}x$ puede ser remplazado por $\tilde{A}x = \dot{x} - W P \tilde{x} - B \tilde{u} + W \xi - r_1$ en la ecuación (6.83), y por lo tanto el segundo término en el funcional de costo se convierte en

$$\mathcal{J}_{est}^2 = - \int_0^{t_f} \left(2\xi' (\dot{x} + W \xi - r_1) - \xi' W \xi + 2x' \dot{\xi} - r' \hat{Q} r \right) dt \quad (6.89)$$

$$= - \int_0^{t_f} \left(2\xi' \dot{x} - 2\xi' r_1 + 2x' \dot{\xi} + \xi' W \xi - r' \hat{Q} r \right) dt, \quad (6.90)$$

De la ecuación (6.47), y $\dot{\sigma} = \xi' W \xi - r' \hat{Q} r$

$$\mathcal{J}_{est}^2 = - \int_0^{t_f} \left(2\xi' \dot{x} - 2\xi' r_1 + 2x' \dot{\xi} + \dot{\sigma} \right) dt. \quad (6.91)$$

Ahora, a partir del cálculo de Ito [111],

$$\mathbb{E} \left\{ \int_0^{t_f} \xi' r_1 dt \right\} = 0, \quad (6.92)$$

$$d(2x' \xi) = 2x' \dot{\xi} dt + 2\xi' dx. \quad (6.93)$$

Mediante la introducción de las últimas implicaciones y después de la adición \mathcal{J}_{est}^2 a \mathcal{J}_{est}^3 , la siguiente ecuación es obtenida

$$\begin{aligned} & \mathbb{E}\{\mathcal{J}_{est}^2 + \mathcal{J}_{est}^3\} = \tag{6.94} \\ & = \mathbb{E}\left\{ \int_0^T -\frac{d}{dt}(2x'\xi)dt - 2x(T)'Sr(T) - \int_0^T \frac{d}{dt}(\sigma)dt + r(T)'Sr(T) + \int_0^T \tilde{u}'R\tilde{u}dt \right\} \\ & = 2x(0)'\xi(0) + \sigma(0) + \int_0^T \tilde{u}'R\tilde{u} dt. \tag{6.95} \end{aligned}$$

Finalmente, si a la última ecuación se le agrega \mathcal{J}_{est}^1 , el funcional de costo total resulta

$$\begin{aligned} \mathcal{J}_{est}(\tilde{u}) & = \mathbb{E}\left\{ \int_0^{t_f} \tilde{u}'R\tilde{u} dt \right\} + x(0)'P(0)x(0) + 2x(0)'\xi(0) + \sigma(0) \\ & \quad + \int_0^{t_f} Tr(P\hat{Q}) dt + Tr(P(0)\Lambda_0) + \int_0^{t_f} Tr(P\hat{W}P\Pi)dt, \tag{6.96} \end{aligned}$$

el cual alcanza su valor mínimo si y solo si $\tilde{u} = 0$. Esto prueba la hipótesis y establece el ‘principio de separación para el problema de tracking estocástico LQ’.

6.2.2 Aplicación numérica: Dos calderas en paralelo

Consideremos el siguiente sistema de dos calderas produciendo vapor en paralelo:

$$\begin{aligned} A_1 & = \begin{pmatrix} -0,112 & 0,05 \\ -0,2 & -0,09 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -0,2 & 0,05 \\ -0,2 & -0,155 \end{pmatrix}, \tag{6.97} \\ B_1 & = \begin{pmatrix} 0,05 \\ 0,315 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 0,125 \\ 0,4 \end{pmatrix}, \quad C_1 = C_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

Estas realizaciones son canónicas, los autovalores de A_1, A_2 son $-0,1013 \pm 0,0993i$ y $-0,1775 \pm 0,0974i$, respectivamente, teniendo la primera caldera una respuesta mas rápida que la segunda. Para cada caldera, la salida en estado estacionario y_∞ correspondiente a una entrada constante $u(\cdot) \equiv k$ es $y_i^\infty = -A_i^{-1}B_i k$,

$$y_1^\infty = \begin{pmatrix} 1,00847 \\ 1,25896 \end{pmatrix} k, \quad y_2^\infty = \begin{pmatrix} 0,960366 \\ 1,34146 \end{pmatrix} k. \tag{6.98}$$

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

Esto demuestra que la caldera mas lenta (segunda) está sujeta a un off-set (primer componente de y^∞) de alrededor del 4%, mucho más grande que el off-set de la primera. Sin embargo, los coeficientes de cada realización transmiten suficiente información sobre el rendimiento esperado de cada caldera controlada con el PID. Por ejemplo, una medida de la eficiencia estática de cada unidad puede ser calculada como

$$\eta_i^\infty = \frac{y_i^\infty(1)}{y_i^\infty(2)} = \frac{x_{2i-1}^\infty}{x_{2i}^\infty}. \quad (6.99)$$

6.2.2.1 Parámetros de la función de costo para el control determinista

Como se anunció en la sección 6.2.1.2 el índice η fue maximizado con el fin de encontrar los mejores valores de Q , R y S para ser usados en el funcional del costo, y por simplicidad se toma $Q = S$. El horizonte de optimización es fijado en $t_f = 20$. En la figura 6.9 se ilustra variaciones de los parámetros q_1, q_2 para determinar el valor del índice donde se localice el máximo. Éste es encontrado en $q_1 = 3,6$ y $q_2 = 0,1$. En la figura 6.10 se muestran variaciones de los parámetros q_1, r , y el máximo es alcanzado para el valor de $r = 1,41$.

Estos valores fueron los utilizados para llevar a cabo la simulación: en primer lugar P, ξ, σ , son calculados, y luego introducidos en la dinámica para obtener el control óptimo u^* que se muestra en la figura 6.12. Los parámetros utilizados en la simulación son

$$r = 1,41, q_1 = s_1 = 3,6, q_2 = s_2 = 0,1, \tilde{x} = 0,5, t_f = 20, \alpha = 150.$$

Durante la etapa inicial del proceso, el control resultante pone una gran demanda en la segunda caldera. En el medio, los controles de ambas calderas evolucionan alrededor de $\alpha/2$ como un patrón no trivial, y al final del tiempo la segunda caldera es preferida de nuevo. La evolución de los estados resultantes de aplicar el control óptimo es ilustrado en la figura 6.11. El costo óptimo es $J^* = 209706$.

6.2.2.2 Simulaciones de control estocástico.

Las simulaciones realizadas para el caso de las dos calderas, en presencia de perturbaciones en la señal y de ruido ambiental, se ilustran en las figuras 6.13 y 6.14.

6.2 Asignación dinámica en calderas

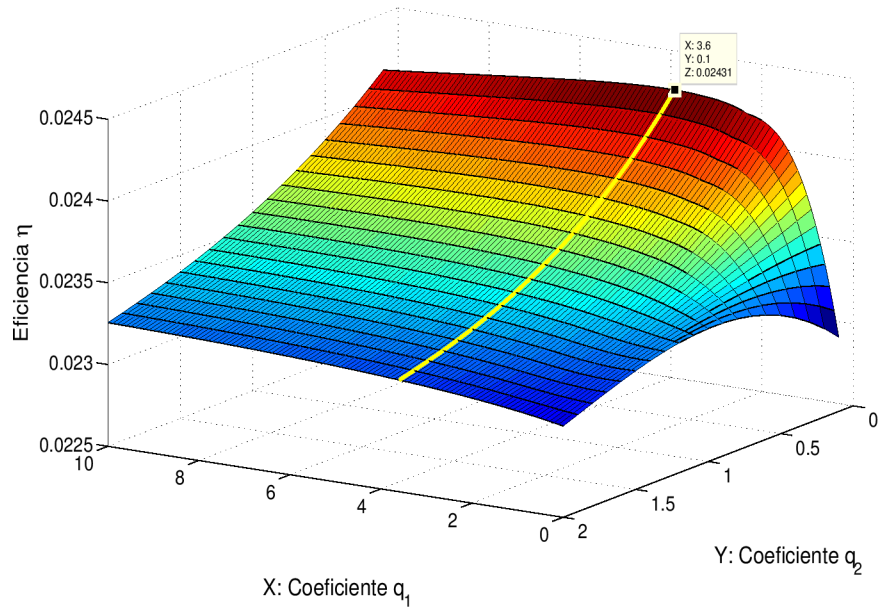


Figura 6.9: η versus q_1 y q_2 .

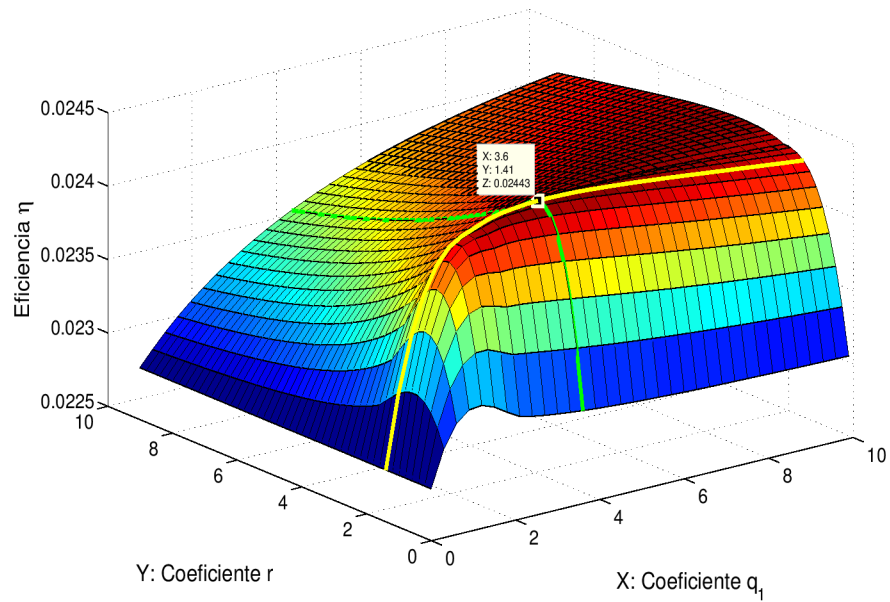


Figura 6.10: η versus q_1 y r .

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

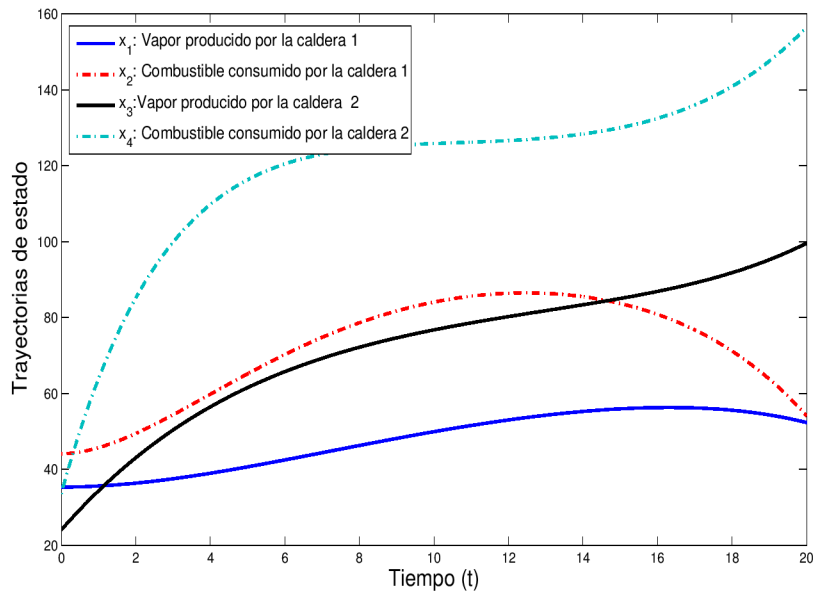


Figura 6.11: Trayectorias de estados.

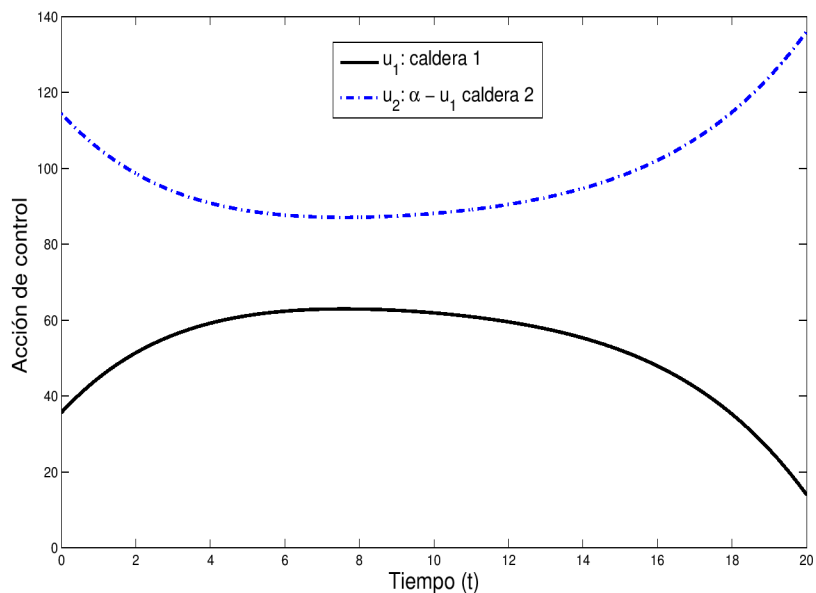


Figura 6.12: Trayectorias de control.

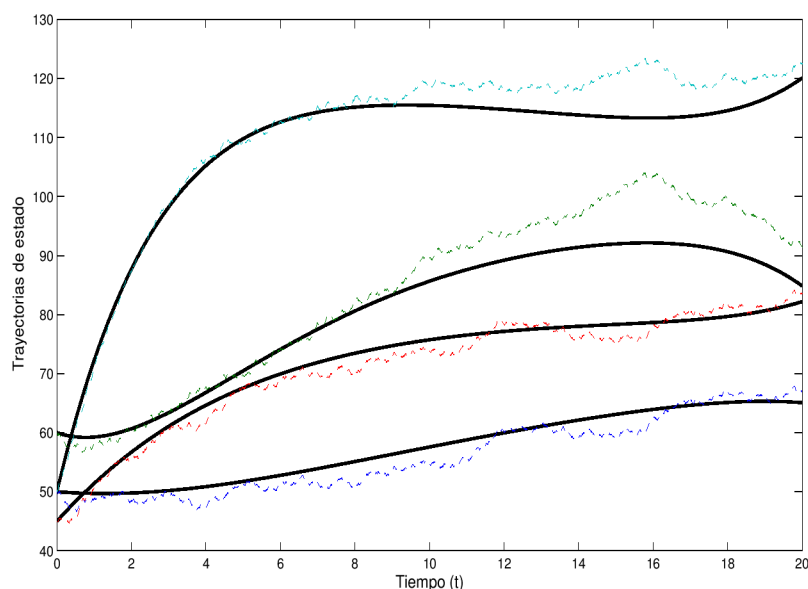


Figura 6.13: Trayectorias de estados, estocástico con líneas solidas, y trayectorias con ruido con líneas discontinuas.

El ruido de medición se simuló con media cero y las siguientes matrices de covarianza,

$$\hat{Q}_1 = \hat{Q}_2 = \begin{pmatrix} 1,34674 & 0,0318417 \\ 0,0318417 & 2,62 \end{pmatrix} \quad (6.100)$$

$$\hat{R}_1 = \hat{R}_2 = \begin{pmatrix} 2,76926 & 0,358896 \\ 0,358896 & 3,67155 \end{pmatrix} . \quad (6.101)$$

El significado de \hat{Q}_i , \hat{R}_i , $i = 1, 2$ se explicó en la subsección 6.2.1.3. Sus valores numéricos se estimaron a partir de datos reales a través de un algoritmo estándar de mínimos cuadrados. Los coeficientes deterministas usados son los mismos que en la subsección anterior 6.2.2.1.

Una confirmación numérica parcial de la optimalidad estocástica de la ley de control ‘feedback’ en la ecuación (6.70) se obtuvo mediante la evaluación de los costos correspondientes a combinaciones apropiadas de las trayectorias en la figura 6.14; en términos precisos,

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

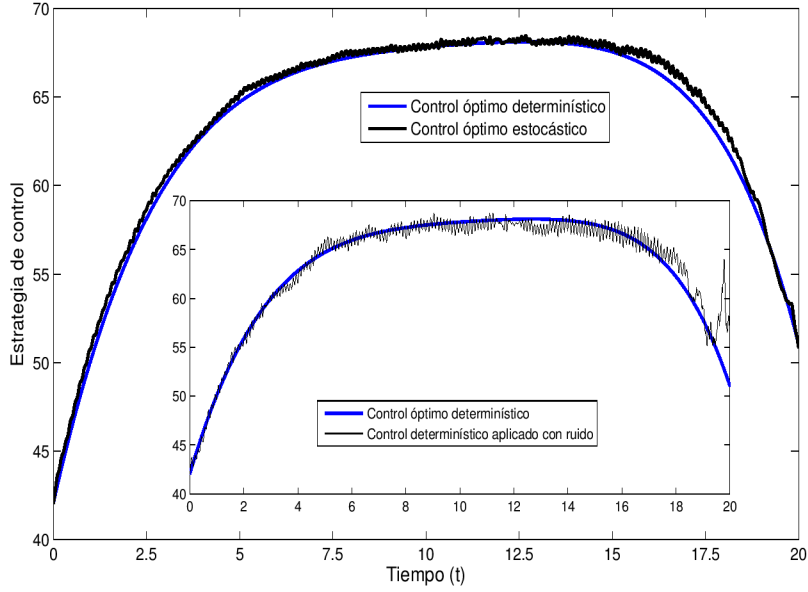


Figura 6.14: Trayectorias de control para el problema determinístico y el estocástico con los mismos coeficientes.

$$\mathcal{J}_1 := \int_0^{t_f} [(\hat{x}(t) - r(t))' Q (\hat{x}(t) - r(t)) + \hat{u}^{*'}(t) R \hat{u}^*(t)] dt \dots$$

$$+ (\hat{x}(t_f) - r(t_f))' S (\hat{x}(t_f) - r(t_f)) \quad (6.102)$$

$$\mathcal{J}_2 := \int_0^{t_f} [(\tilde{x}(t) - r(t))' Q (\tilde{x}(t) - r(t)) + \tilde{u}'(t) R \tilde{u}(t)] dt \dots$$

$$+ (\tilde{x}(t_f) - r(t_f))' S (\tilde{x}(t_f) - r(t_f)) \quad (6.103)$$

donde el significado de las variables en \mathcal{J}_1 está claro, y en \mathcal{J}_2 :

$$\tilde{u}(t) := -R^{-1} \hat{B}' \left[P(t)x(t) + \alpha \tilde{\xi}(t) \right], \quad (6.104)$$

donde $x(\cdot)$ denota una perturbación numérica de media cero del óptimo $\hat{x}(\cdot)$, y $\tilde{x}(\cdot)$ denota la solución numérica (determinista) de la ecuación de estado (6.55) para las entradas $\tilde{u}_i(\cdot)$, $r_{1i} \equiv 0$, $i = 1, \dots, n$. Los valores resultantes son:

$$\mathcal{J}_{est}(\hat{u}^*) \cong \mathcal{J}_1 = 209706 < 210487 = \mathcal{J}_2 \cong \mathcal{J}_{est}(\tilde{u}). \quad (6.105)$$

El resultado (6.105) muestra la importancia del filtrado de Kalman para la identificación de los estados no medibles y ante la presencia de ruido blanco aditivo. La ganancia de realimentación del error es elegida de forma óptima y consecuentemente los valores del funcional del costo son menores, al tener en cuenta las correcciones de las perturbaciones existentes.

6.2.2.3 Confirmaciones relevantes adicionales

6.2.2.3.1 Evaluando optimalidad

Un experimento numérico simulando el efecto de las perturbaciones sobre la ley de feedback afín variante en el tiempo de la ecuación (6.50) fue realizado a través de la combinación convexa de una familia de variaciones que cubre tanto al control óptimo como a la constante de punto de referencia nominal $u(\cdot) \equiv \bar{x} = \alpha/n$. La expresión es:

$$u = \beta u^* + (1 - \beta)\bar{x}, \quad (6.106)$$

donde, para el caso de las dos calderas, $u = u_1$, $u^* = u_1^*$, $u_2 = \alpha - u_1$, $\alpha = 150$, $\bar{x} = 75$, $\beta \in [-0,5, 2,5]$ y $Q = R = S = 0,5I$.

Algunas de las variaciones del control son representadas en la figura 6.15, y sus correspondientes valores de costo \mathcal{J}_β son comparados contra el costo óptimo $\mathcal{J}^* = \mathcal{J}_1(u^*)$. Los resultados validan la optimalidad de u^* y se representan en la figura 6.15.

6.2.2.3.2 Eficiencia dinámica

En la práctica de la ingeniería la ‘eficiencia’ η^∞ (ver ecuación (6.99)) de una caldera operando en estado estacionario mide la relación entre el calor transmitido generado por el vapor frente al calor asociado con el suministro de combustible. En el contexto presente, una versión dinámica (transitoria) de tal concepto está ensayado mediante la siguiente relación variante en el tiempo, para cada caldera $i = 1 \dots n$:

$$\eta_i(t) := \frac{x_{2i-1}(t)}{x_{2i}(t)} \simeq \frac{\text{Vapor producido al tiempo } t}{\text{Suministro de combustible al tiempo } t}, \quad (6.107)$$

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

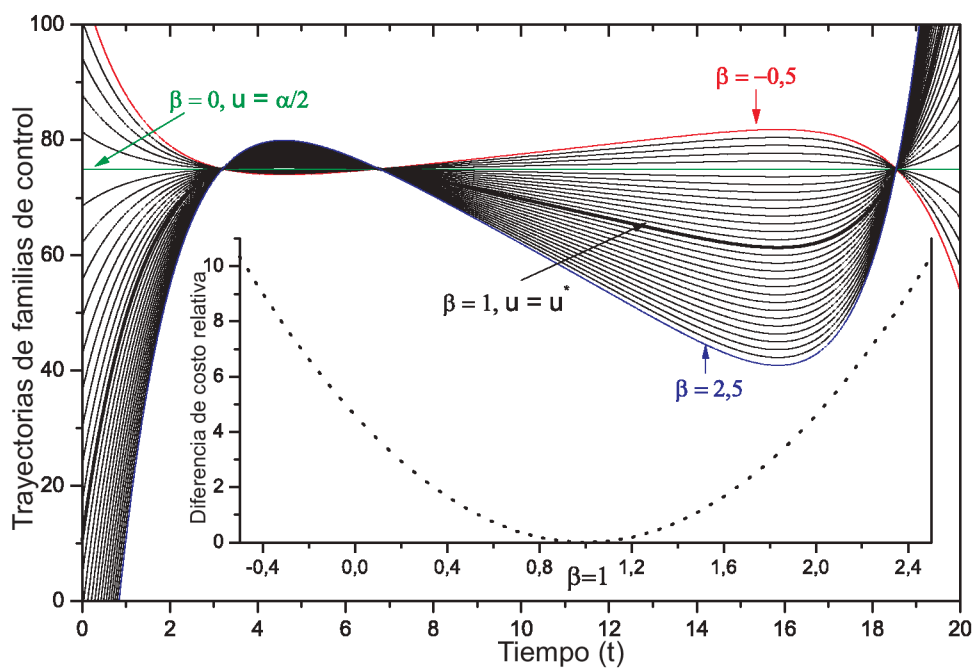


Figura 6.15: Marco principal: Variaciones alrededor del control óptimo. Marco pequeño: Diferencias relativas $(\mathcal{J}_\beta - \mathcal{J}^*) / \mathcal{J}^*$ entre el valor del costo generado por cada variación de control \mathcal{J}_β y el costo óptimo \mathcal{J}^* . El comportamiento parabólico del costo bajo combinaciones lineales de controles refleja la naturaleza lineal-cuadrática del problema.

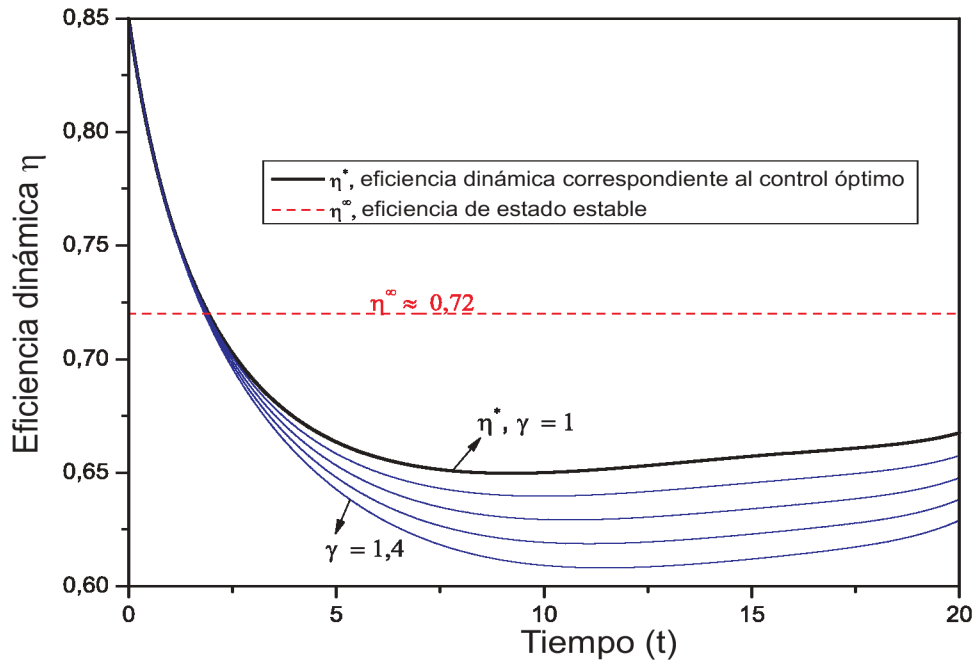


Figura 6.16: Eficiencias dinámicas correspondientes a varios miembros de la familia de control $\{u = \gamma u^*, 1 \leq \gamma \leq 1,4\}$ pero manteniendo una demanda total de $\alpha = 150$ para las dos calderas.

donde las variables son tratadas de manera adecuada sin dimensiones. Para el grupo de calderas, la eficiencia dinámica global puede ser entonces definida como

$$\eta(t) := \frac{\sum_{i=1}^n x_{2i-1}(t)}{\sum_{i=1}^n x_{2i}(t)} . \quad (6.108)$$

Si se realizan intentos de generar más vapor que la asignación óptima, entonces, correspondientemente, más combustible se deberá suministrar. Esto conduce a eficiencias dinámicas más pequeñas, como se refleja en la figura 6.16, la cual muestra que los resultados LQR-óptimos también son más ‘eficientes’ cuando se adapta la definición clásica estacionaria al análisis de transitorios.

6.2.3 Conclusiones

Una estrategia de control óptimo para cambiar dinámicamente los valores establecidos para un grupo de equipos de servicio con n unidades en paralelo se demuestran

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

y se ilustran.

La ley de realimentación resultante minimiza la suma de dos objetivos de costo en disputa: la salida de la producción de vapor, y el consumo de combustible durante el período de optimización. Esto proporciona una asignación óptima online de la demanda a cada una de las unidades en operación. El control tiene $n - 1$ grados de libertad, ya que todos los objetivos individuales deberán sumar a la demanda total proveniente del control supervisor de la planta. La falta del último grado de libertad, introduce una estructura lineal afín para la dinámica del problema cuando es planteado para todo el grupo, a través de un cambio adecuado de variables, el problema es convertido a un problema lineal de tracking.

La combinación de los costos individuales cuadráticos para las unidades resultan en un costo cuadrático total para el grupo. Entonces el tratamiento para las dinámicas lineales ampliadas y el costo cuadrático difiere de la configuración usual del LQR, estas se asemejan a las ecuaciones asociadas a un problema de tracking. También se introduce un método heurístico para elegir los pesos de costos. El método busca los mejores parámetros Q , R y S de tal manera que el índice de eficiencia es maximizado.

En este caso la solución para el problema determinístico fue encontrada en términos de la RDE más un vector variable en el tiempo que puede ser calculado una sola vez, entonces los cambios asignados en la demanda total son manejados de manera simple por introducción de los nuevos valores como un factor en la ley de control en forma de feedback. Los parámetros del control son calculados offline y no es necesario que sean recalculados después de los cambios en la demanda total.

La elección de modelos lineales para cada unidad también permite refinar los resultados determinísticos resultante en un problema de control óptimo desde el punto de vista estocástico, haciendo frente a las perturbaciones generales, tales como cambios en la composición del combustible y ruidos de medición. Esto es posible por una aplicación rigurosa del principio de separación y la adición de un filtro de Kalman.

En algunas situaciones, el ahorro en el costo de la estrategia óptima ha demostrado ser importante con respecto a los generados por estrategias de control constante a trozos, sobre todo cuando el control óptimo indica que se deben tomar valores diferentes a una partición equitativa de la demanda total. La eficiencia

6.2 Asignación dinámica en calderas

dinámica $\eta^*(t)$ correspondiente al control óptimo es siempre mejor que aquellos correspondientes a estrategias dirigidas a objetivos dinámicos de vapor adicional ($u(t) > u^*(t)$). En conclusión, los resultados presentados pueden resultar atractivos para implementaciones prácticas en plantas químicas.

6. OTROS PROBLEMAS ADICIONALES DE CONTROL ÓPTIMO

6.3 Discusión, conclusiones y perspectivas

En este capítulo se muestran dos aplicaciones prácticas, un tanto apartadas de la línea central de la tesis, utilizando estrategias de control óptimo. La primera de ellas a través del uso de programación dinámica híbrida explicada en la sección 2.2, donde se tienen en cuenta explícitamente las restricciones en los controles y estados, la cual es aplicada a un sistema lineal de bajo orden, que modela únicamente los eventos nocturnos del paciente. La segunda aplicación muestra un adecuado uso de la estrategia de control óptimo sin restricciones para repartir la demanda de producción de una planta industrial entre varios equipos encargados de la generación de vapor. Se usa un modelo linealizado de las calderas individuales. El modelo del grupo se convierte en un sistema lineal afín. En este tratamiento también se presentan inclusión de ruidos en el ambiente y en las mediciones de los dispositivos, convirtiéndolo en un problema estocástico pero basándose en la solución del control óptimo determinístico.

Este no es el fin, ni siquiera es el comienzo del final. Pero, posiblemente, sea el fin del comienzo.

Winston Churchill

CAPÍTULO

7

Conclusiones y perspectivas

El trabajo realizado en esta tesis doctoral muestra una fundamentación teórica para calcular el control óptimo de sistemas con ciertas restricciones y la forma en que puede ser aplicado a diferentes procesos de ingeniería, con especial énfasis en la exploración del uso de computación en paralelo para el tratamiento numérico de problemas que surgen en las aplicaciones. Se generan y validan resultados tanto para sistemas lineales como para sistemas no lineales que presentan limitaciones físicas en la variable de control. Los sistemas que son tratados con mayor énfasis a lo largo de la tesis, son aquéllos expresados en modelos determinísticos en tiempo continuo (como se demarca en la figura 2.1) de dimensión finita lineales y no lineales, autónomos o variantes en el tiempo; aunque en el capítulo 6 se estudia un ejemplo con un modelo estocástico. El problema de control óptimo busca la minimización del índice de desempeño, en general expresado como la integral del Lagrangiano más una penalización final. El Lagrangiano fue considerado de forma general. La penalización final de las estrategias de horizonte finito en muchos de los casos fue forzada a ser cuadrática para simplificar su tratamiento, sobre todo para resolver la ecuación diferencial de Riccati. Para las estrategias planteadas a lo largo de la tesis, las soluciones analíticas, incluso para el problema LQR, no son posibles, por lo que se vuelve imprescindible recurrir a soluciones aproximadas por intermedio de métodos numéricos (por ejemplo utilizando diferentes

7. CONCLUSIONES Y PERSPECTIVAS

formas del método del gradiente, o a través de variaciones espaciales del control). Al no poseer una receta que permita encontrar las trayectorias de control restringido en tiempo real, se ensayó computación en paralelo para realizar la búsqueda a partir de distintos intentos numéricos, con el fin de encontrar la mejor trayectoria de control de los sistemas restringidos. Se obtuvieron controles subóptimos en muchos de los casos mejores que la restricción del problema de control óptimo irrestricto.

Cuando se presentan estas limitaciones físicas, el control en la mayoría de los procesos industriales se satura, lo cual no necesariamente corresponde a la mejor decisión o la mejor estrategia a implementar. Si la restricción aparece limitando zonas de operación muy pequeñas, en algunos casos puede ser solucionado cambiando las variables manipuladas por aquéllas que puedan operar en rangos más amplios, sin embargo cuando las limitaciones son ineludiblemente innatas del sistema las estrategias deben ser construidas de modo que éstas se tengan en cuenta de antemano. Cuando las saturaciones son frecuentes el sistema opera como si estuviera en lazo abierto, ya que el controlador sigue su límite máximo o mínimo de forma independiente de la salida del controlador, hasta que ésta no regrese a su 'zona de operación'. Si se conocen previamente las características de la variable manipulada y del sistema, se pueden saber los tiempos en que se entra o sale de la saturación (o tiempos de conmutación). En la tesis es posible observar cómo el conocimiento de estos puntos permite también realizar diferentes variaciones modificando los tiempos de conmutación.

7.1 Conclusiones generales

Para sistemas lineales:

- Se trabajó con la modificación de la matriz de penalización final para el problema fantasma, construida a partir de los valores de fase finales relacionados con un problema irrestricto. Dicho problema sustituto tiene el mismo funcional de costo pero diferentes condición inicial y matriz de penalización final (utilizada para la solución de la RDE).
- Se encontraron fórmulas algebraicas eficientes para calcular las cruciales matrices α , β (ver ecuaciones 3.6 de la sección 3.2), a través de la exponencial

de la matriz Hamiltoniana \mathbf{H} (ver 3.3). Se utilizan modificaciones novedosas de las ideas de Van Loan [143] para la actualización de las matrices.

- También se obtuvieron fórmulas algebraicas para la determinación de las integrales de ciertos objetos matemáticos vinculados con la actualización de los parámetros (ver ecuaciones 3.25, 3.26, 3.27) [143].
- En el procedimiento de actualización de los valores de fase finales se evitó la integración numérica de la dinámica y del funcional del costo por la incorporación de ciertos objetos matemáticos relevantes (ver subsección 3.3.1.3).
- Se realizó un estudio sobre problemas que pueden presentar varios períodos regulares, analizando un ejemplo en el que se presentan dos períodos, proponiendo soluciones en las cuales se hacen optimizaciones individuales de acuerdo al número de arcos regulares, pero teniendo en cuenta siempre la reducción total del funcional del costo (ver sección 3.5).

Para sistemas no lineales, se trabajó con base a dinámicas linealizadas alrededor de trayectorias de control y estados, para aprovechar características de lazo cerrado y robustez ante la aparición de perturbaciones:

- Se exploró la relación con el método del proceso fantasma utilizado en el tratamiento de sistemas lineales. Siguiendo esa línea se llegó a resultados subóptimos (ver 4.1).
- Se trabajó con éxito con perturbaciones de control espaciales para la actualización de las trayectorias de control, con lo que se arribó siempre a un control sub-óptimo mejor que el semilla. El control semilla se propone aún con el desconocimiento del proceso en general, o de manera teórica por intermedio de la solución de la HJB.

Sobre computación en paralelo en control óptimo:

- Se aplicó de manera novedosa el cálculo en paralelo a las estrategias de control óptimo cuando se tienen restricciones en la variable manipulada.
- Se comprobaron los beneficios de distribuir la cantidad de cálculos en problemas de control.

7. CONCLUSIONES Y PERSPECTIVAS

- Realizar cálculos de manera mas rápida y eficiente permite trabajar con sistemas o procesos en tiempo real, como se mostró al hacer divisiones en intervalos de muestra.
- Se analizaron diferentes arquitecturas de computadores y se ejecutaron cálculos numéricos en computadoras paralelas tanto en procesadores multinúcleo, como en los denominados cluster de procesadores (grupo de procesadores interconectados que son utilizados como una computadora paralela).
- Se analizó y utilizó OpenMP para programar código en maquinas de memoria compartida, trabajando dos estrategias diferentes para paralelizar el problema de actualización de los valores de fase finales: (i) distribuyendo por medio de secciones, y (ii) por intermedio de realizar paralelización sobre los bucles involucrados en los barridos de los valores de fase finales.
- Se evidenció para el problema de actualización de los valores de fase finales que el aumento en la dimensión n del modelo, provocaba menor eficiencia cuando se analizaba sobre la arquitectura paralela. Empero, al observar la ganancia total de tiempo es posible apreciar que resulta ventajoso utilizar varios procesadores con respecto a trabajar con un solo procesador. Por lo tanto, se puede concluir que, en casos en que se decide aumentar la precisión en los cálculos a través de un refinamiento espacial o temporal, el cálculo en paralelo siempre ahorra tiempo.
- Se exhibió la facultad de trabajar con diferentes herramientas, lenguajes y programas permitiendo ser vinculados a los sistemas de control.

En otros problemas de control óptimo:

- Se aplicó la técnica “Programación Dinámica Híbrida” al control de niveles de glucosa a pacientes con diabétes tipo I. El principal resultado de esta aplicación es la regulación de eventos nocturnos donde el paciente se encuentre en hiperglucemia. Las acciones de control complementan a la bomba de insulina continua que asiste al paciente.

- Se logró diseñar estrategias que reducen en promedio la inyección de insulina, al mismo tiempo que disminuye un “costo médico total” entre todas las terapias aceptables.
- En la aplicación a calderas, se diseñó una estrategia de control óptima para cambiar dinámicamente los valores de consigna establecidos para un grupo de unidades de servicio trabajando en paralelo para la generación de vapor.
- Se realizó una conversión desde un problema lineal afín a través de un cambio de variable a un problema lineal de seguimiento.
- Se demostró que estrategias de control óptimo funcionan de mejor manera que asignaciones estáticas de la demanda para un grupo de calderas trabajando en paralelo.
- Se trabajó con un problema estocástico con el cual se hizo frente a las perturbaciones generales, tales como cambios en la composición del combustible y a los ruidos de medición, siendo posible por una aplicación rigurosa del principio de separación y la adición de un filtro de Kalman (ver subsección 6.2.1.4).
- Se exhibió que los ruidos que afectan a la salida y los estados son independientes de aquéllos que afectan otras unidades, siendo la covarianza de ruidos entre las unidades cero. Esto permitió que los problemas de filtrado y estimación puedan ser resueltos para cada unidad del grupo de forma independiente (ver subsección 6.2.1.3).

7.2 Algunas perspectivas

A partir de los resultados de la tesis sobre el uso de estrategias en control óptimo, diversas líneas de investigación pueden ser profundizadas, tales como:

- Se impone la extensión y simulación de problemas que involucren un número finito de arcos regulares entre pares de tiempos de conmutación, para obtener conclusiones más relevantes sobre el método de optimización.

7. CONCLUSIONES Y PERSPECTIVAS

- Ante la necesidad de generar trayectorias óptimas en línea con el proceso, otra investigación podría fomentar el estudio en paralelo de distintos Lagrangianos. En el caso por ejemplo del tratamiento para pacientes diabéticos tipo 1 un costo adecuado puede ser ideado para penalizar de diferente forma los eventos de hiperglucemia y de hipoglucemia.
- En el problema de manejo de restricciones, aunque se presentaron estrategias tanto en sistemas lineales como en no lineales, aún sigue siendo un problema abierto, más teniendo en cuenta que estos tratamientos se vuelven numéricos cuando las limitaciones aparecen. Es posible entonces plantear nuevos métodos numéricos para los problemas restringidos, algunos utilizando cálculo en paralelo para encontrar soluciones óptimas, y sea posible comparar los resultados.
- Utilizar estrategias MPC para el control de nivel de glucemia mediante el suministro de insulina en pacientes diabéticos tipo 1, pero en este caso asociando el denominado control por zonas, sujeto a restricciones con costos asimétricos, en el que se penalicen las desviaciones respecto a los rangos seguros de operación.
- En problemas no lineales el uso de computación en paralelo resulta útil tanto para mejorar la eficiencia en las evaluaciones de la dinámica como para determinar los costos y los gradientes de los costos asociados a la evaluación de diferentes puntos de operación, seleccionados después de realizar una linealización del modelo para satisfacer exigencias de control (restricciones en la variable manipulada), en el que se represente adecuadamente una zona de operación del sistema no lineal tratado, y tomando en cuenta el modelo que más se aproxima a el comportamiento real del sistema no lineal.

Bibliografía

- [1] Abraham, R., & Marsden, J. E. (1978). *Foundations of Mechanics*. Reading, Massachusetts: Benjamin / Cummings, second ed. 7, 26
- [2] Abu-Rmileh, A., & Garcia-Gabin, W. (2011). *Hypoglycemia Prevention in Closed-Loop Artificial Pancreas for Patients with Type-1 Diabetes*. InTech Open Access. 149
- [3] Agrachev, A. A., & Sachkov, Y. L. (2004). *Control Theory from the Geometric Viewpoint*. Berlin-Heidelberg: Springer-Verlag. 33, 48
- [4] Ahnert, K., & Mulansky, M. (2011). Odeint-solving ordinary differential equations in c++. *API Conf. Proc.*, 1389, 1586–1589. 126
- [5] Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, (pp. 483–485). ACM. 135
- [6] Anderson, J., A (1994). A parallel architecture for distributed adaptive control. *Journal of Microcomputer Applications*, 17, 17–34. 8
- [7] Arruabarrena, A., & Mugerza, J. (2012). Computadores paralelos: Computación de alta velocidad. <http://www.sc.ehu.es/acwarfra/arpar/AP/AP.fitxategiak/AP-apunteak/ComPar.1.pdf>. [Online; disponible 17-Octubre-2016]. xv, 121
- [8] Athans, M., & Falb, P. (2006). *Optimal Control: An Introduction to the Theory and Its Applications*. New York, USA: Dover. 27, 29, 33, 132

BIBLIOGRAFÍA

- [9] Atlas, E., Nimri, R., Miller, S., Grunberg, E. A., & Phillip, M. (2010). Md-logic artificial pancreas system. *Diabetes care*, 33(5), 1072–1076. 149
- [10] Banks, S. P., & Dinesh, K. (2000). Approximate optimal control and stability of nonlinear finite-and infinite-dimensional systems. *Annals of Operations Research*, 98(1-4), 19–44. 83
- [11] Banks, S. P., & McCaffrey, D. (1998). Lie algebras, structure of nonlinear systems and chaotic motion. *International Journal of Bifurcation and Chaos*, 8(07), 1437–1462. 83
- [12] Barceló, A., Aedo, C., Rajpathak, S., & Robles, S. (2003). The cost of diabetes in latin america and the caribbean. *Bulletin of the world health organization*, 81(1), 19–27. 147
- [13] Baxter, M. J., Tokhi, M. O., & Fleming, P. (1997). A generic approach to parallelizing and developing control algorithms for heterogeneous architectures. *International Journal of Adaptive Control and Signal Processing*, 11, 443–460. 127
- [14] Bellman, R., & Kalaba, R. (1963). A note on Hamilton's equations and invariant imbedding. *Quarterly of Applied Mathematics*, XXI, 166–168. 7
- [15] Bellman, R. E. (1957). *Dynamic Programming*. Princeton, NJ, USA: Dover. 22
- [16] Benner, P., Ezzatti, P., Mena, H., Quintana-Ortí, E., & Remon, A. (2011). Solving differential riccati equations on multi-gpu platforms. In *10th International Conference on Computational and Mathematical Methods in Science and Engineering CMMSE11*, (pp. 178–188). 127
- [17] Benner, P., Ezzatti, P., Mena, H., Quintana-Ortí, E. S., & Remón, A. (2010). Numerical solution of differential riccati equations on hybrid cpu-gpu platforms. *differential equations*, 2, 6. 127
- [18] Bernhard, P. (1972). Introducción a la teoría de control Óptimo. Tech. Rep. Cuaderno Nro. 4, Instituto de Matemática “Beppo Levi”, Rosario, Argentina. 21, 27, 28, 36

- [19] Bramanti, E., Bramanti, M., Stiavetti, P., & Benedetti, E. (1994). A frequency deconvolution procedure using a conjugate gradient minimization method with suitable constraints. *Journal of Chemometrics*, 8, 409–421. 34, 46
- [20] Bruni, C., Di Pillo, G., & Koch, K. (1974). Bilinear systems: An appealing class of ‘nearly linear’ systems in theory and applications. *IEEE Transactions on Automatic Control*, AC(19), 334–348. 6
- [21] Bryson, A., & Ho, Y. (1975). *Applied Optimal Control*. New York: John Wiley and Sons, revised printing ed. 6, 25, 154, 168
- [22] Bujak, J. (2009). Optimal control of energy losses in multi-boiler steam systems. *Energy*, 34(9), 1260–1270. 162, 163, 164, 165
- [23] Camacho, E. F., & Bordons, C. (2004). *Model Predictive Control*. London, UK: Springer, 2nd ed. 35
- [24] Caramia, M., & Dell’ Olmo, P. (2008). *Multi-objective Management in Freight Logistics*. London, UK: Springer. 60
- [25] Chapman, B., Jost, G., & Van Der Pas, R. (2008). *Using OpenMP*. MIT Press. 9, 122
- [26] CIMEC (2008). Descripción del Cluster Aquiles. <http://www.cimec.org.ar/twiki/bin/view/Cimec/AquilesDescription>. [Online; disponible 26-Agosto-2016]. 126
- [27] Cimen, T., & Banks, S. P. (2004). Global optimal feedback control for general nonlinear systems with nonquadratic performance criteria. *Systems & Control Letters*, 53(5), 327–346. 83
- [28] Cobelli, C., Renard, E., & Kovatchev, B. (2011). Artificial pancreas: past, present, future. *Diabetes*, 60(11), 2672–2682. 148
- [29] Coddington, A., & Levinson, N. (1955). *Theory of ordinary differential equations*. International series in pure and applied mathematics. McGraw-Hill. 18, 99

BIBLIOGRAFÍA

- [30] Collins, D., & Lang, R. (1998). Economic impact of control and optimization on industrial utilities. *Energy engineering*, 95(1), 10–21. 162
- [31] Control, D., Group, C. T. R., et al. (1993). The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. *N Engl J Med*, 1993(329), 977–986. 147
- [32] Control, D., Group, C. T. R., et al. (1994). Effect of intensive diabetes treatment on the development and progression of long-term complications in adolescents with insulin-dependent diabetes mellitus: Diabetes control and complications trial. *The Journal of Pediatrics*, 125(2), 177–188. 147
- [33] Costanza, V. (2007). Finding initial costates in finite-horizon nonlinear-quadratic optimal control problems. *Optimal Control Applications & Methods*, 29, 225–242. 35
- [34] Costanza, V. (2007). Parametric uncertainty and disturbance attenuation in the suboptimal control of a nonlinear electrochemical process. *Optimal Control Applications & Methods*, 28, 209–228. 6
- [35] Costanza, V. (2008). Finding initial costates in finite-horizon nonlinear-quadratic optimal control problems. *Optimal Control Applications and Methods*, 29(3), 225–242. 8
- [36] Costanza, V. (2008). Regular optimal control problems with quadratic final penalties. *REVISTA DE LA UNIÓN MATEMÁTICA ARGENTINA*, 49. 7, 8
- [37] Costanza, V., & Gómez Múnera, J. A. (2016). Cost Reduction for Nonlinear Systems Under Restrictions on the Manipulated Variables. *5th International Conference on Engineering Optimization*. Foz de Iguazú, Brasil. 10, 12
- [38] Costanza, V., & Neuman, C. E. (2006). Optimal control of nonlinear chemical reactors via an initial-value hamiltonian problem. *Optimal Control Applications & Methods*, 27, 41–60. 33, 84

- [39] Costanza, V., & Neuman, C. E. (2009). Partial differential equations for missing boundary conditions in the linear-quadratic optimal control problem. *Latin American Applied Research*, 39, 207–212. 35
- [40] Costanza, V., & Neuman, C. E. (2009). Partial differential equations for missing boundary conditions in the linear-quadratic optimal control problems. *Latin American Applied Research*, 39, 207–212. 33
- [41] Costanza, V., & Picó, M. (2005). Control hamiltoniano de sistemas no lineales en tiempo real. *XI RPIC*. Río Cuarto, Argentina. 89, 92
- [42] Costanza, V., & Rivadeneira, P. S. (2008). Finite-horizon dynamic optimization of nonlinear systems in real time. *Automatica*, 44, 2427–2434. 84
- [43] Costanza, V., & Rivadeneira, P. S. (2013). Optimal saturated feedback laws for LQR problems with bounded controls. *Computational and Applied Mathematics*, 32, 355–371. 8, 29, 34, 37, 48, 78, 85, 86, 130
- [44] Costanza, V., & Rivadeneira, P. S. (2014). *Enfoque Hamiltoniano al control óptimo de sistemas dinámicos*. Saarbrücken, Alemania: OmniScriptum. 7, 24, 25, 33, 36, 48
- [45] Costanza, V., & Rivadeneira, P. S. (2014). Online suboptimal control of linearized models. *System Science & Control Engineering*, 2, 379–388. 8, 34, 37, 48, 50, 54, 77, 78
- [46] Costanza, V., & Rivadeneira, P. S. (2014). Partially-Regular Bounded-Control Problems for Nonlinear Systems. *XXIV AADECA; Conference Proceedings*. Buenos Aires, Argentina. 29, 83, 84, 115, 144
- [47] Costanza, V., & Rivadeneira, P. S. (2015). Optimal supervisory control of steam generators operating in parallel. *Energy*, 93, 1819–1831. 165, 172
- [48] Costanza, V., Rivadeneira, P. S., & Gómez Múnera, J. A. (2015). Control sub-óptimo de sistemas no lineales con restricciones en la variable manipulada. *XVI RPIC; Conference Proceedings*. Córdoba, Argentina. 10, 12

BIBLIOGRAFÍA

- [49] Costanza, V., Rivadeneira, P. S., & Gómez Múnera, J. A. (2015). Numerical Treatment of the Bounded-Control LQR Problem By Updating the Final Phase Value. *XVI RPIC; Conference Proceedings*. Córdoba, Argentina. 10, 12
- [50] Costanza, V., Rivadeneira, P. S., & Gómez Múnera, J. A. (2016). An Efficient Cost Reduction Procedure for Bounded-Control LQR Problems. *Computational and Applied Mathematics*. 10, 11
- [51] Costanza, V., Rivadeneira, P. S., & Gómez Múnera, J. A. (2016). Numerical Treatment of the Bounded-Control LQR Problem By Updating the Final Phase Value. *IEEE Latin America Transactions*. 10, 11
- [52] Costanza, V., Rivadeneira, P. S., & González, A. H. (2013). Minimizing control-energy in a class of bounded-control lqr problems. *Optimal Control Applications & Methods*, 35(3), 361–382. 48
- [53] Costanza, V., Rivadeneira, P. S., & Spies, R. D. (2009). Equations for the missing boundary values in the hamiltonian formulation of optimal control problems. *Journal of Optimization Theory and Applications*, 149, 26–46. 36
- [54] Das, S., Pan, I., Halder, K., Das, S., & Gupta, A. (2013). Optimum weight selection based lqr formulation for the design of fractional order $PI^\lambda D^\mu$ controllers to handle a class of fractional order systems. In *Computer Communication and Informatics (ICCCI), 2013 International Conference on*, (pp. 1–6). IEEE. 168
- [55] Dhamo, V., & Tröltzsch, F. (2011). Some aspects of reachability for parabolic boundary control problems with control constraints. *Computational Optimization and Applications*, 50, 75–110. 43
- [56] Doyle, F. J., Huyett, L. M., Lee, J. B., Zisser, H. C., & Dassau, E. (2014). Closed-loop artificial pancreas systems: engineering the algorithms. *Diabetes care*, 37(5), 1191–1197. 149

- [57] Doyle III, F. J. (2012). Zone model predictive control of an artificial pancreas. In *Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, China*. 149
- [58] Elleri, D., Allen, J., Nodale, M., Wilinska, M., Acerini, C., Dunger, D., & Hovorka, R. (2010). Suspended insulin infusion during overnight closed-loop glucose control in children and adolescents with type 1 diabetes. *Diabetic Medicine*, 27(4), 480–484. 149
- [59] Ezzatti Infante, P. M., Quintana-Ortí, E. S., & Remón Gómez, A. (2011). Computación de alto desempeño para la reducción de modelos. *Reportes Técnicos 11-10*. 127
- [60] Fernández Baldomero, F. J. (2001). *Entorno paralelo de modelado y simulación bajo MATLAB. Aplicación a sistemas de visión*. Ph.D. thesis, Universidad de Granada, España. 122
- [61] Fleming, W. H., & Rishel, R. (1975). *Deterministic and Stochastic Optimal Control*. New York, USA: Dover. 97, 173
- [62] Fletcher, R., & Reeves, C. M. (1964). Function minimization for conjugate gradients. *The Computer Journal*, 7(2), 149–154. 34, 46
- [63] Gallardo Hernandez, A. G. (2011). *Controlador de glucosa en sangre*. Ph.D. thesis, Universidad Nacional Autónoma de México. 147
- [64] Gómez Múnera, J. A., & Costanza, V. (2014). Computación en paralelo para control óptimo basado en modelos. *XXIV AADECA; Conference Proceedings*. Buenos Aires, Argentina. 10, 12, 116
- [65] Gómez Múnera, J. A., Rivadeneira, P. S., & Costanza, V. (2017). A Cost Reduction Procedure for Control-Restricted Nonlinear Systems. *Computational Optimization and Applications*, enviado. 10, 11
- [66] Gondhalekar, R., Dassau, E., Zisser, H. C., & Doyle III, F. J. (2014). Periodic-zone model predictive control for diurnal closed-loop operation of an artificial pancreas. *Journal of Diabetes Science and Technology*, 7(6), 1446–1460. 149, 152

BIBLIOGRAFÍA

- [67] Goodwin, G. C., Graebe, S. F., & Salgado, M. E. (2001). *Control system design*, vol. 240. Prentice Hall New Jersey. 29, 30
- [68] Grosman, B., Dassau, E., Zisser, H. C., Jovanovic, L., & Doyle III, F. J. (2010). Zone model predictive control: a strategy to minimize hyper- and hypoglycemic events. *Journal of Diabetes Science and Technology*, 4(4), 961–975. 149
- [69] Gustafson, J. L. (1988). Re-evaluating amdhal's law. *Communications of the ACM*, 31(5), 532–533. 135
- [70] Härkegård, O., & Glad, S. T. (2005). Resolving actuator redundancy—optimal control vs. control allocation. *Automatica*, 41(1), 137–144. 164
- [71] Hatzopoulos, D., Koutsopoulos, I., Koutitas, G., & Van Heddeghem, W. (2013). Dynamic virtual machine allocation in cloud server facility systems with renewable energy sources. In *2013 IEEE International Conference on Communications (ICC)*, (pp. 4217–4221). IEEE. 164
- [72] Havlena, V. (2004). Coordination in multilayer process control and optimization schemes. US Patent 6,832,134. 164, 165
- [73] Hearn, G., & Grimble, M. J. (2010). Temperature control in transport delay systems. In *The 2010 American Control Conference*, (pp. 6089–6094). Baltimore, MD, USA. 50
- [74] Hennessy, J. D., & Patterson, D. A. (1993). *Arquitectura de computadores: Un enfoque cuantitativo*. McGraw-Hill. 9, 120, 135
- [75] Hernández García, V., Quintana Ortí, E. S., & Quintana Ortí, G. (2009). Métodos numéricos para la resolución del problema lineal-cuadrático de control óptimo. 127
- [76] Hirsch, M. W., Smale, S., & Devaney, R. L. (2004). *Differential equations, dynamical systems, and an introduction to chaos*. USA: Elsevier, 2nd ed. 18, 20

-
- [77] Horová, E., Mazoch, J., HiIgartová, J., Kvasnička, J., Škrha, J., Šoupal, J., & Prázný, M. (2012). Acute hyperglycemia does not impair microvascular reactivity and endothelial function during hyperinsulinemic isoglycemic and hyperglycemic clamp in type 1 diabetic patients. *Experimental diabetes research*, 2012, Article ID 851487. 151
- [78] Hossian, M. A., Kabir, U., & Tokhi, M. O. (2002). Impact of data dependencies in real-time high performance computing. *Microprocessors and Microsystems*, 26, 253–261. 121
- [79] Hovorka, R., Canonico, V., Chassin, L. J., Haueter, U., Massi-Benedetti, M., Federici, M. O., Pieber, T. R., Schaller, H. C., Schaupp, L., Vering, T., et al. (2004). Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiological measurement*, 25(4), 905–920. 149
- [80] Howlett, P. (2000). The optimal control of a train. *Annals of Operations Research*, 98(1-4), 65–87. 88
- [81] Howlett, P. G., Pudney, P. J., & Vu, X. (2009). Local energy minimization in optimal train control. *Automatica*, 45(11), 2692–2698. 88
- [82] Hugues-Salas, O., & Banks, S. P. (2008). Optimal control of chaos in nonlinear driven oscillators via linear time-varying approximations. *International Journal of Bifurcation and Chaos*, 18(11), 3355–3374. 83
- [83] Isidori, A. (1999). *Nonlinear Control Systems II*. London: Springer-Verlag. 6
- [84] Itik, M. (2016). Optimal control of nonlinear systems with input constraints using linear time varying approximations. *NONLINEAR ANALYSIS-MODELLING AND CONTROL*, 21(3), 400–412. 83
- [85] Jamshidi, M., Tarokh, M., & Shafai, B. (1992). *Computer-aided Analysis and Design of Linear Control Systems*. Prentice-Hall International editions. Prentice-Hall International. 16

BIBLIOGRAFÍA

- [86] Kadish, A. H. (1963). Automation control of blood sugar a servomechanism for glucose monitoring and control. *ASAIO Journal*, 9(1), 363–367. 148
- [87] Kailath, T. (1980). *Linear systems*, vol. 156. Prentice-Hall Englewood Cliffs, NJ. 31
- [88] Kalman, R. E., Falb, P. L., & Arbib, M. A. (1969). *Topics in Mathematical System Theory*. New York: McGraw-Hill. 7, 19, 22, 24, 30
- [89] Kanellakopoulos, I., Kokotovic, P. V., & Marino, R. (1991). An extended direct scheme for robust adaptive nonlinear control. *Automatica*, 27(2), 247–255. 6
- [90] Kantor, J. C. (1987). An overview of nonlinear geometric methods for process control. In *Shell Process Control Workshop*. Butterworth. 6
- [91] Khalil, H. K., & Grizzle, J. (1996). *Nonlinear systems*, vol. 3. Prentice hall New Jersey. 81
- [92] Kim, J. H., & Han, C. (2001). Short-term multiperiod optimal planning of utility systems using heuristics and dynamic programming. *Industrial & engineering chemistry research*, 40(8), 1928–1938. xvi, 162, 163, 164
- [93] Kleinberg, J., & Tardos, É. (2013). *Algorithm Design*. Pearson New international edition. Pearson. 22
- [94] Kojima, A., & Morari, M. (2004). Lq control for constrained continuous-time systems. *Automatica*, 40(7), 1143–1155. 33
- [95] Kovatchev, B., Cobelli, C., Renard, E., Anderson, S., Breton, M., Patek, S., Clarke, W., Bruttomesso, D., Maran, A., Costa, S., et al. (2010). Multinational study of subcutaneous model-predictive closed-loop control in type 1 diabetes mellitus: summary of the results. *Journal of diabetes science and technology*, 4(6), 1374–1381. 149
- [96] Krener, J. C. (1974). Linearization and bilinearization of control systems. In *Proc. 12 12th, Annual Allerton Conf. on Circuit and Systems Theory*. 6

- [97] Kuo, B. C. (1996). *Sistemas de control automático*. Pearson Educación. 81
- [98] Kwakernaak, H., & Sivan, R. (1972). *Linear optimal control systems*, vol. 1. Wiley-interscience New York. 6
- [99] Lara, F. (2007). *Derivación Matricial*. Fundación Universitaria Konrad Lorenz. 27
- [100] León-Vargas, F., Garelli, F., De Battista, H., & Vehí, J. (2015). Postprandial response improvement via safety layer in closed-loop blood glucose controllers. *Biomedical Signal Processing and Control*, 16, 80–87. 149
- [101] Levine, W. S. (1996). *The control handbook*. CRC press. 29
- [102] Lewis, F. L., & Syrmos, V. L. (1995). *Optimal control*. John Wiley & Sons. 6, 22
- [103] Li, J. (2004). *The Dynamics of Glucose-Insulin Endocrine Metabolic Regulatory System*. Ph.D. thesis, ARIZONA STATE UNIVERSITY. 155
- [104] Loutseiko, M., Voskanyan, G., Keenan, D. B., & Steil, G. M. (2011). Closed-loop insulin delivery utilizing pole placement to compensate for delays in subcutaneous insulin delivery. *Journal of diabetes science and technology*, 5(6), 1342–1351. 151
- [105] Magdelaine, N., Chaillous, L., Guilhem, I., Poirier, J.-Y., Krempf, M., Moog, C. H., & Le Carpentier, E. (2015). A long-term model of the glucose insulin dynamics of type 1 diabetes. *IEEE Transactions on Biomedical Engineering*, 62(6), 1546–1552. 150, 151, 156
- [106] Magni, L., Raimondo, D. M., Dalla Man, C., De Nicolao, G., Kovatchev, B., & Cobelli, C. (2009). Model predictive control of glucose concentration in type i diabetic patients: An in silico trial. *Biomedical Signal Processing and Control*, 4(4), 338–346. 149
- [107] Marchetti, G., Barolo, M., Jovanovic, L., Zisser, H., & Seborg, D. E. (2008). An improved pid switching control strategy for type 1 diabetes. *IEEE Transactions on Biomedical Engineering*, 55(3), 857–865. 149

BIBLIOGRAFÍA

- [108] MathWorks (2016b). Parallel computing toolbox. http://www.mathworks.com/help/pdf_doc/distcomp/distcomp.pdf. [Online; disponible 20-Octubre-2016]. xv, 9, 78, 123
- [109] MohammadRidha, T., & Moog, C. H. (2015). Model free control for type-1 diabetes: A fasting-phase study. *IFAC-PapersOnLine*, 48(20), 076–081. 149
- [110] Ogata, K. (2003). *Ingeniería de control moderna*. Pearson Educación. 81
- [111] Oksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media. 177, 178
- [112] Orchard, T. J., Nathan, D. M., Zinman, B., Cleary, P., Brillon, D., Backlund, J.-Y. C., & Lachin, J. M. (2015). Association between 7 years of intensive treatment of type 1 diabetes and long-term mortality. *Jama*, 313(1), 45–53. 147
- [113] Pantelides, C. C., & Renfro, J. G. (2013). The online use of first-principles models in process operations: Review, current status and future needs. *Computers and Chemical Engineering*, 51, 136–148. 8, 127
- [114] Pardalos, P., & Pytlak, R. (2008). *Conjugate Gradient Algorithms In Non-convex Optimization*. New York, USA: Springer. 34
- [115] Phillip, M., Battelino, T., Atlas, E., Kordonouri, O., Bratina, N., Miller, S., Biester, T., Avbelj Stefanija, M., Muller, I., Nimri, R., et al. (2013). Nocturnal glucose control with an artificial pancreas at a diabetes camp. *New England Journal of Medicine*, 368(9), 824–833. 149
- [116] Pickup, J., Mattock, M., & Kerry, S. (2002). Glycaemic control with continuous subcutaneous insulin infusion compared with intensive insulin injections in patients with type 1 diabetes: meta-analysis of randomised controlled trials. *Bmj*, 324(7339), 705. 148
- [117] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., & Mishchenko, E. F. (1964). *The Mathematical Theory of Optimal Processes*. New York, USA: Macmillan. 7, 25, 29, 30, 33, 101, 132

- [118] Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, *11*, 733–764. 33
- [119] Quintana-Ortí, E. S., & Hernandez, V. (1998). Parallel solution of riccati matrix equations with the matrix sign function. *Automatica*, *34*(2), 151–156. 127
- [120] Quintana-Ortí, E. S., & Van De Geijn, R. (1999). Parallel solution of selected problems in control theory. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*. Citeseer. Citeseer. 127
- [121] Rao, A. V., Benson, D. A., Huntington, G. T., Francolin, C., Darby, C. L., & Patterson, M. A. (2008). User's manual for gops: A matlab package for dynamic optimization using the gauss pseudospectral method. Tech. rep., University of Florida. 35
- [122] Retnakaran, R., Hochman, J., DeVries, J. H., Hanaire-Broutin, H., Heine, R. J., Melki, V., & Zinman, B. (2004). Continuous subcutaneous insulin infusion versus multiple daily injections the impact of baseline a1c. *Diabetes care*, *27*(11), 2590–2596. 148
- [123] Revert, A., Garelli, F., Picó, J., De Battista, H., Rossetti, P., Vehí, J., & Bondía, J. (2013). Safety auxiliary feedback element for the artificial pancreas in type 1 diabetes. *IEEE Transactions on Biomedical Engineering*, *60*(8), 2113–2122. 149
- [124] Rivadeneira, P. S., Gómez Múnera, J. A., & Costanza, V. (2016). Programación Híbrida para el Manejo de la Glucemia en Pacientes Diabéticos Tipo I. *XXV AADECA; Conference Proceedings*. Buenos Aires, Argentina. 12
- [125] Rivadeneira, P. S., Múnera, J. A. G., & Costanza, V. (2016). Dynamic allocation of industrial utilities as an optimal stochastic tracking problem. *Chemical Engineering Science*. 11
- [126] Robinson, J. D. (1990). A linear quadratic regulator weight selection algorithm for robust pole assignment. Tech. rep., DTIC Document. 168

BIBLIOGRAFÍA

- [127] Rojas, O. J., Goodwin, G. C., Serón, M. M., & Feuer, A. (2004). An svd based strategy for receding horizon control of input constrained linear systems. *International Journal of Robust and Nonlinear Control*, 14(13-14), 1207–1226. 33
- [128] Sanderson, C., & Curtin, R. (2016). Armadillo: a template-based c++ library for linear algebra. *Journal of Open Source Software*. 136
- [129] Sato, J., Hirose, T., & Watada, H. (2012). Continuous glucose monitoring system: Is it really accurate, safe and clinically useful? *Journal of diabetes investigation*, 3(3), 225–230. 148
- [130] Scheel, C., & McInnis, B. (1981). Parallel processing of optimal-control problems by dynamic programming. *Information Sciences*, 25(2), 85–114. 127
- [131] Slotine, J.-J. E., & Li, W. (1991). *Applied nonlinear control*, vol. 199. prentice-Hall Englewood Cliffs, NJ. 29, 30
- [132] Snir, M., Otto, S. W., Walker, D. W., Dongarra, J., & Huss-Lederman, S. (1995). *Mpi: The complete reference*. 9
- [133] Snir, M., Steve, O., Huss-Lederman, S., Walker, D., & Dongarra, J. (1996). *MPI: The Complete Reference*. MIT Press. 122, 126
- [134] Sontag, E. D. (1998). *Mathematical Control Theory*. New York: Springer. 6, 7, 17, 19, 21, 22, 25, 27, 28, 30, 33, 36, 96, 97, 98, 99, 116, 154
- [135] Sonzogni, V. E., Yommi, A. M., Nigro, N. M., & Storti, M. A. (2002). A parallel finite element program on a beowulf cluster. *Advances in Engineering Software*, 33(7), 427–443. 120, 125
- [136] Soyly, S., Danişman, K., Saçu, İ. E., & Alçı, M. (2013). Closed-loop control of blood glucose level in type-1 diabetics: A simulation study. In *Electrical and Electronics Engineering (ELECO), 2013 8th International Conference on*, (pp. 371–375). IEEE. 149

- [137] Speyer, J. L., & Jacobson, D. H. (2010). *Primer on Optimal Control Theory*. Philadelphia, USA: SIAM Books. 33
- [138] Tanenbaum, A. S. (2009). *Sistemas operativos modernos. Tercera edición*. México: Pearson Educación. 120, 127
- [139] Tokhi, M., Hossain, M. A., & Shaheed, M. (2003). *Parallel Computing for Real-Time Signal Processing and Control*. Springer. 9, 120, 134
- [140] Tokhi, M. O., Hossian, M. A., & Chambers, C. (1997). Performance evaluation of dsp and trasnputer based systems in sequential real-time applications. *Microprocessors and Mycrosystems*, 21, 237–248. 9, 127
- [141] Troutman, J. L. (1996). *Variational Calculus and Optimal Control*. New York, USA: Springer. 33, 97, 101
- [142] Tyner, D. (2007). *Geometric Jacobian Linearization*. Ph.D. thesis, Queen's University, Kingston, Ontario, Canada. 97, 100
- [143] Van Loan, C. F. (1978). Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3), 395–404. 42, 193, 213
- [144] Villegas Perrasse, A., Abad, S. B., Faciolince, S., Hernández, N., Maya, C., Parra, L., Rivas, E., & Vallejo, P. (2006). Controlling diabetes mellitus and its complications in medellín, colombia, 2001-2003. *Revista Panamericana de Salud Pública*, 20(6), 393–402. 147
- [145] Weinzimer, S. A., Steil, G. M., Swan, K. L., Dziura, J., Kurtz, N., & Tamborlane, W. V. (2008). Fully automated closed-loop insulin delivery versus semiautomated hybrid control in pediatric patients with type 1 diabetes using an artificial pancreas. *Diabetes care*, 31(5), 934–939. 149
- [146] Wilinska, M. E., Budiman, E. S., Taub, M. B., Elleri, D., Allen, J. M., Acerin, C. L., Dunger, D. B., & Hovorka, R. (2009). Over night closed-loop insulin delivery with model predictive control: assessment of hypoglycemia and hyperglycemia risk using simulation studies. *Journal of Diabetes Science and Technology*, 3(5), 1109 – 1120. 149

BIBLIOGRAFÍA

- [147] Xu, J., & Zeng, Z. (2010). Applying optimal control model to dynamic equipment allocation problem: case study of concrete-faced rockfill dam construction project. *Journal of Construction Engineering and Management*, 137(7), 536–550. 164, 165
- [148] Zhang, B., Liu, M., Li, Y., & Wu, L. (2013). Optimization of an industrial air compressor system. *Energy Engineering*, 110(6), 52–64. 162
- [149] Zhang, S. (2009). Matlab: Parallel computing toolbox. https://www.msi.umn.edu/sites/default/files/Matlab_PCT.pdf. [Online; disponible 12-septiembre-2016]. 123
- [150] Zinman, B. (2015). Type 1 diabetes study demonstrates that optimal control can reduce mortality. <http://research.lunenfeld.ca/rss/?page=2121>. 147



Cálculo de Integrales Expresadas con Matrices Exponenciales

Las integrales (3.19 y 3.21) pueden obtenerse de forma casi inmediata de acuerdo a [143], es necesario únicamente expresar adecuadamente las matrices involucradas.

A.1 Integral 3.20

$$\begin{aligned}
 \hat{\Psi}(t, \tau) &:= \int_{\tau}^t \Psi'(\sigma, \tau) Q e^{A(\sigma-\tau)} d\sigma \\
 &= \int_{\tau}^t \left(\int_{\tau}^{\sigma} e^{A'(\sigma-s)} ds \right) Q e^{A(\sigma-\tau)} d\sigma \\
 &= \left[\int_{\tau}^t \left(\int_{\tau}^{\sigma} e^{-A's} ds \right) e^{A'\sigma} Q e^{A\sigma} d\sigma \right] e^{-A\tau}, \quad (\text{A.1})
 \end{aligned}$$

utilizando integración por partes

$$\int u \cdot \dot{v} d\sigma = u \cdot v - \int \dot{u} \cdot v d\sigma, \quad (\text{A.2})$$

A. CÁLCULO DE INTEGRALES EXPRESADAS CON MATRICES EXPONENCIALES

con

$$u = \int_{\tau}^{\sigma} e^{-A's} ds, \quad \dot{v} = e^{A'\sigma} Q e^{A\sigma}.$$

Para u se tiene que

$$\begin{aligned} u &= \int_{\tau}^{\sigma} e^{-A's} ds \\ &= \int_0^{\sigma-\tau} e^{-A'(s+\tau)} ds \\ &= \left(\int_0^{\sigma-\tau} e^{-A's} ds \right) e^{-A'\tau}, \end{aligned} \quad (\text{A.3})$$

teniendo que

$$\begin{aligned} \Psi'(t, \tau) &= \left(\int_0^{\sigma-\tau} e^{-A's} ds \right) e^{A'(t-\tau)} \\ \Psi'(t, \tau) \left(e^{-A'(t-\tau)} \right) &= \int_0^{\sigma-\tau} e^{-A's} ds, \end{aligned} \quad (\text{A.4})$$

reemplazando la ecuación (A.4) en (A.3) se obtiene

$$u = \Psi'(t, \tau) \left(e^{-A'(t-\tau)} \right) e^{-A'\tau} = \Psi'(t, \tau) \left(e^{-A't} \right). \quad (\text{A.5})$$

Para obtener v se procede de la siguiente forma

$$\begin{aligned} v &= \int_{\tau}^t e^{A'\sigma} Q e^{A\sigma} d\sigma \\ &= \int_0^{t-\tau} e^{A'(\sigma+\tau)} Q e^{A(\sigma+\tau)} d\sigma \\ &= e^{A'\tau} \left(\int_0^{t-\tau} e^{A'\sigma} Q e^{A\sigma} d\sigma \right) e^{A\tau} \\ &= e^{A'\tau} \check{\Psi}(t, \tau) e^{A\tau}. \end{aligned} \quad (\text{A.6})$$

De las ecuaciones (A.5) y (A.6) se obtiene $[u \cdot v]$ de la ecuación (A.2), y multiplicando por el $e^{-A\tau}$ faltante respecto a (A.1) queda

$$[u \cdot v] e^{-A\tau} = \Psi'(t, \tau) \left(e^{-A't} \right) e^{A'\tau} \check{\Psi}(t, \tau) e^{A\tau} e^{-A\tau} = \Psi'(t, \tau) e^{-A'(t-\tau)} \check{\Psi}(t, \tau). \quad (\text{A.7})$$

Retomando (A.2) y multiplicando por el $e^{-A\tau}$, la segunda parte de la ecuación se define como sigue:

$$\begin{aligned}
 \left(\int_{\tau}^t \dot{u} \cdot v \, d\sigma \right) e^{-A\tau} &= \left(\int_{\tau}^t \int_{\tau}^{\sigma} e^{-A'\sigma} e^{A's} Q e^{As} \, ds d\sigma \right) e^{-A\tau} \\
 &= \left(\int_{\tau}^t \int_{\tau}^{\sigma} e^{-A'(\sigma-s)} Q e^{As} \, ds d\sigma \right) e^{-A\tau} \\
 &= \left(\int_0^{t-\tau} \int_{\tau}^{\sigma+\tau} e^{-A'(\sigma+\tau)} e^{A's} Q e^{As} \, ds d\sigma \right) e^{-A\tau} \\
 &= \left(\int_0^{t-\tau} \int_{\tau}^{\sigma} e^{-A'(\sigma-s)} Q e^{As} \, ds d\sigma \right). \tag{A.8}
 \end{aligned}$$

Definiendo

$$K(\Delta) = \int_0^{\Delta} \int_0^{\sigma} e^{-A'(\sigma-s)} Q e^{As} \, ds d\sigma, \tag{A.9}$$

cuyo valor puede ser obtenido convenientemente conforme a la matriz exponencial (3.24). La matriz relacionada $\hat{\Psi}(t, \tau)$ queda determinada en (3.27) por las ecuaciones (A.7 y A.9).

Códigos de cálculo en paralelo

En este apéndice se incluyen los códigos fuentes para trabajar los problemas en arquitectura paralela estudiados en el capítulo 5 correspondientes al frenado óptimo de un tren y a un proceso de laminación con actualización de los valores de fase finales.

B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

B.1.1 Código en MATLAB con las divisiones de los intervalos de muestra

En este programa se muestra el código fuente utilizado en MATLAB para trabajar de manera local, con un $t_f = 10$ y realizando una división de intervalos de muestra de duración $\Delta t = 1$.

Configuración local:

```
pmode start 3
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

Código B.1: Pmodebidimensional

```
1  %Inicializacion del Programa
2  clear all
3  clc
4
5  global A B R Q W X0 Pv Ptemp tspan Th Umin Umax n Xre Usat stat
6
7  mpiprofile on
8
9  %problema bidimensional
10 n=2;
11
12 %Tiempo horizonte
13 Th = 10;
14 T0 = 0;
15
16 %Datos para el problema bidimensional
17 A = [0 1/Th;0 0];
18 B = [0; 1/Th];
19
20 Q = 10*[1 0;0 1];
21 R = 0.5;
22 S = 100*[1 0; 0 1];
23
24 %Trabaja hasta con 15 decimales
25 format long
26
27 %Divisiones para obtener la solucion en los intervalos de
28 %muestra fijos de 1
29 divf=1;
30
31 %divisiones
32 div=0.01;
33
34 %Cotas del control
35 Umin = 0;
36 Umax = 3;
37
38 %W
39 W = B*inv(R)*B';
40
41 %tspanr
42 tspanf=[T0:divf:Th];
43 %tspan
44 tspan = [T0:div:Th];
45
46 %Condicion inicial
47 X0 = [1; -1];
48
49 %Matriz S vectorizada
50 Sv = S(:);
```

B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```
51
52 % Solucion del problema de Ricatti
53 % Vector solucion de P como un unico vector , P vectorizado
54 [Tspan Pv] = ode45('Ricattib',fliplr(tspanf),Sv);
55 Pv = flipud(Pv);
56
57 % Desde aca necesito realizar el ciclo para que se
58 % utilice solo una ode subdivision
59 sub = Th/divf;
60 barri = fix((1/div));
61
62 % Utilizo barri para el almacenamiento de los estados ,
63 % es igual al tamaño del vector en cada pasada
64 Ptemp = Pv(1,:);
65 flag = 0;
66 flag2 = 0;
67
68 % Reordenamiento del vector para la multiplicacion matricial
69 P = reshape(Ptemp, size(A));
70
71 % Control semilla
72 U = -Th*inv(R)*B'*P*X0;
73
74 % Control son las cotas de saturacion
75 Usat = min(max(U,Umin),Umax);
76
77 % Para manejar la variable guardada
78 aux=0;
79 aux2=2;
80 aux3=1;
81 Ualma = zeros(1,11);
82 Ualma(1)=Usat;
83
84     numlabs;
85     for i=1:1:sub;
86
87         tiempoinipre = clock;
88         tiempoinipost = clock;
89         if labindex==1
90             tiempoinipl = clock;
91 % Condicion inicial para el estado en la primera
92 % toma de muestreo
93
94 % Recibe el valor de control U desde el otro lab
95         if (i>1)
96             Usat=labReceive(2,21);
97         end
98
99         if i==1
100             suma = 0;
101         end
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
102
103         tspan1=[flag : div : flag+(Th/sub)];
104         [T XL] = ode45('statebmue',tspan1,[X0;0]);
105
106         % Obtencion de la X separada del lagrangiano
107         X = XL(:,[1:n]);
108
109         % Almacenado de X
110         Xalma((aux*barri+1):(aux*barri+1)+barri,1:n) = X;
111
112         L = XL(length(T),n+1);
113         suma = suma + L;
114
115         % Talma((aux*barri+1):(aux*barri+1)+barri,1)=T;
116         aux=aux+1;
117
118         X0=transpose(X(barri+1,:))
119
120         % Aca se debe mandar el estado X0
121         labSend(X0,2,20);
122
123         % Se realiza la asignacion del nuevo estado X0 para la proxima
124         % corrida
125
126         flag=flag+(Th/sub);
127
128         if i==sub
129             Xfinal = XL(length(T),[1:n])';
130         % costo total
131             J = suma + Xfinal'*S*Xfinal
132         end
133
134         tiempofinp1 = etime(clock,tiempoinip1)
135     end
136
137     if labindex==2
138
139         tiempoinip2 = clock;
140         if (i==1)
141             Xre=X0;
142         end
143
144         % Timer de tiempo(Para calculo del control)
145         tiempo_muestra = timer('TimerFcn','stat=false;', 'StartDelay',0.5);
146         start(tiempo_muestra)
147         stat=true;
148
149         % Reordenamiento del vector para la multiplicacion matricial
150         P = reshape(Ptemp,size(A));
151
152         % Control a mandar si no realiza el promedio en el delta
```


B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```

153         Umo = min(max(-Th*inv(R)*B'*P*Xre, Umin), Umax);
154
155         Ptemp=Ptemp(:);
156         % Para mirar la cantidad de veces que realiza la ode
157         count_ode = 0;
158         l=div;
159         tic
160         while (stat==true)&&(count_ode <=50)
161
162             tspanc=[flag2:1:flag2+(Th/sub)];
163         % Solucion de la ode para la prediccion
164         [~,XPcon] = ode45('statecon', tspanc, [Xre;Ptemp]);
165
166         if (stat==true)
167             Xcon=XPcon(:,1:n);
168             Pcon=XPcon(:,n+1:2*n+2);
169         % Tamano del vector
170             M = length(Xcon);
171
172         % Para visualizar en cada corrida hasta que ode realiza
173             tama(i) = M;
174
175         % Inicializar contador y promedio
176             count1=0;
177             promedio=0;
178
179         for m=1:M
180             P=reshape(Pcon(m,1:2*n), size(A));
181             Xp=transpose(Xcon(m,1:n));
182             Up = -Th*inv(R)*B'*P*Xp;
183             promedio = promedio+Up;
184             count1 = count1+1;
185         end
186
187             promedio/count1;
188             count_ode = count_ode+1;
189             l=l/2;
190
191         end
192     end
193
194     Xul=transpose(Xcon(length(Xcon),1:n))
195
196     switch aux3
197     case 1
198         Xdc1=Xcon;
199     case 2
200         Xdc2=Xcon;
201     case 3
202         Xdc3=Xcon;
203     case 4

```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
204             Xdc4=Xcon;
205         case 5
206             Xdc5=Xcon;
207         case 6
208             Xdc6=Xcon;
209         case 7
210             Xdc7=Xcon;
211         case 8
212             Xdc8=Xcon;
213         case 9
214             Xdc9=Xcon;
215         case 10
216             Xdc10=Xcon;
217
218     end
219
220     time = toc;
221
222     if (1-time > 0)
223         U = promedio/count1
224         Usat = min(max(U,Umin),Umax);
225
226         %Ualma=Usat;Almacenamiento del control para la utilizacion
227         % posterior
228         Ualma(aux2) = Usat;
229
230     else
231         Ualma(aux2) = Umo;
232
233     end
234
235     flag2=flag2+divf;
236     aux2=aux2+1;
237     aux3=aux3+1;
238
239     %Aca se debe recibir el estado X0
240     Xre=labReceive(1,20);
241
242     Ptemp = Pv(i+1,:);
243     stat;
244
245     %Aca se debe enviar el valor del control U nuevo que recalcula
246     % este worker
247
248     if (1-time > 0)
249         labSend(Usat,1,21);
250     else
251         labSend(Umo,1,21);
252     end
253
254     wait(tiempo_muestra)
```

B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```

255         tiempofinp2 = etime(clock , tiempoinip2)
256         delete (tiempo_muestra)
257
258     end
259
260     if labindex==3
261
262         if i==1
263             % Matriz calculada con la variacion de S con el gradiente
264             Sac = [64.94 -2.11; -2.11 99.99];
265
266             % Matriz S vectorizada
267             Sacv = Sac (:);
268
269             % Solucion del problema de Ricatti
270             % Vector solucion de P como un unico vector , P vectorizado
271             [~, Pacv] = ode45('Ricattib', flip1r(tspanf), Sacv);
272             Pacv = flipud(Pacv);
273
274             J = 0;
275             sum = 0;
276         end
277
278         tspan1=[flag : div : flag +(Th/sub)];
279         [T XL] = ode45('statebmuecos', tspan1 , [X0;0]);
280
281         X = XL(:, [1:n]);
282         L = XL(length(T), n+1);
283         sum = sum + L;
284
285         X0 = transpose(X(barri+1, :));
286
287         % Reordenamiento del vector para la multiplicacion matricial
288         P = reshape(Pacv(i+1,:), size(A));
289
290         % Control
291         Usat = min(max(-Th*inv(R)*B'*P*X0, Umin), Umax);
292
293         Ualma(aux2) = Usat;
294         aux2 = aux2+(Th/sub);
295         flag = flag + (Th/sub);
296
297         if i==sub
298             Xfinal = XL(length(T), [1:n])';
299             % costo total
300             J = sum + Xfinal'*Sac*Xfinal
301
302         end
303     end
304
305     tiempofinpre = etime(clock , tiempoinipre)

```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
306 %Tiempo de espera para sincronizar
307     pause(1-tiempofinpre)
308     tiempofinpost = etime(clock, tiempoinipost)
309
310 end
311
312 if labindex==2
313     concatena
314 end
315
316 mpiprofile off
317 mpiprofile viewer
```

B.1.2 Código en C++ para estudiar la evolución del costo ante el refinamiento en la división de los intervalos de muestra

Este programa exhibe el código fuente utilizado en el cluster para el problema trabajado realizando la división en intervalos de muestra, inicialmente con un valor de $\Delta t = 1$. Se incluyó la cabecera `mpi.h` que permite el uso estándar de librerías de paso de mensajes. En el programa se utilizaron funciones de comunicación punto a punto para comunicar entre sí a los dos procesadores reservados, uno para la emulación del modelo y otro para el cálculo del control.

Para la compilación y ejecución del programa son necesarios los siguientes comandos:

```
$ mpicxx bidimensionalC.cpp -o Cpar.bin
$ mpiexec -f machi.dat -n 2 Cpar.bin
```

Código B.2: Bidimensional C++

```
1  /*Problema bidimensional
2  The cheapest stop of a train
3  Version MPI
4  John Anderson Gómez Múnera
5  */
6
7  #include <iostream>
8  #include <boost/numeric/odeint.hpp>
9  #include <boost/numeric/ublas/vector.hpp>
10 #include <boost/numeric/ublas/matrix.hpp>
11 #include <boost/numeric/ublas/io.hpp>
12 #include <fstream>
13 #include "gnuplot_i_calnum.h"
```

B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```
14 #include <cmath>
15 #include <iomanip>
16 #include <fstream>
17 #include "algebraLineal.h"
18 #include <mpi.h>
19
20 using namespace CALNUM;
21 #define SLEEP_LGTH 3 //la figura permanece 10 segundos en pantalla
22
23 typedef boost::numeric::ublas::vector< double> state_type;
24
25 using namespace std;
26 using namespace boost::numeric::odeint;
27 using namespace boost::numeric::ublas;
28
29 //Variables globales
30 int dim = 100;
31 double p1[101] = {0};
32 double p2[101] = {0};
33 double p3[101] = {0};
34 double p4[101] = {0};
35 float Usat;
36 double x1[11] = {0};
37 double x2[11] = {0};
38
39 //Prototipos de funciones
40 void rica( const state_type &p , state_type &dpdt , double t);
41 void write_rica ( const state_type &p , const double t);
42 void inverso(double *p1, double *p2, double *p3, double *p4);
43 void write_txt(double *p1, double *p2, double *p3, double *p4);
44 void statebmue( const state_type &x , state_type &dxdt , double t);
45 void write_state ( const state_type &x , const double t);
46
47 void rica( const state_type &p , state_type &dpdt , double t)
48 {
49     dpdt[0] = -1*( 0.2*p[1]*p[2] - 1 );
50     dpdt[1] = -1*(0.2*p[1]*p[3] - 0.1*p[0]);
51     dpdt[2] = -1*(0.2*p[3]*p[2] - 0.1*p[0]);
52     dpdt[3] = -1*( 0.2*p[3]*p[3] - 0.1*p[2] - 0.1*p[1] - 1);
53 }
54
55 void write_rica ( const state_type &p , const double t)
56 {
57     static int i = 0;
58     cout << t << '\t' << p[0] << '\t' << p[1] << '\t' \
59 << p[2] << '\t' << p[3] << '\t' << i << endl;
60
61     p1[i] = p[0];
62     p2[i] = p[1];
63     p3[i] = p[2];
64     p4[i] = p[3];
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
65
66     i++;
67 }
68
69 void inverso(double *p1, double *p2, double *p3, double *p4)
70 {
71     //int dim =100;
72     for(int l = 0; l < dim; l++, dim--)
73     {
74         double aux1 = p1[l];
75         double aux2 = p2[l];
76         double aux3 = p3[l];
77         double aux4 = p4[l];
78
79         p1[l] = p1[dim];
80         p2[l] = p2[dim];
81         p3[l] = p3[dim];
82         p4[l] = p4[dim];
83
84         p1[dim] = aux1;
85         p2[dim] = aux2;
86         p3[dim] = aux3;
87         p4[dim] = aux4;
88     }
89 }
90
91 void write_txt(double *p1, double *p2, double *p3, double *p4)
92 {
93     static float tt=0;
94     FILE *ERica;
95     ofstream fout("datos.txt");
96     fout << "P1\t" << "P2\t" << "P3\t" << "P4\t" << endl;
97     ERica = fopen(" Ricatti.txt", "w");
98     for(int l=0; l<=dim; l++)
99     {
100         fprintf(ERica, "%f\t%f\t%f\t%f\t%f\t\n", tt, p1[l], p2[l], p3[l], p4[l]);
101         fout <<tt<<'\t'<< p1[l] << "\t"<< p2[l]<< "\t"<<p3[l] <<"\t"<< p4[l]<<endl;
102         tt+=0.1;
103     }
104     fclose ( ERica);
105     fout.close();
106 }
107
108 //Solución de la ecuación de estados en el primer procesador
109 void statebmue( const state_type &x , state_type &dxdt , double t)
110 {
111     dxdt[0] = (0.1 * x[1]);
112     dxdt[1] = (0.1 * Usat);
113 }
114
115 //Se obtienen los valores de los vectores de estado
```

B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```
116 void write_state ( const state_type &x , const double t )
117 {
118     static int i = 0;
119     x1[i] = x[0];
120     x2[i] = x[1];
121     i++;
122 }
123
124 int main(int argc , char **argv)
125 {
126     state_type p(4);
127     p[0] = 100; p[1] = 0; p[2] = 0; p[3] = 100;
128     typedef runge_kutta_dopri5< state_type> stepper;
129     integrate_const(make_dense_output<stepper>(1E-6 ,1E-6),rica ,p\
130 ,0.0,10.0,0.1, write_rica);
131
132     inverso(p1 ,p2 ,p3, p4);
133     cout <<'\n';
134     int nd = (sizeof(p1)/sizeof(p1[0]));
135     Vector<double> t(nd), pp1(nd), pp2(nd), pp3(nd), pp4(nd);
136     for(int l = 0; l<nd;l++){
137         cout << p1[l] <<'\t' << p2[l] <<'\t' << p3[l] <<'\t' \
138 << p4[l] <<'\t'<<1*0.1<<endl;
139         t[l] = l*0.10;
140         pp1(l) = p1[l];
141         pp2(l) = p2[l];
142         pp3(l) = p3[l];
143         pp4(l) = p4[l];
144     }
145     cout <<"\nsize:\t" << nd << endl;
146
147     write_txt(p1, p2, p3, p4 );
148
149     //Dibujo de puntos yy versus xx
150     Gnuplot gl = Gnuplot();
151     gl.set_style("lines");
152     gl.set_xlabel("t");
153     gl.set_ylabel("P");
154     gl.plot_xy(t, pp1, "P1_");
155     sleep(SLEEP_LGTH);
156     gl.plot_xy(t, pp2, "P2_");
157     sleep(SLEEP_LGTH);
158     gl.plot_xy(t, pp3, "P3_");
159     sleep(SLEEP_LGTH);
160     gl.plot_xy(t, pp4, "P4_");
161
162     sleep(SLEEP_LGTH);
163     //la figura permanece SLEEP_LGTH segundo
164
165     //declaración de algunas constantes , tiempo horizonte
166     int Th = 10;
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
167     int T0 = 0;
168
169     //divisiones
170     double div = 0.01;
171
172     //cotas de control
173     float Umin = 0;
174     float Umax = 3;
175     int sub = Th;
176     float barri = 1/div;
177     double flag = 0;
178     float flag2= 0;
179     float R = 0.5;
180
181     //B transpuesto
182     matrix<double> Bt (1,2);
183     Bt(0,0) = 0;
184     Bt(0,1) = -2;
185
186     //X0 semilla
187     matrix<double> X0 (2,1);
188     X0(0,0) = 1;
189     X0(1,0) = -1;
190
191     //Xre para la multiplicación;
192     matrix<double> Xre(2,1);
193
194     //Ricatti semilla
195     matrix<double> Ps(2,2);
196     Ps(0,0) = pp1(1);
197     Ps(0,1) = pp2(1);
198     Ps(1,0) = pp3(1);
199     Ps(1,1) = pp4(1);
200
201     //Ricatti para multiplicar en el procesador 2
202     matrix<double> Ptem(2,2);
203
204     matrix<double> tem = prod(Ps,X0);
205     //float tem1 = -Th*(1/R);
206
207     //control semilla
208     matrix<double> U = prod(Bt,tem);
209
210     float Us = U(0,0);
211
212     Usat = min(max(Us,Umin),Umax);
213
214     //Manejo de variables guardadas
215     int aux = 0;
216     int aux2 = 2;
217     int aux3 = 1;
```


B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```
218     Vector<double> Ualma(Th+1);
219
220     //Para recibir
221     float Xre1, Xre2;
222
223     //int ierror;
224     int myid;    /* Identificador del proceso*/
225     int size;   /* Numero de procesos*/
226     double startwtime1 = 0.0, startwtime2=0.0, \
227     endwtime1, endwtime2;    //Contadores de tiempo de inicio
228
229     int tag=0;
230     int tag1=1;
231     int source, dest;
232
233     //Iniciar el vector de estados
234     state_type x(2);
235     MPI_Status status;
236     MPI_Init(&argc, &argv);
237     MPI_Comm_rank(MPLCOMM_WORLD, &myid);
238     MPI_Comm_size(MPLCOMM_WORLD, &size);
239
240     for (int i = 0; i <= sub; i++)
241     {
242         if(myid == 0)
243         {
244             startwtime1 = MPI_Wtime();
245             source = 1;
246             dest = 1;
247
248             if(i>1)
249                 Usat = MPI_Recv(&Usat, 1, MPLFLOAT, 1, tag, MPLCOMM_WORLD, &status);
250
251             x[0] = X0(0,0); x[1] = X0(1,0);
252             typedef runge_kutta_dopri5< state_type > stepper;
253             integrate_const( make_dense_output< stepper >(1E-6, 1E-6), \
254             statebmue, x, flag, flag+ 1.0, 0.1, write_state );
255
256             int ndest = (sizeof(x1)/sizeof(x1[0]));
257             cout << "Estado_final_pasada:_" << x[0] << endl;
258             Vector<double> tst(ndest), xx1(ndest), xx2(ndest), Usatv(ndest);
259             for (int l = 0; l<ndest; l++){
260                 cout << x1[l] << '\t' << x2[l] << '\t' << flag + l*0.1 << endl;
261                 tst[l] = flag+l*0.10;
262                 xx1[l] = x1[l];
263                 xx2[l] = x2[l];
264                 Usatv[l] = Usat;
265             }
266             MPI_Send(&x, 2, MPLDOUBLE, 1, tag1, MPLCOMM_WORLD);
267             flag = flag + (1/sub); //Para aumentar el intervalo de muestra
268             X0(0,0) = x[0]; \
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
269         X0(1,0) = x[1];
270 //Se reasignan los valores para la próxima pasada, los estados
271 //quedan con el último valor de la solución de la ODE
272
273         //Dibujo de puntos X "Estados" versus t "tiempo"
274         Gnuplot g2 = Gnuplot();
275         g2.set_style("lines");
276         g2.set_xlabel("t");
277         g2.set_ylabel("X");
278         g2.plot_xy(tst , xx1 , "X1");
279         g2.plot_xy(tst , xx2 , "X2");
280         g2.plot_xy(tst , Usatv , "Control");
281
282         endwtime1 = MPI_Wtime();//Cálculo del tiempo del procesador 1
283         sleep(SLEEP_LGTH);
284         //la figura permanece SLEEP_LGTH segundo
285
286         cout << "Tiempo de ejecución procesador 1:" << \
287 startwtime1-endwtime1 << endl;
288     }//Acá sale del procesador 1
289
290     if (myid == 1)
291     {
292         startwtime2 = MPI_Wtime();
293         if (i = 1){
294             Xre = X0;
295             Ptem = Ps;
296         }
297         tem = prod(Ptem , Xre);
298         //control semilla
299         U = prod(Bt , tem);
300         Us = U(0 , 0);
301         Usat = min(max(Us , Umin) , Umax);
302
303         source = 0;
304         dest = 0;
305         //Para recibir el estado
306         MPI_Recv(&x , 2 , MPI_DOUBLE , source , tag1 , MPI_COMM_WORLD , &status);
307
308         //Para rearmar P
309         Ptem(0,0) = pp1(10*i + 1);
310         Ptem(0,1) = pp2(10*i + 1);
311         Ptem(1,0) = pp3(10*i + 1);
312         Ptem(1,1) = pp4(10*i + 1);
313
314         MPI_Send(&Usat , 1 , MPI_FLOAT , dest , tag , MPI_COMM_WORLD);
315         endwtime2 = MPI_Wtime();
316     }//Fin del procesador 2
317 }//Cierre del for de intervalos
318 MPI_Finalize();
319 return 0;
```

B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

320 }

B.1.3 Código en MATLAB para el problema fantasma

Este programa muestra el código de MATLAB al trabajar el problema fantasma buscando los valores de fase finales ocultos y sin dividir en intervalos de tiempo. Para habilitar la sección paralela como configuración local se utiliza el comando:

```
pmode start 4
```

Código B.3: Fantasma

```
1  %Bidimensional, problema del tren resolucio
2  %a traves del Problema fantasma
3
4  clc
5  clear
6  global A B R Q W Th Umin Umax n Pv tspan Up1 Up2 Up3 Up4 Up5 Up6 Up7 Up8;
7
8  tic
9  %problema bidimensional
10 n=2;
11
12 %Tiempo horizonte
13 Th = 10;
14 T0 = 0;
15
16 %Datos para el problema bidimensional
17 A = [0 1/Th;0 0];
18 B = [0; 1/Th];
19
20 Q = 10*[1 0;0 1];
21 R = 0.5;
22 S = 100*[1 0; 0 1];
23
24 %Matriz de penalizacion inicial
25 Sini = 1*[1 0; 0 1];
26
27 %Cotas del control
28 Umin = 0;
29 Umax = 3;
30
31 %Trabaja hasta con 15 decimales
32 format long
33
34 %Estados iniciales
35 X0 = [1; -1];
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
36
37 %W
38 W = B*inv(R)*B';
39
40 % divisiones
41 div = 0.01;
42 % tspan
43 tspan = T0:div:Th;
44
45 % Matriz S vectorizada
46 Sv = S(:);
47
48 % Solucion del problema de Ricatti
49 % Vector solucion de P como un unico vector, P vectorizado
50 [Tspan Pv] = ode45('Ricattib',fliplr(tspan),Sv);
51 Pv = flipud(Pv);
52
53 % Costo semilla inicial
54 J0 = 0;
55
56 % Solucion de la ecuacion de estados
57 [T, XJ] = ode45('statef',tspan,[X0;J0]);
58
59 Xfinalseed = XJ(length(T),1:n)';
60 Lambdafinalseed = 2*S*Xfinalseed;
61
62 %%
63 %%      *
64 %%      |
65 %%  |<--- h --->|
66 %% *-----@-----*
67 %%      |
68 %%      *
69 %El valor de h es la separacion(variacion) de la semilla
70 %En X(Estados)
71
72 h = 0.1;
73
74 %%      *
75 %%      |
76 %%  |<--- k --->|
77 %% *-----@-----*
78 %%      |
79 %%      *
80 %El valor de k es la separacion(variacion) de la semilla
81 %En lambda
82 k = 0.5;
83
84 %Definicion de epsilon para {Xn;Lan} = {Xv;Lav} - epsilon*grad(J)
85 epsilon = 2e-3;
86
```

B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```

87 %Inicializacion de los gradientes
88 Jt = zeros(1,8,codistributorId());
89
90 % Iteraciones
91 iter = 30;
92
93 % Jalma inicializacion del vector para el costo
94 Jalma = zeros(8,iter);
95
96 %Inicializacion de los vectores del gradiente
97 dJX1 = zeros(1,iter);
98 dJX2 = zeros(1,iter);
99
100 dJL1 = zeros(1,iter);
101 dJL2 = zeros(1,iter);
102
103 for i = 1:iter
104
105     for j = drange(1:8) %Comando para distribuir en los cores
106
107         if (labindex==1)
108
109             rho = Xfinalseed + [h 0]';
110             mu = Lambdafinalseed;
111
112             % Calculo de la nueva S condicion final
113             Sn = (1/2)*((mu*mu')/(rho'*mu));
114             Snv = Sn(:);
115
116             % Calculo de la nueva u(final)
117             uff = -(R^-1)*B'*Sn*rho;
118
119             % Ecuacion de resolucio para obtener el control hacia atras
120             [~, PXU] = ode45(@Ratras, flip1r(tspan), [Snv; rho; uff]);
121
122             % Obtencion del control
123             Up1 = flipud(PXU(:,4*n-1));
124
125             % Costo
126             C0 = 0;
127
128             % Ecuacion de estados resuelta hacia adelante
129             [Tha, XJp] = ode45('statefp', tspan, [X0; C0]);
130
131             % Estados finales
132             Xfinal = XJp(length(Tha), 1:n)';
133
134             % Costo de trayectoria
135             Jp1 = XJp(length(Tha), n+1);
136
137             % Costo debido a la penalizacion final

```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
138         Jp2 = Xfinal'*S*Xfinal;
139
140         % Costo total
141         Jt(j) = Jp1 + Jp2 ;
142
143     end
144
145     if (labindex==2)
146         rho = Xfinalseed - [h 0]';
147         mu = Lambdafinalseed;
148
149         % Calculo de la nueva S condicion final
150         Sn = (1/2)*((mu*mu')/(rho'*mu));
151         Snv = Sn(:);
152
153         % Calculo de la nueva u(final)
154         uff = -(R^-1)*B'*Sn*rho;
155
156         % Ecuacion de resolucio n para obtener el control hacia atras
157         [~, PXU] = ode45(@Ratras, tspan, [Snv; rho; uff]);
158
159         % Obtencion del control
160         Up2 = flipud(PXU(:,4*n-1));
161
162         % Costo
163         C0 = 0;
164
165         % Ecuacion de estados resuelta hacia adelante
166         [Tha, XJp] = ode45('statefp2', tspan, [X0; C0]);
167
168         % Estados finales
169         Xfinal = XJp(length(Tha), 1:n)';
170
171         % Costo de trayectoria
172         Jp1 = XJp(length(Tha), n+1);
173
174         % Costo debido a la penalizacion final
175         Jp2 = Xfinal'*S*Xfinal;
176
177         % Costo total
178         Jt(j) = Jp1 + Jp2 ;
179
180     end
181
182     if (labindex==3)
183         rho = Xfinalseed + [0 h]';
184         mu = Lambdafinalseed;
185
186         % Calculo de la nueva S condicion final
187         Sn = (1/2)*((mu*mu')/(rho'*mu));
188         Snv = Sn(:);
```

B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```

189
190     % Calculo de la nueva u(final)
191     uff = -(R^(-1))*B'*Sn*rho;
192
193     % Ecuacion de resolucio para obtener el control hacia atras
194     [~, PXU] = ode45(@Ratras, flip1r(tspan), [Snv; rho; uff]);
195
196     % Obtencion del control
197     Up3 = flipud(PXU(:, 4*n-1));
198
199     % Costo
200     C0 = 0;
201
202     % Ecuacion de estados resuelta hacia adelante
203     [Tha, XJp] = ode45('statefp3', tspan, [X0; C0]);
204
205     % Estados finales
206     Xfinal = XJp(length(Tha), 1:n)';
207
208     % Costo de trayectoria
209     Jp1 = XJp(length(Tha), n+1);
210
211     % Costo debido a la penalizacion final
212     Jp2 = Xfinal'*S*Xfinal;
213
214     % Costo total
215     Jt(j) = Jp1 + Jp2;
216
217 end
218
219 if (labindex==4)
220     rho = Xfinalseed - [0 h]';
221     mu = Lambdafinalseed;
222
223     % Calculo de la nueva S condicion final
224     Sn = (1/2)*((mu*mu')/(rho'*mu));
225     Snv = Sn(:);
226
227     % Calculo de la nueva u(final)
228     uff = -(R^(-1))*B'*Sn*rho;
229
230     % Ecuacion de resolucio para obtener el control hacia atras
231     [~, PXU] = ode45(@Ratras, flip1r(tspan), [Snv; rho; uff]);
232
233     % Obtencion del control
234     Up4 = flipud(PXU(:, 4*n-1));
235
236     % Costo
237     C0 = 0;
238
239     % Ecuacion de estados resuelta hacia adelante

```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
240     [Tha,XJp] = ode45('statefp4',tspan,[X0;C0]);
241
242     % Estados finales
243     Xfinal = XJp(length(Tha),1:n)';
244
245     % Costo de trayectoria
246     Jp1 = XJp(length(Tha),n+1);
247
248     % Costo debido a la penalizacion final
249     Jp2 = Xfinal'*S*Xfinal;
250
251     % Costo total
252     Jt(j) = Jp1 + Jp2 ;
253
254 end
255
256 if (labindex==5)
257     rho = Xfinalseed;
258     mu = Lambdafinalseed + [k 0]';
259
260     % Calculo de la nueva S condicion final
261     Sn = (1/2)*((mu*mu')/(rho'*mu));
262     Snv = Sn(:);
263
264     % Calculo de la nueva u(final)
265     uff = -(R^-1)*B'*Snrho;
266
267     % Ecuacion de resolucio para obtener el control hacia atras
268     [~, PXU] = ode45(@Ratras, flip1r(tspan),[Snrho;uff]);
269
270     % Obtencion del control
271     Up5 = flipud(PXU(:,4*n-1));
272
273     % Costo
274     C0 = 0;
275
276     % Ecuacion de estados resuelta hacia adelante
277     [Tha,XJp] = ode45('statefp5',tspan,[X0;C0]);
278
279     % Estados finales
280     Xfinal = XJp(length(Tha),1:n)';
281
282     % Costo de trayectoria
283     Jp1 = XJp(length(Tha),n+1);
284
285     % Costo debido a la penalizacion final
286     Jp2 = Xfinal'*S*Xfinal;
287
288     % Costo total
289     Jt(j) = Jp1 + Jp2 ;
290
```


B.1 Códigos para el primer acercamiento, caso de estudio: frenado óptimo del tren

```

291     end
292
293     if (labindex==6)
294         rho = Xfinalseed;
295         mu = Lambdafinalseed - [k 0]';
296
297         % Calculo de la nueva S condicion final
298         Sn = (1/2)*((mu*mu')/(rho'*mu));
299         Snv = Sn(:);
300
301         % Calculo de la nueva u(final)
302         uff = -(R^-1)*B'*Snrho;
303
304         % Ecuacion de resolucio para obtener el control hacia atras
305         [~, PXU] = ode45(@Ratras, flip1r(tspan), [Snv; rho; uff]);
306
307         % Obtencion del control
308         Up6 = flipud(PXU(:, 4*n-1));
309
310         % Costo
311         C0 = 0;
312
313         % Ecuacion de estados resuelta hacia adelante
314         [Tha, XJp] = ode45('statefp6', tspan, [X0; C0]);
315
316         % Estados finales
317         Xfinal = XJp(length(Tha), 1:n)';
318
319         % Costo de trayectoria
320         Jp1 = XJp(length(Tha), n+1);
321
322         % Costo debido a la penalizacion final
323         Jp2 = Xfinal'*S*Xfinal;
324
325         % Costo total
326         Jt(j) = Jp1 + Jp2 ;
327
328     end
329
330     if (labindex==7)
331         rho = Xfinalseed;
332         mu = Lambdafinalseed + [0 k]';
333
334         % Calculo de la nueva S condicion final
335         Sn = (1/2)*((mu*mu')/(rho'*mu));
336         Snv = Sn(:);
337
338         % Calculo de la nueva u(final)
339         uff = -(R^-1)*B'*Snrho;
340
341         % Ecuacion de resolucio para obtener el control hacia atras

```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
342     [~, PXU] = ode45(@Ratras, flip1r(tspan), [Snv; rho; uff]);
343
344     % Obtencion del control
345     Up7 = flipud(PXU(:, 4*n-1));
346
347     % Costo
348     C0 = 0;
349
350     % Ecuacion de estados resuelta hacia adelante
351     [Tha, XJp] = ode45('statefp7', tspan, [X0; C0]);
352
353     % Estados finales
354     Xfinal = XJp(length(Tha), 1:n)';
355
356     % Costo de trayectoria
357     Jp1 = XJp(length(Tha), n+1);
358
359     % Costo debido a la penalizacion final
360     Jp2 = Xfinal' * S * Xfinal;
361
362     % Costo total
363     Jt(j) = Jp1 + Jp2 ;
364
365 end
366
367 if (labindex==8)
368     rho = Xfinalseed;
369     mu = Lambdafinalseed - [k 0]';
370
371     % Calculo de la nueva S condicion final
372     Sn = (1/2)*((mu*mu')/(rho'*mu));
373     Snv = Sn(:);
374
375     % Calculo de la nueva u(final)
376     uff = -(R^-1)*B'*Snv*rho;
377
378     % Ecuacion de resolucio para obtener el control hacia atras
379     [~, PXU] = ode45(@Ratras, flip1r(tspan), [Snv; rho; uff]);
380
381     % Obtencion del control
382     Up8 = flipud(PXU(:, 4*n-1));
383
384     % Costo
385     C0 = 0;
386
387     % Ecuacion de estados resuelta hacia adelante
388     [Tha, XJp] = ode45('statefp8', tspan, [X0; C0]);
389
390     % Estados finales
391     Xfinal = XJp(length(Tha), 1:n)';
392
```

B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

```
393         % Costo de trayectoria
394         Jp1 = XJp(length(Tha),n+1);
395
396         % Costo debido a la penalizacion final
397         Jp2 = Xfinal'*S*Xfinal;
398
399         % Costo total
400         Jt(j) = Jp1 + Jp2 ;
401
402     end
403 end
404     % Recoleccion de los gradientes
405     Jalma(:,i) = gather(Jt)';
406
407     % Uso de la diferencia centrada
408     %
409     % {J(X(T)+ h) - J(X(T) - h)}/(2h)
410     %
411
412     % Derivada respecto a la coordenada X1
413     dJX1(i) = (Jalma(1,i) - Jalma(2,i))/(2*h);
414
415     % Derivada respecto a la coordenada X2
416     dJX2(i) = (Jalma(3,i) - Jalma(4,i))/(2*h);
417
418     % Derivada respecto a la coordenada X1
419     dJL1(i) = (Jalma(5,i) - Jalma(6,i))/(2*k);
420
421     % Derivada respecto a la coordenada X2
422     dJL2(i) = (Jalma(7,i) - Jalma(8,i))/(2*k);
423
424     % calculo de los estados para la proxima corrida
425     Xfinalseed = Xfinalseed - epsilon*[dJX1(i) dJX2(i)]';
426     Lambdafinalseed = Lambdafinalseed - epsilon*[dJL1(i) dJL2(i)]';
427 end
428 toc
```

B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

Inicialmente se presenta el código principal en el cual se hace la definición del modelo estudiado y también el código fuente que contiene las funciones utilizadas para el cálculo del problema.

B. CÓDIGOS DE CÁLCULO EN PARALELO

B.2.1 Programa principal para el problema de dimensión $n = 100$

El programa principal para el código fuente presentado es el correspondiente al de dimensión $n = 100$, aunque para las variaciones de las diferentes dimensiones es suficiente con cambiar el n , el h y utilizar una semilla apropiada.

Código B.4: Principal

```
1 // John Anderson Gómez Múnera
2 #include "funciones.hpp"
3
4 //Declaración de variables globales
5 int n=100;
6 mat H(2*n,2*n);
7 mat S(n,n);
8 mat A(n,n);
9 mat Q(n,n);
10 vec B(n,1);
11 vec X0(n,1);
12 double Umin, Umax, tf, R, dt;
13
14 // Todo el vector X
15 mat X(n+1,5001); //tiene dimensiones como n+1 porque incluye el costo \
16 y 101 es la división de las muestras en la ode
17 //Variable de estados para la integración, usando odeint
18 typedef vector<double> state_type;
19
20 int main (int argc, char **argv) {
21
22     wall_clock timer;
23     timer.tic();
24     omp_set_num_threads(1);
25
26     Umin = -1.5, Umax = -0.5, tf = 0.5;
27     R = 100.0;
28     dt = 0.0001;
29     vec X0J0(n+1,1);
30
31     //variables del problema
32     double V0 = 1.0; //velocidad cerca del estado estable
33     double h = 0.10; // Pasos de discretización
34     double a = 1.001; // peso en la ecuación del coeficiente de transferencia \
35 de calor por conducción
36     double theta_a = 20; //temperatura ambiente
37     double theta_0 = 700; // temperatura inicial
38
39     //estados finales
40     vec Xf;
41
```

B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

```
42 //Iniciando el problema
43 // matriz A
44 A.zeros();
45 for (int k = 0; k < n-1 ; k++){
46     A(k,k) = ((V0/h)-a);
47     A(k,k+1) = -(V0/h);
48 }
49 A(n-1,n-1) = -(a+(V0/h));
50 A(n-1,n-2) = (V0/h);
51
52 //vector B
53 for (int k = 0; k < n; k++){
54     B(k) = ((a/V0)*(theta_0-theta_a)*exp(-(a/V0)*k));
55 }
56
57 //Condicion inicial
58 X0.ones();
59 X0 = 100*X0;
60
61 //Condicion inicial y costo inicial
62 X0J0.zeros();
63 X0J0(span(0,n-1)) = X0;
64
65 // matriz de penalización final
66 S = 15* eye(n,n);
67 //S.print("S:");
68 // matriz de penalización de estados
69 Q = 0.05*eye(n,n);
70 // W
71 mat W = B*(1.0/R)*B.t();
72
73 //matriz hamiltoniana
74 H.zeros();
75 H.submat(span(0,n-1),span(0,n-1)) = A;
76 H.submat(span(0,n-1),span(n,2*n-1)) = -0.5*W;
77 H.submat(span(n,2*n-1),span(0,n-1)) = -2.0*Q;
78 H.submat(span(n,2*n-1),span(n,2*n-1)) = -A.t();
79
80 //-----
81 //semillas del estado y coestado del fantasma
82 vec rho(n,1);
83 rho.load("rho100.dat", auto_detect);
84
85 vec mu(n,1);
86 mu = 2*S*rho;
87
88 //----- taus semillas
89 double tau1, tau2;
90 tau1 = 1.3166549e-01;
91 tau2 = 1.5594532e-01;
92
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
93 //----- variables para el bucle, inicialización-----/
94 vec dJtotal;
95 vec taus;
96
97 //Para el gradiente
98 vec rhog;
99 vec mug;
100
101 //-----variables para el bucle, almacenar inicialización ----/
102 double iter = 1;
103 double gamma = 1.0;
104
105 //taus gradientes
106 double tau1grad, tau2grad;
107
108 //Almacenamiento de los taus
109 vec tau1g(iter);
110 vec tau2g(iter);
111
112 //----- Inicio del bucle -----/
113
114 for (int k = 0; k < iter; k++){
115     // cálculo de los  $dJ/\rho_{\mu}(k, l)$ 
116     dJtotal = dJdeta(tau1, tau2, rho, mu);
117
118     //delta/rho
119     rhog = dJtotal(span(0, n-1));
120
121     //delta/mu
122     mug = dJtotal(span(n, 2*n-1));
123
124     //uso del gradiente
125     rho = rho - gamma*rhog;
126     mu = mu - gamma*mug;
127
128     // taus gradiente
129     tau1grad = dJdtau1(tau1, tau2, rho, mu);
130     tau2grad = dJdtau2(tau1, tau2, rho, mu);
131
132     //taus nuevos
133     tau1 = tau1 - tau1grad;
134     tau2 = tau2 - tau2grad;
135
136     //Almacenando los taus
137     tau1g(k) = tau1;
138     tau2g(k) = tau2;
139 }
140
141 cout << "\nEl valor de gamma es: " << gamma << endl;
142
143 double tiempo = timer.toc();
```

B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

```
144     cout << "\n#_de_segundos:_\n" << tiempo << endl;
145
146     return 0;
147 }
```

B.2.2 Código fuente para las funciones

Código fuente con las funciones utilizadas para la realización del programa principal

Código B.5: Código fuente de las funciones

```
1  #include "funciones.hpp"
2  // John Anderson Gomez Munera
3  // Funcion para guardar el coste en un archivo.txt
4  // Calcula la exponencial de la matriz hamiltoniana
5  mat Emat(double t1, double t2){
6      mat U = expmat(H*(t2-t1));
7      return U;
8  }
9  // Funcion que obtiene alpha y beta
10 mat albe(mat U, vec rho, vec mu){
11     mat albe;
12     mat Smoc = Smo(rho, mu);
13     mat I(n, n);
14     I.eye();
15     mat IS;
16     IS = join_vert(I, 2*Smoc);
17     albe = U*IS;
18     return albe;
19 }
20 //Calcula S por intermedio de rho y mu
21 mat Smo(vec rho, vec mu){
22     mat Smo;
23     double div = as_scalar(rho.t()*mu);
24     Smo = (1.0/2.0)*(mu*mu.t())/div;
25     return Smo;
26 }
27 // Función para calcular Riccati
28 mat Ricca(double t1, double t2, vec rho, vec mu){
29     mat P;
30     mat U = Emat(t1, t2);
31     mat alphabeta;
32     alphabeta = albe(U, rho, mu);
33     mat alpha = alphabeta(span(0, n-1), span(0, n-1));
34     mat beta = alphabeta(span(n, 2*n-1), span(0, n-1));
35     P = (1.0/2.0)*beta*alpha.i();
36     return P;
37 }
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
38 // Matriz fundamental o matriz de transición de estados
39 mat Mfund(double t, double tau){
40     mat I(n,n);
41     I.eye();
42     mat tmp(2*n,2*n);
43     tmp.zeros();
44     tmp.submat(span(0,n-1),span(n,2*n-1)) = I;
45     tmp.submat(span(n,2*n-1),span(n,2*n-1)) = -A;
46     mat etmp(2*n,2*n);
47     etmp = expmat(tmp*(t-tau));
48     mat tmp2(n,n);
49     tmp2 = etmp.submat(span(0,n-1),span(n,2*n-1));
50     return expmat(A*(t-tau))*tmp2;
51 }
52 // Formula algebraica matriz relacionada 1
53 mat MR_1(double t, double tau){
54     mat I(n,n);
55     I.eye();
56     mat tmp(3*n,3*n);
57     tmp.zeros();
58     tmp.submat(span(0,n-1),span(n,2*n-1)) = I;
59     tmp.submat(span(n,2*n-1),span(n,2*n-1)) = -A.t();
60     tmp.submat(span(n,2*n-1),span(2*n,3*n-1)) = Q;
61     tmp.submat(span(2*n,3*n-1),span(2*n,3*n-1)) = A;
62     mat etmp(3*n,3*n);
63     etmp = expmat(tmp*(t-tau));
64     mat tmp2(n,n);
65     tmp2 = etmp.submat(span(0,n-1),span(2*n,3*n-1));
66     return trans(Mfund(t,tau))*expmat(-A.t()*(t-tau))*MR_2(t,tau)-tmp2;
67 }
68 // Matriz relacionada 2
69 mat MR_2(double t, double tau){
70     mat tmp(2*n,2*n);
71     tmp.zeros();
72     tmp.submat(span(0,n-1),span(0,n-1)) = -trans(A);
73     tmp.submat(span(0,n-1),span(n,2*n-1)) = Q;
74     tmp.submat(span(n,2*n-1),span(n,2*n-1)) = A;
75     mat etmp(2*n,2*n);
76     etmp = expmat(tmp*(t-tau));
77     mat tmp2(n,n);
78     tmp2 = etmp.submat(span(0,n-1),span(n,2*n-1));
79     mat tmp3(n,n);
80     tmp3 = etmp.submat(span(n,2*n-1),span(n,2*n-1));
81     mat tmp4(n,n);
82     tmp4 = tmp3.t();
83     return tmp3.t()*tmp2;
84 }
85 /*****Para calcular X(tau1) *****/
86 // Funcion matriz fundamental para integrar
87 vec Xtau1(double t1){
88     return expmat(A*t1)*X0 + Mfund(t1,0)*B*Umin;
```


B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

```

89 }
90 // Funcion para calcular X(tau1,tau2,rho,mu) retorna un vector
91 vec Xtau2(double t1,double t2, vec rho, vec mu)
92 {
93     //Se obtienen las componentes para construir X(tau2)
94     mat U = Emat(t1,t2);
95     mat U1, U2;
96     U1 = U.submat(0,0,n-1,n-1);
97     U2 = U.submat(0,n,n-1,2*n-1);
98     // Se obtiene Riccati en P(tau1)
99     mat Pt1;
100    Pt1 = Ricca(tf,t1,rho,mu);
101    return (U1 + 2*U2*Pt1)*Xtau1(t1);
102 }
103 /*****X(tf)*****/
104 vec Xtf(double t1,double t2, vec rho, vec mu)
105 {
106     return expmat(A*(tf-t2))*Xtau2(t1,t2,rho,mu) + Mfund(tf,t2)*B*Umax;
107 }
108 //calcula la derivada de S respecto a rho(i) en un sistema de n*n
109 mat dSrho(vec rho, vec mu)
110 {
111     double v = as_scalar(rho.t()*mu);
112     mat M = mu*mu.t();
113     mat dSr;
114     dSr.zeros(n,n*n);
115     //ciclo para llenar la matriz de salida
116     for(int k = 0; k < n; k++)
117     {
118         dSr.row(k) = reshape(-(M/(2*v*v))*mu(k),1,n*n);
119     }
120     return dSr;
121 }
122 //calcula la derivada de S respecto a mu(i) en un sistema de n*n
123 mat dSm(vec rho, vec mu)
124 {
125     double v = as_scalar(rho.t()*mu);
126     mat M = mu*mu.t();
127     mat dSmu;
128     dSmu.zeros(n,n*n);
129     mat Z;
130     Z.zeros(n,n);
131     //ciclo para llenar la matriz de salida
132     for(int j = 0; j < n; j++)
133     {
134         for(int k = 0; k < n; k++) {
135             for(int l = 0; l < n; l++){
136                 if(k == j && l == j) Z(k,l) = 2.0*mu(j);
137                 else if(l == j && k != j) Z(k,l) = mu(k);
138                 else if(k == j && l != j) Z(k,l) = mu(l);
139                 else Z(k,l) = 0.0;

```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
140         } //end for l
141     } //end for k
142     dSmu.row(j) = reshape((Z/(2.0*v))-(M/(2.0*v*v))*rho(j),1,n*n);
143 } //end for j
144 return dSmu;
145 }
146 // Funcion para calcular la derivada de riccati respecto a rho y mu
147 mat DeRicca(double t1, double t2, vec rho, vec mu)
148 {
149     mat U = Emat(t1,t2); //Obtiene la exponencial matricial
150     //Matrices involucradas
151     mat U2 = U(span(0,n-1),span(n,2*n-1));
152     mat U4 = U(span(n,2*n-1),span(n,2*n-1));
153     mat alphabeta;
154     alphabeta = albe(U,rho,mu);
155     mat alpha = alphabeta(span(0,n-1),span(0,n-1)); // Obtiene alpha
156     mat beta = alphabeta(span(n,2*n-1),span(0,n-1)); // obtiene beta
157     mat dSr;
158     dSr = dSrho(rho,mu); // Derivadas de S respecto a rho(i)
159     mat dSmu;
160     dSmu = dSm(rho,mu); // Derivadas de S respecto a mu(i)
161     // Inicialización de las derivadas de Riccati respecto a cada componente
162     mat dPr, dPm;
163     dPr.zeros(n,n*n);
164     dPm.zeros(n,n*n);
165     for(int k = 0; k < n; k++)
166     {
167         //Llena los valores de la derivada respecto a rho
168         dPr.row(k)=reshape((1.0/2.0)*2.0*U4*(reshape(dSr.row(k),n,n))\
169 *inv(alpha)-(1.0/2.0)*beta*alpha.i()*2.0*U2*(reshape(dSr.row(k),n,n))\
170 *alpha.i(),1,n*n);
171         //Llena los valores de la derivada respecto a mu
172         dPm.row(k)=reshape((1.0/2.0)*2.0*U4*(reshape(dSmu.row(k),n,n))\
173 *inv(alpha)-(1.0/2.0)*beta*alpha.i()*2.0*U2*(reshape(dSmu.row(k),n,n))\
174 *alpha.i(),1,n*n);
175     }
176     return join_vert(dPr,dPm);
177 }
178 //calcula el valor de la derivada de X respecto a rho y mu en tau2
179 mat dXr_mt2(double t1, double t2, vec rho, vec mu)
180 {
181     mat U = Emat(t1,t2); //Obtiene la exponencial matricial
182     //Matriz involucradas
183     mat U2 = U(span(0,n-1),span(n,2*n-1));
184
185     // Derivada de la funcion de Riccati
186     mat dPr_mt1;
187     dPr_mt1 = DeRicca(tf,t1,rho,mu);
188     // Inicializacion del ciclo
189     mat dXrm;
190     dXrm.zeros(n,2*n);
```

B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

```

191   for(int k = 0; k < 2*n; k++){
192       dXrm.col(k) = 2*U2*reshape(dPr_mt1.row(k),n,n)*Xtau1(t1);
193   }
194   return dXrm;
195 }
196 //calcula el valor de la derivada de X respecto a rho y mu en tf
197 mat dXr_mt(double t1,double t2, vec rho, vec mu)
198 {
199     // Derivada del estado en tau2
200     mat dXr_m_t2;
201     dXr_m_t2 = dXr_mt2(t1,t2,rho,mu);
202     // Inicializacion del ciclo
203     mat dXrm_tf;
204     dXrm_tf.zeros(n,2*n);
205     for(int k = 0; k < 2*n; k++){
206         dXrm_tf.col(k) = expmat(A*(tf-t2))*dXr_m_t2.col(k);
207     }
208     return dXrm_tf;
209 }
210 /*****Calculo de los dJ respecto a las variables *****/
211 //dJ1 = 0
212 // dJ2
213 vec dJ2deta(double t1,double t2, vec rho, vec mu)
214 {
215     // Inicializacion del ciclo
216     vec dJ2 = zeros<vec>(2*n);
217     mat U = Emat(t1,t2); //Obtiene la exponencial matricial
218     //Matrices involucradas
219     mat U1 = U(span(0,n-1),span(0,n-1));
220     mat U2 = U(span(0,n-1),span(n,2*n-1));
221     // derivada de ricatti completa en taul(para todas las variables)
222     mat dPr_m1;
223     dPr_m1 = DeRicca(tf,t1,rho,mu);
224     // derivada de ricatti completa en tau2(para todas las variables)
225     mat dPr_m2;
226     dPr_m2 = DeRicca(tf,t2,rho,mu);
227     for(int k = 0; k < 2*n; k++){
228         dJ2(k) = as_scalar(Xtau1(t1).t()*(reshape(dPr_m1.row(k),n,n)-\
229 4*trans(U1+2*U2*Ricca(tf,t1,rho,mu))*Ricca(tf,t2,rho,mu)*U2*\
230 reshape(dPr_m1.row(k),n,n)-trans(U1+2*U2*Ricca(tf,t1,rho,mu))*\
231 reshape(dPr_m2.row(k),n,n)*(U1+2*U2*Ricca(tf,t1,rho,mu))*Xtau1(t1));
232     }
233     return dJ2;
234 }//End function
235 // dJ3
236 vec dJ3deta(double t1,double t2, vec rho, vec mu)
237 {
238     // Inicializacion del ciclo
239     vec dJ3 = zeros<vec>(2*n);
240     mat U = Emat(t1,t2); //Obtiene la exponencial matricial
241     //Matrices involucradas

```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```

242     mat U2 = U(span(0,n-1),span(n,2*n-1));
243     // derivada de ricatti completa en tau1(para todas las variables)
244     mat dPr_m1;
245     dPr_m1 = DeRicca(tf,t1,rho,mu);
246     //ciclo para todas las variables
247     for(int k = 0; k < 2*n; k++){
248         dJ3(k) = as_scalar(4*(trans(Xtau2(t1,t2,rho,mu))*MR_2(tf,t2)+\
249 Umax*B.t()*MR_1(tf,t2))*U2*reshape(dPr_m1.row(k),n,n)*Xtau1(t1));
250     }
251     return dJ3;
252 } //End Function
253 // dJ4
254 vec dJ4deta(double t1,double t2, vec rho, vec mu)
255 {
256     // Inicializacion del ciclo
257     vec dJ4 = zeros<vec>(2*n);
258     mat U = Emat(t1,t2); //Obtiene la exponencial matricial
259     //Matrices involucradas
260     mat U2 = U(span(0,n-1),span(n,2*n-1));
261     // derivada de ricatti completa en tau1(para todas las variables)
262     mat dPr_m1;
263     dPr_m1 = DeRicca(tf,t1,rho,mu);
264     //ciclo para todas las variables
265     for(int k = 0; k < 2*n; k++){
266         dJ4(k) = as_scalar(4*Xtf(t1,t2,rho,mu).t()*S*expmat(A*(tf-t2))*\
267 U2*reshape(dPr_m1.row(k),n,n)*Xtau1(t1));
268     }
269     return dJ4;
270 } //End Function
271 // La suma de los costos de las componentes
272 vec dJdeta(double t1,double t2, vec rho, vec mu)
273 {
274     // Inicializacion del ciclo, para la derivada de etha(i) total
275     vec dJdetat = zeros<vec>(2*n);
276     //diferentes vectores para los calculos de los dJ
277     vec dJ2 = zeros<vec>(2*n);
278     vec dJ3 = zeros<vec>(2*n);
279     vec dJ4 = zeros<vec>(2*n);
280     dJ2 = dJ2deta(t1,t2,rho,mu);
281     dJ3 = dJ3deta(t1,t2,rho,mu);
282     dJ4 = dJ4deta(t1,t2,rho,mu);
283     dJdetat = dJ2 + dJ3 + dJ4;
284     return dJdetat;
285 }
286 /*****Calculo de los dJ respecto a tau1 *****/
287 // dJ1dtau1
288 double dJ1dtau1(double t1,double t2, vec rho, vec mu)
289 {
290     //derivada del costo respecto a tau1
291     return R*(Umin*Umin) + as_scalar(Xtau1(t1).t()*Q*Xtau1(t1));
292 }

```

B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

```

293 // dJ2dtau1
294 double dJ2dtau1(double t1, double t2, vec rho, vec mu)
295 {
296     // W dentro de la funcion
297     mat W = B*(1/R)*B.t();
298     //derivada del costo respecto a tau1
299     return as_scalar(2*Xtau1(t1).t()*Ricca(tf, t1, rho, mu)*B*Umin) + \
300 as_scalar(Xtau1(t1).t()*Ricca(tf, t1, rho, mu)*W*Ricca(tf, t1, rho, mu)*Xtau1(t1))\
301 -as_scalar(Xtau1(t1).t()*Q*Xtau1(t1))-as_scalar(2*Xtau2(t1, t2, rho, mu).t()*\
302 Ricca(tf, t2, rho, mu)*dXtau2dtau1(t1, t2, rho, mu));
303 }
304 // dJ3dtau1
305 double dJ3dtau1(double t1, double t2, vec rho, vec mu)
306 {
307     //derivada del costo respecto a tau1
308     return as_scalar((Xtau2(t1, t2, rho, mu).t()*MR_2(tf, t2) + \
309 Umax*B.t()*MR_1(tf, t2))*dXtau2dtau1(t1, t2, rho, mu));
310 }
311 // dJ4dtau1
312 double dJ4dtau1(double t1, double t2, vec rho, vec mu)
313 {
314     //derivada del costo respecto a tau1
315     return as_scalar(2*Xtf(t1, t2, rho, mu).t()*S*expmat(A*(tf-t2))\
316 *dXtau2dtau1(t1, t2, rho, mu));
317 }
318 //dJdtau1
319 double dJdtau1(double t1, double t2, vec rho, vec mu)
320 {
321     double dJ1dtault, dJ2dtault, dJ3dtault, dJ4dtault;
322     dJ1dtault = dJ1dtau1(t1, t2, rho, mu);
323     dJ2dtault = dJ2dtau1(t1, t2, rho, mu);
324     dJ3dtault = dJ3dtau1(t1, t2, rho, mu);
325     dJ4dtault = dJ4dtau1(t1, t2, rho, mu);
326     //derivada del costo respecto a tau1
327     return dJ1dtault + dJ2dtault + dJ3dtault + dJ4dtault;
328 }
329 /***** Calculo de los dJ respecto a tau2 *****/
330 // dJ1dtau2
331 double dJ1dtau2(double t1, double t2, vec rho, vec mu)
332 {
333     //derivada del costo respecto a tau2
334     return 0;
335 }
336 // dJ2dtau2
337 double dJ2dtau2(double t1, double t2, vec rho, vec mu)
338 {
339     // W dentro de la funcion
340     mat W = B*(1/R)*B.t();
341     //derivada del costo respecto a tau2
342     return as_scalar(Xtau2(t1, t2, rho, mu).t()*(Ricca(tf, t2, rho, mu)\
343 *W*Ricca(tf, t2, rho, mu) + Q)*Xtau2(t1, t2, rho, mu));

```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
344 }
345 // dJ3dtau2
346 double dJ3dtau2(double t1, double t2, vec rho, vec mu)
347 {
348     //derivada del costo respecto a tau2
349     return -(R*(Umax*Umax) + as_scalar(Xtau2(t1, t2, rho, mu).t()\
350 *Q*Xtau2(t1, t2, rho, mu)));
351 }
352 // dJ4dtau2
353 double dJ4dtau2(double t1, double t2, vec rho, vec mu)
354 {
355     // W dentro de la funcion
356     mat W = B*(1/R)*B.t();
357     //derivada del costo respecto a tau2
358     return as_scalar(-2*Xtf(t1, t2, rho, mu).t()*S*expmat(A*(tf-t2))\
359 *(W*Ricca(tf, t2, rho, mu)*Xtau2(t1, t2, rho, mu)+ B*Umax));
360 }
361 double dJdtau2(double t1, double t2, vec rho, vec mu)
362 {
363     double dJ1dtau2t, dJ2dtau2t, dJ3dtau2t, dJ4dtau2t;
364     dJ1dtau2t = dJ1dtau2(t1, t2, rho, mu);
365     dJ2dtau2t = dJ2dtau2(t1, t2, rho, mu);
366     dJ3dtau2t = dJ3dtau2(t1, t2, rho, mu);
367     dJ4dtau2t = dJ4dtau2(t1, t2, rho, mu);
368     return dJ1dtau2t + dJ2dtau2t + dJ3dtau2t + dJ4dtau2t;
369 }
370 /***** derivada de X(tau2) respecto a tau1*****/
371 vec dXtau2dtau1(double t1, double t2, vec rho, vec mu)
372 {
373     mat U = Emat(t1, t2); //Obtiene la exponencial matricial
374     //Matrices involucradas
375     mat U1 = U(span(0, n-1), span(0, n-1));
376     mat U2 = U(span(0, n-1), span(n, 2*n-1));
377     // W dentro de la función
378     mat W = B*(1/R)*B.t();
379     return U1*(A*Xtau1(t1)+B*Umin)-(A-W*Ricca(tf, t2, rho, mu))*\
380 Xtau2(t1, t2, rho, mu)+2*U2*(Ricca(tf, t1, rho, mu)*\
381 (W*Ricca(tf, t1, rho, mu)*Xtau1(t1)+B*Umin)-(A.t()*Ricca(tf, t1, rho, mu)+Q)*Xtau1(t1));
382 }
```

El header ‘funciones.hpp’ al que se hace referencia contiene los prototipos de las funciones y los demás llamados de ‘.hpp’ utilizados para correr el programa.

B.2.3 Makefile para la compilación de manera local

Los ficheros Makefile son utilizados para escribir instrucciones que son llamadas dependencias, se usan con el comando Make y sirven para dirigir la recompilación automática de los códigos fuentes de un programa determinado. Es empleado en

B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

sistemas operativos tipo Unix/Linux.

Código B.6: Makefile para la compilación de manera local

```
1 PLUS = g++
2 CFLAGS = -I/home/anderson/Documentos/Control/Versiones_boost/boost_1_62_0
3
4 main20par:20_par.cpp
5     $(PLUS) -fopenmp $(CFLAGS) -pg 20_par.cpp funciones.cpp \
6 -o 20_par.bin -O3 -funroll-loops -larmadillo
7
8 ejecompar:
9     OMP_NUM_THREADS=1 OPENBLAS_NUM_THREADS=1 ./20_par.bin
10
11 clean:
12     rm -f *.bin *.cpp~ *.out
```

B.2.4 Porción de código paralelizado con ‘omp parallel sections’ subsección 5.3.1.1

Porción de código modificada del código fuente ‘funciones.cpp’ para realizar la programación sobre las secciones.

Código B.7: omp parallel sections

```
1 // La suma de los costos de las componentes
2 vec dJdeta(double t1, double t2, vec rho, vec mu)
3 {
4     // Inicializacion del ciclo, para la derivada de etha(i) total
5     vec dJdetat = zeros<vec>(2*n);
6
7     //diferentes vectores para los calculos de los dJ
8     vec dJ2 = zeros<vec>(2*n);
9     vec dJ3 = zeros<vec>(2*n);
10    vec dJ4 = zeros<vec>(2*n);
11
12    #pragma omp parallel
13    {
14        #pragma omp sections
15        {
16            #pragma omp section
17            {
18                dJ2 = dJ2deta(t1, t2, rho, mu);
19            } // Fin Section
20            #pragma omp section
21            {
22                dJ3 = dJ3deta(t1, t2, rho, mu);
23            } //Fin section
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
24     #pragma omp section
25     {
26         dJ4 = dJ4deta(t1,t2,rho,mu);
27     } //Fin section
28 } // Fin sections
29 } //Fin parallel
30 dJdetat = dJ2 + dJ3 + dJ4;
31
32 return dJdetat;
33 }
34
35 //dJdtaul
36 double dJdtaul(double t1,double t2, vec rho, vec mu)
37 {
38     double dJ1dtault, dJ2dtault, dJ3dtault, dJ4dtault;
39     #pragma omp parallel
40     {
41         #pragma omp sections
42         {
43             #pragma omp section
44             {
45                 dJ1dtault = dJ1dtaul(t1,t2, rho, mu);
46             } // Fin section 1
47             #pragma omp section
48             {
49                 dJ2dtault = dJ2dtaul(t1,t2, rho, mu);
50             } // Fin section 2
51             #pragma omp section
52             {
53                 dJ3dtault = dJ3dtaul(t1,t2, rho, mu);
54             } // Fin section 3
55             #pragma omp section
56             {
57                 dJ4dtault = dJ4dtaul(t1,t2, rho, mu);
58             } // Fin section 4
59         } // Fin Sections
60     } // Fin parallel
61     //derivada del costo respecto a taul
62     return dJ1dtault + dJ2dtault + dJ3dtault + dJ4dtault;
63 }
64
65 double dJdtau2(double t1,double t2, vec rho, vec mu)
66 {
67     double dJ1dtau2t, dJ2dtau2t, dJ3dtau2t, dJ4dtau2t;
68     #pragma omp parallel
69     {
70         #pragma omp sections
71         {
72             #pragma omp section
73             {
74                 dJ1dtau2t = dJ1dtau2(t1,t2, rho, mu);
```


B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

```
75     } // Fin section 1
76     #pragma omp section
77     {
78         dJ2dtau2t = dJ2dtau2(t1,t2, rho, mu);
79     } // Fin section 2
80     #pragma omp section
81     {
82         dJ3dtau2t = dJ3dtau2(t1,t2, rho, mu);
83     } // Fin section 3
84     #pragma omp section
85     {
86         dJ4dtau2t = dJ4dtau2(t1,t2, rho, mu);
87     } // Fin section 4
88     } // Fin Sections
89 } // Fin parallel
90 //derivada del costo respecto a tau1
91 return dJ1dtau2t + dJ2dtau2t + dJ3dtau2t + dJ4dtau2t;
92 }
```

B.2.5 Porción de código paralelizado con ‘omp parallel for’ subsección 5.3.1.2

Porción de código modificada del código fuente ‘funciones.cpp’ para realizar la programación sobre los bucles.

Código B.8: omp parallel sections

```
1 // Función que calcula la derivada de S respecto a los rho(i)
2 // de un sistema de n*n
3 mat dSrho(vec rho, vec mu)
4 {
5     double v = as_scalar(rho.t()*mu);
6     mat M = mu*mu.t();
7     mat dSr;
8     dSr.zeros(n, n*n);
9
10    //ciclo para llenar la matriz de salida
11    #pragma omp parallel for //para paralelizar
12    for(int k = 0; k < n; k++)
13    {
14        dSr.row(k) = reshape(-(M/(2*v*v))*mu(k), 1, n*n);
15    }
16    return dSr;
17 }
18
19 // Función que calcula la derivada de S respecto a los mu(i)
20 // de un sistema de n*n
21 mat dSm(vec rho, vec mu)
```

B. CÓDIGOS DE CÁLCULO EN PARALELO

```
22 {
23     double v = as_scalar(rho.t()*mu);
24     mat M = mu*mu.t();
25     mat dSmu;
26     dSmu.zeros(n,n*n);
27     mat Z;
28     Z.zeros(n,n);
29     int j,k,l;
30
31 #pragma omp parallel default(none) \
32     shared(rho,mu,n,Z,M,v,dSmu) private(j,k,l)
33 {
34     #pragma omp for
35     //---ciclo para llenar la matriz de salida
36     for(int j = 0; j < n; j++)
37     {
38         for(int k = 0; k < n; k++) {
39             for(int l = 0; l < n; l++){
40                 {
41                     if(k == j && l == j) Z(k,l) = 2.0*mu(j);
42                     else if(l == j && k != j) Z(k,l) = mu(k);
43                     else if(k == j && l != j) Z(k,l) = mu(l);
44                     else Z(k,l) = 0.0;
45                 }// end pragma omp
46             }//end for l
47         }//end for k
48
49         {
50             dSmu.row(j) = reshape((Z/(2.0*v))-(M/(2.0*v*v))*rho(j),1,n*n);
51         }
52     }//end for j //---- end of omp for -----/
53 } // end pragma
54 return dSmu;
55 }
```

B.2.5.1 Makefile para compilar en el cluster 'coyote' y agregar armadillo

Código B.9: makefile para la compilación de manera local

```
1 PLUS = g++
2 ARMA_LIB = -I/usr/local/armadillo/7.800/include
3 ARMA_DIR = -L/usr/local/armadillo/7.800/lib/
4
5 main100par:100_par.cpp
6     $(PLUS) $(ARMA_LIB) $(ARMA_DIR) -fopenmp -pg $< funciones.cpp \
7 -o $@.bin -O3 -funroll-loops -larmadillo
8
9 clean:
10     rm -f *.bin *.cpp~ *.out
```

B.2 Paralelización con OpenMP. Problema de reducción del costo por actualización de los valores de fase finales 3.1

B.2.5.2 ‘script’ para correr el programa en el cluster ‘coyote’

Para correr el programa en el cluster ‘coyote’ se debe enviar a una cola con SLURM, y además crear un script (un runjob) con el siguiente contenido (para correr en un nodo con 8 procesadores).

Código B.10: script para crear el job

```
1  #! /usr/bin/bash
2  #SBATCH --nodes=1 --ntasks-per-node=1 -J AGM --partition=test -t 30
3
4  module load armadillo
5  DIRECC = /u/guest01/ejemplo/Anderson
6  cd $DIRECC
7  $DIRECC/main100par.bin
```

Para usar el SLURM y submitir el job se tipea

```
$ sbatch runjob
```

con lo cual es asignado un JOBID (un identificador del job).

El trabajo de tesis se realizó a partir 2012 a 2017 bajo la supervisión de los Doctores Vicente Costanza y Mario Storti en la Universidad Nacional del Litoral (UNL).

Escrita en \LaTeX bajo la distribución TeXLive con el editor TeXmaker y la plantilla de la Universidad de Deusto, en Santa Fe el 5 de abril de 2017

Esta página se ha dejado en blanco intencionalmente