

UNIVERSIDAD DE ALMERÍA

ESCUELA SUPERIOR DE INGENIERÍA
MÁSTER EN INFORMÁTICA INDUSTRIAL

IMPLEMENTACIÓN DE CONTROL PREDICTIVO
MULTIOBJETIVO

Curso 2011/2012

Alumno/a:

Francisco Manuel Arrabal Campos

Director/es:

Dr. José Domingo Álvarez Hervás

Dr. José Luis Guzmán Sánchez



UNIVERSIDAD DE ALMERÍA
ESCUELA SUPERIOR DE INGENIERÍA
Departamento de Lenguajes y Computación



TRABAJO FIN DE MÁSTER
MÁSTER EN INFORMÁTICA INDUSTRIAL
POSGRADO EN INFORMÁTICA

IMPLEMENTACIÓN DE CONTROL PREDICATIVO
MULTIOBJETIVO

Francisco Manuel Arrabal Campos

Dirigida por: Dr. José Luis Guzmán Sánchez y Dr. José Domingo Álvarez Hervás

Almería, Septiembre 2012

TRABAJO FIN DE MÁSTER
MÁSTER EN INFORMÁTICA INDUSTRIAL
POSGRADO EN INFORMÁTICA



IMPLEMENTACIÓN DE CONTROL PREDICTIVO
MULTIOBJETIVO

por

Francisco Manuel Arrabal Campos

Para la obtención del
Título del Máster en Informática Industrial
Posgrado en Informática

Director

Director

Autor

¡Tú gran astro!
¡Qué sería de tu felicidad si no tuvieras a quienes iluminas!

Agradecimientos

Con este trabajo comienza un nuevo ciclo de mi vida. He de expresar mi profundo agradecimiento a todas aquellas personas que me han dado la oportunidad de desarrollarme, tanto intelectual como personalmente. Su influencia es de una importancia capital para poder llegar al punto en el que me encuentro, finalizando un posgrado y el trabajo culmen del mismo. No puedo nombrar a todos, pero si quiero reconocer específicamente el valor a algunos de ellos. Comenzaré con mi familia, por su apoyo y empuje incondicional.

José Luis Guzmán y Domingo Álvarez, mis directores del trabajo, cuya importancia es vital en el desarrollo del mismo y me ha aportado un punto de vista del mundo ingenieril muy interesante a partir de su experiencia, ellos han sido y serán una inspiración inestimable, un referente tanto a nivel personal como profesional. Mis compañeros de clase, los cuales me han motivado a hacerlo lo mejor posible, con sus ejemplos de trabajo y excelencia en las distintas áreas del posgrado; y hacer que este tiempo de estudio hayan “volado”. Mis amigos de toda la vida, de un valor incalculable, quienes como decía Sócrates: “Amigo es no solo quien perdona un error, sino también quien ayuda a que no vuelva a cometerlo”. Los profesores de la universidad, que han hecho posible alcanzar un nivel de conocimientos técnicos necesarios para la realización del trabajo. A José Luis Torres, compañero de clase, por ayudarme e inspirarme en la realización de estos agradecimientos, los cuales son, a veces, tan difíciles de concretar. A mi mujer Mónica Pascual por ser incondicional. Y a todos aquellos que olvido nombrar.

Implementación de Control Predictivo Multiobjetivo

Índice

1.	Introducción.....	5
1.1	Estrategia del MPC	6
1.2	Historia del MPC industrial	7
1.3	Limitaciones de la tecnología existente	10
1.4	El futuro de la tecnología del MPC	10
2.	Optimización multiobjetivo	11
2.1	Algunos métodos para resolver problemas de optimización multiobjetivo	12
2.2	Herramientas de optimización disponibles	13
3.	Optimización convexa	15
3.1	Propiedades y definiciones básicas convexas	15
3.2	YALMIP herramienta de optimización convexa	16
4.	Implementación de MPC multiobjetivo	17
4.1	Formulación del problema	17
4.2	Esquema propuesto MPC multiobjetivo	18
4.3	Algoritmo multiobjetivo MPC	19
4.4	Problema implementado	19
4.5	Comparación con los controladores H_∞ y H_2	20
4.6	Variaciones del problema planteado	21
5.	Conclusiones	23
6.	Trabajos futuros	24
7.	Agradecimientos	24
8.	Referencias	25
	Anexo	27

Implementación de Control Predictivo Multiobjetivo

Francisco Manuel Arrabal Campos

Máster en Informática Industrial

Postgrado en Informática

Escuela Superior de Ingeniería, Universidad de Almería

Abstract: *Model predictive control (MPC) has been a control strategy widely used and investigated by industry and research community. The main reason of this control strategy came by his great capacity of work with all kinds of processes and primarily by handling process constraints implicitly. In the great majority of cases, the optimization problem formulated in MPC is given for a single objective problem. In this work, the solution of formulate a multiobjective MPC control is analyzed. The fundamental idea of this project is to replicate the recent results in this area and analyze implementations issues about on this topic. This study will allow open the research way to apply this kind of optimization problem to various real problems like optimum control of energy process or bioprocess.*

Resumen: *El control predictivo basado en modelo (MPC) ha sido una estrategia de control ampliamente utilizada y estudiada por las comunidades del sector industrial y del sector académico. La principal potencia de esta estrategia de control ha venido por su gran capacidad de trabajar con todo tipo de procesos y fundamentalmente por el manejo implícito de restricciones de proceso. En la gran mayoría de casos, el problema de optimización formulado en MPC viene dado por un problema de un único objetivo. En este trabajo, se analiza la solución de formular un control MPC multiobjetivo. La idea fundamental de este proyecto consiste en reproducir los resultados recientes de investigación en este ámbito y analizar los aspectos de implementación relacionados. El estudio de este tipo de problemas permitirá abrir una vía de investigación para poder aplicar este tipo de problema de optimización a diversos problemas reales, tales como el control óptimo de procesos energéticos o bioprocesos.*

Keywords: Multiobjective MPC, optimization MPC, MPC, control multiobjective MPC, Pareto set..

Palabras Clave: Multiobjetivo MPC, optimización MPC, control MPC multiobjetivo, conjunto de Pareto.

1. INTRODUCCIÓN

En los últimos 30 años se ha asistido a un crecimiento continuo en el uso de los ordenadores para el control avanzado de plantas industriales. La rápida mejora del hardware ha posibilitado el desarrollo de esta técnica de control. El control predictivo basado en modelo (MPC – Model Predictive Control) es actualmente la tecnología avanzada más común para el control de procesos que está implantada en la mayoría de las instalaciones industriales [29]. El MPC no especifica una estrategia de control concreta sino que más bien es un amplio abanico de métodos de control que usan de forma explícita el modelo del proceso para obtener la señal de control minimizando una función de coste. Los distintos métodos de control difieren en el modelo usado que representa el proceso, los ruidos y función de coste a minimizar.

Las ventajas del MPC sobre otros métodos son [3] [4]:

- Es efectivo cuando los operadores tienen un conocimiento limitado en control y su ajuste al proceso es sencillo con respecto a otros métodos.
- Amplio espectro de aplicaciones desde procesos sencillos con una sola entrada hasta procesos con dinámicas complejas como tiempos muertos, fase no mínima, procesos inestables o procesos multivariados.

- Compensan las perturbaciones medibles.
- Es fácilmente implementable, y aunque posee un alto coste computacional, existen métodos para calcular su controlador explícitamente de forma que la computación sea mínima.
- El tratamiento de las restricciones es conceptualmente muy simple y pueden ser incluidas en cualquier momento para el cálculo de la ley de control.

Esta técnica de control también tiene sus inconvenientes. El más importante de todos ellos es que aunque su implementación no es compleja es un poco más compleja que el clásico controlador Proporcional Integral Derivativo (PID). Si la dinámica es simple y no hay muchas restricciones el controlador se puede calcular de antemano pudiéndose aplicar a dinámicas rápidas. En caso contrario su implementación resulta imposible o prácticamente inviable por la cantidad de cálculos que hay que realizar para obtener la acción de control. La mayor dificultad radica en obtener el modelo del proceso dado que para obtenerlo se necesita de un especialista en control, aun así ha demostrado ser un controlador razonable y se ha aplicado de forma exitosa en numerosas aplicaciones.

1.1 Estrategia del MPC

Los controladores control predictivo basado en modelo se componen de forma general como se observa en la figura 1, de una salida $y(t)$ conocida, una acción de control $u(t)$ conocida, una referencia a seguir $s(t|k)$ determinada, un horizonte de predicción H_p , las acciones de control futuras $u(t+k|t)$ y la predicción de salida $\hat{y}(t+k|t)$ ¹. Las salidas futuras determinadas en el horizonte de predicción H_p son predichas a cada instante t usando el modelo del proceso. Los valores predichos $\hat{y}(t+k|t)$ para $k=1 \dots H_p$ dependen del valor actual en el instante t y las señales de control $u(t+k|t)$ para $k=0 \dots H_p-1$ que son calculadas y enviadas al sistema. El conjunto de señales de control futuras son calculadas optimizando una regla determinada la cual intenta mantener la salida del proceso lo más cerca de la referencia. Existen un conjunto de reglas que estas suelen ser de la forma $1/Norma$, $2/Norma$ o $\infty/Norma$ [3] [5] aunque pudiera ser cualquier regla siempre y cuando cumpla una serie de condiciones para asegurar que el problema tuviere

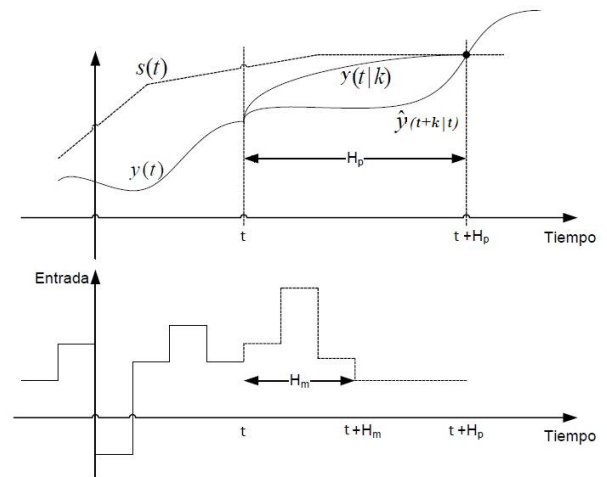


Figura 1. Estrategia del MPC

El modelo del proceso suele ser lineal aunque se puede generalizar para modelos no lineales utilizando funciones afines a trozos donde se puede aproximar el modelo no lineal a un modelo lineal a regiones. Toda la estrategia puede estar supeditada a restricciones como por ejemplo tener una acción de control máxima o mínima tanto así como tener una salida máxima o mínima (figura2), acciones de control supeditadas a condiciones de estabilidad, regiones de exploración terminal o criterios de convergencia para un horizonte de predicción que tiene al infinito.

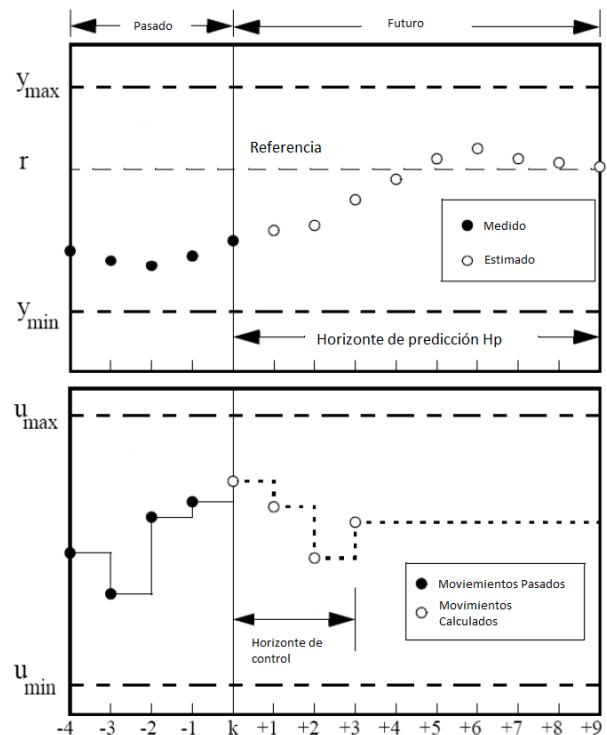


Figura 2. Restricciones en la salida del proceso y en la señal de control

¹ La notación indica el valor predicho de la variable en el instante $t+k$ calculada en el instante t .

La señal de control calculada en el instante actual para el instante siguiente $u(t + 1|t)$ es utilizada la señal de control para el momento actual y las otras señales de control son desechadas ya que en el siguiente instante de muestreo $y(t + 1)$ es conocida y por tanto se repite el proceso con la nueva información de la planta. $u(t + 1|t + 1)$ es calculada con información diferente al actual y por tanto su predicción también será diferente a la anterior. En cada instante t se resuelve un problema de control óptimo sobre un horizonte finito futuro H_p donde se resuelve,

$$\begin{aligned} \min f(|y - r|, |u|) \text{ sujeto a: } & (1) \\ y_{min} \leq y \leq y_{max} \\ u_{min} \leq u \leq u_{max} \\ \Delta u_{min} \leq \Delta u \leq \Delta u_{max} \\ \text{condiciones de estabilidad} \end{aligned}$$

Aplicándose sólo el primer movimiento de control óptimo, $u^*(t)$, se repite con las nuevas medidas de la planta y se optimiza el problema mínimo planteado.

La estructura básica de implementación del MPC se observa en la figura 3. Se usa un modelo para predecir la salida futura a partir de las entradas y salidas pasadas y actuales y de las futuras salidas predichas. El optimizador calcula las acciones de control futuras dada un función de coste y sus restricciones. El modelo elegido determina la eficacia del control, por lo que éste debe ser lo suficientemente adecuado como para poder predecir la evolución del proceso de una forma razonable y al mismo tiempo lo suficientemente simple como para poder implementarlo. Las distintas metodologías difieren en la forma que se implementa el modelo del proceso.

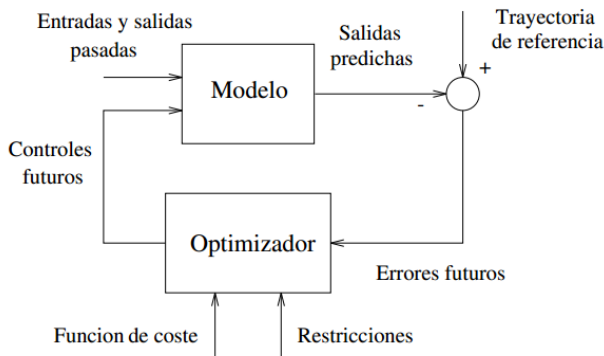


Figura 3. Estructura del MPC

En la industria uno de los métodos más usados para obtener el modelo es el denominado modelo de respuesta del impulso truncado (truncated impulse

response model) ya que es muy simple de obtener dado que sólo hay que medir la salida para un impulso dado. Es ampliamente aceptado en la industria ya que es simple y puede ser usado para procesos multivariables.

El modelo dado como función de transferencia es usado por la comunidad científica pero el cálculo del controlador se hace más tedioso requiriendo más parámetros. Como ventaja añadir el tiempo muerto es más fácil siendo un punto no despreciable ya que en la industria se da esta circunstancia frecuentemente.

El modelo de espacio de estados es el más utilizado en la comunidad académica e investigadora dado que el cálculo del controlador es fácil incluso para el caso multivariable. La descripción de los procesos por el modelo de espacio-estado permite conceptualizar criterios de estabilidad y aplicar control robusto. En el presente trabajo se representará y desarrollará la formulación para este caso exclusivamente.

El optimizador es otro elemento fundamental en la estrategia del MPC, de hecho proporciona la acción de control futura. Las funciones objetivo y las restricciones identificarán el método de resolución.

1.2 Historia del MPC Industrial

En este apartado se describirá una breve historia de la tecnología del MPC. En la figura 4 se muestra la mayoría de los algoritmos del MPC donde se ilustra su genealogía.

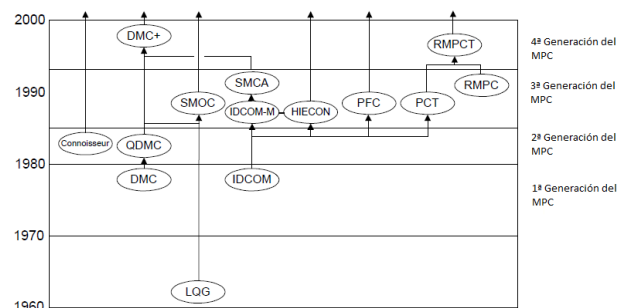


Figura 4. Genealogía del MPC

El desarrollo de los conceptos modernos de control fue introducido por el trabajo de Kalman [6] [7] [8] al principio de los sesenta. Kalman y sus colaboradores describieron los procesos como un modelo lineal en espacio de estados,

$$x_{k+1} = Ax_k + Bu_k + Gw_k \quad (2)$$

$$y_k = Cx_k + \xi_k \quad (3)$$

El vector u representa las variables manipulables, el vector y representa la salida medida del proceso. El valor de x representa los estados a ser controlado. Se incluye w como perturbación y ζ como ruido aleatorio que sigue una distribución Gaussiana independiente de media cero.

La función objetivo penaliza las desviaciones con respecto al cuadrado del estado deseado y el valor al cuadrado de las variables manipulables. La desviación con respecto a la salida deseada y las acciones de control se ponderan por dos matrices \mathbf{Q} y \mathbf{R} .

$$J = \sum_1^{\infty} (\|x_{k+i}\|_Q^2 + \|u_{k+j}\|_R^2) \quad (4)$$

Implícitamente se asume que todas las variables están escritas en función del estado deseado. La solución a este problema fue llamado *linear quadratic gaussian (LQG)*.

El horizonte infinito de predicción del algoritmo LQG dotó propiedades de estabilidad. Para el caso de que el modelo de un proceso fuera perfecto se comportaba de forma estable siempre que $Q \succcurlyeq 0$ y $R \succcurlyeq 0$, es decir, que fueran matrices semidefinidas positivas.

Desde el punto de vista práctico se logró el control de la salida del proceso, controlar el error en régimen permanente y el cálculo del estado. Sin embargo, las restricciones en la entrada, salidas y otras no fueron introducidas en la teoría del controlador, lo cual producía efectos inesperados [6].

En poco tiempo la teoría LQG fue aceptada como estándar para resolver problemas de control. En [9] se estiman que hay miles de aplicaciones del LQG con aproximadamente 400 patentes al año basados en el filtro de Kalman. Pero ha tenido poco impacto en las aplicaciones industriales debido a,

- No incluir las restricciones.
- No tratar procesos no lineales.
- No incluir un modelo de incertidumbre.
- Razones culturales.

Pero la mayor razón para no poder aplicar es algoritmo estaba relacionada con la cultura en la comunidad de los ingenieros de instrumentación y control, lo cuales establecieron que los conceptos del LQG eran impracticables. En este entorno industrial se generaliza un modelo general adaptado a los problemas de optimización dinámica donde se calculan dentro de

un horizonte de predicción no infinito. Se define de forma explícita las dinámicas de los procesos y se incluyen las restricciones. Ahora se calcula en cada instante de tiempo la acción de control óptima, la cual se aplica al proceso para medir el comportamiento y volver a calcular el problema de optimización. En este contexto nace una nueva metodología para el modelado y control industrial que fue llamada tecnología MPC [6]. La primera aplicación industrial de control MPC fue presentada en [10]. Su propuesta fue llamada control predictivo heurístico basado en modelo (MPHC – Model Predictive Heuristic Control), su solución comercial fue llamada IDCOP (acrónimo de identificación y dirigir). Sus características principales fueron modelo de respuesta al impulso de la planta, función cuadrática a optimizar con inclusión de restricciones y usó de métodos heurísticos para la solución al problema de optimización. La representación elegida como modelo del proceso fue $y_{k+j} = \sum_1^N h_i u_{k+j-i}$.

A principios de los 70 se acercaba una crisis de petróleo. Los ingenieros de una de las mayores compañías petrolíferas del mundo Shell Oil desarrollan su propio controlador MPC independiente. Cutler y Ramaker presentaron los detalles del algoritmo de control multivariable sin restricciones el cual llamaron control matricial dinámico (DMC - dynamix matrix control). Sus características principales fueron modelo lineal de respuesta escalón, función cuadrática en un horizonte finito de predicción y la solución fue calculada como un problema de mínimos cuadrados. La expresión elegida como modelo del proceso para representar la respuesta en escalón fue:

$$y_{k+j} = \sum_1^{N-1} s_i \Delta u_{k+j-i} + s_N u_{k+j-N}$$

Los iniciales IDCOP y DMC presentan la primera generación del MPC y tuvieron una enorme repercusión en la industria donde establecieron un nuevo paradigma en la ingeniería de control. Al poco tiempo, otra vez los ingenieros de Shell Oil mejoraron su controlador DMC añadiendo la capacidad de configurar restricciones del controlador. Se denominó control matricial dinámico cuadrático (Quadratic Dynamic Matrix Control – QDMC), aunque este controlador es uno de los más avanzados, un problema cuadrático QP es uno de los problemas más estudiados, el único requisito es que el hessiano del problema debe ser semidefinido positivo para que el problema sea convexo y por lo tanto tenga solución.

El control MPC fue ganando adeptos hasta que se produce una explosión de soluciones tecnológicas a las deficiencias y a los problemas planteados. Los ingenieros de las principales compañías (Adersa, Setpoint Inc., Shell (Francia)) desarrollaron nuevas versiones del algoritmo del MPC (tablas 1 y 2).

Setpoint Inc. (compañía de control) desarrolló IDCOM-M, IDCOM-S y SMCA. Adersa desarrolla HEICON. En los finales de los años 80 Shell Research en Francia desarrolla Shell Multivariable Optimizing Controls (SMOC) donde comienza a usar el modelo espacio de estados en el algoritmo MPC.

También aparecieron otros como tecnología de control predictivo (Predictive Control Technology – PCT) (Profimatics) y el algoritmo RMPC de Honeywell. A finales del año 1995 Honeywell compró Profimatics combinando sus algoritmos del MPC dando como resultado RMPCT, al año siguiente Aspen Technology Inc. compra Setpoint y DMC corporation dando como resultado el algoritmo DMC-plus. Ambos algoritmos son los más representativos de la cuarta generación de algoritmos de MPC, las cuales son soluciones industriales que se encuentran en el mercado.

Compañía	Nombre de Producto	Descripción
Adersa	HIECON	Hierarchical constraint control
	PFC	Predictive functional control
	GLIDE	Identification package
Aspen Tech	DMC-plus	Dynamic matrix control package
	DMC-plus model	Identification package
Honeywell	RMPCT	Robust model predictive control technology
Hi-Spec		
Shell Global Solutions	SMOC-II ^a	Shell multivariable optimizing control
Invensys	Connoisseur	Control and identification package

Tabla 1. Compañías y productos con tecnología MPC lineal. [29]

Area	Aspen Technology	Honeywell Hi-Spec	Adersa	Invensys	SGS	Total
Refining	1200	480	280	25		1985
Petrochemicals	450	80	—	20		550
Chemicals	100	20	3	21		144
Pulp and paper	18	50	—	—		68
Air & Gas	—	10	—	—		10
Utility	—	10	—	4		14
Mining/Metallurgy	8	6	7	16		37
Food Processing	—	—	41	10		51
Polymer	17	—	—	—		17
Furnaces	—	—	42	3		45
Aerospace/Defense	—	—	13	—		13
Automotive	—	—	7	—		7
Unclassified	40	40	1045	26	450	1601
Total	1833	696	1438	125	450	4542
First App.	DMC:1985 IDCOM-M:1987 OPC:1987	PCT:1984 RMPCT:1991	IDCOM:1973 HIECON:1986	1984	1985	
Largest App.	603 × 283	225 × 85	—	31 × 12	—	

Tabla 3. Resumen de aplicaciones del MPC lineal [6].

Compañía	Nombre de Producto	Descripción
Adersa	PFC	Predictive functional control
Aspen Tech	Aspen Target	Nonlinear MPC package
Continental Controls, Inc.	MVC	Multivariable control
DOT Products	NOVA-NLC	NOVA nonlinear controller
Pavilion Technologies	Process Perfecter	Nonlinear control

Tabla 2. Compañías y productos con tecnología MPC no lineal [29].

En las tablas 3 y 4 se resumen todas las aplicaciones industriales aproximadamente hasta la actualidad. Las aplicaciones descritas son definidas por los vendedores por lo que hay que tener cuidado en sacar conclusiones de estos datos ya que la tecnología del MPC es una tecnología general donde el mismo controlador es puede servir para distintas aplicaciones.

Más de 4600 aplicaciones [6] están resumidas en la tablas 3 y 4, esto es debido a que la tecnología del MPC aplicada está creciendo continuamente.

La tecnología del MPC se encuentra en una amplia variedad de áreas, aunque aproximadamente el 67% de los controladores están orientados a refinerías. Esto es debido a que esta tecnología comenzó a aplicarse en estas áreas.

Otro gran grupo donde se encuentra habitualmente este controlador son las empresas químicas. Un grupo donde está creciendo esta tecnología son las empresas de papel, procesamiento de alimentos, aeroespacial e industrias del automóvil.

Area	Adersa	Aspen Technology	Continental Controls	DOT Products	Pavilion Technologies	Total
Air and Gas			18			18
Chemicals	2		15		5	22
Food Processing					9	9
Polymers		1		5	15	21
Pulp & Paper					1	1
Refining					13	13
Utilities		5	2			7
Unclassified	1		1			2
Total	3	6	36	5	43	93

Tabla 4. Resumen de aplicaciones del MPC no lineal [6].

1.3. Limitaciones de la tecnología existente

La mayoría de los MPC industriales disponibles tienen limitaciones debido a la herencia de la tecnología original DMC e IDCOM. Los problemas son los siguientes,

- Elección limitada de modelos.
- Retroalimentación óptima.
- Falta de estabilidad nominal.
- Ineficiente solución cuando se pretende una optimización dinámica.

La elección limitada de modelos es un problema ya que éstos están limitados estrictamente a procesos estables. No es posible representar un proceso no estable mediante estos modelos. Todos estos problemas pueden ser solucionados usando un modelo paramétrico en la forma modelo de espacio de estados o el modelo ARX.

El problema de la retroalimentación óptima se produce porque el controlador asume que no hay presencia de perturbaciones no caracterizadas, es decir, que no hay perturbaciones estocásticas que afectan al estado del proceso. Muske y Rawlings [6] demostraron que la mejor forma de afrontar este problema fue el desarrollo del modelo de espacio de estados, por ello no es casual que el algoritmo SMOC utilice un filtro Kalman como retroalimentación.

Ajustar los controladores para un modo de operación estable sin violar las restricciones puede ser muy difícil incluso suponiendo que el modelo es perfecto. Actualmente se está desarrollando un algoritmo que incluya la estabilidad para todos los casos, aunque a nivel industrial todavía no están incorporados todas estas innovaciones (Chen y Allogöwer [6] presentaron método del horizonte casi infinito, quasi-infinite horizon method, para sistemas no lineales que abordaban estos problemas).

El modelo de incertidumbre aplicado a la tecnología MPC no está desarrollado actualmente como se debería. La mayoría de paquetes de identificación proviene de estimaciones del modelo de incertidumbre. Sólo Honeywell suministra una pequeña solución en su configuración del diseño del control. La tecnología hasta el momento desarrollada para la identificación de modelos tienen las siguientes deficiencias,

- Las pruebas que se hacen a los procesos para saber su dinámica suele ser mediante respuesta escalón y necesitan la experiencia de los ingenieros.
- No hay una herramienta que gestione los datos de forma adecuada que represente el modelo del proceso para el diseño de controladores MPC, generalmente las plantas están sobre testeadas.
- Los algoritmos de identificación son limitados.
- No existe un criterio para seleccionar el tipo de algoritmo a utilizar.
- No existe el MPC adaptativo excepto en raras ocasiones.
- No hay un enfoque sistemático para obtener modelos no lineales de los procesos.
- No hay un criterio en la validación de modelos de los procesos.

1.4 El futuro de la tecnología del MPC

El futuro de la tecnología es prometedor. Las necesidades industriales y los éxitos en los controladores actuales hacen que la investigación de esta tecnología no pare de crecer en distintas áreas. Existen pocos algoritmos MPC adaptativos a pesar de que la industria actual incentiva la auto-sintonización. Excepto pequeñas investigaciones teóricas no existen verdaderas teorías adaptativas para la tecnología MPC.

Otra área es el MPC robusto, donde excepto las investigaciones de Honeywell, los experimentos se basan en simulaciones en el sentido literal “a lo bruto” de plantas desajustadas “para ver que pasa”. Que el MPC sea robusto garantiza estabilidad y gestiona incertidumbres dentro del proceso de la planta con lo que, reduce el tiempo de sintonización de un algoritmo MPC.

La tecnología del MPC que permita incluir modelos no lineales será una combinación entre conocimiento del modelo y una gestión de datos operativos. El modelo de la planta será un espacio afín definido por regiones donde su comportamiento será definido por su espacio de estados en operación. Los ensayos en lazo cerrado se realizan mediante señales binarias de secuencia pseudo-aleatoria (Pseudo Random Bit Secuencias - PRBS) con unos requerimientos mínimos de tiempo para realizar el ensayo. La base de datos del modelo no lineal que gestionen dichos datos será una de las claves para el desarrollo de esta tecnología. Por último siendo, el objeto del presente trabajo será una reformulación del problema que exprese todos los elementos importantes para la optimización de las funciones de coste, la próxima generación de MPC será aquella que sea capaz de optimizar múltiples funciones objetivo donde se convertirá esta cualidad en un estándar dentro de la tecnología del MPC.

En los apartados anteriores se ha desarrollado una revisión de la tecnología del MPC desde dos puntos de vista: ‘qué es’ y ‘por qué’. De aquí en adelante se desarrollará las tecnologías teóricas del control predictivo multiobjetivo basado en modelo haciendo una revisión de los métodos de resolución del problema a optimizar y la soluciones que hay en el mercado, la reformulación del problema monoobjetivo a multiobjetivo y la reproducción de un algoritmo multiobjetivo desarrollado por Alberto Bemporad y David Muñoz de la Peña [1].

2. OPTIMIZACIÓN MULTI OBJETIVO

En la ingeniería de control la optimización multiobjetivo ha sido utilizada durante mucho tiempo. Sin embargo sólo en los últimos diez años se ha desarrollado intensivamente. No sólo ha sido desarrollado para los modelos avanzados de control sino que también para las arquitecturas clásicas de control como es el PID o PI [11]. En [11] se ha tratado tanto el problema multiobjetivo del MPC como el problema multiobjetivo del PID/PI.

A pesar de que no existe mucha literatura la optimización multiobjetivo ha sido utilizada para multitud de casos y aplicaciones de todo tipo. Un ejemplo como se describe en [12], donde se desarrollan modelos financieros para la optimización multiobjetivo utilizando programación paramétrica.

Otro ejemplo es descrito en [13] donde se aplica una función de coste multiobjetivo y es tratada mediante redes neuronales para la resolución del mismo. En 2005 se propuso un control paramétrico de la concentración de glucosa en sangre para combatir la diabetes de tipo 1, todo esto se hizo mediante minimización de funciones multiobjetivo [14].

Por otro lado, en el trabajo [2] se presenta el control multiobjetivo para el crecimiento del cultivo bajo invernadero, mostrando resultados satisfactorios.

En el diseño de controladores basados en minimización de funciones de coste está muy desarrollada en la ingeniería de control. El control convencional establecía una sólo función de coste normalmente definida como,

$$\min J = f(x(t), u(t)) \text{ sujeto a: (5)}$$

$$g(x(t), u(t)) = 0 \text{ (6)}$$

$$h(x(t), u(t)) \leq 0 \text{ (7)}$$

donde (5) es la función de coste, (6) y (7) son las restricciones. La función $J(x, u)$ es minimizada con respecto a $u(t)$ que son las acciones de control. Las acciones de control óptimas $u^*(t)$ es lo que se denomina *ley de control óptimo*. Algunas veces las soluciones de este tipo de problemas de optimización pueden ser resueltas analíticamente mediante una solución elegante, pero otras soluciones y sobre todo aquellas que presentan restricciones se resuelven mediante métodos numéricos.

La optimización multiobjetivo (también llamado multicoste, multicriterio o vector de optimización) fue introducida por V. Pareto en [18] y puede ser definida como el problema de encontrar las variables de decisión o parámetros, los cuales satisfacen las restricciones y optimiza el vector, donde cada uno de los elementos representan las funciones objetivo. La formulación de un problema típico es,

$$\min J(x, u) = [J_1, J_2, \dots, J_n] \text{ sujeto a: (8)}$$

$$g(x(t), u(t)) = 0 \text{ (6)}$$

$$h(x(t), u(t)) \leq 0 \text{ (7)}$$

donde $J: R^s \times R^m \rightarrow R^l$ es el vector de funciones objetivo siendo una aplicación bilineal de con dos espacios de dimensión s y m cada uno para dar un espacio de dimensión l , $u \in U \subset R^p$, n es el número de funciones objetivo, p es el número de variables de decisión. Optimizar significa minimizar o maximizar según sea la aplicación. Si todas la J_i son convexas, las restricciones que son inecuaciones son convexas y las restricciones que son una igualdad son afines se dice que el problema de optimización es convexo [16].

Definición 1: un punto factible U^P es óptimo de Pareto si y sólo si no existe otro punto U factible tal que $J_i(x, U) \leq J_i(x, U^P)$, $\forall i = 0, \dots, l$ y $J_i(x, u) < J_i(x, U^P)$ para al menos uno de los índices $j \in \{0, \dots, l\}$.

Algunas veces es útil tener una herramienta que defina subpuntos óptimos que son más fáciles de investigar por los algoritmos que son suficientemente buenos para las aplicaciones prácticas, esto es lo que se llama el punto de parte débil.

Definición 2: un punto $U^P \in U \subset R^p$ es óptimo de Pareto débil con respecto a U si no existe otro punto, $u \in U$, tal que $J(x, u) \leq J(x, U^P)$.

El conjunto de parteo óptimo es un subespacio contenido en U , $\mathcal{U} \subset U$, el cual es a su vez la imagen de la aplicación lineal $J: \mathcal{U} \rightarrow \mathcal{S}$ sobre el espacio de funciones de coste \mathcal{S} , este es llamado frente de Pareto.

Definición 3: dado un vector de funciones coste y el conjunto de Pareto, el frente de Pareto se define como,

$$\mathcal{S}f \triangleq \{J(x, u_i) \in \mathcal{S} \subseteq R^n | u_i \in \mathcal{U}, \forall i\}$$

Definición 4: un punto, J^* es un punto utópico si $J^* = \min J_i(x, u) \forall u$. Figura 5.

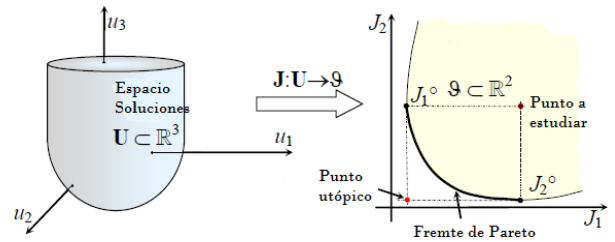


Figura 5. Problema multiobjetivo, frente de Pareto.

2.1 Algunos métodos para resolver problemas de optimización multiobjetivo

Los métodos para resolver problemas de optimización multiobjetivo pueden clasificarse por multitud de criterios. Existen dos grandes grupos:

- Métodos por escalarización: estos tipos de métodos normalmente usan coeficientes. Son muy eficientes pero por otro lado no siempre encuentran la solución aceptable debido a conflicto de intereses entre las funciones objetivo. Suelen buscar óptimos locales y no globales.
- Métodos de Pareto: estos métodos buscan soluciones directamente en el espacio de soluciones. Estos métodos utilizan los conceptos de dominancia y no dominancia. La desventaja es que necesitan mucha capacidad de computación y la convergencia al resultado suele ser muy lenta.
- Métodos que no son ni de Pareto ni de escalarización.

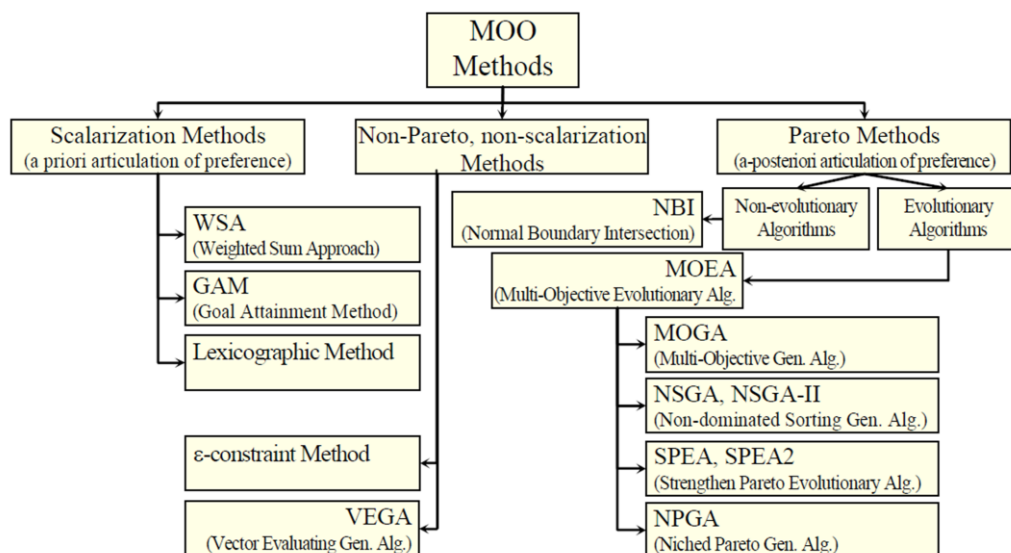


Figura 6. Clasificación general de los métodos de resolución para problemas multiobjetivo.

ene distintos

algoritmos para la resolución del problema planteado. Para el caso de métodos de escalarización se tiene algoritmos como suma de peso (Weighted Sum Approach - WSA), logro del objetivo (Goal Attainment Method - GAM) y métodos lexicográficos. En la figura 6 se muestra una ilustración donde se resumen algunos métodos.

Los métodos de Pareto se clasifican a su vez en algoritmos no evolutivos y algoritmos evolutivos. En los algoritmos evolutivos se establece la supervivencia dentro de una evolución biológica del punto más apto. La población se renueva mediante operadores genéticos, recombinación, mutación, y selección. Los más importantes son MOGA (Multiple Objective Genetic Algorithm), NSGA-II (Non-dominated sorting genetic algorithm), SPEA2 (Strength Pareto evolutionary algorithm) y NPGA-II (Niched Pareto genetic algorithm). La mayor ventaja de estos algoritmos es que pueden resolver problemas no convexos.

abarcado no es un criterio exacto, porque muchos *resolvedores numéricos* pueden tratar más de un tipo de problemas. Por ejemplo, las herramientas de resolución cuadrática resuelven también problemas lineales, ya que estos se pueden ver como un caso particular de los anteriores. La mayoría de algoritmos que se aplican para resolver problemas de optimización se pueden clasificar en algoritmos convexos y algoritmos evolutivos, y en general son estos los que se emplean en las aplicaciones comerciales. La utilización de los métodos de punto interior es común en la gran mayoría de las aplicaciones convexas, se resumen en descomponer el problema original en una secuencia de problemas cuadráticos que a su vez se descomponen en una serie de sistemas lineales de ecuaciones.

Las herramientas de problemas lineales se resumen en la tabla 5 aunque el entorno más accesible es utilizar MATLAB[®] con el comando *linprog* del complemento Matlab Optimization Toolbox.

Nombre	Algoritmos	Problemas	Licencia	Limitaciones en demo
<i>lp_solve</i>	símplex ramificar y acotar	LP MILP	LGPL ^t	
<i>GLPK</i>	símplex primal símplex dual p. interior primal-dual ramificar y cortar	LP MILP	GPL	
<i>BPMPD</i>	p. interior primal-dual	LP QP	libre	
<i>MINOS</i>	símplex gradiente reducido lagrangiano proyectado	LP no lineales	comercial	3 meses no gratuita

Tabla 5. Herramientas lineales convexas.

2.2 Herramientas de optimización disponibles

En este apartado se hace una revisión de las herramientas más importantes de optimización con el fin de implementar la mejor herramienta. Si se desea profundizar en este apartado se pueden consultar las webs en [19] y [20]. Se describen las características de cada programa de forma que se pueda conocer cuál es más adecuado para el problema que necesitemos resolver, en función de diferentes propiedades como su complejidad o facilidad de uso, condiciones comerciales de distribución, y la variedad de problemas y formatos de datos que admita.

La cantidad de herramientas disponibles en el mercado para resolución de problemas de optimización es muy grande. La selección por el tipo de problemas

Para resolver problemas cuadráticos-cónicos de grandes dimensiones existen las herramientas, tablas 6 y 7, *CPLEX*, *Mosek*, *SeDuMi*, *SDPT3*, *Gurobi*, *Xpress-Optimizer* y *OOQP*. También se pueden resolver problemas cuadráticos utilizando programas orientados a problemas cónicos.

Estas herramientas sirven también para resolver problemas lineales, ya que los LPs son un subconjunto de los problemas cuadráticos. Por otro lado, el complemento *Matlab Optimization Toolbox* nos permite utilizar el comando *quadprog* para resolver problemas cuadráticos con restricciones lineales.

Nombre	Algoritmos	Problemas	Licencia	Limitaciones en demo
<i>CPLEX</i>	símplex primal símplex dual p.interior barrera ramificar y cortar	LP QP redes QCQP MILP MIQP ¹⁵	comercial	90 días limitada en tamaño
<i>Gurobi</i>	símplex primal símplex dual barrera paralelo plano cortante heurísticas	LP QP MILP MIQP	comercial	500 variables y 500 restricciones. Demo académica: sin limitaciones
<i>Xpress-Optimizer</i>	símplex primal símplex dual p.interior barrera ramificar y acotar	LP QP MILP MIQP	comercial	30 días Demo académica: 400 filas, 800 col. 5000 coef. de mat. y 400 var. enteras
<i>OOQP</i>	p. interior primal-dual	QP	libre	

Tabla 6. Herramientas cuadráticas convexas.

Nombre	Algoritmos	Problemas	Licencia	Limitaciones en demo
<i>Mosek</i>	símplex primal símplex dual p. interior id. base ramificar, acotar y cortar	LP QP convexos MILP MIQP	comercial	30 días Demo académica: 60 días sin limitaciones
<i>SeDuMi</i>	p. interior	LP QP SDP cónicos	GPL	
<i>SDPT3</i>	camino no viable	LP QP SDP	GPL	
<i>PENSDP</i>	p. interior	SDP	comercial	1 mes no renovable Demo académica: 3 meses renovable

Tabla 7. Herramientas cónicas convexas.

Los sistemas de modelado proporcionan un lenguaje comprensible que permite al usuario expresar su problema centrandolo su esfuerzo en definirlo correctamente, dejando en manos del programa la conversión a otro formato compatible con las herramientas de resolución que forman parte del núcleo del sistema de modelado.

Todas estas transformaciones del modelo descrito por el usuario producen otro problema o conjunto de problemas en un formato estándar equivalentes, y su mecanismo interno consiste, básicamente en transformaciones algebraicas de variables, eliminar o añadir restricciones, utilizar variables de holgura, y otros recursos. La definición del modelo en *CVX* es inmediata a partir del problema matemático, mientras que para utilizar directamente el *solver SeDuMi*, necesitamos modificar la formulación.

Después de todas estas transformaciones, el sistema de modelado se encarga de elegir el *resolvedor numérico* más adecuado para el tipo de problema tratado. A partir de ahí, interpreta el resultado, indicando si es o no viable, y devuelve la solución óptima si es posible, adaptándola de nuevo al formato original.

Existen cuatro herramientas sobre las que se encuentran mayor número de referencias sobre optimización convexa: *CVX*, *AMPL*, *GAMS* y *YALIMP(MPT)*. Si bien este criterio de selección no es demasiado riguroso, ya que quedan sin tratar herramientas de gran calado en el mercado como *AIMMS*, *TOMLAB*, *LINGO*, etc. En la tabla 8 se muestra un esquema de las herramientas descritas de modelado.

MATLAB[®] proporciona herramientas de optimización multiobjetivo por medio de su toolbox “*optimization toolbox*™”.

Este toolbox incluye el método de los pesos (weighted sum method), restricción del épsilon (ϵ -constraint method) y logro del objetivo (goal attainment method). Otro entorno de desarrollo de propósito general es TOMLAB[®], el cual soporta a través FORTAN[®] rutinas mediante interfaz MEX[®]. GEA[®] Toolbox desarrollada para MATLAB[®] proporciona un entorno para algoritmos evolutivos y logro del objetivo (goal attainment).

3. OPTIMIZACIÓN CONVEXA

Cuando se plantea un problema de optimización lo primero que hay que estudiar si el problema planteado tiene solución, de ahí que el estudio de las condiciones necesarias y suficientes sean investigadas de forma exhaustiva. La solución se puede enfocar desde varios puntos de vista desde condiciones estrictas hasta condiciones débiles.

Los métodos de resolución de problemas de optimización suelen ser resueltos por métodos numéricos, existen unas condiciones que determinan que el problema tiene solución. Si el problema de optimización es convexo o tiene condiciones convexas se puede asegurar a priori que el problema tiene solución. En este apartado se muestran las principales de estas condiciones ya que el esquema del modelo propuesto en este trabajo resuelve el problema multiobjetivo con condiciones convexas de ahí que se pueda desarrollar un controlador explícito para esta condiciones.

3.1 Propiedades y definiciones básicas convexas

A continuación se van a describir las propiedades y definiciones [16] resumidas de la optimización convexa:

- Líneas y segmentos: dados dos puntos $x_1 \neq x_2 \in R^n$. La línea de estos dos puntos se define como $y = \Theta x_1 + (1 - \Theta)x_2$.

Nombre	Solvers	Licencia	Limitaciones en demo
CVX	SeDuMi, SDPT3	GPL	
AMPL	BPMD, CPLEX, LAMPS, LOQO, lp_solve, MINOS, Mosek, OSL, SOPT, XA, Xpress-MP, MINTO, CONOPT, DONLP2, FILTER, FSQP, IPOPT, KNITRO, LANCELOT, NPSOL, PENNON, SNOPT, MINLP ²³ , Gurobi	comercial	300 Variables y 300 restricciones y objetivos
GAMS	ALPHAECF, BARON, BDMLP, COIN-OR, CPLEX, DECIS, DICOPT, Gurobi, KNITRO, LGO ²⁴ , LINDOGLOBAL, LOGMIP, MILES, MINOS, Mosek, MSNLP, NLPEC, OQNLP, PATH, SBB, SCIP, SNOPT, XA, Xpress	comercial	300 restricciones y variables, con 2000 elementos distintos de 0, 50 variables discretas. En caso de utilizar un solver global está limitado a 10 restricciones y variables
YALIMP	CDD, GLPK, lp_solve, QSOPT, BINTPROG, CPLEX, Gurobi, linprog, Mosek, BPMPD, CLP, OOQP, QPC, NAG, quadprog, Xpress, CSDP, SDPT3, SeDuMi, DSDP, SDPA, SDPLR, LMILAB, PENBMI, PENSDF, fmincon, GPPSY, IPOPT, KYPD, LMIRANK, MPT, PENNON, SNOPT, VSDP,	libre	

Tabla 8. Herramientas de modelado.

- Conjunto Afín: un conjunto $\mathcal{C} \subseteq R^n$ es afín si una línea cualquiera que une dos puntos que pertenecen a \mathcal{C} siguen perteneciendo al conjunto.
- Conjunto convexo: un conjunto $\mathcal{C} \subseteq R^n$ es convexo si el segmento que une dos puntos de \mathcal{C} sigue perteneciendo al conjunto.
- Hiperplano y semiespacios: un hiperplano es un conjunto de la forma $\{x|a^T = b\}$ donde $a \in R^n, a \neq 0, b \in R$ y semiespacios son los huecos que deja el hiperplano.
- Poliedro: es un conjunto finito de hiperplanos y semiespacios.

Definición 5: una aplicación lineal $f: R^n \rightarrow R$ es convexa si el dominio de f es un conjunto convexo y si para todos los x e y pertenecen al dominio de f $\forall \theta \in [0,1]$ se tiene $f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$.

Condición 1: se supone una aplicación lineal que es diferenciable. Entonces f es convexo si y sólo si el dominio f es convexo y además $f(y) \geq f(x) + \nabla f(x)^T(y - x)$.

Condición 2: se supone una aplicación lineal doblemente diferenciable, esto es, que existe su Hessiano $\nabla^2 f$ para cada punto del dominio de f . Entonces f es convexa sí y solo sí el Hessiano de f es semidefinida positiva.

Definición 6: los problemas de optimización convexa se define por que tienen las funciones convexas, las inecuaciones son convexas y las igualdades son afines.

Los problemas de optimización convexa aseguran que haya un conjunto factible solución del problema que es a su vez convexo. La teoría de dualidad de Lagrange proporciona una herramienta para determinar el óptimo dentro del conjunto convexo. En el presente trabajo no se desarrolla esta teoría pero si se hará mención a las condiciones Karush-Kuhn-Tucker, las cuales aseguran el óptimo. Estas condiciones son importantes ya que todos los algoritmos de optimización convexa se apoyan en estas condiciones.

Condición 3: se consideran los problemas primal y dual. Dado el conjunto óptimo de los respectivos puntos. La holgura implica que z^* minimiza $L(z, \alpha^*, \beta^*)$. Siendo f, g_i, h_i diferenciables entonces el gradiente de $L(z, \alpha^*, \beta^*)$ en z^* debe ser 0,

$$\begin{aligned} \nabla f(z^*) + \sum \alpha_i^* \nabla g_i(z^*) + \sum \beta_j^* \nabla h_j(z^*) &= 0 \quad (9) \\ \alpha_i^* g_i(z^*) &= 0 \quad (9.a) \\ u_i^* &\geq 0 \quad (9.b) \\ g_i(z^*) &\leq 0 \quad (9.c) \\ h_j(z^*) &= 0 \quad (9.d) \end{aligned}$$

Estas condiciones son las llamadas condiciones Karush-Kuhn-Tucker o condiciones KKT. Estas condiciones son necesarias para encontrar el óptimo. Algunos algoritmos de optimización convexa están desarrollados sobre esta condiciones [16].

3.2 YALMIP herramienta de optimización convexa

YALMIP se utiliza como un complemento (toolbox) gratuito para MATLAB®, y sirve para modelar problemas de optimización convexas (aunque también se utiliza para problemas no convexas).

El lenguaje es consistente con la sintaxis de Matlab, por lo que es muy fácil de aprender para usuarios familiarizados con este entorno. Implementa una gran cantidad de recursos de modelado, permitiendo que el usuario se concentre en el modelo a alto nivel, mientras que YALMIP se ocupa del modelado a bajo nivel para obtener modelos eficientes y numéricamente satisfactorios.

Soporta varios tipos de problemas lineales, cuadráticos, cónicos de orden 2, semidefinidos, geométricos, y otros más como por ejemplo los problemas cónicos con mezcla de variables enteras.

También se puede usar YALMIP para calcular soluciones explícitas mediante el interfaz “Multi-Parametric Toolbox” (MPT), este está orientado a controladores con restricciones lineales, no lineales y sistemas híbridos.

3.2.1 Instalación de YALMIP

La instalación de esta herramienta es muy sencilla. Se descarga el paquete desde <http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Tutorials.Installation>. Una vez descargado se descomprime en el directorio de trabajo. Una vez allí se añade al directorio de MATLAB®. Seleccionar “File” y en el desplegable seleccionar “Set path”. En la figura 7 se muestra lo descrito.

Una vez seleccionado añadir directorio o “set path” aparecerá una ventana como se muestra en la figura 8.

Se selecciona añadir con todos los subdirectorios y seleccionar la carpeta ../yalmip/ donde este ubicada. Una vez seleccionado la carpeta donde está YALMIP se pulsa guardar. Para comprobar que todo ha ido correctamente en la ventana de comandos del MATLAB[®] se ejecuta “yalmiptest”.

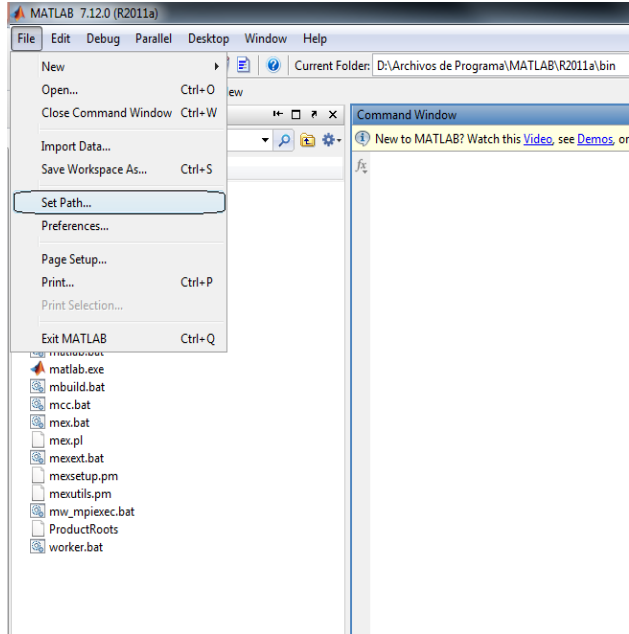


Figura 7. Establecer directorio.

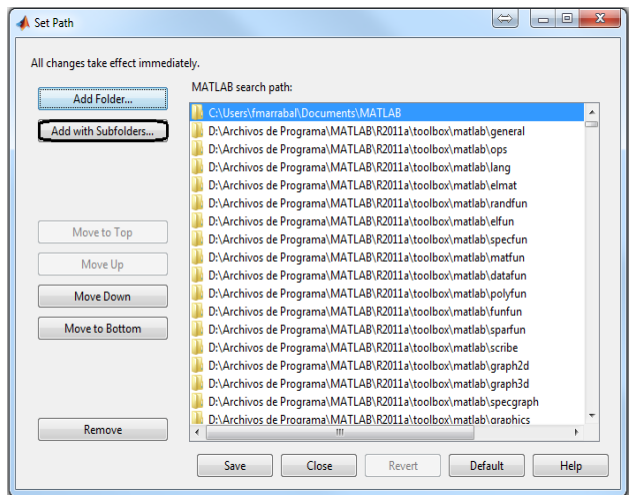


Figura 8. Ventana emergente para añadir directorios.

Si lo que se desea es utilizar la herramienta MPT, lo primera actividad a realizar será obtener en paquete que se puede descargar desde la siguiente web <http://control.ee.ethz.ch/~mpt/downloads/>. Una vez descargado se descomprime en la carpeta que será el área de trabajo. Se siguen los mismos pasos que para YALMIP aunque para poder utilizarlo hay que inicializar el paquete ejecutando en la ventana de comandos del MATLAB[®] “mpt_init”.

4. IMPLEMENTACIÓN DEL MULTI-OBJECTIVE MPC

Los problemas de control multiobjetivo han sido estudiados desde hace tiempo. Normalmente las funciones de coste entran en conflicto entre ellas donde los controladores H_2/H_∞ son los más utilizados. En esta implementación se considera un algoritmo que proporciona una de las soluciones del conjunto de Pareto óptimo. Existen muchas soluciones de Pareto óptimas que solucionan el problema para los distintos pesos y garantizan la estabilidad.

La implementación de este esquema de optimización multiobjetivo se basa en programación lineal multiparamétrico multiobjetivo (mp-moLP) y programación cuadrático multiparamétrico multiobjetivo (mp-moQP) donde se utilizan funciones de coste convexas afines.

En el trabajo realizado en [1] se hace un estudio sobre el MPC explícito orientado a obtener un controlador multiobjetivo. También propone un esquema de resolución de optimización multiobjetivo mediante un algoritmo propuesto en el mismo trabajo. En el presente trabajo se limitará a la reproducción del algoritmo de optimización multiobjetivo. En [1] se aprovechan de las condiciones KKT en la ecuación (9) para la escalarización del problema multiobjetivo. La importancia del trabajo realizado en [1] se debe que a que establece un algoritmo que selecciona un conjunto de Pareto óptimo y a su vez garantiza la estabilidad en lazo cerrado.

4.1 Formulación del problema

Se considera el sistema discreto e invariante en el tiempo, dado como una representación de espacio de estados:

$$x(t + 1) = f(x(t), u(t)) \quad (10)$$

donde $x(t)$ es el vector de estado, $x(t+1)$ es el vector de estado en el siguiente tiempo de muestreo y $u(t)$ es la variable de control de entrada. Además, $f(\cdot, \cdot)$ es la función de estado actualizado que depende del tipo de modelo, el cual describe el sistema invariante en el tiempo (LTI) el cual es

$$f(x(t), u(t)) = Ax(t) + Bu(t) \quad (11)$$

Por otro lado, se asume que el sistema de variables está sujeto a las siguientes restricciones,

$$x(t) \in \mathcal{X}, u(t) \in \mathcal{U}, \forall t \geq 0 \quad (12)$$

donde \mathcal{X} y \mathcal{U} son conjuntos de poliedros convexos.

Se considera el siguiente problema de control óptimo multiobjetivo,

$$\begin{aligned} \min_u J(U, x) \quad (13) \quad \text{sujeto a:} \\ x(t+1) = Ax(t) + Bu(t), \quad x_0 = x \quad (14) \\ x(t) \in \Omega \quad (15) \end{aligned}$$

$J(U, x) = [J_0(U, x), J_1(U, x), \dots, J_l(U, x)]: R^s \times R^n \rightarrow R^{l+1}$ donde $l \geq 1, s = Nm, U = [u'_0, \dots, u'_{N-1}]'$ es la secuencia de movimientos futuros, x_k es k pasos en el futuro de predicción del estado inicial $x = x(t)$ y Ω es el poliedro de la región terminal. Cada función de coste se define como sigue,

$$J_i(U, x) = \sum_{k=0}^{N-1} L_i(x(t), u(t)) + F_i(x(N)) \quad (16),$$

Donde L es una aplicación lineal $L_i: R^{n+m} \rightarrow R$ y el coste terminal es otra aplicación lineal $F_i: R^n \rightarrow R$ donde se asumen que son convexas, donde L_i y F_i pueden ser cualquier norma o función y pueden estar ponderadas o no. Este criterio es descrito en [1].

La solución del problema (13) es planteado como una variación del método de los pesos ya que se desea una escalarización de la función multicriterio para obtener el argumento mínimo de U ,

$$U^*(x, \alpha) = \arg \min_U \alpha' J(U, x) \quad (17)$$

Sujeto a: (11. a)

$$\alpha = [\alpha_0, \dots, \alpha_l]', \quad \alpha_i \geq 0, \quad \sum_0^l \alpha_i = 1 \quad (18)$$

4.2 Esquema propuesto MPC multiobjetivo

La elección de las funciones de coste se realiza conforme a una serie de criterios como pueden ser la estabilidad, robustez, velocidad de convergencia, etc. Los pesos de las funciones de coste determinan todas estas características. El esquema propuesto realiza un ajuste dinámico de estos pesos en función de un peso objetivo α_d para cada instante de tiempo siendo ajustado.

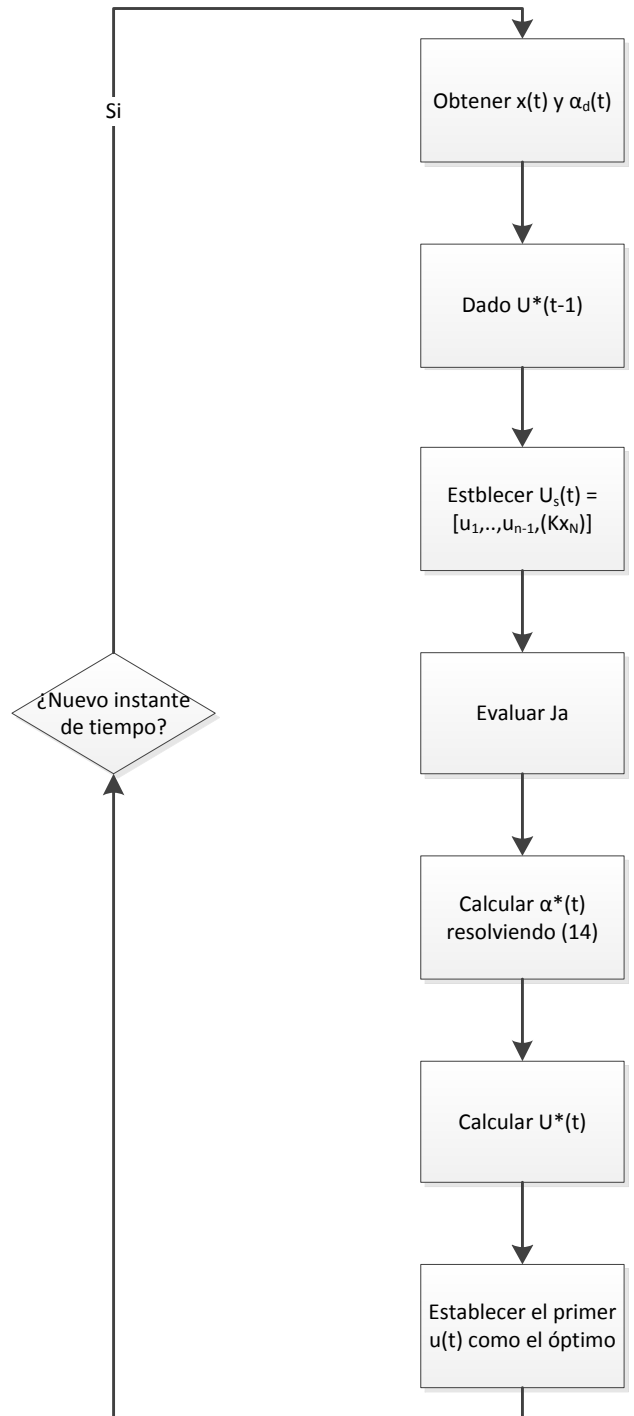


Figura 9. Algoritmo propuesto

Este peso objetivo da prioridad a las distintas funciones de coste, donde el esquema intenta acercar lo más posible el peso de la función objetivo al peso objetivo garantizando la estabilidad del controlador,

$$\alpha^* = \arg \min f(\alpha - \alpha_d) \quad (19)$$

$$s. a V^*(x, \alpha) \leq J_a \quad (20)$$

$$\sum \alpha_i = 1, \alpha_i \geq 0 \quad (21)$$

$$\text{donde } V^*(x, \alpha) = \alpha' J(U^*(x, \alpha), x) \quad (22),$$

$$J(U, x) = [J_1, J_2, \dots, J_j] \text{ y } \alpha = [\alpha_1, \alpha_2, \dots, \alpha_j] \quad (23)$$

4.3 Algoritmo multiobjetivo MPC

Se ejecuta para cada instante de tiempo. El algoritmo se utilizada para obtener,

$$\alpha^*(t) = \alpha^*(x(t), \alpha_d(t), J_a(t)) \text{ y } U^*(t)U^*(x(t), \alpha^*(t)) \quad (24)$$

Primero se obtienen los valores de $x(t)$ y α_d que son directamente medidos. Dado que se dispone de todos los valores óptimos del instante anterior de $U^*(t-1)$ se puede establecer un $U_s(t) = [u'_1, \dots, u'_{N-1}, (kx_N)]$ y se puede evaluar $J_a(t) = \alpha^*(t-1)'J(U_s(t), x(t))$. Una vez que se tiene J_a , se puede calcular $\alpha^*(t)$ resolviendo (19). En este momento se puede calcular $U^*(t)$ resolviendo (17) y se establece el $u(t)$ igual al primer óptimo del vector $U^*(t)$. En la figura 9 se muestra lo descrito y en la siguiente tabla 9.

Algoritmo MPC multiobjetivo

1. Obtener $x(t), \alpha_d$;
2. Dado $U^*(t-1) = [u'_0, \dots, u'_{N-1}]'$ ser la secuencia óptima predicha en el tiempo $t-1$ que comienza en el estado $x(t-1)$, y dado x_N ser el óptimo correspondiente para el estado $t-1+N$;
3. Se establece $U_s(t) = [u'_0, \dots, u'_{N-1}, (Kx_N)]'$;
4. Se evalúa $J_a(t) = \alpha^*(t-1)'J(U_s(t), x(t))$;
5. Calcular $\alpha^*(t)$ resolviendo (19) para $x = x(t), \alpha_d = \alpha_d(t), J_a = J_a(t)$;
6. Calcular $U^*(t)$ resolviendo (17) para $x = x(t), \alpha_d = \alpha_d(t)$;
7. Establecer $u(t)$ igual al primer óptimo de la secuencia $U^*(t)$;
8. Fin.

Tabla 9. Resumen del algoritmo MPC.

Este algoritmo garantiza la estabilidad en lazo cerrado del sistema, la comprobación está desarrollada por los autores en [1].

4.4 Problema implementado

Se considera el sistema lineal (11) con las siguientes características:

- Dos funciones de coste J_0 y J_1 , una de ellas cuadrática y la otra lineal.
- $|x(t)| \leq 10, |u(t)| \leq 10$.
- En la región terminal $u = kx$ con $k = [-0.5, -1.4]$, garantizando la estabilidad.

- Se considera la región terminal

$$\Omega = x \in R^2 \left\{ \begin{array}{l} \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right] x \leq 10 \\ \left[\begin{array}{cc} 0 & -1 \\ -0.5 & -1.4 \end{array} \right] x \leq 10 \end{array} \right.$$

A continuación se muestran los valores de las matrices A, B, Q, R, P,

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}; B = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}; Q_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$R_0 = 0.2; Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}; R_1 = 0.1$$

El controlador MPC generado se denomina h_{mo}^{QP} y está basado en un vector de funciones objetivo $J = [J_0, J_1]^T$. El objetivo del vector de pesos es,

$$\alpha_d(t) = \left[\frac{1}{1+0.02\|x(t)\|_2^2}, \frac{0.02\|x(t)\|_2^2}{1+0.02\|x(t)\|_2^2} \right].$$

Esto es, si se cerca del valor de referencia se da más importancia a una de las funciones objetivo. En caso contrario se da más importancia a la otra función objetivo.

Las funciones objetivo se definen como sigue,

$$L_0(x, u) = \|Q_0 x\|_\infty + \|R_0 u\|_\infty, F_0(x) = \|P_0 x\|_\infty \quad (25)$$

$$L_1(x, u) = x' Q_1 x + u'^{R_1} u, F_1(x) = x' P_1 x \quad (26)$$

La función que minimiza el valor de α no está definida en el trabajo de [1], y por ello se define la función $f(\alpha - \alpha_d) = \|\alpha - \alpha_d\|_\infty$ (27) para la resolución del problema planteado. Los resultados obtenidos en la simulación son iguales a los mostrados en [1], tal y como se describen a continuación:

- Los valores de x_1 y x_2 , así como de la entrada se muestran en la siguiente figura 10:

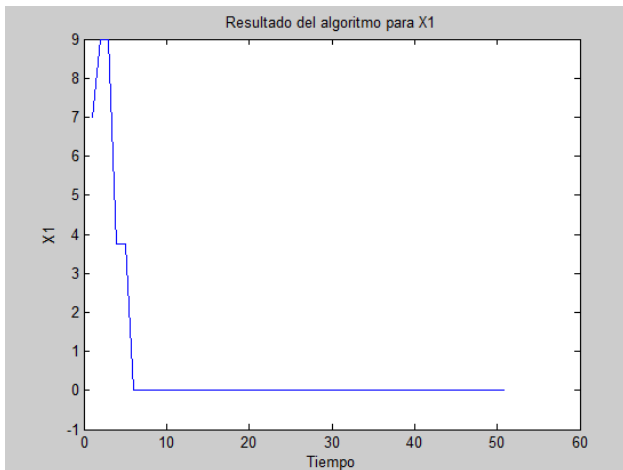


Figura 10.a. Resultado del algoritmo para x_1

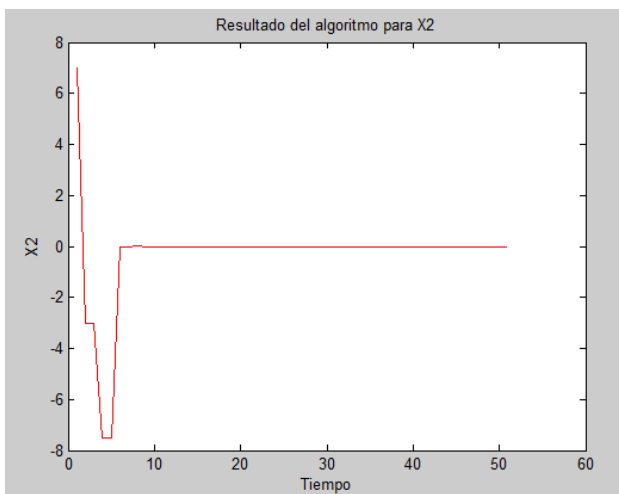


Figura 10.b. Resultado del algoritmo para x_2

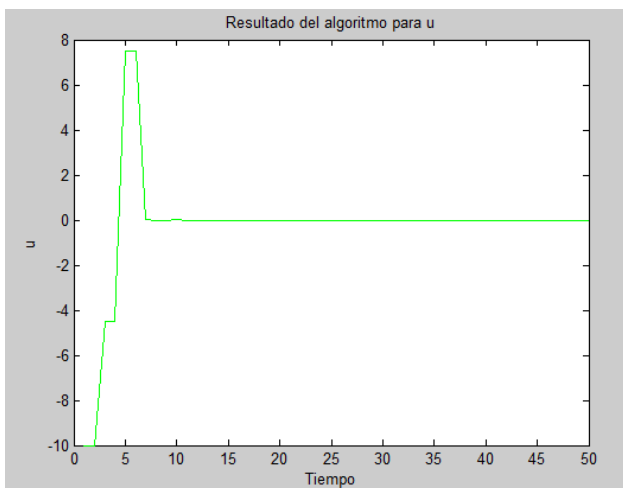


Figura 10.c. Resultado del algoritmo para u

A la luz de los resultados se observan que son muy parecidos a los presentados en [1]. Las diferencias son motivadas, como se ha comentado anteriormente, a que se desconoce que $f(\alpha - \alpha_d)$ se ha utilizado, también se desconoce que valor de α inicial se ha utilizado para realizar la simulación.

4.5 Comparación con los controladores H_∞ y H_2

En este apartado se hace una comparativa entre el controlador multiobjetivo y los controladores H_∞ y H_2 . Los controladores H_∞ y H_2 han sido calculados con el toolbox *mpt*. Una vez seguido los pasos anteriores de instalación se inicializa en el cuadro de comando de MATLAB – *mpt_init* – y se modela el problema mediante dos comandos *sysStruct* y *probStruct*,

```

sysStruct.A = [1 1;0 1];
sysStruct.B = [0.5;1];
sysStruct.C = [1 0;0 1];
sysStruct.D = [0;0];
sysStruct.umin = -10;
sysStruct.umax = 10;
sysStruct.ymin = [-10;-10];
sysStruct.ymax = [10;10];
sysStruct.xmin = [-10;-10];
sysStruct.xmax = [10;10];
sysStruct.Uset{1} = [-Inf Inf];
probStruct.norm = Inf;
probStruct.subopt_lev = 0;
probStruct.N = 5;
probStruct.Q = [0.1 0;0 1];
probStruct.R = 0.2;
probStruct.tracking = 0;

```

En el *probStruct.norm* se puede definir la norma añadiendo 1,2 o Inf. Luego en la ventana de comandos se ejecuta,

```

ctrl=mpt_control(sysStruct,probStruct)
[X,U,Y] = sim(ctrl,[7;7],50)

```

Con estos dos comandos se calcula el controlador de forma explícita y se simula. De esta forma se pueden obtener los datos de la simulación y poder trabajar con ellos.

La convergencia al punto de referencia se realiza más rápidamente para el controlador multiobjetivo. En la figura 11 se observa qué sucede tanto para x_1 y x_2 . El comportamiento de la acción de control u es más brusco con respecto a los otros controladores pero está dentro de los límites de operación., tal y como se puede observar en la figura 11.c. Este comportamiento es esperado ya que el controlador multiobjetivo converge antes que los otros controladores por el que conlleva una acción de control más agresiva.

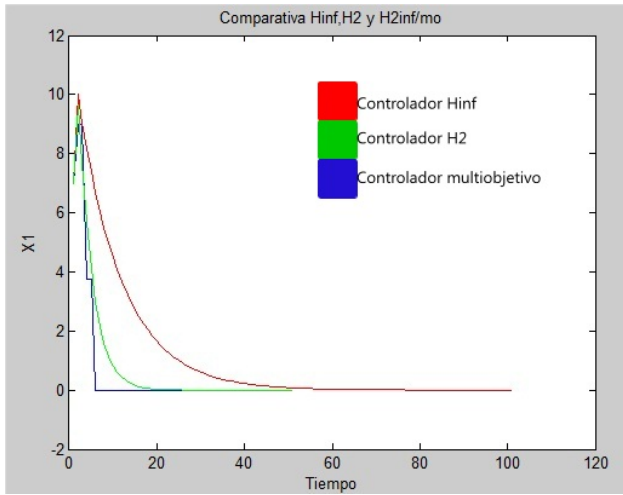


Figura 11.a. Comparativa de x_1 para H_1^{mLP} , H_2^{mQP} y H_{mo}^{mQP}

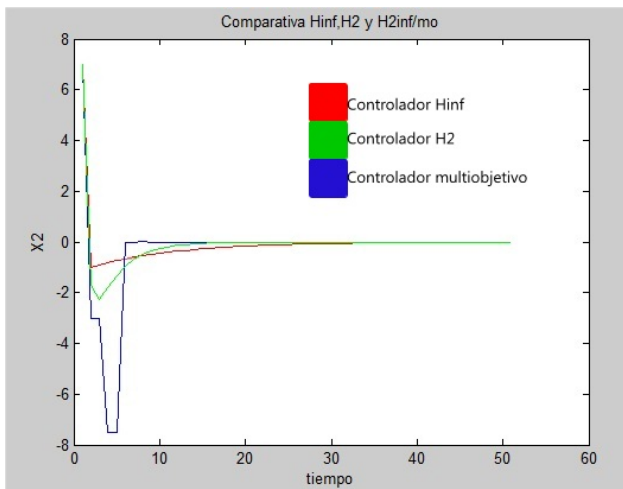


Figura 11.b. Comparativa de x_2 para H_1^{mLP} , H_2^{mQP} y H_{mo}^{mQP}

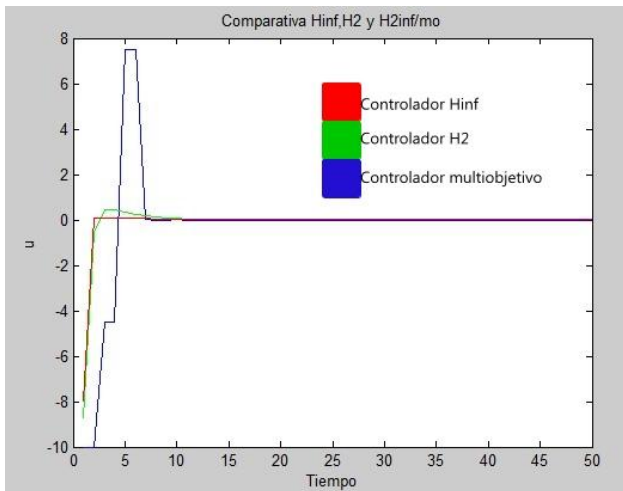


Figura 11.c. Comparativa de u para H_1^{mLP} , H_2^{mQP} y H_{mo}^{mQP}

4.6 Variaciones del problema planteado

En este apartado se hará una variación del problema inicial. Primero se cambiará $f(\alpha - \alpha_d)$ haciéndola un igual a la norma 2. Posteriormente se hará el algoritmo para $L_0(x, u) = \|Q_0x\|_1 + \|R_0u\|_1, F_0(x) = \|P_0x\|_1$,

para terminar al problema original se añade al modelo una incertidumbre normalmente aleatoria de media cero.

- $f(\alpha - \alpha_d) = \|\alpha - \alpha_d\|_2$, se obtienen resultados parecidos excepto para x_2 que presenta una pequeña oscilación en el momento de converger. Figura 12.

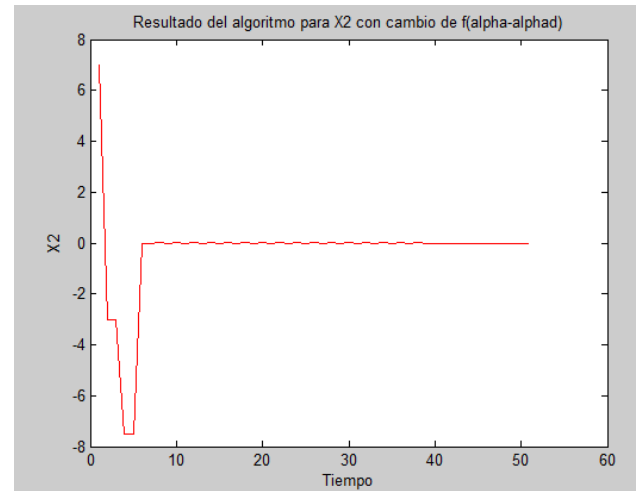


Figura 12. Valor de x_2 cambiando $f(\alpha - \alpha_d) = \|\alpha - \alpha_d\|_2$

- $L_0(x, u) = \|Q_0x\|_1 + \|R_0u\|_1, F_0(x) = \|P_0x\|_1$, se obtiene resultados sensiblemente peores para las variables de estado y sensiblemente mejor para la acción de control. La figura 13 muestra el resultado para x_1 y x_2 . También se muestra el resultado de la acción de control u .

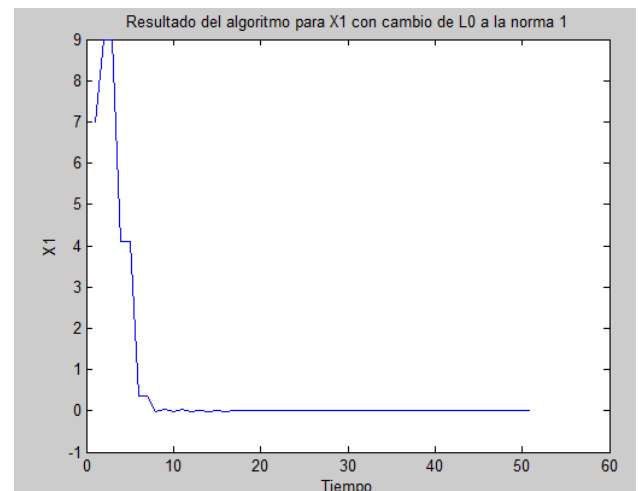


Figura 13.a. Resultado del valor de x_1

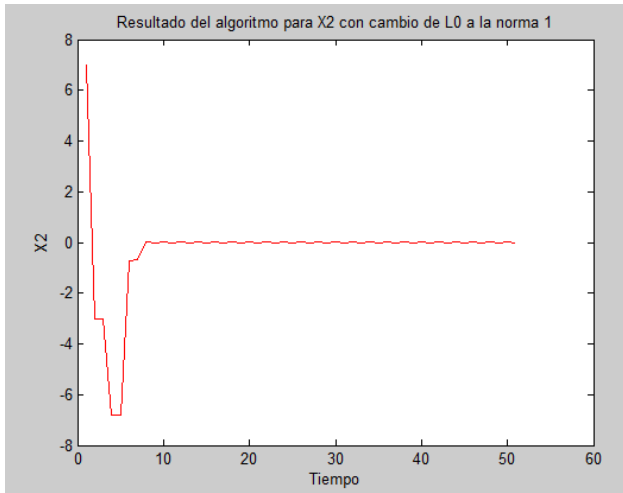


Figura 13.b. Resultado del valor de x_2

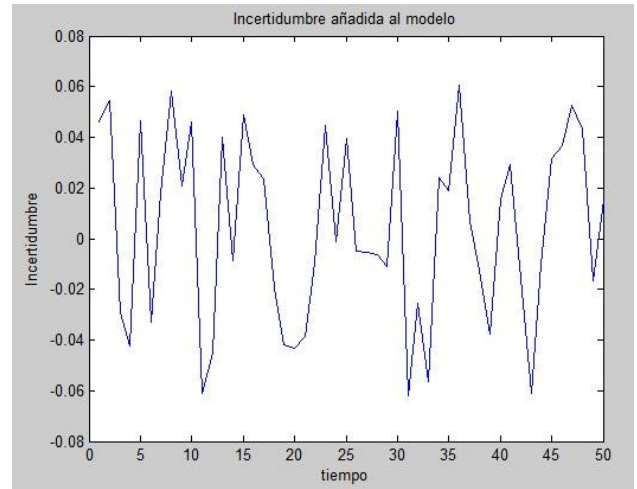


Figura 14. Incertidumbre añadida al modelo

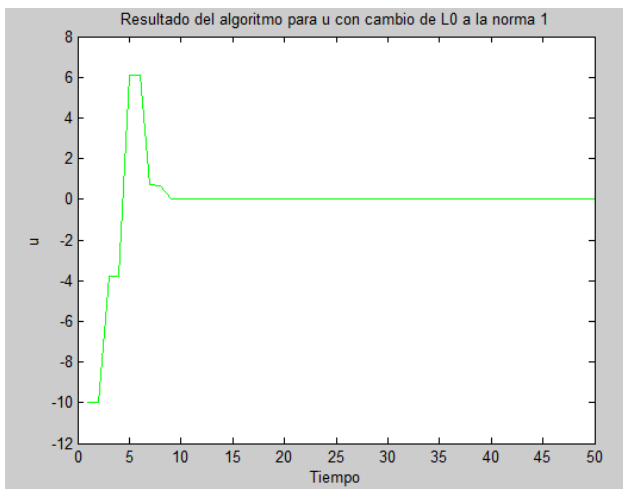


Figura 13.c. Resultado del valor de u

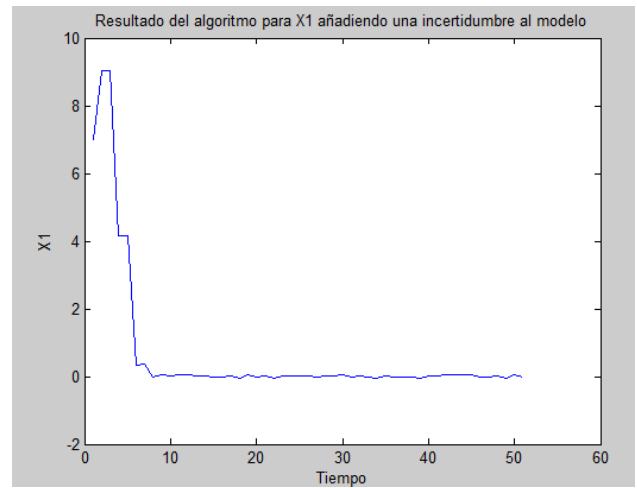


Figura 15.a. Resultado x_1 con una incertidumbre en el modelo

- Al problema original (11) se le añade una variable aleatoria de media cero,

$$x(t+1) = f(x(t), u(t))$$

$$f(x(t), u(t)) = Ax(t) + Bu(t) + W(t)$$

donde $W(t)$ es una incertidumbre que en cada instante de tiempo toma un valor entre $-0,0625$ y $0,0625$. En la figura 14 se observa la incertidumbre añadida al modelo del sistema.

Se pueden hacer simulaciones de robustez del método (sin entrar en detalles dado que no es objeto del trabajo). Se han obtenido los siguientes resultados y se muestran en las siguientes figuras 15.a, 15.b y 15.c donde se observan los efectos de esta incertidumbre.

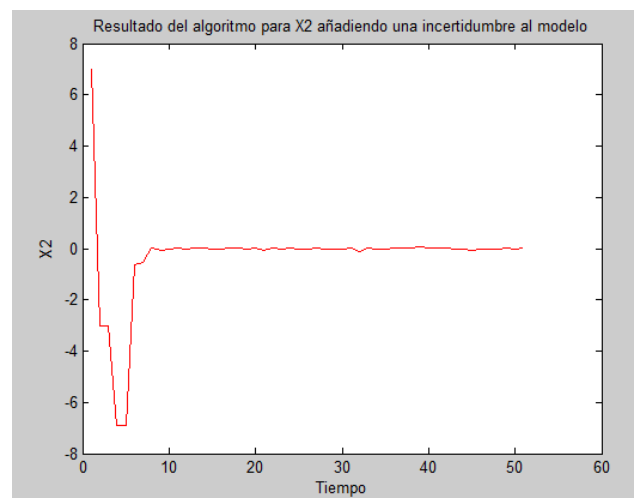


Figura 15.b. Resultado de x_2 con una incertidumbre en el modelo

La incertidumbre en el modelo no afecta a la convergencia del proceso ya que el controlador es capaz de corregir esa incertidumbre en el proceso de optimización por lo que no hay variaciones en el comportamiento del proceso. En régimen permanente sí se observa pequeñas variaciones con respecto a la referencia.

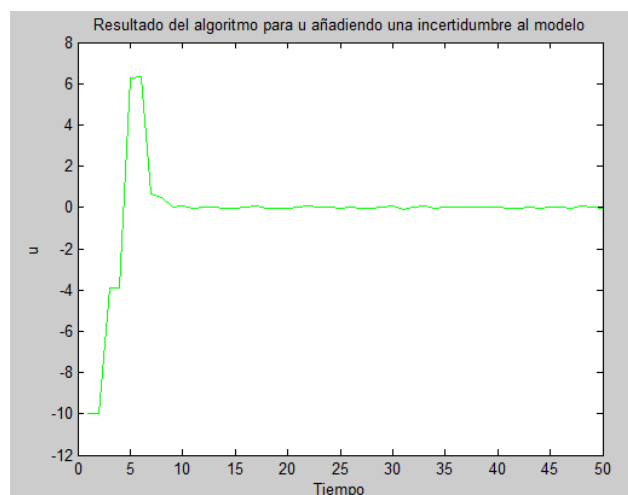


Figura 15.c. Resultado de u con una incertidumbre en el modelo

5. CONCLUSIONES

El artículo, en el cual se basa este trabajo, propone una técnica novedosa referente a un esquema de control predictivo basado en modelo para optimización multiobjetivo. Se ha simulado tal y como los autores los describen el algoritmo propuesto en [1] dando los mismos resultados aunque puede haber cierta discrepancia dado que no se conocen las condiciones de inicialización del problema y no se conocen los resolvidores numéricos utilizados aunque como se describe en la segunda parte de su trabajo se ha utilizado un controlador calculado de forma explícita. En el presente trabajo se ha calculado la simulación siguiendo exactamente el algoritmo propuesto en la figura 9 que es el mismo que se ha utilizado en [1].

Los resultados obtenidos mejoran considerablemente otros controladores como el H_2 y H_∞ tal y como se describe en el apartado 4.5. Sin duda, según las simulaciones, el controlador multiobjetivo converge considerablemente más rápido que cualquiera de los otros dos aunque produce una gran variación en la acción de control pero sin salirse de las restricciones planteadas. Esta situación es de esperarlo ya que la acción de control es una parte que determina la evolución del sistema en el siguiente instante.

Las variaciones del problema inicial también da buenos resultados. El cambio de una de las funciones objetivo de norma infinito a norma 1 no ha producido grandes cambios salvo en algunas transiciones como se observan en la figura 17 y 18 con respecto a las figura 13 y 14. Añadir la incertidumbre al modelo no produce ningún empeoramiento en la fase de convergencia del proceso. Tampoco es destacable la variación del estado de la planta en estabilidad con respecto a la referencia.

Por último, considerar todas las condiciones de estabilidad que garantizan la convergencia del método. En el algoritmo se han implementado condiciones de estabilidad tanto para la región terminal como para el del proceso en lazo cerrado que aseguran la solución del problema de optimización.

6. FUTUROS TRABAJOS

El algoritmo de optimización multiobjetivo implementado se ha realizado solucionando un problema de optimización en cada instante de tiempo. Esto hace muy costosa la computación del problema donde en procesos con dinámicas muy rápidas requiere de un controlador con un tiempo de muestreo muy pequeño. Por ello, el estudio e investigación de controladores explícitos multiobjetivo será en el futuro una de las líneas de investigación más importantes ya que reduce el control a un sistemas tabulado de acciones de control según regiones definidas por estados.

En el caso del presente trabajo el punto de vista elegido ha sido la optimización convexa mediante programación multiparamétrica. Esto es la obtención de la solución del problema de optimización mediante regiones que corresponden a las variables de estado, siguiendo el punto de vista sería investigar para la obtención de funciones afines de la forma

$$V^* = \varphi_i x + \omega_i \alpha + \gamma_i, \quad \text{si } \mathcal{H}_{ix} x \leq \mathcal{K}_{ix}$$

donde se añade un nuevo parámetro α que es utilizado para ser minimizado como se propone en el algoritmo utilizado.

7. AGRADECIMIENTOS

Este trabajo ha sido financiado con el Proyecto Nacional DPI2011-27818-C2-01 del Ministerio de Ciencia e Innovación y Fondos FEDER.

8. REFERENCIAS

- [1] A. Bemporad, D. Muñoz de la Peña. 2009. Multiobjective model predictive control. Elsevier Automatica 45(2009) 2823-2830.
- [2] A. Ramírez, F. Rodríguez, J.L. Guzmán, M. Berenguel. Multiobjective hierarchical control arquitectura for greenhouse crop growth. Automatica, 48, 490-498, 2012.
- [3] E.F. Camacho y C. Bordons. 2004. Model Predictive Control Second Edition. ISBN: 1852336943.
- [4] E.F. Camacho y C. Bordons. 2004. Control predictivo: pasado, presente y futuro. CEA-IFAC.
- [5] F. Borrelli, A. Bemporad, M. Morari. 2011. Predictive control for linear and hybrid systems. Cambridge.
- [6] S. Joe Qin, T.A. Badgwell. 2003. A survey of industrial model predictive technology. Pergamon. Control Engineering Practice 733-764.
- [7] R.E. Kalman. 1960. Contributions to the theory of optimal control. Bulletin de la Societe Mathematique de Mexicana, 5, 102-119.
- [8] R.E. Kalman. 1960. A new approach to linear filtering a predictions problem. Transactions of ASME, Journal of Basic Engineering, 87, 35-45.
- [9] G.C. Goodwin, S.F. Graebe, M.E. Salgado. (2001). Control system design. Englewood Cliffs, Prentice Hall.
- [10] J. Richalet, A. Rault, J.L. Testud, J. Papon. (1976). Algorithmic control of industrial processes. In Proceedings of the 4th IFAC symposium on identification and system parameter estimation. (pp. 1119-1167).
- [11] G.P. Liu, J.B. Yang, J.F. Whidborne. 2003. Multiobjective Optimisation and control. Research studies Press Ltd. Exerter.
- [12] O. Romank. 2010. Parametric y multiobjective optimization with applications in finance. McMaster University.
- [13] H. HAgelogiannaki. Aprioritized multiobjective MPC configuration using adaptive rbf networks y evolutionary computation. School of Chemical Engineering, NTUA, 15780 Athens, Greece.
- [14] P. dua, F.J. Doyle III. 2005. Multiobjective parametric control of blood glucose concentration for Type 1 Diabetes. 44th IEEE Conferece on Decision and Control, Seville Spain, 12/15/2005.
- [15] F. Borrelli, A. Bemporad, M. Morari. 2011. Predictive control for linear and hybrid systems.
- [16] S. Boyd, L. Vandenberghe. 2004. Convex optimization. Cambridge university press. ISBN 0521833787.
- [17] Z. Baotic. 2005. Optimal control piecewise affine systemas – a multiparametric approach –. Dissertation of Swiss federal institute of technology.
- [18] V. Pareto. 1906. Manuale di economia política, societa editrice librería, Milano. Translated into english by A.S. Schwier as Manual of political economy. Macmillan, New York.
- [19] C.A. Coello-Coello. 2006. *EMOO website*: <http://www.lania.mx/~ccoello/EMOO>.
- [20] K.M. Miettinen. 2006. Links connected to optimization, operations research and other interesting areas. <http://www.mit.jyu.fi/miETTINE/lista.html>.
- [21] P. Tondel, T.A. Johansen, A. Bemporad. An Algorithm for Multi-Parametric Quadratic Programming and Explicit MPC Solutions.
- [22] X. Prats, V. Puig, J. Quevedo, F. Nejjari. 2009. Optimal departure aircraft trajectories minimizing population annoyance. Technical University of Catalonia (UPC).

- [23] T.A. Johansen, A. Grancharova. 2004. Approximate explicit model predictive control implemented via orthogonal search tree partitioning. Department of engineering cybernetics, Norwegian University of Science and Technology.
- [24] M.M. Islam Chy, G. Boulet. 2011. A model predictive controller of plastic sheet temperature for a thermoforming process. American Control Conference O'Farrel Street, San Francisco, CA, USA.
- [25] E. Pistikopoulos, M. Georgiadis, V. Dua. 2007. Multiparametric Programming. WILEY-VCH Verlag GmbH & KGaA. ISBN: 9783527316915.
- [26] V. Miranda, C. de Prada. 2006. Controlador Predictivo Explícito de un evaporador. Departamento de Sistemas y automática. Universidad de Valladolid.
- [27] V. Exadaktylos, J. Taylor. 2010. Multi-objective performance optimisation for model predictive control by goal attainment. Lancaster University. International Journal of Control Vol. 83, N° 7, 1374-1386.
- [28] L. Feng, J. Wang, F. Liao. 2007. Multi-objective Robust Model Predictive Control: Trajectory Tracking Problem Through LMI Formulation. Proceedings of the 2007 American Control Conference.
- [29] M. Nikolaou. 1999. Model Predictive Controllers: A Critical Synthesis of Theory and Industrial Needs. Chemical Engineering Dept. University of Houston.

ANEXO

Código desarrollado en el entorno MATLAB[®] y YALMIP

El entorno de desarrollo de modelado para la optimización de problemas de control y otros utiliza lo que se denomina variables simbólicas. YALMIP permite trabajar constantemente con este tipo de variables, de hecho todo el código desarrollado es utilizado por este tipo de variables. El comando *sdpvar* define variables de decisión simbólicas y *solvesdp* es la función común para la solución de problemas de optimización. A continuación se presentará el código MATLAB[®] programado con el cual se han realizado todas las simulaciones.

- Código básico del algoritmo: contiene el código para la resolución del problema planteado multiobjetivo en [1].

%Se inicializa el problema, las matrices A,B,Q,R%%%

%También se inicializan las restricciones%%%

```
A=[1 1;0 1];
B = [0.5;1];
Qo = [0.1 0;0 1];
Ro = 0.2;
Po = [0.5649 0.4054;0.4054 1.6027];
Ql = [1 0;0 0.1];
Rl = 0.1;
Pl = [9.6085 1.1401;-0.2965 9.4107];
```

%Se inicializa la región final para asegurar la estabilidad%%%

```
K_final = [-0.5;-1.4];
H = [0 1;1 0;0 -1;-1 0;-0.5 -1.4;0.5 1.4];
K = [10;10;10;10;10;10];
```

%Horizonte de predicción%%%

```
N = 5;
```

%Cantidad de simulaciones a realizar%%%

```
Numero_simulaciones = 50;
```

%Se inicializan las variables simbólicas%%%

```
x = sdpvar(2,N+1);
U = sdpvar(N,1);
alpha = sdpvar(2,1);
```

% Punto inicial como en el problema planteado%%%

```
x(1,1) = 7;
x(2,1) = 7;
objective = 0;
J1 = 0;
J0 = 0;
```

%Se establecen los límites de operación tanto para el estado como para la acción de control%%%

```
F = set(-10<=x<=10);
F = F + set(-10<=U<=10);
```

%Se establecen el modelo en el problema y se resuelve para el momento inicial 0%%%

```
for k = 1:N-1
```

%Se establecen las funciones de coste para cada instante dentro del horizonte de predicción%%%

```
J0 = J0 + norm(x(:,k)')*Qo,inf)+norm( U(k)')*Ro,inf);
J1 = J1 + x(:,k)')*Ql*x(:,k) + U(k)')*Rl*U(k);
F = F + set(x(:,k+1) == A*x(:,k) + B*U(k));
End
```

%Condición de la región final para procurar la estabilidad

```
F = F + set(H*x(:,N)<= K);
```

%Se añade el valor de la solución de la ecuación de Ricatti para asegurar la estabilidad % % % % % % % % % % % %

```
J0 = J0 + norm(x(:,N))*Po,inf);
J1 = J1 + x(:,N)*Pl*x(:,N);
```

%Se inicializa alpha % % % % % % % % % % % %

```
alpha(1,1) = 0.2;
alpha(2,1) = 0.8;
```

%Se compone la función de coste multiobjetivo % % % %

```
J = double(alpha(1,1))*J0 + double(alpha(2,1))*J1;
objective = J;
```

%Se resuelve el problema para el estado inicial % % % % %

```
sol = solvesdp(F,objective);
```

%Se inicializan las variables escalares que son la solución del óptimo y el valor del estado en k+1 % % % % % % % % % %

```
U_actual = zeros(1,Numero_simulaciones);
x_actual = zeros(2,Numero_simulaciones+1);
```

%Se convierte a escalar la variable de simbólica para el primer instante % % % % % % % % % % % % % % % %

```
x_actual(:,1) = double(x(:,1));
```

%Simulamos tantas veces como instantes de tiempo se desean, para el caso particular del trabajo se han realizado 50 simulaciones para un horizonte de predicción 5 % % %

```
for k = 1:Numero_simulaciones
```

%Se convierte el primer instante de las variables simbólicas a escalares % % % % % % % % % % % %

```
U_actual(1,k) = double(U(1,1));
x_actual(:,k+1) = double(x(:,2));
```

```
x_siguiente = double(x(:,2));
```

%Se inicializa para volver a calcular en cada simulación, U_a se utiliza para establecer el Us que se utiliza para calcular Ja % % % % % % % % % % % % % % % %

```
U_a = sdpvar(N,1);
J0_a = 0;
J1_a = 0;
```

%Se reinician las variables para cada instante de tiempo calculado % % % % % % % % % % % % % % % %

```
for kk = 1:N-1
    U_a(kk,1) = U(kk+1,1);
End
```

% Se establece el valor de U para el caso en lazo cerrado % % % % % % % % % % % % % % % % % %

```
U_a(N,1) = K_final'*x(:,N);
```

% Se calcula Ja, condición para el cálculo del óptimo de alpha % % % % % % % % % % % % % % % %

```
for kk = 1:N
    J0_a = J0_a + norm(x(:,kk))*Qo,1) + norm(U_a(kk)*Ro,1);
    J1_a = J1_a + x(:,kk)*Ql*x(:,kk) + U_a(kk)*Rl*U_a(kk);
End
```

%Se sigue el algoritmo establecido según lo descrito en secciones anteriores % % % % % % % % % % % % % %

```
J_a =
double(alpha(1,1))*double(J0_a) + double(alpha(2,1))*double(J1_a);
```

%Se inicializa alpha como variable simbólica % % % % %

```
alpha = sdpvar(2,1);
```


%Se establece el valor al que alpha tiene que converger según la limitaciones del problema y la dinámica del modelo

```
alphad =
[1/(1+0.2*norm(double(x(:,2)),2));0.2*norm(double(x(:,2)),
2)/(1+0.2*norm(double(x(:,2)),2))];
```

%Se inicializan las restricciones para alpha

```
F_s =
set(alpha(1,1)*double(J0)+alpha(2,1)*double(J1)<J_a);
F_s = F_s + set(alpha>0);
F_s = F_s + set(sum(alpha)==1);
```

%Se establece el la function objetivo de alpha

```
objective = norm(alpha-alphad,1);
```

%Se soluciona el problema según la función de coste y sus restricciones

```
sol_alpha = solvesdp(F_s,objective);
```

%Se inicializan las variables de estado y acciones de control según su horizonte de predicción

```
x = sdpvar(2,N+1);
U = sdpvar(N,1);
J1 = 0;
J0 = 0;
F = 0;
```

%Se inicializa el valor actual del estado ya que este evoluciona en cada simulación

```
x(1,1) = x_actual(1,k);
x(2,1) = x_actual(2,k);
```

%Se vuelven a establecer las restricciones

```
F = set(-10<=x<=10);
F = F + set(-10<=U<=10);
```

%Se establecen las funciones de coste para cada instante dentro del horizonte de predicción

```
for kk = 1:N-1
    J0 = J0 + norm(x(:,kk))*Qo,1)+norm( U(kk)*Ro,1);
    J1 = J1 + x(:,kk)*Ql*x(:,kk) + U(kk)*Rl*U(kk);
    F = F + set(x(:,kk+1) == A*x(:,kk) + B*U(kk));
End
```

%Condición de la región final para procurar la estabilidad

```
F = F + set(H*x(:,N)<= K );
```

%Se añade el valor de la solución de la ecuación de Ricatti para asegurar la estabilidad

```
J0 = J0 + norm(x(:,N))*Po,1);
J1 = J1 + x(:,N)*Pl*x(:,N);
```

%Se compone la función de coste multiobjetivo y se resuelve el problema multiobjetivo planteado

```
J = double(alpha(1,1))*J0+double(alpha(2,1))*J1;
objective = J;
sol = solvesdp(F,objective);
end
```

%Fin

- Código para cambiar la función objetivo de norma infinito a norma 1: la siguiente línea de código sustituirá la valor de J0 en la composición de la función de coste.

$$J0 = J0 + \text{norm}(x(:,k)'*Qo,1) + \text{norm}(U(k)*Ro,1)$$

- Código para añadir la incertidumbre al modelo: este código se añade en la composición de las restricciones ya que la dinámica del proceso se introduce como una restricción más.

$$E = [0.0625; 0];$$

$$F = F + \text{set}(x(:,k+1) == A*x(:,k) + B*U(k) + E*(-1 + 2*rand(1)));$$

Resumen—El control predictivo basado en modelo es una técnica de control que se ha expandido en los últimos años. La gestión de costes con restricciones con el objeto de optimizar variables de estado se ha empleado desde hace mucho tiempo hasta ahora. Dando un enfoque nuevo se pueden controlar sistemas ajustado a este concepto. El control predictivo basado en modelo clásico optimiza un solo criterio u objetivo, esta técnica ha dado buenos resultados. Dado que se desea ir más allá el presente trabajo trata el problema desde un punto de vista multiobjetivo y describe la técnica utilizada. Se caracterizan las herramientas que hay al alcance y explica el algoritmo utilizado reproduciendo el ejemplo descrito en un trabajo novedoso y actual. Además, se realizan pequeñas variaciones para comprobar el algoritmo utilizado. El código utilizado está escrito para el programa de ingeniería MATLAB® con la ayuda de YALMIP, siendo un toolbox para el modelado de sistemas que brinda una interfaz para la optimación.

