

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“PLATAFORMA BASADA EN MATLAB® PARA
ENTRENAMIENTO EN LA SINTONÍA
DE CONTROLADORES PREDICTIVOS”

TESIS DE GRADO

Previa a la obtención del Título de:

MAGÍSTER EN AUTOMATIZACIÓN Y CONTROL
INDUSTRIAL

Presentado por:

Julio César Zambrano Abad

Guayaquil – Ecuador

2014

AGRADECIMIENTO

Primeramente a Dios por estar a mi lado en este largo camino, por bendecirme con salud y darme la fortaleza para seguir en los momentos más difíciles.

A mi directora de tesis, Dra. Ana Isabel González por su dedicación, paciencia y sobre todo por ser el faro que alumbró mi camino con sus conocimientos transmitidos.

A mis profesores del programa, de una manera especial al MSc. Efrén Herrera, al MSc. Carlos Salazar y al Dr. Orestes Llanes, por la motivación, sus consejos y enseñanzas a lo largo de todo el programa.

A mi esposa Mary y a mi hijo Julio Alejandro por prestarme el tiempo que les pertenecía.

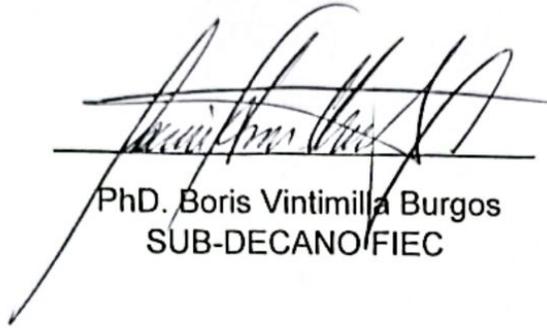
DEDICATORIA

A mi esposa Mary quien con mucho amor y paciencia me alentó a seguir y me ayudó a levantarme en los momentos tan difíciles que viví durante este trayecto. A mi hijo Julio Alejandro quien fue la fuente de inspiración para superarme profesionalmente.

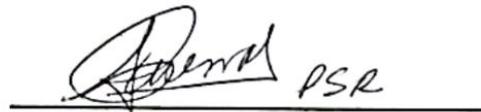
A mi padres Mariana y Julio quienes forjaron en mí el deseo de superación y me enseñaron desde pequeño a luchar para alcanzar mis metas. Les dedico este trabajo por darme todo lo que estuvo a su alcance para que me eduque y sea un hombre de bien.

A mi tía Rosa quien me permitió robarme un espacio de su hogar y me recibió como un hijo más mientras realizaba mis estudios.

TRIBUNAL DE SUSTENTACIÓN



PhD. Boris Vintimilla Burgos
SUB-DECANO FIEC



Dra. Ana Isabel González
DIRECTORA DE TESIS



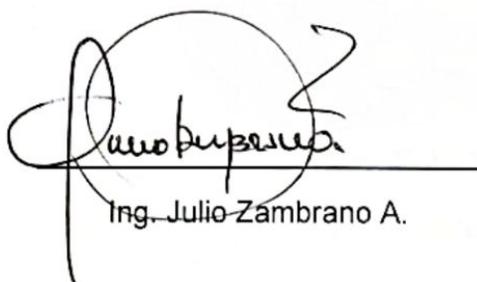
MSc. Efrén Herrera
CO-DIRECTOR DE TESIS



MSc. Carlos Salazar L
MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".



Ing. Julio Zambrano A.

RESUMEN

La presente tesis consiste en la implementación de una plataforma para sintonía y simulación de controladores predictivos. La metodología parte del conocimiento de las diversas variantes y los aspectos algorítmicos del MPC. Posteriormente y producto del estudio se construye en MATLAB® la plataforma de simulación. A continuación se detalla a manera de resumen el contenido capitular de la presente tesis.

En el capítulo 1 se presenta la motivación, los objetivos de la tesis y la solución adoptada. Por último se detalla la estructura capitular de la tesis.

En el capítulo 2 se presenta un estado del arte de la metodología de control. Se analiza la evolución del control predictivo reflejada en las variantes que presenta. Posteriormente se presentan las alternativas comerciales para la industria. Finalmente se da una breve revisión de las herramientas de sintonía y simulación con las que se cuenta en la actualidad, haciendo énfasis en MPCTool de MATLAB®.

En el capítulo 3 se presenta el fundamento de la estrategia MPC. El análisis parte de una introducción en la que se anota una idea muy generalizada del

control predictivo. Posteriormente se estudia el fundamento matemático del algoritmo adicionando aspectos relacionados a la estimación de estados y la incorporación de restricciones. Seguidamente se presenta una solución al problema de optimización utilizando programación cuadrática. Por último se presenta detalles del controlador predictivo con incorporación de restricciones en las variables manipuladas y variables de salida.

En el capítulo 4 se presenta la implementación de la plataforma de simulación SIMUMPC. Esta plataforma se basa en la utilización de MATLAB® como herramienta de programación. Se inicia proporcionando detalles del protocolo de comunicación para el intercambio de información entre el PC-planta y el PC-controlador. Posteriormente se dan detalles del algoritmo para obtener el modelo identificado, el mismo que puede servir al PC-controlador para la ejecución del algoritmo MPC. A continuación se presenta la implementación del algoritmo MPC escogido. Al final de este capítulo se presentan los resultados de las pruebas realizadas.

Finalmente se presentan conclusiones y recomendaciones producto de la investigación realizada.

ÍNDICE GENERAL

ABREVIATURAS	X
SIMBOLOGÍA	XI
ÍNDICE DE FIGURAS	XII
ÍNDICE DE TABLAS	XV
INTRODUCCIÓN	XVI
1. MOTIVACIÓN Y OBJETIVOS DE LA TESIS.....	1
1.1 Motivación.....	1
1.2 Objetivos de la tesis y solución adoptada.....	6
1.3 Organización de la tesis.....	9
2. ESTADO DEL ARTE.....	12
2.1 Variantes de control predictivo.....	13
2.2 Control predictivo en la industria.....	21
2.3 Herramientas de simulación para control predictivo.....	27
3. CONTROL PREDICTIVO BASADO EN MODELO (MPC).....	49
3.1 Introducción.....	50
3.2 Fundamentación del control predictivo basado en modelo...	52
3.3 Estimador de estados.....	64
3.4 Formulación de restricciones.....	67
3.5 Solución numérica utilizando programación cuadrática.....	72
3.6 Control predictivo con restricciones en las variables manipuladas y variables de salida.....	82

3.7 Diseño del experimento.....	87
4. IMPLEMENTACIÓN DE LA PLATAFORMA SIMUMPC Y PRUEBAS DEL SISTEMA.....	89
4.1 Protocolo de comunicación.....	90
4.2 Herramienta para identificación de sistemas.....	96
4.3 Algoritmo MPC.....	102
4.4 Pruebas del sistema y análisis de resultados.....	121

ABREVIATURAS

DMC:	Control con matriz dinámica.
GPC:	Control predictivo generalizado.
HCE:	Herramienta de control y estimación de mpctool
LP:	Programación lineal.
MIMO:	Múltiples entradas, múltiples salidas.
MPC:	Control predictivo basado en modelo.
MPHC:	Control predictivo heurístico basado en modelo.
OPC:	OLE para control de procesos.
PFC:	Control predictivo funcional.
QP:	Programación cuadrática.
RMPCT:	Tecnología de control predictivo multivariable robusto
SISO:	Simple entrada, simple salida
UDP:	Protocolo de datagrama de usuario.
UTP:	Par trenzado no blindado.

SIMBOLOGÍA

M:	Horizonte de predicción.
N:	Horizonte de control.
$r\omega$:	Factor de peso.
R:	Matriz de pesos.
$\bar{r}s$:	Vector de puntos de ajuste.
u:	Variable de control.
Δu :	Incremento de la variable de control.
x:	Estado del proceso.
y:	Salida del proceso.
A:	Matriz de estados.
B:	Matriz de entrada en un sistema de espacio de estados.
C:	Matriz de salida en un sistema de espacio de estados.
D:	Matriz de transmisión directa.
F, Φ :	Matrices de inicialización de la formulación predictiva.
H:	Matriz Hessiana.
J:	Función de coste.
P:	Matriz de covarianza de error.
λ :	Multiplicadores de Lagrange

ÍNDICE DE FIGURAS

Figura 1.1 Esquema de interconexión de la plataforma SIMUMPC.....	7
Figura 2.1 Vista inicial de la herramienta de control y estimación (MPCTool).....	30
Figura 2.2 Simulación de vuelo de la aeronave ‘Cessna Citation 500’ para una consigna de 40m en la altura del avión.....	37
Figura 2.3 Simulación de vuelo de la aeronave ‘Cessna Citation 500’ con restricción en la tasa de altitud.....	38
Figura 2.4 Detalles del controlador predictivo creado con el comando mpc de MATLAB®.....	39
Figura 2.5 Simulación de vuelo de la aeronave ‘Cessna Citation 500’ ejecutada desde MATLAB®.....	42
Figura 2.6 Bloques MPC desde Simulink®.....	43
Figura 2.7 Bloque MPC para la creación de un sistema de control en Simulink®.....	44
Figura 2.8 Simulación de vuelo de la aeronave ‘Cessna Citation 500’ utilizando Simulink®.....	46
Figura 3.1 Idea de control predictivo basado en modelo.....	51
Figura 3.2 Controlador MPC con observador de estados.....	67
Figura 4.1 Esquema de configuración para la comunicación UDP entre 2 PCs.....	92
Figura 4.2 Transmisión y recepción de datos a través de UDP.....	93
Figura 4.3 Señales PRBS para identificación de sistema con tres entradas.....	98
Figura 4.4 Respuestas del sistema ante señales de entrada PRBS.....	99
Figura 4.5 Gráfica del comando n4sid para sugerir el orden de modelo identificado.....	101
Figura 4.6 Validación del proceso de identificación.....	101
Figura 4.7 Diagrama esquemático de la estrategia de control predictivo.....	102

Figura 4.8 Diagrama esquemático para la implementación del modelo aumentado.....	103
Figura 4.9 Coste computacional de la fase de inicialización para un caso específico.....	108
Figura 4.10 Formulación de restricciones para el caso SISO con $N=5$ y $M=3$	110
Figura 4.11 Formulación de restricciones en Δu para el caso MIMO.....	111
Figura 4.12 Proceso iterativo del algoritmo MPC para el caso sin restricciones.....	112
Figura 4.13 Coste computacional del algoritmo MPC sin restricciones...	113
Figura 4.14 Proceso iterativo del algoritmo MPC con restricciones.....	115
Figura 4.15 Coste computacional para un algoritmo MPC con restricciones.....	116
Figura 4.16 Ventana de inicio de la herramienta SIMUMPC.....	119
Figura 4.17 Ambiente simMPC.....	120
Figura 4.18 Ambiente simMPCejec.....	121
Figura 4.19 MODELO SISO: Servomecanismo de posición.....	123
Figura 4.20 Ejecución del algoritmo MPC sobre planta SISO para dos conjuntos diferentes de horizonte de control y predicción.....	124
Figura 4.21 Ejecución del algoritmo MPC sobre planta SISO para dos conjuntos diferentes del factor de peso R	125
Figura 4.22 Desempeño del controlador predictivo cuando el modelo de predicción difiere del modelo de la planta.....	126
Figura 4.23 MPC SISO con restricciones en Δu	127
Figura 4.24 Incumplimiento de restricciones en el algoritmo MPC.....	128
Figura 4.25 Ejecución del algoritmo MPC con restricciones en Δu y u	129
Figura 4.26 Turbina - caldera tipo tambor para pruebas del controlador MPC MIMO.....	131
Figura 4.27 Respuestas del sistema turbina - caldera tipo tambor (MPC sin restricciones).....	132
Figura 4.28 Acciones de control para el sistema turbina - caldera tipo tambor (MPC sin restricciones).....	132

Figura 4.29 Salidas del sistema turbina - caldera tipo tambor (MPC con restricciones).....	135
Figura 4.30 Acciones de control para el sistema turbina - caldera tipo tambor (MPC con restricciones).....	135
Figura 4.31 Modelo de un motor dc.....	136
Figura 4.32 Simulación en línea: MPC sin restricciones para control de velocidad motor dc.....	137
Figura 4.33 Simulación en línea: MPC con restricciones para control de velocidad motor dc.....	138
Figura 4.34 Comparación entre modo en línea y fuera de línea para servomecanismo de posición.....	139
Figura 4.35 Reactor químico multivariable.....	140
Figura 4.36 Simulación en línea: MPC para reactor químico no lineal.....	142
Figura 4.37 Comparación entre modo en línea y fuera de línea para reactor químico no lineal multivariable.....	143

ÍNDICE DE TABLAS

Tabla 3.1 Formulación de restricciones para el caso MIMO.....	70
Tabla 4.1 Resultado de la Implementación de un integrador embebido.....	104
Tabla 4.2 Dimensión de las matrices F y Φ	106

INTRODUCCIÓN

El control predictivo basado en modelo es una estrategia de control altamente atractiva que puede tratar con procesos de dinámica compleja, no lineal y multivariable. Hoy en día existen diversas variantes de esta estrategia y estas dependen del formato de modelo utilizado por el controlador para realizar las predicciones de salida. En este trabajo se presenta la implementación de una plataforma basada MATLAB® para el entrenamiento en la sintonía de controladores predictivos. Esta plataforma tendrá la particularidad de que puede funcionar a través de una comunicación remota por medio de un protocolo de comunicación UDP. De esta manera la plataforma bajo esta modalidad requerirá de dos computadores personales (PC) uno que haga las veces de controlador y otro de planta.

Con el presente trabajo se logrará mejorar las características que presenta la herramienta de simulación mpctool de MATLAB® que permite la simulación de controladores predictivos con la planta y el controlador residiendo en el mismo computador. Al disponer de la planta y el controlador en PCs diferentes las mejoras frente a mpctool vendrán dadas por la inclusión de parámetros que están implícitos en la adquisición de datos como retardos y ruido.

El desarrollo de este trabajo parte con el estado de arte de la estrategia de control. Posteriormente se estudia el fundamento teórico y los aspectos matemáticos del control predictivo basado en modelo. En base al estudio desarrollado se utilizará un interfaz gráfico de usuario (GUIDE) y las herramientas de programación que brinda MATLAB® para implementar el algoritmo predictivo.

CAPÍTULO 1

MOTIVACIÓN Y OBJETIVOS DE LA TESIS.

1.1 Motivación.

La evolución del mercado y las estrictas exigencias marcadas por normativas para el cumplimiento de altos estándares de calidad han obligado a las industrias de todo el mundo a acogerse a técnicas de control modernas que permitan marchar en consonancia con los requerimientos que surgen día a día. Conforme han ido surgiendo necesidades dentro del área de control de procesos industriales, el avance de la ciencia y la tecnología ha venido ofreciendo un amplio abanico de técnicas de control para solventar dichas necesidades. En 1999 [1] presentó un reporte desde el enfoque de la teoría de control sobre el desempeño de la industria japonesa en los últimos años. En este reporte se deja ver claramente como para el año 1989 la inestabilidad, la no-linealidad, las variaciones, el tiempo muerto, las interacciones y las perturbaciones eran problemas claramente existentes, cuyos algoritmos de control no podían mitigar por completo sus efectos. Para el año 1995 algunos de estos problemas como la estabilidad y la no-

linealidad disminuyeron pero otros como la interacción aumentaron, dejando ver un desempeño creciente en las técnicas de control y a la vez la marcada aparición de procesos complejos en la industria.

En la actualidad no existe un reporte en cifras de los problemas que afronta la industria desde el punto de vista de la teoría de control, sin embargo del análisis de la literatura existente se puede decir que las técnicas de control moderno ayudan a mitigar mayormente los problemas que enfrenta la industria, pues, gestionan situaciones complejas y reales donde el control clásico requiere mejoras sustanciales para controlar un proceso. El Control Predictivo basado en Modelo (MPC) es una metodología capaz de ofrecer soluciones a todos los problemas previamente señalados. Esta poderosa herramienta fue introducida a través de la industria petro-química en 1978 y presentada como Control Predictivo Heurístico basado en Modelo MPHC [2]. Desde sus inicios ha revolucionado la ingeniería de control dando solución a procesos con comportamiento dinámico complejo. En los últimos años su aplicación se ha extendido hacia otros sectores de la industria debido a las grandes ventajas que presenta con respecto al control tradicional y a otras técnicas de control moderno (ver [3], [4] y [5]). Entre sus principales ventajas se puede citar:

- Facilidad para tratar con sistemas de múltiples entradas y múltiples salidas.

- Incorporación de restricciones en el actuador.
- Permisividad para trabajar en zonas muy cercanas a las restricciones.
- Variedad de algoritmos dependiendo del formato del modelo de predicción.
- Operatividad en procesos continuos y procesos por lotes, tratamiento con plantas inestables y de fase no mínima.

En contraparte la desventaja más considerable es la complejidad del algoritmo que concurre en un problema de optimización lo cual requiere un tiempo elevado de procesamiento y la utilización de ordenadores para su implementación. Esto ha hecho que el nacimiento de esta estrategia se dé en procesos con dinámica lenta. Sin embargo en los últimos años, el avance de la ciencia y la tecnología conjuntamente con el interés de científicos e investigadores están dejando de lado este inconveniente. [6, p. 2] Cita a [7] quien manifiesta que:

En los últimos diez años el incremento de la velocidad de hardware conjuntamente con las mejoras de algoritmos de optimización ha incrementado en un factor 10^6 el tiempo de solución del problema de optimización. Es decir un problema que hace 10 años tomaba 10 minutos en ser resuelto hoy se puede hacer en 600 microsegundos.

Adicionalmente [8] describe una serie de métodos para mejorar el tiempo de respuesta del algoritmo MPC. A diferencia de otras técnicas, esta

herramienta de control avanzado hace uso del modelo dinámico del proceso y de una estrategia de optimización. Por la facilidad de llevar el algoritmo generalmente se utilizan modelos lineales obtenidos a partir de pruebas experimentales basadas en la obtención de parámetros o a través de métodos de identificación de sistemas (ver [9] y [10]). No obstante en los últimos años se ha dado una amplia aceptación a algoritmos que utilizan modelos no lineales. Abundante literatura respecto al tema se puede encontrar en: [11], [12], [13] y [14]. En consideración a la utilización del modelo, esta estrategia de control no se enmarca dentro de una teoría estricta, pues como se ve existen muchas variantes para representar la dinámica de un proceso, cada una de estas viene asociada con un determinado algoritmo (ver [15, pp. 2-4 y 12-15]).

En la actualidad existen muchos reportes en los que se presentan ventajas cuantitativas debido al remplazo de un controlador tradicional PID por un MPC dentro de un lazo de control. Sin embargo, este no es su único nicho, pues un controlador predictivo puede funcionar en un nivel jerárquico superior proporcionando valores óptimos del punto de ajuste a un nivel inferior en el que puede residir un controlador tradicional PID. El cálculo de estos puntos de ajuste no solamente está vinculado con el aspecto operacional de la planta, puede incorporar aspectos económicos y

ambientales. A esta técnica de control se le conoce como control predictivo supervisorio (ver [16] y [17]).

Desde sus inicios el control predictivo basado en modelo tuvo gran aceptación pese a carecer de un sustento para garantizar la estabilidad de los sistemas. No obstante la motivación de los investigadores ha permitido disponer hoy en día de una serie de formulaciones de control predictivo con estabilidad garantizada mediante la utilización de la teoría de Lyapunov. En [6, pp. 167-185] se puede ver una investigación sobre las diversas maneras de garantizar la estabilidad, por otra parte [18] resume los ingredientes necesarios para satisfacer las condiciones de estabilidad en el MPC.

La gran aceptación a nivel industrial y su evolución reflejada en las variantes que oferta (ver [4]) ha provocado que este conjunto de métodos se constituya en el foco de interés para programas de estudio a nivel superior que persiguen la formación de especialistas en el área de control. No obstante por la esencia de la metodología en la etapa de preparación es difícil contar con plantas o procesos reales que permitan aterrizar los conceptos teóricos. Hoy en día se cuenta con la herramienta "MPCTool" de MATLAB[®] (ver [19] y [20]) para sintonizar y simular el comportamiento de un controlador predictivo para un determinado proceso, sin embargo por la concepción de la técnica es

necesario contar con el modelo del proceso e involucrar una serie de parámetros para que el resultado se apegue más al entorno real.

1.2 Objetivos de la tesis y solución adoptada.

1.2.1 Objetivo general.

Diseñar e implementar una plataforma de software para aplicaciones simuladas de control predictivo, basada en la interconexión de dos ordenadores (PC), uno que emule el comportamiento de una planta y otro que realice las tareas de control.

1.2.2 Objetivos específicos.

- Implementar una plataforma de comunicación con Protocolo de Datagrama de Usuario (UDP) [21] entre dos PCs. Una PC tendrá el controlador implementado y otra PC hará de planta, pasándole en cada período de muestreo las mediciones al controlador.
- Estudiar los aspectos matemáticos que fundamentan el control predictivo basado en modelo con y sin restricciones.
- Implementar un algoritmo que permita identificar el modelo de la planta para utilizarlo en el controlador.
- Implementar el algoritmo controlador utilizando el lenguaje de programación de MATLAB[®] y brindado al usuario la flexibilidad necesaria para manipular los parámetros de ingreso que requiere algoritmo.

- Evaluar el desempeño del controlador utilizando diversas plantas simuladas y los reportes generados por el algoritmo.

1.2.3 Solución adoptada (plataforma SIMUMPC).

La plataforma de simulación SIMUMPC ofrece al usuario una variante para la sintonía de controladores predictivos. El presente trabajo de investigación presenta detalles del diseño e implementación de esta plataforma basada en la comunicación de dos PCs, uno para simular el comportamiento de la planta y otro para ejecutar el algoritmo de control. La figura 1.1 muestra un esquema que ilustra la interconexión que tendrá la plataforma SIMUMPC.

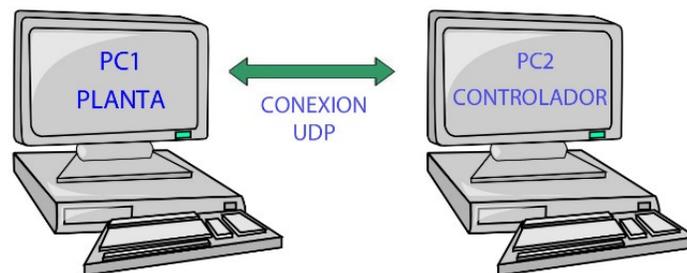


Figura 1.1 Esquema de interconexión de la plataforma SIMUMPC

El comportamiento dinámico de la planta se transmite a través de sus variables medibles hacia el PC-controlador. Mediante los valores numéricos de estas variables en este PC se puede realizar la identificación del modelo, la ejecución del algoritmo MPC y si es necesaria una estimación de estados.

La comunicación entre los dos PCs se realiza mediante un cable UTP cat6 [22] con configuración cruzada. La información se intercambia en cada período de muestreo utilizando el protocolo de comunicación UDP, respetando las condiciones de cada elemento de la comunicación, pues la planta al ser simulada por un computador puede recibir grandes paquetes de información en un solo período, caso que no ocurre en una planta real, sin embargo se realiza los ajustes necesarios para que el sistema funcione apegado a la realidad, sobre todo incluyendo parámetros representativos generados en la adquisición de datos. A continuación se detallan las características de los PCs vinculados en la plataforma de simulación:

PC1 (Planta):

- Contiene una gran variedad de modelos de plantas en formato de ecuación diferencial. Estos modelos son tomados de libros y artículos científicos disponibles.
- Posibilidad de que el usuario pueda ingresar nuevos modelos de predicción.
- A los valores de las variables de salida que fluyen hacia el PC controlador se pueden agregar ruido aleatorio para apegar el funcionamiento del sistema a la realidad.
- Los datos que ingresan son procesados y salen de uno en uno, simulando de mejor manera el comportamiento de una planta real.

PC2 (Controlador):

- Brinda la opción de identificar sistemas lineales SISO y MIMO, mediante la técnica de sub-espacios.
- Brinda la posibilidad de realizar estimaciones de estado.
- Realiza el algoritmo de control predictivo basado en modelo, pudiendo utilizarse como alternativa un modelo identificado o uno ingresado manualmente por el usuario.
- Presenta las gráficas de respuesta del sistema controlado.
- Brinda flexibilidad para cambiar los parámetros del controlador, como por ejemplo: horizonte de predicción, horizonte de control, restricciones, etc.

1.3 Organización de la tesis.

La presente tesis consiste en la implementación de una plataforma para sintonía y simulación de controladores predictivos. La metodología parte del conocimiento de las diversas variantes y los aspectos algorítmicos del MPC. Posteriormente y producto del estudio se construye la plataforma de simulación. A continuación se detalla el contenido capitular de la presente monografía.

En el capítulo 2 se presenta un estado del arte de esta metodología de control. Se analiza la evolución del control predictivo reflejada en las

variantes que presenta. Posteriormente se presentan las alternativas comerciales para la industria. Finalmente se da una breve revisión a MPC TOOL de MATLAB[®], una de las alternativas de simulación más completas para la sintonía de este tipo de controladores.

En el capítulo 3 se presenta el fundamento de la estrategia MPC. El análisis parte de una introducción en la que se anota una idea muy generalizada del control predictivo. Posteriormente se estudia el fundamento matemático del algoritmo adicionando aspectos relacionados a la estimación de estados y la incorporación de restricciones. Seguidamente se presenta una solución al problema de optimización utilizando programación cuadrática. Por último se presenta detalles del controlador predictivo con incorporación de restricciones en las variables manipuladas y variables de salida.

En el capítulo 4 se presenta la implementación de la plataforma de simulación SIMUMPC. Esta plataforma se basa en la utilización de MATLAB[®] como herramienta de programación. Se inicia proporcionando detalles del protocolo de comunicación para el intercambio de información entre el PC-planta y el PC-controlador. Posteriormente se dan detalles del algoritmo para obtener el modelo identificado, el mismo que puede servir al PC-controlador para la ejecución del algoritmo MPC. A continuación se presenta la

implementación del algoritmo MPC escogido. Al final de este capítulo se presentan los resultados de las pruebas realizadas.

Finalmente se presentan conclusiones y recomendaciones producto de la investigación realizada.

CAPÍTULO 2

ESTADO DEL ARTE

El control predictivo basado en modelo se puede considerar como un conjunto de estrategias que están sujetas en mayor o menor medida a ideas comunes que constituyen la esencia del mismo. En resumen estas ideas se pueden expresar de la siguiente manera:

- Utilización de un modelo para predecir las salidas del proceso.
- Minimización de una función de coste para el cálculo de las señales de control.
- Utilización de un horizonte de predicción que se va desplazando hacia el futuro (estrategia de horizonte deslizante)

En este capítulo se presenta una breve síntesis del fundamento de las variantes que conforman la familia de controladores predictivos. Posteriormente se analizan las variantes comerciales de control predictivo que hoy en día se implementan en la industria. Por último se analiza

MPCTOOL de MATLAB[®] una de las alternativas de simulación más completas para la sintonía de este tipo de controladores.

2.1 Variantes de control predictivo.

Tras la aparición del control predictivo se ha venido dando una marcada evolución que ha puesto a disposición de la industria y la comunidad científica un grupo de variantes que conforman la familia de controladores predictivos. Estas variantes principalmente difieren en la forma de representar el modelo dinámico del proceso (véase [23]). A continuación se realiza un breve análisis de las variantes de controladores predictivos, anotando sus rasgos más sobresalientes.

2.1.1 Control con matriz dinámica (DMC).

Esta técnica de control apareció poco después de la formulación precursora de Richalet et. al, que se le conoció como control predictivo heurístico basado en modelo MPHC. Se atribuye la idea de DMC a Cluter y Ramaker [24] trabajadores de la Shell Oil Co., y su éxito se debe a la gran facilidad para tratar con procesos multivariables de gran dimensión. Esta técnica utiliza un modelo de respuesta al escalón como se muestra en la ecuación 2.1.

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t - i) \quad (2.1)$$

El algoritmo consiste en la formulación de una matriz que se le conoce como matriz dinámica, de ahí el origen de su nombre. Esta matriz está conformada por los coeficientes g_i de la respuesta al escalón y su dimensión depende del horizonte de control m . La ecuación 2.2 muestra la estructura de la matriz dinámica G para un sistema SISO:

$$G = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_1 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_m & g_{m-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_p & g_{p-1} & \ddots & g_{p-m+1} \end{bmatrix} \quad (2.2)$$

Las variables de control se calculan minimizando una función de coste cuadrática como se muestra en la ecuación 2.3. Los factores p y m se les conoce como horizontes de predicción y control respectivamente, \hat{y} es la salida predicha, ω es la trayectoria de referencia y Δu representa los movimientos de la señal de control.

$$J = \sum_{j=1}^p [\hat{y}(t+j|t) - \omega(t+j)]^2 + \sum_{j=1}^m \lambda [\Delta u(t+j-1)]^2 \quad (2.3)$$

En el caso de que no exista restricciones el problema de control se resuelve de manera analítica. El resultado del procedimiento permite disponer del vector u que contiene los futuros incrementos de la señal de control (véase la ecuación 2.4).

$$u = (G^T G + \lambda I)^{-1} G^T (\omega - f) \quad (2.4)$$

Cabe indicar que este procedimiento se realiza en cada período de muestreo y únicamente se aplica a la planta el primer término del vector u . En [25, pp. 33-38] se puede ver en detalle la formulación del algoritmo y el procedimiento para ampliar al caso multivariable. Cuando se incorporan restricciones la minimización de la función de coste se convierte en un problema de programación cuadrática (QP) [26].

En la actualidad existen numerosas aplicaciones de DMC tanto a nivel industrial como a nivel de investigación científica en plantas didácticas de laboratorio. Por ejemplo [27] presenta la implementación de DMC para una caldera y sistema de turbina. El autor realiza un estudio comparativo utilizando dos alternativas: un modelo linealizado y otro con los datos de la respuesta al escalón del proceso. Pese a ser una estrategia ampliamente utilizada algunos autores sostienen que esta técnica carece de sustento en cuanto a estabilidad y opciones para el cálculo de parámetros de sintonía. Esto ha incentivado a que sobre la base de DMC se desarrollen variantes para satisfacer estos requerimientos. En [28] se propone la implementación de una nueva ecuación de sintonización que permita gobernar la agresividad del controlador. Por otra parte [29] plantea una formulación de DMC con estabilidad de Lyapunov.

El coste computacional del algoritmo DMC implica la utilización de ordenadores y dependiendo de la dinámica del proceso puede que su sistema operativo este obligado a trabajar en tiempo real. En [30] se presenta una propuesta de implementación donde se mide el desempeño de un algoritmo DMC que se ejecuta en una plataforma de tiempo real. No obstante cuando la dinámica del proceso lo exige esta no es la única alternativa, [31] presenta la implementación del algoritmo utilizando una plataforma embebida FPGA.

2.1.2 Control predictivo generalizado (GPC).

El control predictivo generalizado fue propuesto por Clarke et al., en 1987 [32] y hoy en día constituye una de las variantes más populares dentro del control predictivo. A diferencia de DMC que utiliza un modelo de respuesta al escalón, GPC utiliza un modelo CARIMA (*controller autoregressive and integrated moving-average*) para representar el comportamiento de la planta (véase la ecuación 2.5). Este modelo es óptimo cuando las perturbaciones del sistema no son estacionarias.

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})\frac{e(t)}{\Delta} \quad \text{con } \Delta = 1 - z^{-1} \quad (2.5)$$

En la ecuación 2.5 $u(t)$ y $y(t)$ son la señal de control y la variable controlada respectivamente, $e(t)$ es el ruido blanco con media cero y d es el tiempo

muerto del sistema. Los polinomios A , B y C son polinomios en el operador con desplazamiento hacia atrás z^{-1} (véase las ecuaciones 2.6, 2.7 y 2.8).

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na} \quad (2.6)$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb} \quad (2.7)$$

$$C(z^{-1}) = 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{nc}z^{-nc} \quad (2.8)$$

En palabras de Camacho y Bordons [25]:

El GPC consiste en el cálculo de una secuencia de futuras acciones de control de forma que se minimice una función de costo multipaso. El índice a minimizar es una función cuadrática que mide por un lado la distancia entre la salida predicha del sistema y una cierta trayectoria de referencia hasta el horizonte de predicción, y por otro el esfuerzo de control necesario para obtener dicha salida (véase la ecuación 2.9).

Con este algoritmo se puede abordar con naturalidad problemas de control en plantas de fase no mínima y plantas inestables en lazo abierto garantizando un cierto grado de robustez.

$$J = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - \omega(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2 \quad (2.9)$$

En la ecuación 2.9 N_1 y N_2 son los valores mínimo y máximo de la ventana u horizonte de predicción y N_u es el horizonte de control. Los parámetros $\delta(j)$ y

$\lambda(j)$ son las secuencias de ponderación o pesos, que según [3] en muchas situaciones se les considera igual a uno y constante respectivamente, $\omega(t + j)$ es el conjunto de trayectorias de referencia futuras. La fórmula resultante para la obtención de las predicciones óptimas se muestra en la ecuación 2.10, en esta ecuación $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$ y $F(z^{-1})$ provienen de una ecuación diofántica (véase [25] para mayor detalle de la formulación).

$$\hat{y}(t + j|t) = G_j(z^{-1})\Delta u(t + j - d - 1) + F(z^{-1})y(t) \quad (2.10)$$

En el caso de que no existan restricciones se utiliza la ecuación 2.9 conjuntamente con la ecuación 2.10 en un problema analítico que permita despejar las acciones de control u . En el caso de que existan restricciones la solución se obtiene utilizando un método de programación cuadrática. Al igual que en el DMC este procedimiento se realiza en cada período de muestreo y únicamente se aplica a la planta el primer término del vector u .

En [33] se puede ver la incorporación de restricciones de entrada y salida en la formulación del GPC. En este mismo artículo se presenta una alternativa para transformar el problema de optimización cuadrática en un problema lineal complementario. Por otra parte en [34] se puede ver los resultados de aplicaciones industriales en un molino de cemento y una torre de secado por

pulverización, casos de estudio que demuestran la efectividad y la fácil utilización del GPC.

2.1.3 Control predictivo funcional (PFC).

Esta metodología de control predictivo fue propuesta por Richalet y ADERSA [35], utiliza un modelo en espacio de estados para representar las características del proceso (véase las ecuaciones 2.11 y 2.12). En [25, pp. 43-50] y [23] se puede ver detalles de la formulación y las principales características de esta variante de control predictivo. En este epígrafe se toma como base las referencias citadas para presentar una síntesis de la formulación.

$$x_m(t) = Mx_m(t-1) + Nu(t-1) \quad (2.11)$$

$$y_m(t) = Qx_m(t) \quad (2.12)$$

La señal de control se obtiene a través de la combinación lineal de funciones base B_i seleccionadas acorde a la naturaleza del proceso y la referencia deseada (véase la ecuación 2.13).

$$u(t+k) = \sum_{u=1}^{nB} u_i(t)B_i(k) \quad (2.13)$$

Normalmente estas funciones son de tipo polinomial:

- $B_1(k) = 1$ para escalón.
- $B_2(k) = k$ para rampa.
- $B_3(k) = k^2$ para parábola.

Todas las referencias pueden ser expresadas como una combinación de estas funciones. La selección de las mismas define el perfil de entrada y puede asegurar una característica determinada. La función de costo a ser minimizada se muestra en la ecuación 2.14.

$$J = \sum_{j=1}^{nH} [\hat{y}(t+j) - \omega(t+j)]^2 \quad (2.14)$$

La trayectoria de referencia $w(t+j)$ no necesariamente debe coincidir con la trayectoria real $r(t+j)$, por lo general es una aproximación de primer orden como la que se muestra en la ecuación 2.15.

$$\omega(t+j) = r(t+j) - \alpha^k [r(t) - y(t)] \quad (2.15)$$

En la ecuación 2.15 el parámetro k es un parámetro de aproximación y puede tomar valores entre cero y uno. Por otra parte los errores de predicción no son considerados a lo largo de todo el horizonte, estos se consideran solo en ciertos puntos a los que se les conoce como puntos de coincidencia. Estos puntos pueden ser considerados como parámetros de

sintonía y deben ser seleccionados tomando en cuenta la influencia de la estabilidad y la robustez del sistema de control. El número de puntos de coincidencia debe ser igual al número de funciones base Bi .

En [36] se puede ver detalles sobre la incorporación de restricciones al problema de control predictivo funcional. Adicionalmente se presenta una síntesis sobre los principios básicos de esta técnica. En lo que respecta a aplicaciones, actualmente se puede encontrar muchos reportes de implementación del PFC en la industria. En [37] se puede ver un reporte de tres aplicaciones industriales distintas: control de espesor en un molino de laminación en frío, control de nivel de agua en una presa y control de posición en un robot.

2.2 Control predictivo en la industria.

Desde hace varias décadas el control predictivo ha venido llamando la atención de científicos e investigadores de todo el mundo, es por ello que hoy en día se puede contar con una gran variedad de controladores no sólo a nivel de investigación científica, pues muchos de estos han sido patentados y actualmente están funcionando bajo el nombre de una determinada marca comercial. Existe un gran número de nombres que denotan las variantes comerciales de plataformas para control predictivo tanto para sistemas lineales como para sistemas no lineales (ver [4] y [5]). A continuación se

describen algunas variantes comerciales de control predictivo utilizadas en la industria sin que el orden implique el grado de desempeño y popularidad.

2.2.1 cpmPlus Expert Optimizer®.

Esta plataforma flexible desarrollada por ABB [38] soporta muchas tecnologías de control, incluyendo el control predictivo basado en modelo. Utiliza una programación basada en diagramas de bloques e interconexiones físicas. Entre sus amplias herramientas destacan aquellas que sirven para implementar sensores virtuales y aplicaciones con inteligencia artificial. Esta no es una plataforma exclusiva para control predictivo, sin embargo brinda la opción de controlar procesos mediante esta técnica. La conexión entre la planta y la plataforma se realiza mediante OPC (OLE para control de procesos), adicionalmente brinda la opción de estimar estados del proceso mediante la técnica de filtro de Kalman [39].

Según [40] esta plataforma utiliza el sistema "Mixed Logical Dynamical" (MLD) para la creación del modelo matemático [41], es decir cuenta con las herramientas necesarias para crear modelos tanto lineales como no lineales y con variables booleanas. De igual manera la función de costo puede ser implementada de acuerdo a las necesidades del usuario. Esta plataforma es más comúnmente utilizada en procesos de fabricación de cemento [42] y plantas de flotación [43].

2.2.23dMPC™.

Esta plataforma de software desarrollada por ABB [44] permite la implementación de controladores predictivos multivariables altamente interactivos. Utiliza la técnica de espacio de estados para estimar el estado actual del proceso y predecir el comportamiento futuro, también es posible generar un conjunto de modelos para procesos no lineales que se pueden ir adaptando a distintos puntos de funcionamiento. La penalización se realiza mediante una función de coste cuadrática. Esta plataforma brinda la posibilidad de trabajar en modo "fuera de línea" mediante un simulador que se instala como un toolbox de MATLAB®. En el modo "en línea" se ejecuta el algoritmo de control y el interfaz lo realiza mediante un servidor OPC instalado en la misma PC con sistema operativo Windows NT 4 o Windows 95/98. A más del algoritmo de control se dispone de herramientas para crear señales de excitación para la identificación de sistemas.

El nombre 3dMPC se debe a sus 3 grados de libertad, esto quiere decir que las respuestas ante cambios en los puntos de ajuste, perturbaciones medidas y retroalimentación de control pueden ser sintonizadas de manera independiente. Más detalles sobre este producto se pueden encontrar en [45] y [46].

2.2.3 DMCplus[®].

Esta plataforma desarrollada por ASPENTECH [47] es otra variante comercial para implementar control predictivo. Esta herramienta de nueva generación apareció en el mercado tras la fusión de Dynamic Matrix Control Corporation y Setpoint Inc., permite trabajar con modelos en espacio de estados tanto para sistemas lineales como para sistemas no lineales. También brinda la posibilidad de realizar identificación mediante la técnica de sub-espacios. Según [48] esta plataforma integra un sistema off-line para diseño y simulación de sistemas y obviamente un sistema on-line para implementación. La herramienta de simulación y diseño es compatible con los estándares Microsoft[®] (Windows 95[®] y Windows NT[®]), por otra parte la herramienta de implementación en línea es compatible con plataformas DEC[®], IBM[®] y UNIX[®]. Esta plataforma está siendo ampliamente utilizada en la industria del petróleo. En [49], [50] y [51] se dejan ver breves reportes de su implementación y evaluación.

En [6, pp. 306-307] se explica el funcionamiento de esta plataforma para el cálculo de la ley de control, lo cual se realiza en dos pasos. En el primer paso se calcula un conjunto de puntos de ajuste y cada uno de éstos corresponde a una variable controlada. Estos puntos de ajuste son calculados sobre una referencia económica o en base a la calidad del producto. Se utiliza una función de costo lineal y se imponen las restricciones en forma de

inecuaciones lineales tanto para la entrada como para la salida. El problema es resuelto a través de programación lineal (LP). En el segundo paso se calcula una trayectoria de control futura sobre un horizonte de control, que es predicha para llevar a la planta a los niveles de ajuste calculados por el optimizador LP.

2.2.4 Profit[®] Controller

Esta plataforma desarrollada por Honeywell [52] brinda la posibilidad de implementar controladores predictivos para sistemas MIMO altamente interactivos, se basa en la Tecnología de Control Predictivo Multivariable Robusto RMPCT (Robust Multivariable Predictive Control Technology[®]) patentada por la misma Honeywell. Esta tecnología tiene como base las características de un MPC tradicional pero adiciona un Algoritmo de Control de Rango RCA (*Range Control Algorithm*) también patentado por Honeywell para proporcionar características de robustez. “RCA minimiza el efecto de incertidumbre del modelo mientras determina los movimientos más pequeños requeridos para cumplir los objetivos de control y optimización de manera simultánea” [53]. Esta herramienta permite utilizar funciones de costo lineales y cuadráticas para brindar mayor flexibilidad al usuario. Al igual que DMCplus[®] esta plataforma está organizada en dos niveles, uno para la optimización y otro para el control. Más detalles del funcionamiento de ésta plataforma se puede encontrar en [6, pp. 308-309].

En modo fuera de línea Profit[®] Controller funciona bajo sistema operativo Windows XP Professional. El fabricante recomienda utilizar un procesador Intel Xeon X5560 o equivalente y memoria RAM de 4 GB. La aplicación en línea puede correr sobre servidores Experion, módulos de aplicación TPS Honeywell o plataformas abiertas DCS bajo Windows XP, Windows 2003 Server y Windows 2008 Server. Más detalles sobre ésta plataforma se pueden encontrar en [54]. En [55] se puede encontrar detalles del algoritmo RMPCT y resultados de una implementación en una unidad de digestión de alumina. En [56] se muestra el resultado de haber utilizado Profit[®] Controller sobre una planta petroquímica de hidrocarburos aromáticos o fraccionador BTX (Benzeno, Tolueno y Xileno). Por otra parte [57] presenta un estudio comparativo de un controlador convencional PID con un controlador predictivo implementado en un cristalizador de vacío utilizando Profit[®] Controller.

2.2.5 PFC[®]

Esta plataforma desarrollada por Richalet y ADERSA [35], utiliza la metodología de control predictivo funcional que se describe en la sección 2.1.3. Hoy en día surge como la segunda generación de controladores predictivos y su nicho radica en procesos no lineales, inestables y de dinámica lenta, tales como los reactores químicos.

Según [58] PFC[®] puede ser implementada sobre sistemas de control distribuido y sobre autómatas programables. En [59] se puede ver detalles de la implementación de esta metodología sobre un intercambiador de calor.

2.2.6 Process Perfect[®]

Esta plataforma desarrollada por Pavilion Technologies Inc., se considera como la primera plataforma de control predictivo para sistemas no lineales que salió al mercado. Permite implementar una estrategia de control predictivo sobre sistemas multivariables y utiliza modelos basados en redes neuronales. En la actualidad está disponible la versión 8 del software que permite el control de sistemas lineales y no lineales.

2.3 Herramientas de simulación para control predictivo.

2.3.1 Model Predictive Control Toolbox[™] (MPCTool)

MPCTool de MATLAB[®] proporciona las herramientas necesarias para el análisis, diseño y sintonización de controladores predictivos que pueden ejecutarse sobre sistemas MIMO con o sin restricciones. El diseño y la simulación pueden realizarse utilizando funciones de MATLAB[®] o bloques en Simulink[®]. Las principales características de esta herramienta se detallan a continuación [60]:

- Personalización de restricciones y pesos con herramientas de asesoramiento para mejorar el rendimiento y la robustez del controlador.
- Solucionador QP.
- Posibilidad de representar modelos diferentes tanto para la planta como para el controlador.
- Posibilidad de manejar diversos escenarios de sintonía y simulación.
- Soporte para la generación de código C con Simulink Coder TM.

Para iniciar con la herramienta es necesario definir dos modelos, uno que represente a la planta controlada y otro que atienda la esencia del controlador predictivo basado en modelo, al que se le denominará modelo para el controlador o modelo de predicción. Estos dos modelos no necesariamente tienen que ser iguales aunque pueden serlo, de esto dependerá en cierto modo el desempeño del controlador. Los modelos pueden ser definidos directamente en el espacio de trabajo de MATLAB[®].

Para objeto de demostración se realizará todo el procedimiento de sintonía utilizando el ejemplo 2.7 de Maciejowski [6, p. 64] y se asumirá que el modelo de la planta es igual al modelo del controlador. Para este problema se cuenta con un modelo linealizado de una aeronave 'Cessna Citation 500', cuando circula a una altitud de 5000m y una velocidad de 128,2m/s. El ángulo de

elevación (rad) es la única entrada, y el ángulo de cabeceo (rad), la altura (m), y la tasa de altitud (m/s) son las salidas. El modelo discretizado con $T_s = 0.5s$ se ilustra a través de las expresiones 2.16, 2.17 y 2.18:

$$A_d = \begin{bmatrix} 0.237 & 0 & 0.178 & 0 \\ -0.376 & 1 & 0.269 & 0 \\ -0.998 & 0 & 0.136 & 0 \\ -48.95 & 64.10 & 2.398 & 1 \end{bmatrix} \quad (2.16)$$

$$B_d = \begin{bmatrix} -0.3 \\ 0 \\ -17 \\ 0 \end{bmatrix} \quad (2.17)$$

$$C_d = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -128.2 & 128.2 & 0 & 0 \end{bmatrix} \quad (2.18)$$

Una vez que se ha definido los modelos de planta y controlador se inicia con la sintonización del controlador, para ello basta con escribir en el espacio de trabajo de MATLAB® el comando “mpctool” y aparecerá la ventana con el nombre “Herramienta de Control y Estimación” (HCE) tal como se muestra en la figura 2.1.

A continuación es necesario importar el modelo de planta que se va a controlar. Es importante saber diferenciar entre modelo de planta y modelo de predicción. El modelo de planta hace referencia al modelo matemático de la planta física que va a ser controlada, por otra parte el modelo de predicción es el modelo que utiliza el controlador para hacer las predicciones.

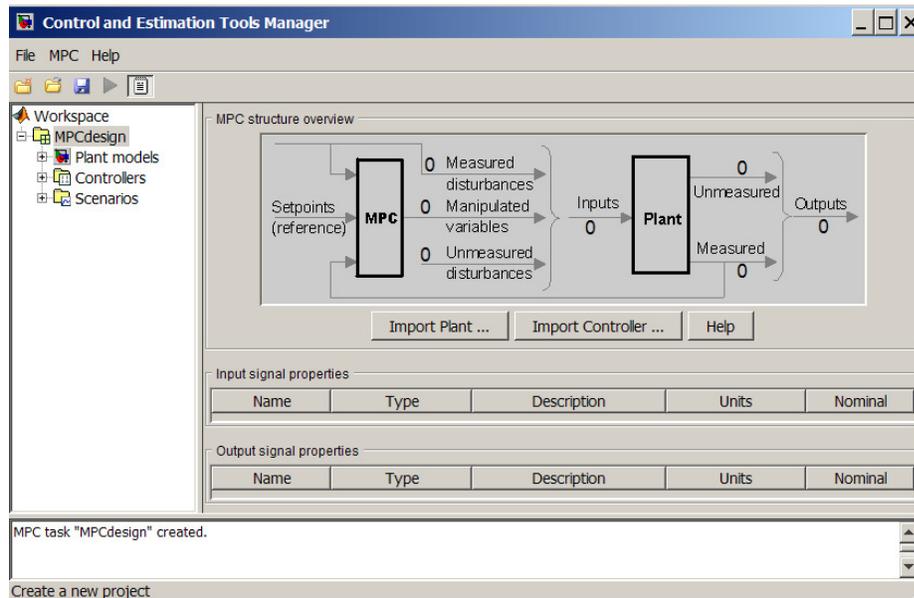


Figura 2.1 Vista inicial de la herramienta de control y estimación (MPCTool)

Para ingresar un modelo se presiona **Importar Planta** y se selecciona entre las dos opciones que brinda el cuadro de diálogo **Importar Modelo de Planta**. Una vez importado el modelo las propiedades de las señales de entrada y salida pueden ser modificadas con los nombres de las variables para evitar confusiones, además se puede dar una descripción y cambiar las unidades de ingeniería. A la izquierda de la ventana principal se distingue un árbol de proyecto con los campos: modelo de planta, controladores y escenarios (véase la figura 2.1).

Al ubicarse sobre **Modelos de Planta** la ventana de diálogo cambia para mostrar información relacionada a la planta previamente importada. Esta

instancia de la HCE deja ver las plantas importadas para el proyecto y los detalles de las mismas. Entre los detalles más significativos constan el tipo de modelo (espacio de estados), el número de entradas (1), el número de salidas (3), el orden (4) y el tiempo de muestreo (0,5s).

Al ubicarse sobre **Controladores** la ventana de diálogo toma un nuevo aspecto. Aquí se puede ver detalles de los controladores utilizados en el proyecto. Al desplegar el árbol de **Controladores** se despliegan los existentes en el proyecto, para este ejemplo demostrativo se visualizará únicamente MPC1. Al posicionarse sobre este se tiene acceso a las características fundamentales para el diseño del mismo. Ahora el cuadro de diálogo toma un nuevo aspecto y presenta las opciones: Modelo y Horizontes, Restricciones, Sintonización de Pesos y Estimación.

En la opción **Modelo y Horizontes** se ilustra el nombre del modelo LTI utilizado para las predicciones. Además existen dos campos importantes: horizonte y bloques. En el horizonte el usuario configura tres parámetros de sintonía del controlador: intervalo de control, horizonte de predicción y horizonte de control. Por otra parte la opción bloques permite actuar sobre los movimientos del controlador, aquí se indica el número de movimientos y la instancia de suceso sobre el horizonte de predicción. Por defecto la opción de bloques viene desactivada. Para el ejemplo demostrativo se va a

considerar inicialmente un intervalo de control de 0.5, un horizonte de predicción de 10 y un horizonte de control de 4.

En la opción **Restricciones** se puede diferenciar dos secciones por separado que permiten personalizar las restricciones de las variables manipuladas y las variables de salida. Para el ejemplo previamente señalado de manera inicial se va asumir que el sistema no tiene restricciones tanto para la entrada como para la salida.

En la opción **Sintonización de Pesos** se puede especificar los pesos que intervienen en la función de coste cuadrática. Esto se analizará con mayor profundidad en la formulación predictiva presentada en el capítulo 4. Esta ventana permite configurar los pesos de entrada y salida del controlador.

Los pesos de salida son factores que afectan la desviación de cada salida predicha con respecto a su referencia. En la HCE por defecto los pesos de salida están configurados con un valor de 1 para todas las salidas. Para el ejemplo demostrativo se considera los pesos por defecto. Por otra parte los pesos de entrada son factores que afectan los ajustes predichos de la variable de control. En la herramienta de control y estimación este parámetro se le reconoce como **Tasa de Peso** y obviamente se encuentra en la sección

de configuración de pesos de entrada. Para el ejemplo demostrativo se considera la tasa de peso por defecto que es de 0.1.

Los pesos de entrada y salida aparecen en la función de coste a minimizar (véase la ecuación 2.9). En el entorno visual de la HCE se le reconoce como **Peso** y obviamente está dentro de la sección de configuración de pesos de entrada. Inicialmente para el ejemplo de demostración se va a asignar el valor por defecto para la única entrada del sistema. Este valor por defecto es igual a 0.

A más de las configuraciones de pesos de entrada y salida existe la sección **General** que brinda la posibilidad de configurar el grado de robustez de las variables controladas a costa de su velocidad de respuesta. Para una configuración óptima se deberá encontrar un equilibrio entre estos dos parámetros.

La opción **Estimación** permite adicionar perturbaciones al proceso de manera que el controlador se adapte y produzca un alto rechazo a las mismas. La opción de estimación brinda tres escenarios de configuración: **Perturbaciones de Salida** que es la opción presentada por defecto, **Perturbaciones de Entrada** y **Ruido en la medición**. Todas las opciones guardan el mismo aspecto de visualización y permiten la configuración de

perturbaciones y ruido en la medición ya sea señal-por-señal o a través de un modelo importado desde el espacio de trabajo de MATLAB®.

Las perturbaciones de entrada y salida pueden ser de tres tipos:

- Paso: simula pasos aleatorios como perturbaciones, se le conoce también como ruido blanco integrado.
- Rampa: simula rampas aleatorias como perturbaciones, se le conoce también como ruido blanco doblemente integrado.
- Blanco: simula ruido blanco como perturbación.

Mayores detalles sobre ruido blanco integrado y ruido blanco doblemente integrado se pueden encontrar en [61]. Para el ejemplo demostrativo se asume perturbaciones tipo paso con una magnitud de 1 para las salidas de ángulo de cabeceo y tasa de altitud, esto quiere decir que se utiliza ruido blanco integrado con una desviación estándar de 1. Para la altura se asume perturbaciones tipo ruido blanco con una magnitud de 1. Además se omiten las perturbaciones de entrada.

Por otra parte el ruido en la medición puede ser configurado como Paso o Blanco. Para el ejemplo demostrativo se asume ruido blanco con magnitud 1 superpuesto en la medición de las tres salidas.

En lo que respecta a la sección **Ganancia de estimación general** según [62] el deslizador determina la agresividad del controlador ante los cambios inesperados en la salida. Para el ejemplo en cuestión se va considerar el valor por defecto.

Por último, la opción **Escenarios** permite las siguientes opciones:

1. Configuraciones de simulación:

- Controlador: Permite la selección de cualquiera de los controladores que se hayan diseñado.
- Planta: Permite la selección del modelo físico que se quiere controlar.
- Duración: Permite configurar el tiempo que va a durar la simulación.
- Lazo cerrado: Cuando esta opción está deshabilitada la simulación correrá en lazo abierto.
- Aplicación de restricciones: Cuando esta opción está deshabilitada las restricciones del controlador son ignoradas.

2. Puntos de ajuste:

Esta sección permite configurar las consignas de las variables controladas. Las consignas pueden ser de tipo constante, escalón, rampa, seno, pulso y

gaussiana. La ocurrencia puede ser especificada en cualquier instante de la simulación.

3. Perturbaciones no medidas:

Esta sección permite especificar detalles de las perturbaciones no medibles, estas perturbaciones pueden ser causadas tanto en las variables manipuladas como en las salidas medidas y pueden ser de tipo constante, escalón, rampa, seno, pulso y gaussiana. La ocurrencia puede ser especificada en cualquier instante de la simulación.

Para el ejemplo demostrativo se considera todas las configuraciones por defecto, únicamente se proporciona una consigna a la altura del avión (40m). En la parte inferior de la ventana escenarios aparece la opción **Simulación**, al presionar sobre este botón se ejecutará la simulación considerando las configuraciones previamente proporcionadas en todos los ambientes visuales de la Herramienta de Control y Estimación. La figura 2.2 a) presenta la evolución temporal de las tres variables de salida. Por otra parte la figura 2.2 b) muestra la acción de control. Cabe recordar que para este ejemplo de simulación el horizonte de predicción es igual a 10, el horizonte de control es igual a 4 y no se han considerado restricciones.

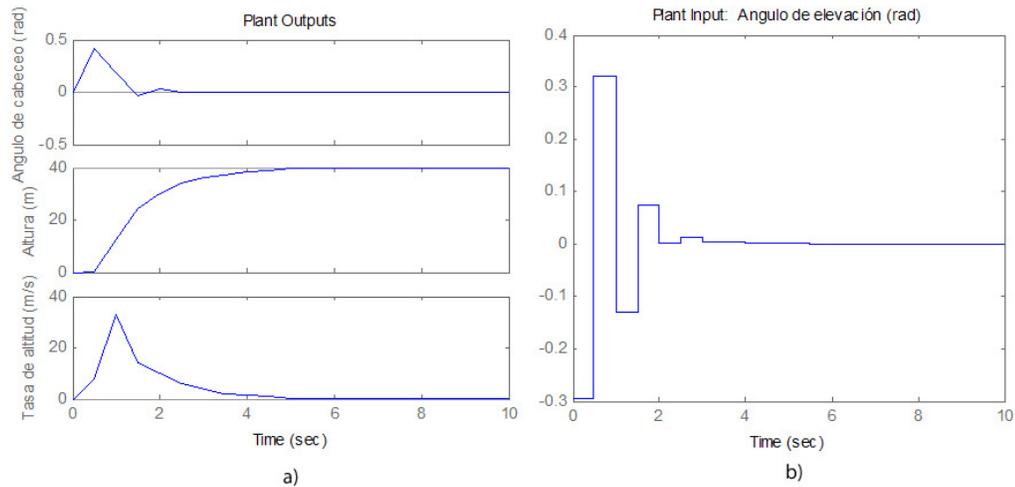


Figura 2.2 Simulación de vuelo de la aeronave ‘Cessna Citation 500’ para una consigna de 40m en la altura del avión.

En la figura 2.2 se puede observar claramente como la altitud llega a la referencia deseada, cabe indicar que este valor está por encima del punto de trabajo al que se linealizó la planta. También se puede observar que la acción de control tiene efecto sobre las otras salidas sin embargo luego de que se extingue el transitorio las variables regresan al valor de cero.

En la ventana de escenarios, en la parte inferior se encuentra a disposición del usuario el **Tutor de Sintonización**. Esta herramienta proporciona una guía para la sintonización de pesos. Con esta herramienta se puede calcular la sensibilidad de la función de coste para pesos individuales y ver si estos deben incrementar o decrementar para mejorar el desempeño del controlador. Mayor información sobre el uso de esta herramienta se puede encontrar en [63].

Una vez que se ha comprendido todas las opciones que brinda la herramienta de control y estimación es posible jugar con cada una de estas para verificar la potencialidad del control predictivo, por ejemplo se puede agregar restricciones a las variables del proceso, modificar los pesos, adicionar perturbaciones, cambiar los horizontes de predicción y control, etc.

La figura 2.3 presenta el comportamiento del sistema cuando se ha añadido una restricción a la tasa de altitud (valor mínimo igual a 0 y valor máximo igual a 8). En 2.3 a) se puede observar claramente como la salida llega a la referencia deseada sin embargo la acción de control (véase 2.3 b)) se presenta modificada debido a la presencia de la restricción. Además se puede notar como la tasa de altitud no sobrepasa de la restricción a diferencia de lo presentado en la figura 2.2.

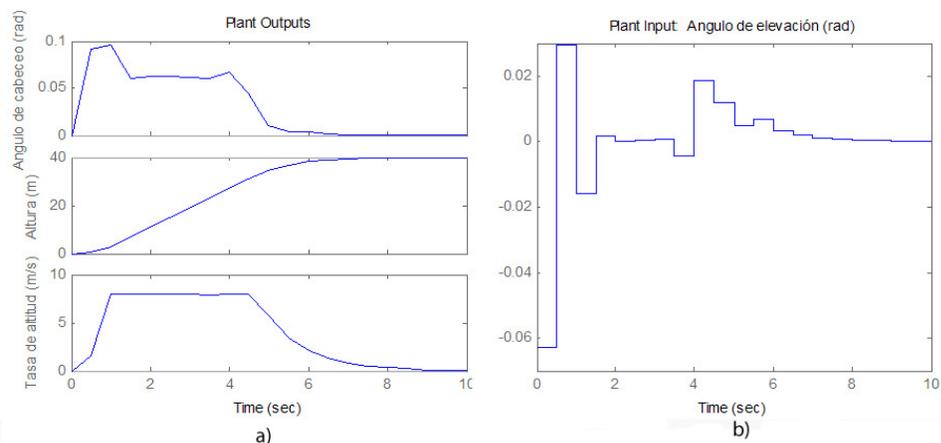


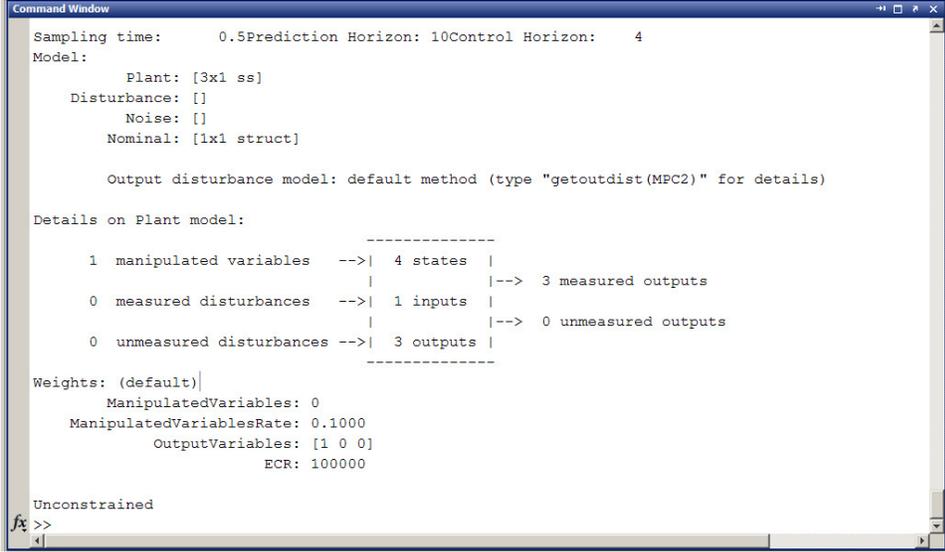
Figura 2.3 Simulación de vuelo de la aeronave 'Cessna Citation 500' con restricción en la tasa de altitud

2.3.2 MPCTool desde MATLAB®.

La herramienta de Control Predictivo basado en Modelo (MPCTool) está provista de una serie de comandos de MATLAB® que permiten la creación, diseño y simulación de controladores predictivos para sistemas lineales y no lineales. Una manera sencilla de crear un controlador es a través del siguiente comando:

$$MPC2 = mpc(sys, Ts, Hp, Hc)$$

Donde $MPC2$ es una estructura que contiene los parámetros del controlador, sys es un sistema definido en espacio de estados, Ts es el tiempo de muestreo, Hp es el horizonte de predicción y Hc es el horizonte de control.



```

Command Window
Sampling time:      0.5 Prediction Horizon: 10 Control Horizon: 4
Model:
  Plant: [3x1 ss]
  Disturbance: []
  Noise: []
  Nominal: [1x1 struct]

  Output disturbance model: default method (type "getoutdist(MPC2)" for details)

Details on Plant model:
-----
1 manipulated variables -->| 4 states |
|                               |--> 3 measured outputs
0 measured disturbances -->| 1 inputs |
|                               |--> 0 unmeasured outputs
0 unmeasured disturbances -->| 3 outputs |
-----

Weights: (default)
  ManipulatedVariables: 0
  ManipulatedVariablesRate: 0.1000
  OutputVariables: [1 0 0]
  ECR: 100000

Unconstrained
fx >>
  
```

Figura 2.4 Detalles del controlador predictivo creado con el comando mpc de MATLAB®

En la figura 2.4 se puede ver el resultado que presenta MATLAB® ante la ejecución de este comando. Aquí se pueden ver detalles como el tiempo de muestreo, horizonte de predicción, horizonte de control, orden del sistema, número de variables manipuladas, número de variables controladas, pesos, restricciones, entre otras. Para la ejecución del comando *mpc* se ha utilizado el modelo continuo de la aeronave 'Cessna Citation 500' presentado en la subsección 1.3.1.

Una vez creada la estructura MPC2 se puede adicionar restricciones y modificar los pesos asignados por defecto. Para modificar los pesos se puede crear una estructura que contenga los pesos de entrada y salida. Se debe tener en cuenta el número de entradas y salidas para formular las matrices de pesos, en este caso se cuenta con una variable manipulada y tres variables controladas:

```
Weights = struct('Input'; [0]; 'InputRate'; [0]; 'Output'; [10 10 10])
```

```
MPC2.Weights = Weights
```

De igual manera se pueden crear estructuras tanto para las variables manipuladas como para las variables de salida. Para el controlador en cuestión se colocan las siguientes restricciones:

```
MV(1) = struct('Min', -0.05, 'Max', 0.05);
```

```
OV(1) = struct('Min', -0.5, 'Max', 0.5);
```

```
OV(3) = struct('Min', 0, 'Max', 12);
```

```
MPC2.MV = MV
```

```
MPC2.OV = OV
```

Para simular el desempeño del controlador se puede utilizar el siguiente comando:

```
[y, t, u] = sim(MPC2, Tsimul, r);
```

Donde *Tsimul* es el tiempo de simulación en segundos ($Tsimul = 40 * Ts$), *r* es un vector de valores constantes que contiene las referencias del sistema ($r = [0 \ 40 \ 0]$), *y* es una matriz que contiene los valores de las variables de salida, su dimensión depende el tiempo de simulación y del número de salidas, *t* es el vector de tiempo y *u* es una matriz que contiene las acciones de control.

Una vez ejecutado el comando se puede graficar las salidas del sistema y la acción de control (véase la figura 2.5 a) y b) respectivamente). Claramente se puede observar como la altitud del avión alcanza la referencia respetando las restricciones establecidas. En [62] se puede obtener más información sobre los comandos relacionados con MPC, además se pueden encontrar ejemplos para que el usuario se familiarice con el uso de este toolbox.

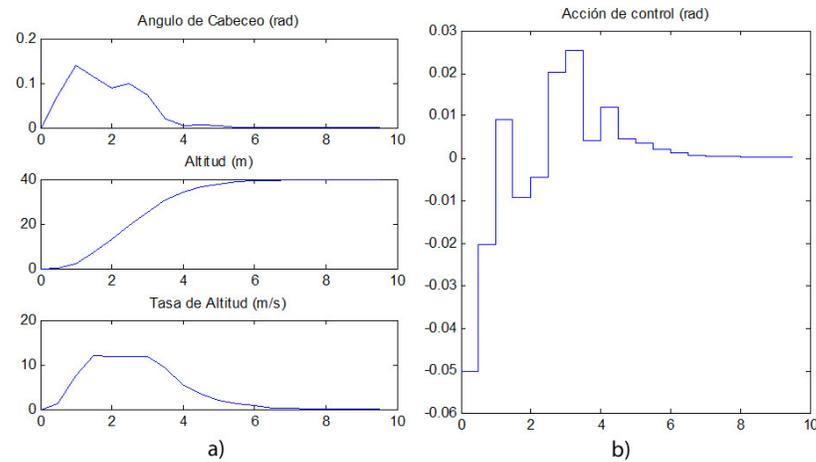


Figura 2.5 Simulación de vuelo de la aeronave ‘Cessna Citation 500’ ejecutada desde MATLAB®.

2.3.3 MPCTool desde Simulink®.

La herramienta de Control Predictivo basado en Modelo (MPCTool) está provista de un bloque en Simulink® para la implementación y simulación de un proceso que puede partir de un objeto MPC previamente diseñado utilizando los comandos de MATLAB®. La ventaja de utilizar Simulink® radica en el hecho de que la simulación puede ser sometida a situaciones más elaboradas utilizando las librerías disponibles. Por ejemplo se puede crear señales personalizadas de entrada y para perturbaciones, por otra parte la arquitectura de diseño basada en la interconexión de bloques brinda mayor flexibilidad para evaluar la sintonía del controlador. Para acceder a las opciones de MPC se crea un ambiente de simulación nuevo y en el **Servidor**

de **Librerías de Simulink®** se accede a la librería **Model Predictive Control Toolbox** (véase la figura 2.6). Esta librería contiene dos bloques uno que permite la creación y edición de un solo controlador y otro que permite la creación y edición de múltiples controladores.

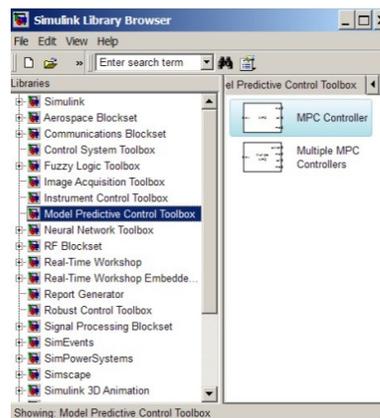


Figura 2.6 Bloques MPC desde Simulink®

La estructura de un bloque MPC se puede observar en el diagrama de la figura 2.7. Por defecto el bloque presenta tres conexiones de entrada y una conexión de salida:

- *ref*: es el punto de ajuste o valor objetivo para la variable controlada.
- *mo*: es la medición de la salida de la planta.
- *md*: para proporcionar las perturbaciones medidas.
- *mv*: es la señal de control.

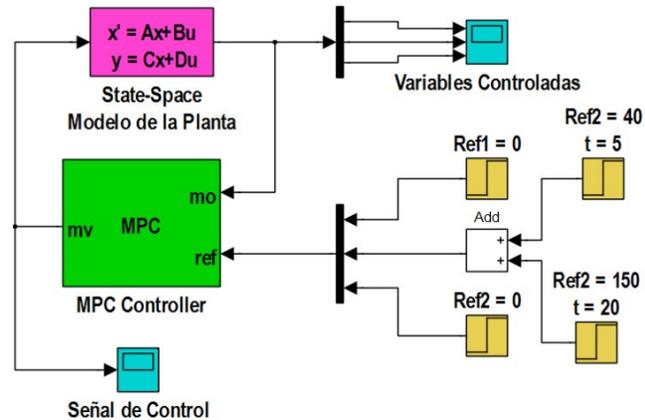


Figura 2.7 Bloque MPC para la creación de un sistema de control en Simulink®

Si el sistema es de múltiples entradas y múltiples salidas se utilizan multiplexores de señal. Para el diagrama de la figura 2.7 se ha utilizado un modelo de planta igual al modelo de predicción, a través de una descripción en espacio de estados. Aprovechando la flexibilidad que brinda Simulink® se proporciona una referencia a la altura del avión de 40m a los cinco segundos de haber iniciado la simulación, posteriormente a los veinte segundos la referencia cambia a 150m. Las referencias para el ángulo de cabeceo y la tasa de altitud permanecen en cero.

Al dar doble clic sobre el bloque de controlador se puede acceder a los parámetros de configuración. En el campo **Controlador MPC** se debe ingresar un objeto previamente creado, en este caso se ingresa MPC2 que

fue el controlador creado en MATLAB® y que se presentó en la sección 1.3.2. Al pulsar sobre **Diseño** se puede editar un controlador existente invocando a la **Herramienta de Control y Estimación** estudiada en la sección 2.3.1. Al finalizar la edición es necesario exportar el controlador al espacio de trabajo de MATLAB® y sobrescribir el existente. En el campo **Condiciones iniciales de estado** se debe ingresar un vector con los estados iniciales del sistema, este vector puede ser definido desde MATLAB®: $xini = zeros(4,1)$.

En la figura 2.8 a) se puede ver las salidas controladas del sistema para un tiempo de simulación de 50 segundos. Aquí claramente se puede observar como la altitud del avión llega a la consigna deseada, sin embargo el acoplamiento del sistema multivariable produce cambios en las otras dos salidas controladas. Esta situación puede ser mejorada manipulando los parámetros de sintonía del controlador. En la figura 2.8 b) se puede observar la acción de control necesaria para llevar el sistema a las referencias deseadas.

En [62] se puede encontrar mayor información sobre la utilización de MPC desde Simulink®, de igual manera la instalación de MATLAB® R2010a con MPCTool incorpora muchos ejemplos de simulación que reposan en el directorio: `c:\Archivos de programa\MATLAB\toolbox\mpc`. Aquí se puede

encontrar cómo implementar un bloque MPC para controlar una planta no lineal.

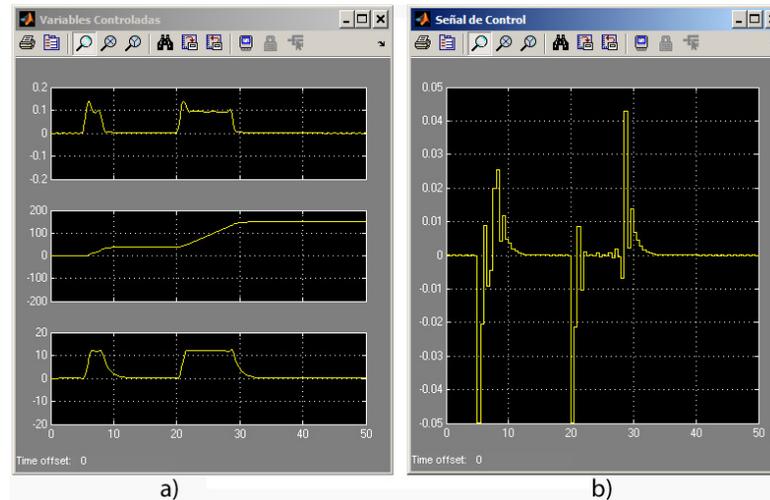


Figura 2.8 Simulación de vuelo de la aeronave ‘Cessna Citation 500’ utilizando Simulink®.

2.3.4 Otras herramientas de simulación.

Sin duda alguna MPCTool de MATLAB® es la herramienta más poderosa y flexible con la que se cuenta hoy en día para la simulación de controladores predictivos, sin embargo, contar con una licencia de MATLAB® puede resultar algo costoso.

Sin contar las herramientas comerciales que se analizaron en la sección 2.2 que brindan la posibilidad de simular fuera de línea el comportamiento de un controlador predictivo, en la actualidad existen simuladores que permiten

probar el desempeño de un controlador predictivo. A continuación se analizan algunos de estos:

- MIMO-GPCIT: Esta herramienta interactiva y de fácil uso puede obtenerse de forma gratuita y no requiere la instalación de ningún paquete informático. Según sus creadores el objetivo de MIMO-GPCIT es facilitar la comprensión del GPC para sistemas multivariables de forma interactiva. En el sitio web (véase [64]) se puede descargar la aplicación para Windows, Mac y Linux así como la documentación de ayuda. Por otra parte en [65] se puede ver el resultado de algunas aplicaciones y un análisis comparativo entre MPCTool de MATLAB® y MIMO-GPCIT.
- MCPC es una herramienta desarrollada en C++ que permite la simulación de controladores predictivos multivariables con restricciones. Fue creada por la universidad de Twente y no está disponible para descargarla de manera gratuita. En [66] se puede ver información detallada sobre su construcción y uso.

A manera de conclusiones finales para el presente capítulo se puede decir que las plataformas comerciales de control predictivo basado en modelo tratan de dar al usuario las comodidades necesarias para hacer uso de la

técnica sin la necesidad de acudir a otras plataformas. Por lo general están vinculadas con otras herramientas como de identificación de sistemas, estimación de estados, linealización, reducción de orden, que en mayor o menor grado pueden ser requeridas por el algoritmo de control. En cuanto a las herramientas de simulación, la investigación deja ver que MPCTool es una herramienta muy poderosa para la sintonía y simulación de controladores predictivos multivariables con restricciones, sin embargo requiere la instalación de los paquetes comerciales MATLAB® y Simulink®. Quizá esta limitante ha hecho que grupos de investigación creen plataformas como MIMO-GPCIT y MCPC que corren sobre ejecutables que no requieren la instalación de paquetes de software, sin embargo de lo que se ha podido ver sobre todo de MIMO-GPCIT que se puede descargar de manera libre no cuenta con la versatilidad y la disponibilidad que brinda MATLAB®, sobre todo en el hecho de que no existe la posibilidad de ingresar nuevos modelos y es muy cerrada en cuanto a detalles del algoritmo, diseño del controlador y forma de interacción entre la planta y el controlador.

CAPÍTULO 3

CONTROL PREDICTIVO BASADO EN MODELO (MPC).

En los capítulos 1 y 2 se han presentado algunos aspectos sobre el fundamento de MPC sin la intención de dar una definición formal de los mismos. Estos aspectos de cierto modo han aportado significativamente al conocimiento de esta técnica y han dejado ver de una manera muy general el potencial de la misma. En este capítulo se presenta una definición formal del fundamento teórico y matemático del MPC como un preámbulo a la implementación de la plataforma de simulación SIMUMPC. El análisis presentado se basa en la formulación en espacio de estados propuesta por Wang [67] y Maciejowski [6]. En primera instancia se presenta la fundamentación de esta técnica como un problema SISO sin restricciones, posteriormente se extiende el problema al caso multivariable. A continuación se presenta el marco teórico y matemático para la estimación de estados, aspecto muy importante en la formulación MPC planteada. Por último se

presenta una formulación con restricciones y una propuesta de solución numérica utilizando programación cuadrática.

3.1 Introducción.

En todas las variantes de MPC el objetivo de la técnica converge en el hallazgo de una trayectoria futura de la variable manipulada u para optimizar la salida de la planta y . Este proceso trae implícito una serie de definiciones que se enmarcan dentro de los aspectos característicos del control predictivo basado en modelo. En términos de Camacho y Bordons [25] una explicación estricta de la estrategia MPC se detalla a continuación:

1. **Utilizando un modelo de predicción para cada instante t se predicen las salidas futuras para un intervalo de tiempo determinado que se le conoce como horizonte de predicción. Las predicciones $\hat{y}(t+k|t)$ para $k = 1 \dots N$ dependen de los valores conocidos hasta el instante t (entradas y salidas pasadas) y de las señales de control futuras $u(t+k|t)$ para $k = 0 \dots N - 1$ que se pretenden mandar a la planta. Idea de control predictivo.**
2. **Las señales de control futuras se calculan mediante la optimización de un determinado criterio de manera que el proceso se mantenga lo más cerca posible de la trayectoria de referencia $w(t + k)$. Este criterio es una función cuadrática de los errores entre la salida predicha y la trayectoria de referencia. Adicionalmente se suele incluir el esfuerzo de control. Si el modelo es lineal y no existen restricciones el problema puede ser resuelto de manera analítica, caso contrario se deberá usar un método iterativo de optimización.**

3. La señal de control $u(t|t)$ es enviada al proceso mientras que las restantes son desechadas. Los valores se actualizan y se vuelve a repetir el proceso.

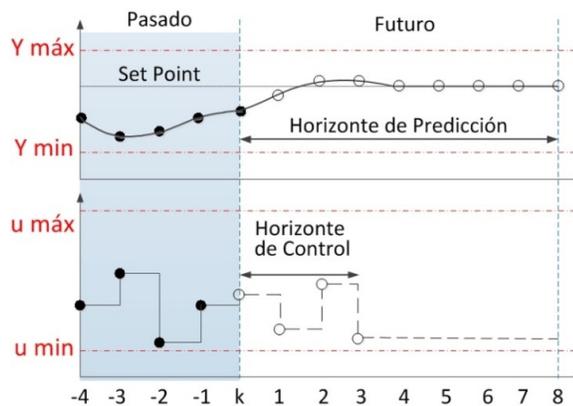


Figura 3.1 Idea de control predictivo basado en modelo.

La expresión $t+k|t$ denota valores futuros de la variable correspondiente basados en las mediciones disponibles en el instante t . La estrategia de control predictivo y la idea de algunos de los términos involucrados se ilustran en la figura 3.1. Aquí el horizonte de predicción N es igual a 8 y el proceso de optimización sobre este intervalo calcula M movimientos. A este parámetro se le conoce como horizonte de control $M(\geq 1, \leq N)$, para el caso de la figura 3.1 M es igual a 3. En esta figura también se puede entender el concepto de estrategia de bloques abordada en la sección 2.3.1, para este caso el controlador realiza 3 movimientos al inicio del horizonte de predicción y después permanece constante.

3.2 Fundamentación del control predictivo basado en modelo.

3.2.1 Formulación predictiva.

Como se pudo ver en el capítulo 2 existen algunas variantes de control predictivo que se diferencian básicamente en el modelo de predicción que se utilice. En el presente epígrafe se aborda el fundamento matemático del control predictivo utilizando un modelo en variables de estado (véase las expresiones 3.1 y 3.2).

$$x(k + 1) = Ax(k) + Bu(k) \quad (3.1)$$

$$y(k) = Cx(k) + Du(k) \quad (3.2)$$

Las expresiones 3.1 y 3.2 corresponde al modelo del proceso y su utilidad radica en la formulación de las predicciones. La matriz A representa la dinámica del sistema (matriz de estados), B corresponde a la matriz de entrada, C corresponde a la matriz de salida y D representa la matriz de transferencia directa ($D = 0$ si $u(k)$ no afecta a $y(k)$ en el mismo intervalo de tiempo), $x \in R^n$ es el vector de estados, $u \in R^m$ es el vector de entrada (control), $y \in R^p$ es el vector de salida.

Atendiendo a la formulación del control predictivo en espacio de estados, un conjunto de predicciones para un horizonte de predicción finito N puede ser representado por las expresiones 3.3 a 3.6.

$$x(1) = Ax(0) + Bu(0) \quad (3.3)$$

$$x(2) = Ax(1) + Bu(1) \quad (3.4)$$

$$x(3) = Ax(2) + Bu(2) \quad (3.5)$$

$$x(N) = Ax(N - 1) + Bu(N - 1) \quad (3.6)$$

Claramente se puede observar como el estado siguiente depende del anterior, lo cual indica que el conjunto de expresiones pueden ser rescritas en función de la condición de estado inicial $x(0)$, (véase las expresiones 3.7 a 3.10).

$$x(1) = Ax(0) + Bu(0) \quad (3.7)$$

$$x(2) = A^2x(0) + ABu(0) + Bu(1) \quad (3.8)$$

$$x(3) = A^3x(0) + A^2Bu(0) + ABu(1) + Bu(2) \quad (3.9)$$

$$x(N) = A^N x(0) + A^{N-1}Bu(0) + \dots + ABu(N - 2) + Bu(N - 1) \quad (3.10)$$

El conjunto de expresiones 3.7 a 3.10 deja ver como las predicciones de estado dependen no solamente del estado inicial, sino también del vector de acciones de control futuras $u(0), u(1) \dots u(N - 1)$. El algoritmo de control predictivo se encarga de encontrar este vector minimizando una función de coste J . Las predicciones de salida (véase las expresiones 3.11 a 3.14)

pueden ser obtenidas mediante el mismo mecanismo que se utilizó para las predicciones de estado.

$$y(1) = CAx(0) + CBu(0) \quad (3.11)$$

$$y(2) = CA^2x(0) + CBu(0) + CBu(1) \quad (3.12)$$

$$y(3) = CA^3x(0) + CA^2Bu(0) + CABu(1) + CBu(2) \quad (3.13)$$

$$y(N) = CA^Nx(0) + CA^{N-1}Bu(0) + \dots + CABu(N-2) + CBu(N-1) \quad (3.14)$$

Hasta el momento el preámbulo de la formulación predictiva indica un problema netamente matricial. Antes de continuar con el análisis es necesario hacer un alto para considerar dos aspectos fundamentales de la formulación. El primer aspecto hace referencia a la incorporación de un integrador para garantizar un error en estado estacionario nulo y el segundo aspecto hace un llamado a la generalización del problema predictivo considerando el concepto de horizonte deslizando.

Existen varias maneras de incluir un integrador en un modelo en espacio de estados. En [6, p. 49] se presenta dos alternativas que acertadamente pueden ser utilizadas en la formulación del control predictivo, una de las alternativas para lograr esto requiere la incorporación de un nuevo vector de estado en la formulación estándar de un sistema en espacio de estados

(véase las ecuaciones 3.15 y 3.16) esta opción también es presentada en [67] y se la utiliza en la creación de un algoritmo de control predictivo.

$$\Delta x(k+1) = A\Delta x(k) + B\Delta u(k) \quad (3.15)$$

$$y(k+1) - y(k) = C.A\Delta x(k) + C.B\Delta u(k) \quad (3.16)$$

Colocando las ecuaciones 3.15 y 3.16 en forma matricial se obtiene el modelo aumentado 3.17 y 3.18:

$$\begin{bmatrix} \overbrace{\Delta x(k+1)}^{x(k+1)} \\ y(k+1) \end{bmatrix} = \overbrace{\begin{bmatrix} A & 0_m^T \\ CA & 1 \end{bmatrix}}^A \begin{bmatrix} \overbrace{\Delta x(k)}^{x(k)} \\ y(k) \end{bmatrix} + \overbrace{\begin{bmatrix} B \\ CB \end{bmatrix}}^B \Delta u(k) \quad (3.17)$$

$$y(k) = \overbrace{\begin{bmatrix} 0_m & 1 \end{bmatrix}}^C \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} \quad (3.18)$$

Donde 0_m es una matriz de ceros cuya longitud depende del orden del sistema. Las nuevas matrices A, B y C se formulan según estas expresiones y se les conoce como matrices del modelo aumentado. Ahora es necesario reformular las predicciones haciendo uso de este nuevo modelo, el procedimiento a seguir es el mismo, pero como se había mencionado anteriormente es necesario implicar en la formulación el concepto de horizonte deslizando.

El conjunto de expresiones 3.7 a 3.10 y 3.11 a 3.14 muestran las predicciones de estado y salida respectivamente, sin embargo esta formulación considera el estado inicial $x(0)$ y las predicciones hasta el horizonte de predicción N (primera ventana de optimización), este proceso aplica para el primer instante de muestreo ($k = 0$). Cuando el próximo período de muestreo se alcanza, el procedimiento debe repetirse calculando una nueva ley de control $\Delta u(k_i), \Delta u(k_i + 1), \dots, \Delta u(k_i + M - 1)$, por ello es necesario generalizar las formulaciones utilizando la idea de horizonte deslizante. Además es necesario considerar para la formulación predictiva el horizonte de control M .

El conjunto de ecuaciones 3.18 a 3.20 y 3.21 a 3.23 muestran la formulación generalizada de las predicciones de estado y salida respectivamente. Las matrices A, B y C son las matrices del modelo con integrador embebido.

$$x(k_i + 1|k_i) = Ax(k_i) + B\Delta u(k_i) \quad (3.18)$$

$$x(k_i + 2|k_i) = A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) \quad (3.19)$$

$$\vdots$$

$$x(k_i + N|k_i) = A^N x(k_i) + A^{N-1}B\Delta u(k_i) + A^{N-2}B\Delta u(k_i + 1) \\ + \dots + A^{N-M}B\Delta u(k_i + M - 1) \quad (3.20)$$

$$y(k_i + 1|k_i) = CAx(k_i) + CB\Delta u(k_i) \quad (3.21)$$

$$y(k_i + 2|k_i) = CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1) \quad (3.22)$$

⋮

$$y(k_i + N|k_i) = CA^N x(k_i) + CA^{N-1}B\Delta u(k_i) + CA^{N-2}B\Delta u(k_i + 1) \dots \\ + CA^{N-M}B\Delta u(k_i + M - 1) \quad (3.23)$$

Las predicciones de salida guardan un determinado patrón que debe ser considerado para la generación del algoritmo. Este patrón puede ser más notorio si la formulación es expresada de forma matricial, para ello es necesario definir los vectores Y y ΔU (véase las expresiones 3.24 y 3.25) tal como lo señala [67]:

$$Y = [y(k_i + 1|k_i) \quad y(k_i + 2|k_i) \quad y(k_i + 3|k_i) \quad \dots \quad y(k_i + N|k_i)] \quad (3.24)$$

$$\Delta U = [\Delta u(k_i) \quad \Delta u(k_i + 1) \quad \Delta u(k_i + 2) \quad \dots \quad \Delta u(k_i + M - 1)]^T \quad (3.25)$$

Utilizando 3.24 y 3.25 en el conjunto de expresiones 3.21 a 3.23 se obtiene la ecuación 3.26, que denota las predicciones expresadas de forma matricial:

$$Y = Fx(k_i) + \Phi \Delta U \quad (3.26)$$

Las matrices F y Φ (véase las expresiones 3.27 y 3.28) están en función de las matrices A, B y C del modelo aumentado. Para un sistema de simple

entrada y simple salida las dimensiones de Y y ΔU son el horizonte de predicción y control respectivamente.

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^N \end{bmatrix} \quad (3.27)$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \dots & CA^{N-M}B \end{bmatrix} \quad (3.28)$$

3.2.2 Optimización sin restricciones.

Con las predicciones que se obtuvieron en la subsección 3.2.1 se puede abordar el problema de control óptimo para cada iteración k_i . Este proceso calcula una secuencia de acciones de control que minimizan una función de costo J y que se aplican al sistema. Continuando con el análisis y tomando como referencia lo que presenta [67], la función de costo que refleja el objetivo de control tiene la estructura de la expresión 3.29:

$$J(y, u) = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (3.29)$$

Esta función de costo cuadrática minimiza el error entre la salida predicha Y y el punto de ajuste R_s teniendo en consideración los incrementos de la señal de control ΔU . \bar{R} es una matriz diagonal $\bar{R} = r\omega I_{M \times M}$ cuya dimensión depende

del horizonte de control, el parámetro $r\omega$ es un parámetro de sintonización que da mayor o menor atención a los incrementos de control ΔU al momento de minimizar el error cuadrático entre la salida predicha y el punto de ajuste, dentro de la ventana de optimización.

El vector de puntos de ajuste contiene los valores de referencia que debe seguir el sistema dentro de la ventana de optimización, de ahí que la dimensión de este vector depende del horizonte de predicción N (véase la ecuación 3.30).

$$R_s^T = [1 \ 1 \dots 1]_N r(k_i) \quad (3.30)$$

En esta sección se aborda el problema de optimización sin restricciones, en este caso el problema puede ser resuelto de manera analítica. Previo a esto es necesario expresar 3.29 en función de los estados $x(k_i)$ y los incrementos de control ΔU (véase la ecuación 3.31).

$$J(y, u) = [R_s - Fx(k_i) - \Phi\Delta U]^T [R_s - Fx(k_i) - \Phi\Delta U] + \Delta U^T \bar{R} \Delta U \quad (3.31)$$

Para encontrar la secuencia de control óptima se iguala a cero la derivada de la función de costo con respecto a la variable de interés y se despeja ΔU . El resultado se puede observar en la ecuación 3.32.

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T [R_s - Fx(k_i)] \quad (3.32)$$

La expresión 3.32 muestra cómo obtener las secuencias de control futuras que llevarán a la salida predicha lo más cerca posible del punto de ajuste para cada intervalo de muestreo dentro de la ventana de optimización. A la expresión $(\Phi^T \Phi + \bar{R})^{-1}$ se le conoce como matriz Hessiana y en la literatura existente se la denota con H .

Como se puede observar todo el procedimiento para resolver la problemática del control predictivo en espacio de estados sin restricciones se resume en un cálculo netamente matricial. En síntesis se puede decir que si los horizontes de control y predicción no cambian durante toda la ejecución del controlador, se calcula por una sola vez las matrices F y Φ que dependen únicamente de las matrices A, B y C del modelo de predicción con integrador embebido. Las matrices F y Φ son utilizadas en la ecuación 3.32 para encontrar el vector de acciones de control, este cálculo se lo debe realizar en cada iteración y únicamente se debe enviar a la planta el primer término del vector ΔU , el resto de términos se desechan.

En [67] se realiza un análisis interesante que demuestra el efecto implícito de una retroalimentación dentro de la formulación predictiva. Si se observa

detenidamente la expresión 3.32 se puede notar dos componentes fundamentales:

- $k_y = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T R_s$ corresponde a los cambios del punto de ajuste.
- $k_{mpc} = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T F$ corresponde a la retroalimentación de estado.

3.2.3 Control predictivo multivariable sin restricciones.

Una de las ventajas más significativas de trabajar con un modelo en espacio de estados es la fácil extensión al caso multivariable a partir del análisis SISO presentado. La diferencia entre un modelo monovariable y un multivariable es la dimensión de las matrices B y C . Esto implica un cambio en la dimensión de las matrices del modelo con integrador embebido (véase las expresiones 3.33 y 3.34)

$$\begin{bmatrix} \overbrace{\Delta x(k+1)}^{x(k+1)} \\ \overbrace{y(k+1)}^{y(k+1)} \end{bmatrix} = \overbrace{\begin{bmatrix} A & 0_m^T \\ CA & I_{qxq} \end{bmatrix}}^A \begin{bmatrix} \overbrace{\Delta x(k)}^{x(k)} \\ \overbrace{y(k)}^{y(k)} \end{bmatrix} + \overbrace{\begin{bmatrix} B \\ CB \end{bmatrix}}^B \Delta u(k) \quad (3.33)$$

$$y(k) = \overbrace{\begin{bmatrix} 0_m & I_{qxq} \end{bmatrix}}^C \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} \quad (3.34)$$

Donde $I_{q \times q}$ es una matriz identidad cuadrada cuya dimensión depende del número de salidas del sistema, o_m es una matriz de ceros cuya dimensión depende del número de salidas y el número de estados del sistema ($q \times n$).

Con las matrices del modelo aumentado se formulan las predicciones de estado y salida utilizando el conjunto de expresiones 3.18 a 3.20 y 3.21 a 3.23 respectivamente. Se debe considerar que para el caso multivariable Y y ΔU deben ser matrices con una estructura tal como se muestra en las expresiones 3.35 y 3.36 respectivamente, esto debido a que el sistema puede tener varias entradas y salidas.

$$Y = [y(k_i + 1|k_i)^T \quad y(k_i + 2|k_i)^T \quad y(k_i + 3|k_i)^T \quad \dots \quad y(k_i + N|k_i)^T]^T \quad (3.35)$$

$$\Delta U = [\Delta u(k_i)^T \quad \Delta u(k_i + 1)^T \quad \Delta u(k_i + 2)^T \quad \dots \quad \Delta u(k_i + M - 1)^T]^T \quad (3.36)$$

Siguiendo el mismo procedimiento que se utilizó para el caso monovariable las predicciones de salida quedan definidas por la ecuación 3.26 con la diferencia de que las matrices F y Φ tienen dimensiones diferentes. El método analítico para resolver el problema de optimización en el caso multivariable es idéntico al monovariable. La secuencia de control queda determinada por la expresión 3.37.

$$\Delta U = \overbrace{[I_m \ o_m \ \dots \ o_m]}^M (\Phi^T \Phi + \bar{R})^{-1} \Phi^T [R_s - Fx(k_i)] \quad (3.37)$$

Dónde I_m es una matriz identidad cuya dimensión depende del número de entradas del sistema y o_m es una matriz de ceros que tiene la misma dimensión que I_m . El cálculo matricial que indica la expresión 3.37 entrega como respuesta un vector que contiene de forma ordenada los incrementos de control de cada variable manipulada para todos los puntos del horizonte de control. Por ejemplo si el horizonte de control es igual a 2 y el sistema tiene 3 variables manipuladas, el vector ΔU guardará la estructura de la expresión 3.38.

$$\Delta U = [\Delta u_1(k_i) \ \Delta u_2(k_i) \ \Delta u_3(k_i) \ \Delta u_1(k_i + 1) \ \Delta u_2(k_i + 1) \ \Delta u_3(k_i + 1)] \quad (3.38)$$

Para el caso multivariable los puntos de ajuste deben ser representados en todos los puntos del horizonte de predicción por un solo vector R_s (véase la ecuación 3.39).

$$R_s = [r_1(k_i) \ r_2(k_i) \ r_3(k_i) \ \dots \ r_q(k_i) \ r_1(k_i + 1) \ r_2(k_i + 1) \ r_3(k_i + 1) \ \dots \ r_q(k_i + 1) \ \dots \\ \dots \ r_1(k_i + N) \ r_2(k_i + N) \ r_3(k_i + N) \ \dots \ r_q(k_i + N)]^T \quad (3.39)$$

3.3 Estimador de estados.

Haciendo referencia a la formulación predictiva, el cálculo de las acciones de control futuras que llevarán a la salida a una referencia deseada, se sirve de los estados x disponibles en el tiempo k_i . Esto implica que las variables de estado deben estar disponibles para el controlador en todo momento, sin embargo en muchos de los casos las variables de estado no tienen una representación física que puede ser medida directamente del proceso, o en el caso de tener un significado físico no existe el sensor adecuado o la presencia de ruido hace difícil la medición.

Una alternativa muy atractiva que permite estimar la evolución de los estados es la presentada por Rudolf E. Kalman en 1958 [68], que hoy en día se le conoce como Filtro de Kalman. Esta técnica de estimación es considerada como una de las más idóneas, pues considera la presencia de ruido (incertidumbre del sensor y del sistema) tanto en la salida como en los estados de la planta. En [39], [69] y [70] se puede ver un análisis completo para la implementación del algoritmo. El estimador de Kalman considera un sistema lineal en tiempo discreto como el que se muestra en las expresiones 3.40 y 3.41.

$$x(k + 1) = A(k)x(k) + B(k)u(k) + G(k)\omega(k) \quad (3.40)$$

$$y(k) = C(k)x(k) + v(k) \quad (3.41)$$

En 3.40 y 3.41 las matrices $A(k), B(k), C(k)$ y $G(k)$ son matrices determinísticas que pueden ser variantes en el tiempo, $\omega(k)$ y $v(k)$ son ruidos aleatorios que actúan en el sistema de forma aditiva con distribución gaussiana y varianza Q y R respectivamente. El filtro de Kalman es un algoritmo óptimo pues minimiza un índice de error cuadrático medio.

Si se define el error como la diferencia entre el valor real y el estimado el algoritmo minimizará la expresión 3.42.

$$P(k + 1) = E\{e(k + 1).e(k + 1)^T\} \quad (3.42)$$

Donde a P se le conoce como matriz de covarianza del error y $\{.\}$ denota la esperanza matemática. El algoritmo se puede sistematizar en cuatro pasos [70]:

1. Se realiza una estimación a priori del estado $\hat{x}(k + 1)$, partiendo de una inicialización de estado tal que $x(0) = E\{x_0\}$ (véase la ecuación 3.43).

$$\hat{x}(k+1|k) = A(k|k)\hat{x}(k) + B(k)u(k) \quad (3.43)$$

2. Se proyecta adelante la covarianza del error (véase la ecuación 3.44).

$$P(k+1) = A(k)P(k)A(k)^T + G(k)QG(k)^T \quad (3.44)$$

3. Se calcula la ganancia de Kalman $L(k+1)$ con una matriz de covarianza inicial $P(0) = E\{e(0)e(0)^T\}$ y se realiza una estimación a posteriori corregida con la salida $y(k+1)$ (véase las ecuaciones 3.45 y 3.46).

$$L(k+1) = P(k+1)C^T(K+1)[C(k+1)P(k+1)C(K+1)^T + R]^{-1} \quad (3.45)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + L(k+1)[y(k+1) - C(k+1)\hat{x}(k+1|k)] \quad (3.46)$$

4. Se actualiza la covarianza del error para la próxima iteración (véase la ecuación 3.47).

$$P(k+1|k+1) = P(k+1|k) - L(k+1)C(k+1)P(k+1|k) \quad (3.47)$$

A los dos primeros pasos se les conoce como ciclos de propagación, mientras que los dos siguientes se les conocen como ciclos de actualización.

En [67] se puede ver un análisis de la incorporación de un observador de estados a la formulación predictiva. Según este análisis el autor concluye que el controlador y el observador pueden ser diseñados por separado aun cuando son considerados juntos al inicio del análisis. Esto también puede ser constatado en el diagrama de bloques presentado en la figura 3.2. Aquí se puede notar claramente como el observador no influye de manera directa en el cálculo de la acción de control.

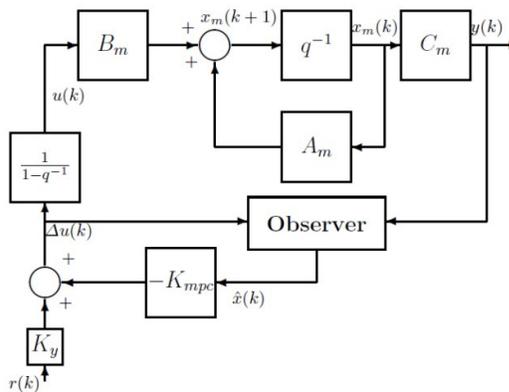


Figura 3.2 Controlador MPC con observador de estados [67, p. 36]

3.4 Formulación de Restricciones.

Una de las ventajas más atractivas que ostenta la técnica de control predictivo es la incorporación de restricciones. Esto quiere decir que la acción de control puede ser calculada bajo ciertas limitaciones reflejadas en impedimentos de carácter numérico tanto para las variables manipuladas como para las salidas del sistema. Por ejemplo una servo-válvula no puede

abrirse más allá del 100% o un amplificador de corriente no podrá brindar una corriente mayor que la de saturación. Al estilo de estos ejemplos, en la práctica todos los procesos están sujetos a restricciones, las cuales al ser consideradas evitan que el algoritmo presente soluciones basadas en situaciones irreales. La gran mayoría de las restricciones están vinculadas con los aspectos que se anotan a continuación:

- Constitución física de los elementos.
- Cuestiones de seguridad.
- Cumplimiento de normas de calidad.
- Respeto al medio ambiente.
- Requerimientos tecnológicos.

Las restricciones pueden ser expresadas en forma de desigualdades. En los problemas de ingeniería de control es muy común encontrarse con tres tipos de restricciones:

- Amplitud de la variable de control (véase la expresión 3.48).
- Tasa de cambio de la variable de control (véase la expresión 3.49).
- Amplitud de la variable de salida o sobre la variable de estado (véase la expresión 3.50).

$$u^{min} \leq u(k) \leq u^{max} \quad (3.48)$$

$$\Delta u^{min} \leq \Delta u(k) \leq \Delta u^{max} \quad (3.49)$$

$$y^{min} \leq y(k) \leq y^{max} \quad (3.50)$$

A las restricciones de amplitud y tasa de cambio de la variable de control comúnmente se les denomina restricciones duras, esto quiere decir que no existe un rango de tolerancia ya que su violación podría incurrir en complicaciones que afectarían notablemente al proceso. Por otra parte las restricciones de salida se les conoce como restricciones blandas, esto quiere decir que existe un rango de holgura y_{soft} sobre el cual puede moverse la variable de salida (véase la expresión 3.51).

$$y^{min} - y_{soft} \leq y(k) \leq y^{max} + y_{soft} \quad (3.51)$$

El cumplimiento del sistema sobre las restricciones de salida usualmente causa grandes cambios tanto en la amplitud como en la tasa de cambio de la variable de control, esto hace que éstas últimas sean violadas y ocurra un conflicto [67].

Es por esta razón que es importante incorporar una banda de relajación en las restricciones de salida.

En el caso de un sistema de múltiples entradas y múltiples salidas las restricciones deben ser especificadas para cada variable, esto se ilustra de

mejor manera en la tabla 3.1. A más de estos tres tipos de restricciones existen las restricciones de “punto final” que han tomado importancia a partir de los análisis de estabilidad del control predictivo. Este tipo de restricciones propuestas por Kwon y Pearson [71] garantizan la estabilidad en lazo cerrado de los sistemas controlables. La idea de estas restricciones es forzar el estado a cero en el final del horizonte de predicción. En [72] se presenta un análisis de estabilidad utilizando este tipo de restricciones.

Tabla 3.1 Formulación de restricciones para el caso MIMO

Amplitud de la variable de control	Tasa de cambio de la variable de control	Amplitud de la variable de salida
$u_1^{min} \leq u_1(k) \leq u_1^{max}$	$\Delta u_1^{min} \leq \Delta u_1(k) \leq \Delta u_1^{max}$	$y_1^{min} \leq y_1(k) \leq y_1^{max}$
$u_2^{min} \leq u_2(k) \leq u_2^{max}$	$\Delta u_2^{min} \leq \Delta u_2(k) \leq \Delta u_2^{max}$	$y_2^{min} \leq y_2(k) \leq y_2^{max}$
\vdots	\vdots	\vdots
$u_n^{min} \leq u_n(k) \leq u_n^{max}$	$\Delta u_n^{min} \leq \Delta u_n(k) \leq \Delta u_n^{max}$	$y_n^{min} \leq y_n(k) \leq y_n^{max}$

Cuando se incorporan restricciones lineales a la formulación, la función de coste no puede ser minimizada de forma analítica, en este caso es necesario acudir a un proceso iterativo que se le denomina Programación Cuadrática (QP) (véase [73] y [74]).

Daniel Rodríguez y Carlos Bordons [75] presentan algunos apuntes sobre el manejo de restricciones dentro del control predictivo. En base a este documento a continuación se presentan algunos inconvenientes relacionados con el tema:

- La incorporación de restricciones al control predictivo puede traer consigo grandes problemas relacionados con la estabilidad de bucle cerrado. Actualmente existen soluciones basadas en la teoría de Lyapunov (véase [76]).
- Pueden existir restricciones que hagan que el problema de optimización no converja en una solución.
- Un problema de optimización es factible cuando la función objetivo es acotada y todas las restricciones sean satisfechas.
- La no factibilidad de un problema de optimización puede aparecer en el transitorio o en régimen permanente.
- El problema de la no factibilidad puede ser resuelto utilizando diversas técnicas: 1) desconexión del controlador pasándolo a forma manual cuando aparezcan incompatibilidades. 2) análisis de factibilidad para eliminar temporalmente restricciones incompatibles. 3) relajación de restricciones que consiste en convertir las restricciones duras en restricciones blandas.

3.5 Solución numérica utilizando programación cuadrática.

La programación cuadrática es un problema de optimización que tiene por objetivo minimizar una función de coste cuadrática multivariable considerando un conjunto de restricciones impuestas. Las restricciones pueden estar sujetas a igualdades o a desigualdades de tipo lineal.

Actualmente existe abundante literatura que aborda el problema de QP. En [77], [78] y [79] se puede ver detalles sobre el procedimiento matemático para encontrar la solución al problema de optimización. Sin la intención de entrar en mayores detalles para el análisis que se presenta en este epígrafe se asume que la solución es factible y que la región de las restricciones están acotadas (véase [77]).

El problema de programación cuadrática con restricciones de igualdad se describe por medio de las ecuaciones 3.52 y 3.53. Donde x es la variable de decisión, E es una matriz simétrica ($n \times n$) que describe los coeficientes de los términos cuadráticos, F es un vector columna que describe los coeficientes de los términos lineales de la función de coste, M es una matriz simétrica ($m \times n$) que define las restricciones impuestas y γ es un vector columna que representa los valores del lado derecho de la igualdad. En el caso de que la matriz E sea igual a cero el término cuadrático desaparece y la optimización se convierte en un problema de programación lineal (LP).

$$J = \frac{1}{2}x^T E x + x^T F \quad (3.52)$$

$$Mx = \gamma \quad (3.53)$$

El problema de optimización con restricciones de igualdad puede ser resuelto a través de la técnica de los Multiplicadores de Lagrange. Este método de optimización es presentado por [67]. En base a lo expuesto por este autor a continuación se presenta una síntesis que permite encontrar la solución al problema de optimización:

El análisis parte de la expresión de Lagrange (véase 3.54) cuya base es la función de costo cuadrática a minimizar.

$$J = \frac{1}{2}x^T E x + x^T F + \lambda^T (Mx - \gamma) \quad (3.54)$$

En la expresión 3.54 λ es un vector columna m -dimensional que contiene un conjunto de valores denominados “Multiplicadores de Lagrange”, estos valores son encontrados paralelamente con los valores de x . El procedimiento de minimización consiste en igualar a cero la primera derivada parcial de la expresión de Lagrange con respecto a x y λ (véase las expresiones 3.55 y 3.56).

$$\frac{\partial J}{\partial x} = Ex + F + M^T \lambda = 0 \quad (3.55)$$

$$\frac{\partial J}{\partial \lambda} = Mx - \gamma = 0 \quad (3.56)$$

Las expresiones 3.55 y 3.56 contienen las condiciones necesarias para minimizar la función objetivo. Despejando las variables de interés se obtiene las expresiones 3.57 y 3.58.

$$\lambda = -(ME^{-1}M^T)^{-1}(\gamma + ME^{-1}F) \quad (3.57)$$

$$x = -E^{-1}(M^T \lambda + F) \quad (3.58)$$

El MPC permite restringir las variables controladas y manipuladas. Estas restricciones deben ser expresadas en formato de desigualdad. El problema de programación cuadrática con restricciones de desigualdad puede ser descrito por las expresiones 3.52 y 3.59.

$$Mx \leq \gamma \quad (3.59)$$

Una restricción de desigualdad puede estar compuesta por restricciones activas e inactivas. Una restricción es activa si $Mx = \gamma$, por otra parte la restricción es inactiva si $Mx < \gamma$. Según [80]:

Las restricciones activas en un punto factible x restringen el dominio factible en la vecindad de x , mientras las otras, restricciones inactivas, no tienen influencia en la vecindad del x . Por lo tanto, al estudiar las propiedades de un mínimo local, es claro que solo se toman en cuenta las restricciones activas.

En esta parte del análisis es necesario tener en cuenta las condiciones de Karush-Kuhn-Tucker¹ (KKT) (véase las expresiones 3.60 a 3.63).

$$Ex + F + M^T \lambda = 0 \quad (3.60)$$

$$Mx - \gamma \leq 0 \quad (3.61)$$

$$\lambda^T (Mx - \gamma) = 0 \quad (3.62)$$

$$\lambda \geq 0 \quad (3.63)$$

Esta es una generalización de la técnica de los Multiplicadores de Lagrange. Este grupo de condiciones son necesarias y suficientes para que exista una solución al problema de optimización. Tomando como referencia el procedimiento que presenta [67], este conjunto de condiciones puede ser expresado en términos de las restricciones activas (véase las expresiones 3.64 a 3.68).

¹ Se denomina así por su creador William Karush (1 Marzo 1917 – 22 Febrero 1997) quien fue profesor de matemáticas en la Universidad Estatal de California en Northridge. Fue el primero en publicar las condiciones necesarias para el problema de restricciones con desigualdad en su tesis de Master: "Minima of Functions of Several Variables with Inequalities as Side Constraints" a pesar de que se hizo famoso luego de presentar un artículo en una conferencia conjuntamente con Harold W. Kuhn y Albert W. Tucker (fuente:Wikipedia).

$$Ex + F + \sum_{i \in S_{act}} \lambda_i M_i^T = 0 \quad (3.64)$$

$$M_i x - \gamma_i = 0 \quad i \in S_{act} \quad (3.65)$$

$$M_i x - \gamma_i < 0 \quad i \notin S_{act} \quad (3.66)$$

$$\lambda_i \geq 0 \quad i \in S_{act} \quad (3.67)$$

$$\lambda_i = 0 \quad i \notin S_{act} \quad (3.68)$$

Donde S_{act} denota el índice del conjunto de restricciones activas, [...] M_i es la i -ésima columna de la matriz M . Para la i -ésima columna $M_i x - \gamma_i = 0$ significa que esta es una restricción de igualdad, por lo tanto es una restricción activa. En contraste $M_i x - \gamma_i < 0$ significa que la restricción es satisfecha, por lo tanto es una restricción inactiva. Para una restricción activa el correspondiente multiplicador de Lagrange es no negativo mientras que el multiplicador de Lagrange es cero si la restricción es inactiva [67].

Considerando que se conoce M_{act} y λ_{act} la solución al problema de optimización con restricciones de desigualdad queda representado a través de las expresiones 3.69 y 3.70.

$$\lambda_{act} = -(M_{act} E^{-1} M_{act}^T)^{-1} (\gamma_{act} + M_{act} E^{-1} F) \quad (3.69)$$

$$x = E^{-1} (F + M_{act}^T \lambda_{act}) \quad (3.70)$$

En conclusión se puede decir que un problema de optimización con restricciones de desigualdad puede ser tratado como un problema con

restricciones de igualdad, siempre y cuando se conozcan las restricciones activas. Entonces el foco de interés en este tipo de problemas está en determinar de todas las restricciones impuestas cuales son activas, por ende las que no lo son se les considerará como restricciones inactivas. Las restricciones inactivas son satisfechas implícitamente con el cumplimiento de una restricción activa, por lo tanto no es necesario considerarlas para el proceso de optimización.

Los “métodos de conjuntos activos” constituyen una alternativa para identificar el conjunto de restricciones activas. A continuación se detalla un proceso iterativo para identificar este tipo de restricciones:

- Armar las matrices M y γ (véase las expresiones 3.75 y 3.76) en base a las restricciones dadas. De manera explicativa se considera un problema con cuatro restricciones (véase las expresiones 3.71 a 3.74), sin embargo el procedimiento puede ser aplicado a más o menos restricciones de desigualdad.

$$a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 \leq z_1 \quad (3.71)$$

$$b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 \leq z_2 \quad (3.72)$$

$$c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \leq z_3 \quad (3.73)$$

$$d_1x_1 + d_2x_2 + d_3x_3 + d_4x_4 \leq z_4 \quad (3.74)$$

$$M = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} \quad (3.75)$$

$$\gamma = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \quad (3.76)$$

- A sabiendas de que existe una solución factible para las restricciones de igualdad se calcula el multiplicador de Lagrange (véase la expresión 3.77).

$$\lambda = -(ME^{-1}M^T)^{-1}(\gamma + ME^{-1}F) = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} \quad (3.77)$$

- Se observa los signos de los elementos del vector columna resultante, si alguno de ellos es negativo quiere decir que la restricción i -ésima correspondiente es una restricción inactiva por lo tanto se deshecha. Los valores positivos corresponden a restricciones activas, por lo tanto son consideradas para la siguiente iteración.
- Con el nuevo conjunto de restricciones se vuelve a formular las matrices M y γ y se vuelve a calcular el multiplicador de Lagrange.

Nuevamente se observan los signos de los elementos correspondientes y se desechan las restricciones inactivas.

- El proceso se vuelve a repetir hasta que el multiplicador de Lagrange tenga solo elementos con signo positivo. Entonces se habrá encontrado las restricciones activas del problema.

Este proceso es altamente iterativo lo cual puede verse reflejado como un coste computacional elevado. El método "Primal-Dual surge como una alternativa altamente atractiva para solventar este inconveniente, este método tiene por objeto la formulación de un problema de optimización que parte del programa principal reduciendo el número de variables de decisión y facilitando el manejo de restricciones, por lo tanto reduce de manera notable el coste computacional.

El modelo dual convierte el problema de minimización en un programa de maximización más sencillo, el planteamiento radica en el cambio de variable de decisión x por la variable λ respetando las restricciones de manera implícita. Según [67] el procedimiento que sustenta la dualidad se detalla a continuación:

El problema parte de la expresión de Lagrange del modelo primal (véase la expresión 3.78)

$$\max_{\lambda \geq 0} \min_x \left[\frac{1}{2} x^T E x + x^T F + \lambda^T (Mx - \gamma) \right] \quad (3.78)$$

La expresión 3.78 puede ser expresada únicamente en términos de los multiplicadores de Lagrange λ , para lo cual la variable x debe ser remplazada utilizando la expresión 3.68. El resultado de este procedimiento se puede ver en la expresión 3.79.

$$\max_{\lambda \geq 0} \left[-\frac{1}{2} \lambda^T M E^{-1} M^T \lambda - \lambda^T (\gamma + M E^{-1} F) - \frac{1}{2} F^T E^{-1} F \right] \quad (3.79)$$

La expresión 3.79 puede ser llevada a términos de una minimización (véase la expresión 3.80).

$$\min_{\lambda \geq 0} \left[\frac{1}{2} \lambda^T \alpha \lambda - \lambda^T \beta - \frac{1}{2} \gamma^T E^{-1} \gamma \right] \quad (3.80)$$

Donde $\alpha = \lambda^T M E^{-1} M^T \lambda$ y $\beta = \gamma + M E^{-1} F$. Claramente se puede observar como el problema de optimización se torna más sencillo de resolverlo ya que la restricción $\lambda \geq 0$ puede ser fácilmente manejable. Considerando que se

conocen las restricciones activas para λ el problema de optimización puede ser resuelto para la variable de decisión x utilizando la expresión 3.81.

$$x = -E^{-1}F - E^{-1}M_{act}^T \lambda_{act} \quad (3.81)$$

Una alternativa para resolver el problema dual es el algoritmo de Hildreth's. Según [67] este algoritmo analiza los componentes individuales λ_i por lo que se convierte en un problema de optimización cuadrática de un simple elemento (véase la expresión 3.82), reduciendo considerablemente el coste computacional.

$$\omega_i^{m+1} = -\frac{1}{h_{ii}} \left[k_i + \sum_{j=1}^{i-1} h_{ij} \lambda_j^{m+1} + \sum_{j=i+1}^n h_{ij} \lambda_j^m \right] \quad (3.82)$$

Donde h_{ij} es el i -ésimo elemento de la matriz α y k_i es i -ésimo elemento del vector β . Las restricciones activas pueden ser encontradas utilizando el criterio mostrado en la expresión 3.83.

$$\lambda_i^{m+1} = \max(0, \omega_i^{m+1}) \quad (3.83)$$

De la expresión 3.83 se obtiene λ^* que contiene un conjunto de valores para la convergencia, donde los ceros corresponden a restricciones inactivas y valores positivos corresponden a restricciones activas. El proceso de

iteración culminará cuando el contador de iteraciones llegue al final. Una vez discriminadas las restricciones se utilizan las activas en la expresión 3.81 para resolver el problema de optimización cuadrática.

3.6 Control predictivo con restricciones en las variables manipuladas y variables de salida.

En la sección 3.5 se analizó el fundamento matemático de una propuesta para la resolución del problema de optimización utilizando programación cuadrática y considerando restricciones de igualdad y desigualdad. En la presente sección se vinculan los aspectos de optimización al análisis del control predictivo con restricciones.

Antes de aplicar el algoritmo QP es necesario que las restricciones sean parametrizadas en función de ΔU . Por ejemplo si se desea restringir la amplitud de la variable manipulada entre u^{min} y u^{max} , se debe buscar una relación entre $u(k_i)$ y $\Delta u(k_i)$. Todas las restricciones están definidas dentro del horizonte de predicción, sin embargo para reducir el coste computacional se selecciona un pequeño conjunto de muestras para imponer las restricciones. La relación entre $u(k_i)$ y $\Delta u(k_i)$ está determinada por la expresión 3.84.

$$u(k_i) = u(k_i - 1) + \Delta u(k_i) \quad (3.84)$$

La expresión 3.85 muestra la estructura del vector ΔU para un horizonte de control M .

$$\Delta U = [\Delta u(k_i) \ \Delta u(k_i + 1) \ \Delta u(k_i + 2) \ \dots \ \Delta u(k_i + M - 1)]^T \quad (3.85)$$

En la primera muestra $\Delta u(k_i)$ queda representado por: $[\overbrace{1 \ 0 \ 0 \ 0 \ \dots \ 0}^M] \Delta U$. Por lo tanto la relación $u(k_i)$ con $\Delta u(k_i)$ para todas las muestras del horizonte de control puede escribirse utilizando el conjunto de expresiones 3.86 a 3.89.

$$u(k_i) = u(k_i - 1) + [\overbrace{1 \ 0 \ 0 \ 0 \ \dots \ 0}^M] \Delta U \quad (3.86)$$

$$u(k_i + 1) = u(k_i - 1) + [\overbrace{1 \ 1 \ 0 \ 0 \ \dots \ 0}^M] \Delta U \quad (3.87)$$

$$u(k_i + 2) = u(k_i - 1) + [\overbrace{1 \ 1 \ 1 \ 0 \ \dots \ 0}^M] \Delta U \quad (3.88)$$

$$\vdots$$

$$u(k_i + N) = u(k_i - 1) + [\overbrace{1 \ 1 \ 1 \ 1 \ \dots \ 1}^M] \Delta U \quad (3.89)$$

El vector ΔU generalmente es llamado “variable de decisión” dentro de la literatura de la optimización. Utilizando esta parametrización la restricción $u^{min} \leq u(k) \leq u^{max}$ se representa de manera general a través de la expresión 3.90.

$$\begin{bmatrix} u^{min} \\ u^{min} \\ \vdots \\ u^{min} \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u(k_i - 1) + \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \Delta U \leq \begin{bmatrix} u^{max} \\ u^{max} \\ \vdots \\ u^{max} \end{bmatrix} \quad (3.90)$$

Por lo tanto las restricciones para la amplitud de la variable manipulada pueden ser expresadas como $U^{min} \leq \rho_1 u(k_i - 1) + \rho_2 \Delta U \leq U^{max}$, donde ρ_1 es un vector de unos y ρ_2 es una matriz triangular inferior. La longitud de los vectores y las matrices depende de los puntos de coincidencia dados como parte de la sintonización del controlador, sin embargo para evitar un gran coste computacional se puede utilizar un conjunto pequeño de muestras para asignar las restricciones. Esta formulación es una generalización basada en una formulación específica que presenta [67] para un conjunto de muestras menor al número de puntos de coincidencia.

Bajo la misma nomenclatura las restricciones para la tasa de cambio de la variable de control pueden ser expresadas de manera directa a través de la expresión 3.91.

$$\begin{bmatrix} \Delta u^{min} \\ \Delta u^{min} \\ \vdots \\ \Delta u^{min} \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \Delta U \leq \begin{bmatrix} \Delta u^{max} \\ \Delta u^{max} \\ \vdots \\ \Delta u^{max} \end{bmatrix} \quad (3.91)$$

Por otra parte las restricciones para la salida pueden ser expresadas utilizando la expresión 3.92.

$$y^{min} \leq Fx(k_i) + \Phi\Delta U \leq y^{max} \quad (3.92)$$

Por lo general dentro del control predictivo las restricciones que se imponen en las variables manipuladas y en las variables de salida son del tipo $a \leq x \leq c$, sin embargo el problema de optimización utilizando QP considera restricciones del tipo $M.x \leq \gamma$. Por lo tanto el primer paso para vincular la formulación del control predictivo al problema de optimización consiste en separar las restricciones impuestas:

- Tasa de cambio de la variable de control (véase las expresiones 3.93 y 3.94).

$$-I\Delta U \leq -\Delta U^{min} \quad (3.93)$$

$$I\Delta U \leq \Delta U^{max} \quad (3.94)$$

- Amplitud de la variable de control (véase las expresiones 3.95 y 3.96).

$$-\rho_2\Delta U \leq \rho_1u(k_i - 1) - U^{min} \quad (3.95)$$

$$\rho_2\Delta U \leq \rho_1u(k_i - 1) + U^{max} \quad (3.96)$$

- Amplitud de la variable de salida (véase las expresiones 3.97 y 3.98).

$$-\Phi\Delta U \leq Fx(k_i) - Y^{min} \quad (3.97)$$

$$\Phi\Delta U \leq -Fx(k_i) + Y^{max} \quad (3.98)$$

Ahora las restricciones del sistema se han convertido en seis tipos de restricciones donde la variable de decisión es ΔU . Estas restricciones deben ser llevadas a la forma $M\Delta U \leq \gamma$ (véase la expresión 3.99)

$$\begin{bmatrix} -I \\ I \\ -\rho_2 \\ \rho_2 \\ -\Phi \\ \Phi \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \\ -\rho_1 u(k_i - 1) - U^{min} \\ \rho_1 u(k_i - 1) + U^{max} \\ Fx(k_i) - Y^{min} - Y_{soft} \\ -Fx(k_i) + Y^{max} + Y_{soft} \end{bmatrix} \quad (3.99)$$

La expresión 3.99 contiene las restricciones para la tasa de cambio y amplitud de la variable manipulada, así como para la amplitud de la salida del sistema considerando una banda de relajación. La función de coste cuadrática bajo estas restricciones puede ser utilizada a través del método de Hildreth's para encontrar las acciones de control en cada período de muestreo. Cabe indicar que para el caso multivariable cada restricción se trata de manera independiente, esto se ve reflejado como un aumento de filas en la expresión 3.99.

3.7 Diseño del Experimento.

Una vez comprendido el fundamento matemático del control predictivo basado en modelo, el procedimiento a seguir para la implementación de la plataforma para el entrenamiento en la sintonía de controladores predictivos se detalla a continuación:

1. Implementación de la comunicación UDP entre dos PCs para simular el intercambio de datos entre la planta y el controlador MPC.
2. Implementación y evaluación de desempeño del algoritmo de identificación de sistemas monovariantes y multivariantes, como alternativa para utilizarlo como modelo de predicción dentro del algoritmo MPC.
3. Revisión bibliográfica para la consecución de modelos de planta para la evaluación de desempeño del algoritmo MPC.
4. Implementación y evaluación de desempeño del algoritmo MPC sin restricciones para sistemas monovariantes.
5. Implementación y evaluación de desempeño del algoritmo MPC sin restricciones para sistemas multivariantes.
6. Implementación y evaluación de desempeño del filtro de Kalman para la estimación de estados.
7. Implementación y evaluación de desempeño del algoritmo MPC sin restricciones con estimación de estados.

8. Implementación y evaluación de desempeño del algoritmo MPC con restricciones para sistemas monovariantes.
9. Implementación y evaluación de desempeño del algoritmo MPC con restricciones para sistemas multivariantes.
10. Diseño e implementación de una plataforma interactiva y flexible para el entrenamiento en la sintonía de controladores predictivos.

CAPÍTULO 4

IMPLEMENTACIÓN DE LA PLATAFORMA SIMUMPC Y PRUEBAS DEL SISTEMA.

El fundamento matemático de la estrategia de control predictivo da las pautas necesarias para la creación de un algoritmo computacional que puede ser implementado sobre un ordenador utilizando diversos entornos de programación. Atendiendo al objetivo del proyecto de tesis se ha implementado una plataforma basada en MATLAB[®] la cual brinda al usuario las herramientas necesarias para manipular los parámetros de sintonía, generar las acciones de control necesarias, aplicarlas a una planta y visualizar su comportamiento.

La plataforma SIMUMPC alberga una serie de herramientas que a más de permitir un entrenamiento en la sintonía de controladores predictivos brinda al usuario la flexibilidad para que se pueda identificar modelos de plantas y estimar sus estados. Estos complementos han sido vinculados a la

plataforma en base a la necesidad del control predictivo de contar con un modelo matemático de predicción y los estados presentes generados en una planta.

Otra característica, quizá la más importante de esta plataforma, es la disponibilidad para sintonizar controladores predictivos separando el controlador y la planta en PCs diferentes y comunicándolos a través de un protocolo UDP. Este modo de sintonía se le denomina “modo en línea” y permite simular el comportamiento controlador-planta atendiendo a ciertos parámetros que ineludiblemente están presentes en una aplicación real, como por ejemplo el ruido, el retardo y el tiempo de muestreo. Sin embargo en el caso de no contar con un PC adicional la plataforma brinda la posibilidad de entrenar controladores predictivos utilizando un sólo PC, donde se encuentra el controlador y la planta, obviamente prescindiendo de los parámetros antes mencionados. A este modo de entrenamiento se le denomina “modo fuera de línea”. En el presente capítulo se muestran detalles sobre la implementación de la plataforma así como los resultados obtenidos con su respectivo análisis.

4.1 Protocolo de Comunicación.

La modalidad de sintonía en línea que ofrece la plataforma SIMUMPC exige un modo de comunicación soportado en MATLAB[®] que garantice la

transmisión y recepción de datos entre el PC-controlador y el PC-planta. Por la sencillez y el bajo consumo de recursos el protocolo UDP (Protocolo de Datagrama de Usuario) constituye una opción altamente atractiva.

UDP es un protocolo basado en el modelo petición - respuesta y está situado en la capa de transporte del modelo de referencia OSI (Interconexión de Sistemas Abiertos). Es un protocolo no orientado a conexión por ende no ofrece garantía en la entrega y recepción de mensajes pues no permite la confirmación y control de flujo de datos (véase [21]), esto genera grandes inconvenientes cuando las aplicaciones demandan altas tasas de bits, sin embargo para el modo de comunicación que requiere la plataforma SIMUMPC estas características dejan de ser inconvenientes pues la cantidad de datos demandada entre el transmisor y receptor es muy baja y la tasa de muestro es relativamente grande.

UDP utiliza los servicios de la capa de red por lo que requiere direcciones IP para el origen y destino, además utiliza puertos para la comunicación entre aplicaciones, el campo de puerto en el datagrama ocupa un espacio de 16 bits, por lo que el rango de valores válidos va de 0 a 65.535. Los 1024 primeros puertos se les conoce como puertos “bien conocidos” y están reservados para aplicaciones estándar.

MATLAB® y Simulink® soportan comunicaciones TCP/IP y UDP a través de la Herramienta de Control e Instrumentación (Instrument Control Toolbox) [81]. UDP permite comunicar instrumentos reales con MATLAB® y Simulink®, de igual manera la comunicación puede estar orientada a aplicaciones de software entre 2 o más PCs para el intercambio de información. En ambos casos se puede enviar y recibir datos de forma sencilla sin la necesidad de conocer la estructura de la trama, estos datos pueden ser procesados y/o graficados en tiempo real.

El modo de configuración requiere la asignación de una dirección IP a cada elemento de la comunicación. Obviamente estas direcciones deben pertenecer a la misma red, de igual manera es necesario especificar el puerto de comunicación. La configuración básica para el intercambio de información entre dos PCs se muestra en la figura 4.1.

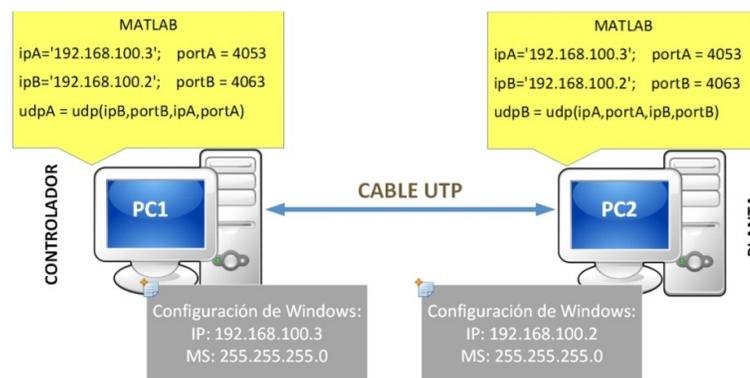


Figura 4.1 Esquema de configuración para la comunicación UDP entre 2 PCs.

Cuando un objeto UDP es creado, el espacio de trabajo en MATLAB® deja ver los parámetros de configuración e información de estado que son asignados al mismo. Los parámetros de configuración permiten modos de comunicación personalizados principalmente atendiendo aspectos relacionados con eventos y llamadas que se generan en la transmisión y recepción de datos. En [81] se puede encontrar información sobre el manejo de estos parámetros así como varios ejemplos de comunicación.

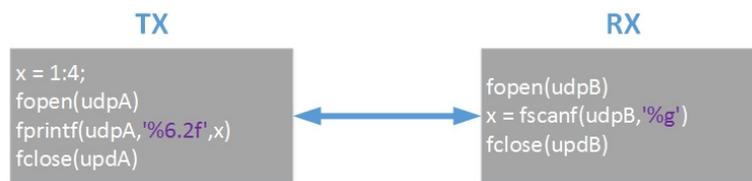


Figura 4.2 Transmisión y recepción de datos a través de UDP

Para transmitir o recibir un dato la conexión UDP debe ser abierta utilizando el comando *fopen* y se puede utilizar el comando *fprintf* o *fwrite* para la escritura de ficheros desde MATLAB®, igualmente para la lectura de datos se puede utilizar *fscanf* o *fread*. Luego de que se hayan transmitido o recibido los datos la comunicación deberá ser cerrada utilizando el comando *fclose*. La figura 4.2 muestra como comunicar dos estaciones remotas con objetos UDP previamente creados.

Para la plataforma SIMUMPC el intercambio de datos entre la planta y el controlador se lo realiza de forma asíncrona, es decir no existe una

coordinación entre los elementos de la comunicación a más del pulso de arranque mediante el cual el controlador ordena funcionar a la planta. El proceso de transferencia bidireccional de datos se describe a continuación:

1. En el PC plata un solucionador de ecuaciones diferenciales resuelve la dinámica de la planta para cada período de refrescamiento. Este valor de tiempo debe ser lo más pequeño posible para que la planta simule un comportamiento continuo. El solucionador utilizado para la plataforma es el ODE45:

```
[t,x]=ode45('planta',[ti tf],xm);
```

Donde *planta* es una función que contiene descrita la dinámica del proceso, t_i y t_f son el tiempo inicial y final de simulación y xm son los estados iniciales. Esta línea de comando debe ser ejecutada periódicamente para que la planta refresque sus estados. Esta situación puede ser administrada a través de un timer:

```
t0=timer('TimerFcn',{@fun,i}, 'ExecutionMode', 'fixedDelay', 'Period', tref);
```

Donde *fun* es una función que contiene la línea de comando del solucionador y *tref* es el período con el cual se quiere refrescar la planta.

Los parámetros t_i y t_f son variables; así para el primer período de refrescamiento $t_i = 0$ y $t_f = t_{ref}$, para el siguiente período $t_i = t_{ref}$ y $t_f = 2t_{ref}$. De igual manera los estados x_m deben ser actualizados con los últimos valores que generó el solucionador.

A medida que se van refrescando los estados de la planta estos son puestos a disposición del controlador. Por último la dinámica del sistema deberá necesariamente depender de una o varias acciones de control las mismas que llegarán a la planta desde el controlador en cada período de muestreo. Es decir antes de ejecutar el solucionador se deberá preguntar por el estado del puerto. En el caso de existir un dato este debe ser considerado como acción de control para refrescar los estados de la planta, caso contrario estos se refrescan con la última acción de control existente.

Todo este proceso inicia bajo la orden del PC controlador y es el único parámetro de sincronización entre ambos elementos.

2. En el PC controlador un *timer* como el mostrado en el punto 1 es utilizado para administrar la ejecución de una función periódicamente. El período configurado es el período de muestreo y determina con que frecuencia se van a adquirir los datos de la planta.

A medida que los estados son adquiridos por el controlador estos son utilizados por el algoritmo MPC para generar la acción de control. Esta a su vez es escrita en el puerto y puesta a disposición de la planta.

4.2 Herramienta para identificación de sistemas.

4.2.1 Introducción.

La estrategia de control predictivo hace uso de un modelo matemático para calcular las predicciones a lo largo de un horizonte de predicción. Este modelo puede tener varias representaciones y en función de esto se distinguen diversas variantes de la estrategia. Este modelo debe residir en el controlador y la exactitud con la que reproduce el comportamiento dinámico real de la planta influye trascendentalmente en el comportamiento del sistema controlado.

Las técnicas de identificación en subespacios proveen modelos LTI multivariantes en espacio de estados. Esta metodología de identificación ha evolucionado considerablemente en los últimos 20 años y en la actualidad presenta diversas variantes para la obtención de un modelo, siendo 4SID (“Subspace-based State Space System IDentification Methods”) el método más comúnmente conocido. La intención de esta subsección no consiste en proporcionar el fundamento matemático y análisis de la familia de técnicas, lo

que se pretende ilustrar es la manera en cómo realizar un proceso de identificación en subespacios utilizando MATLAB®. Sin embargo si se desea ampliar sobre el tema se puede acudir a [82] quien presenta un estado del arte sobre la técnica.

El diseño del experimento requiere el conocimiento funcional básico de la planta (número de entradas y salidas), es necesario generar secuencias binarias pseudo aleatorias 'PRBS' no correlacionadas para cada una de las entradas. La PRBS es una señal de dos niveles cuyo nivel mínimo y máximo depende de los requerimientos de funcionamiento de la planta que se desea identificar. El resultado de la secuencia da una sucesión de pulsos rectangulares modulados en su ancho, es decir se genera excitaciones temporales de tipo escalón con un contenido rico en frecuencias. Por ejemplo para identificar un modelo de tres entradas se genera un set de tres señales PRBS no correlacionadas, esto se lo puede realizar en MATLAB® utilizando la siguiente lista de comandos:

$$u1 = idinput(800, 'PRBS', [0 0.3], [0 1]);$$
$$u2 = idinput(800, 'PRBS', [0 0.6], [0 6]);$$
$$u3 = idinput(800, 'PRBS', [0 0.8], [0 10]);$$
$$u = [u1 u2 u3]$$

El comando *idinput* requiere cuatro parámetros de configuración:

$$u = idinput(N, 'Type', Band, Levels)$$

Donde “N” es la longitud o número de elementos del vector generado, “Tipo” especifica el tipo de señal a utilizar, ya que este comando puede generar otro tipo de señales, “Band” es un vector de dos elementos que especifica el contenido de frecuencia mínimo y máximo normalizado entre 0 y 1 y “Level” es un vector de dos elementos que especifica los valores mínimo y máximo de amplitud de la señal a generar. La figura 4.3 ilustra las tres secuencias temporales generadas.

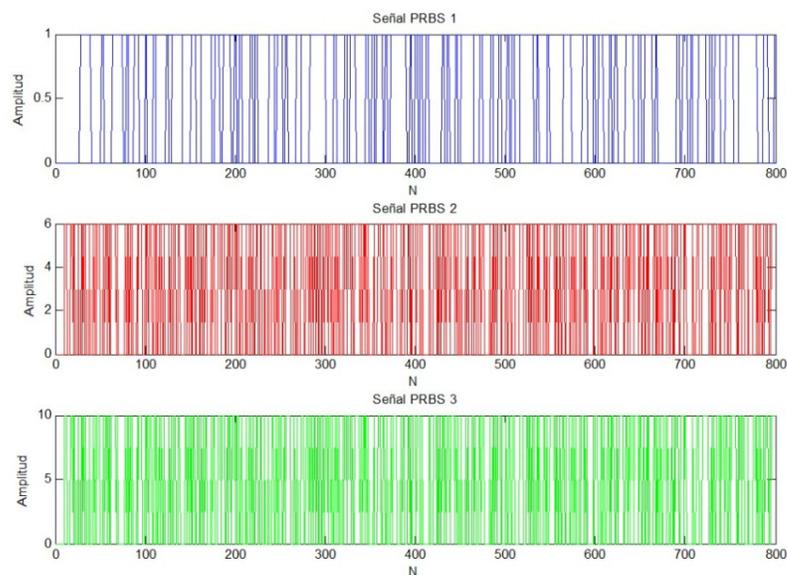


Figura 4.3 Señales PRBS para identificación de sistema con tres entradas

Las secuencias generadas deben ser ingresadas a la planta, en el caso de que sea una simulación como es el caso de la plataforma SIMUMPC la planta deberá ser representada en otro PC a través de un modelo matemático adecuado, este modelo puede ser un conjunto de ecuaciones diferenciales simuladas a través de ODEs [83]. Las salidas generadas deben ser muestreadas con un tiempo determinado que dependerá de las características dinámicas del sistema.

El experimento continúa con el almacenamiento de los datos generados por la planta como respuesta a la secuencias de excitación PRBS. La figura 4.4 muestra las respuestas del sistema ante las señales de entrada generadas. Para este caso ilustrativo se considera una planta de tres entradas y dos salidas.

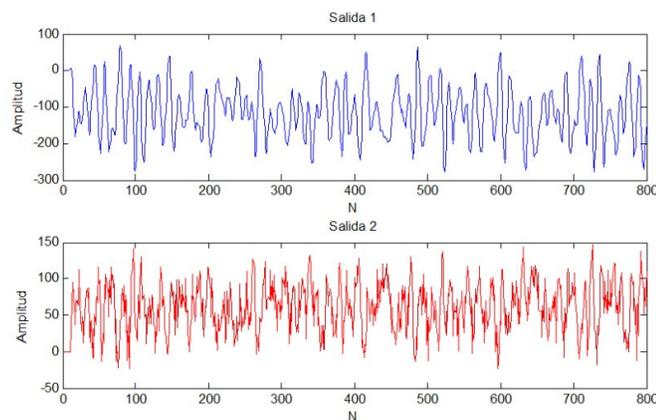


Figura 4.4 Respuestas del sistema ante señales de entrada PRBS

Posteriormente se debe armar un objeto de datos que contenga los vectores de salida y entrada, además se deberá especificar el tiempo de muestreo con el que se realizó el experimento:

$$z = \text{iddata}(y, u, Ts)$$

Este objeto de datos es el que sirve para ejecutar la identificación de la planta a través del comando `n4sid`:

$$\text{sys} = \text{n4sid}(z, \text{orden})$$

El parámetro de configuración “orden” puede ser ingresado directamente en función del orden de modelo al que el usuario desea acomodarse, sin embargo si se coloca 1:10 una gráfica será visualizada previa la identificación (véase la figura 4.5). Bajo esta condición MATLAB® recomendará al usuario el orden del sistema más adecuado considerando un orden máximo de 10. El valor que se decida deberá ser ingresado manualmente de acuerdo al mensaje visualizado en la ventana de comandos.

Una vez ingresado el orden, MATLAB® ejecuta internamente el algoritmo de identificación para devolver el modelo discreto en espacio de estados. Se puede utilizar el comando `compare` para generar una gráfica de validación que especifique el porcentaje de coincidencia entre una porción de datos

reales obtenidos del experimento y una porción de datos obtenidos con el modelo estimado (véase la figura 4.6).

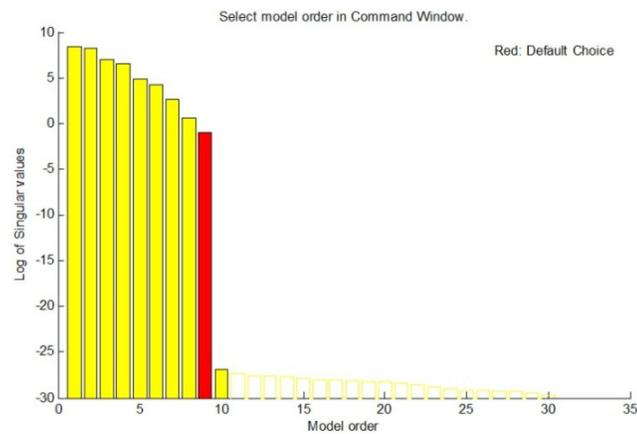


Figura 4.5 Gráfica del comando `n4sid` para sugerir el orden de modelo identificado.

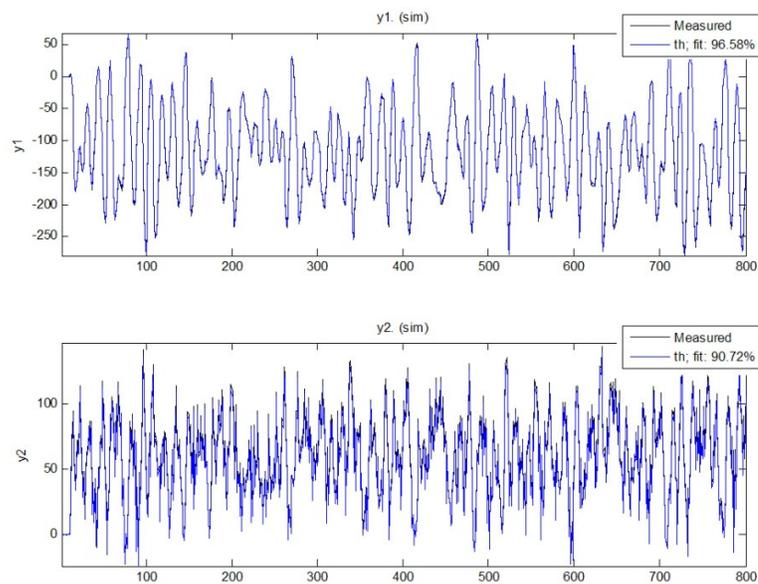


Figura 4.6 Validación del proceso de identificación.

4.3 Algoritmo MPC.

La estrategia de control predictivo que se implementa en SIMUMPC hace uso de un modelo en espacio de estados para la predicción de la salida y el cálculo de la acción de control mediante la minimización de una función de coste. La figura 4.7 muestra un diagrama de flujo que sintetiza la implementación del algoritmo.

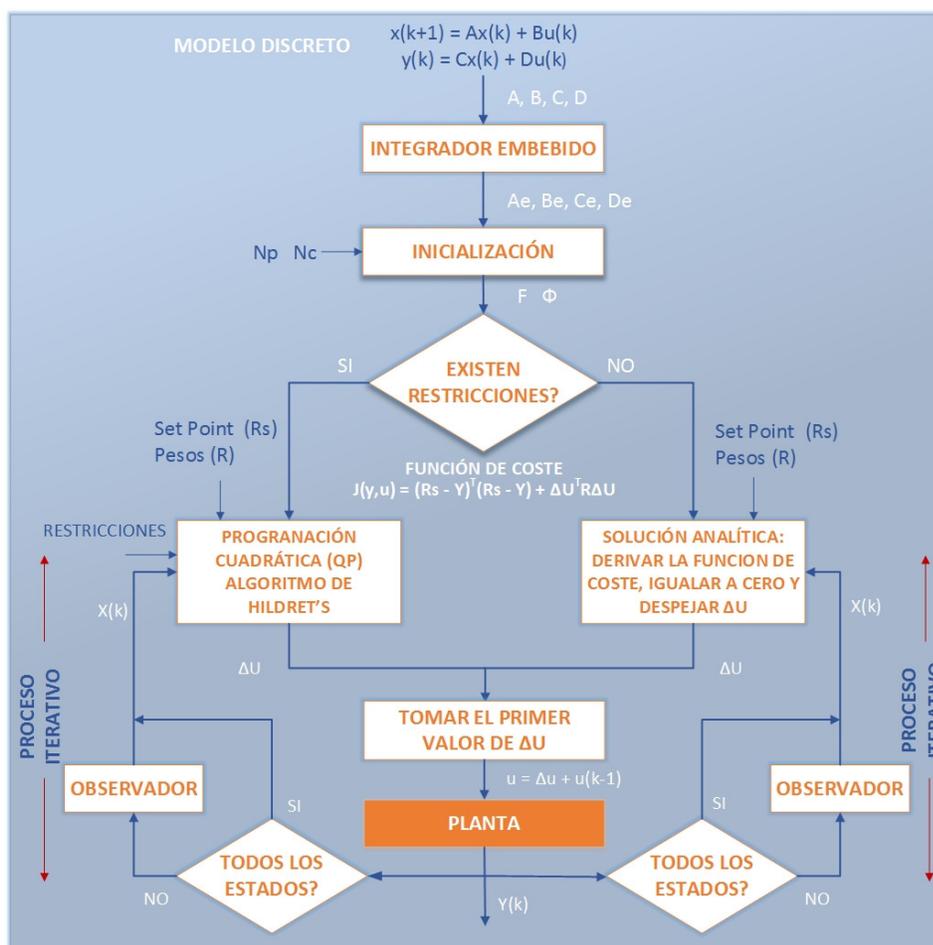


Figura 4.7 Diagrama esquemático de la estrategia de control predictivo.

Según el diagrama mostrado en la figura 4.7 la implementación del controlador predictivo se ha dividido en tres etapas: implementación del modelo aumentado, fase de inicialización y proceso iterativo o fase de ejecución, las mismas que serán detalladas en la presente sección.

4.3.1 Implementación del modelo aumentado.

El algoritmo MPC utiliza un modelo con integrador embebido que también se le conoce como modelo aumentado debido al hecho de que incrementa estados en la representación del sistema. La manera de implementar un integrador en un sistema de espacio de estados se muestra en las expresiones 3.17 y 3.18. Estas expresiones han sido utilizadas para mostrar la representación esquemática de la figura 4.8.

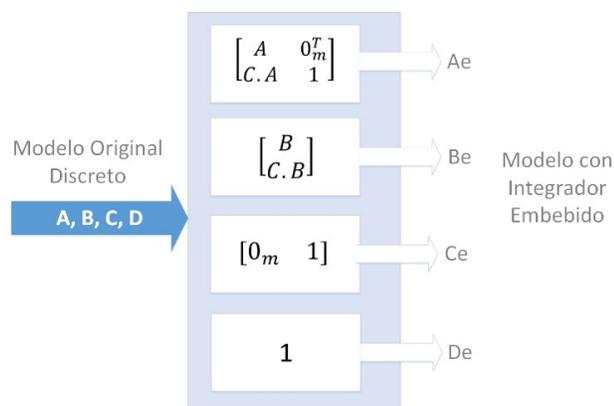


Figura 4.8 Diagrama esquemático para la implementación del modelo aumentado.

Como se puede observar en la figura 4.8 la implementación de un integrador embebido exige la modificación de las matrices de estado, esta modificación se puede lograr fácilmente utilizando un lenguaje de programación. La tabla 4.1 muestra el resultado de la implementación de un integrador embebido en un modelo SISO y MIMO respectivamente.

Tabla 4.1 Resultado de la Implementación de un integrador embebido.

Modelo original	Modelo con integrador embebido
$A = \begin{bmatrix} -0.622 & -0.027 & 0.0810 & 0.007 \\ 2.3980 & -0.594 & -0.119 & 0.035 \\ 0.5570 & 0.3800 & 0.9720 & 0.092 \\ -7.626 & 0.1770 & 0.3810 & 0.028 \end{bmatrix}$ $B = \begin{bmatrix} 0.001 \\ 0.007 \\ 0.037 \\ 0.092 \end{bmatrix} \quad C = [1 \ 0 \ 0 \ 0]$	$Ae = \begin{bmatrix} -0.622 & -0.027 & 0.0810 & 0.007 & 0 \\ 2.3980 & -0.594 & -0.119 & 0.035 & 0 \\ 0.5570 & 0.3800 & 0.9720 & 0.092 & 0 \\ -7.626 & 0.1770 & 0.3810 & 0.028 & 0 \\ -0.622 & -0.027 & 0.0810 & 0.007 & 1 \end{bmatrix}$ $Be = \begin{bmatrix} 0.001 \\ 0.007 \\ 0.037 \\ 0.092 \\ 0.001 \end{bmatrix} \quad Ce = [0 \ 0 \ 0 \ 1]$
$A = \begin{bmatrix} -0.002 & 0.000 & 0.000 \\ 0.0800 & -0.100 & 0.000 \\ -0.007 & 0.000 & 0.000 \end{bmatrix}$ $B = \begin{bmatrix} 0.900 & -0.374 & -0.150 \\ 0.000 & 15.190 & 0.000 \\ 0.000 & -0.148 & 1.658 \end{bmatrix}$ $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$Ae = \begin{bmatrix} -0.002 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.0800 & -0.100 & 0.000 & 0.000 & 0.000 \\ -0.007 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.002 & 0.000 & 0.000 & 1.000 & 0.000 \\ 0.0800 & -0.100 & 0.000 & 0.000 & 1.000 \end{bmatrix}$ $Be = \begin{bmatrix} 0.900 & -0.374 & -0.150 \\ 0.000 & 15.190 & 0.000 \\ 0.000 & -0.148 & 1.658 \\ 0.900 & -0.374 & -0.150 \end{bmatrix}$ $Ce = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

El incremento de estados está en función del número de salidas del sistema, en el caso del modelo SISO se incrementa un estado con respecto al modelo

original, por otra parte para el modelo MIMO de tres entradas y dos salidas se incrementan dos estados con respecto al modelo original.

4.3.2 Fase de Inicialización.

En la fase de inicialización del algoritmo de control se calculan las matrices F y Φ . Estas matrices forman parte de la ecuación de predicción $Y = F \cdot x(k_i) + \Phi \Delta U$ y permanecen constantes durante todo el proceso de ejecución. Esta fase se le denomina así ya que corresponde a la primera etapa del algoritmo predictivo y no forma parte del proceso iterativo, es decir estas matrices son calculadas una sola vez en el controlador sin la necesidad de que haya conexión con la planta.

Las matrices F y Φ dependen del modelo matemático de predicción con integrador embebido (Ae, Be, Ce, De) , además la matriz F depende del horizonte de predicción y la matriz Φ depende del horizonte de control y predicción, es decir a través de ambos horizontes estas matrices en su contenido heredan aspectos de la sintonía del controlador, por lo tanto únicamente si el controlador no satisface las condiciones de diseño la fase de inicialización deberá ser ejecutada nuevamente considerando valores de horizonte de control y predicción diferentes. La tabla 4.2 proporciona información sobre las dimensiones de estas matrices, aquí se deja en claro que las dimensiones de estas matrices pueden ser muy grandes, lo cual

obviamente demanda un coste computacional considerable, por ejemplo para un sistema MIMO de cuarto orden con 2 entradas y 3 salidas y con un horizonte de predicción y control de 12 y 8 respectivamente, las matrices F y Φ tendrían una dimensión de 36 por 5 y 36 por 16 respectivamente. Cabe indicar que el número de estados hace referencia al modelo aumentado.

Tabla 4.2 Dimensión de las matrices F y Φ

Matriz F	Matriz Φ
Número de filas: Número de salidas * Horizonte de predicción	Número de filas: Número de salidas * Horizonte de predicción
Número de columnas: Número de estados	Número de columnas: Número de entradas * Horizonte de control

Es por esta razón que los algoritmos de control predictivo desde su origen han constituido su nicho en ordenadores personales, sin embargo este coste computacional no tiene consecuencia alguna en el desempeño del controlador ya que el cálculo de las matrices F y Φ se realiza fuera de línea en la etapa de inicialización. Las expresiones 4.1 y 4.2 muestran el cálculo de las matrices F y Φ para el modelo SISO mostrado en la tabla 4.1. Se considera valores de 5 y 3 para el horizonte de predicción y control respectivamente.

$$F = \begin{bmatrix} -0.622 & -0.027 & 0.081 & 0.007 & 1 \\ -0.314 & 0.0380 & 0.115 & 0.105 & 1 \\ 0.3500 & 0.0040 & 0.167 & 0.016 & 1 \\ -0.425 & 0.0460 & 0.221 & 0.020 & 1 \\ -0.289 & 0.0440 & 0.264 & 0.025 & 1 \end{bmatrix} \quad (4.1)$$

$$\Phi = \begin{bmatrix} 0.001 & 0.000 & 0.000 \\ 0.004 & 0.001 & 0.000 \\ 0.006 & 0.004 & 0.001 \\ 0.008 & 0.006 & 0.004 \\ 0.011 & 0.008 & 0.006 \end{bmatrix} \quad (4.2)$$

El proceso de inicialización continúa con la formulación de la matriz Hessiana $H = (\Phi' \cdot \Phi + R)^{-1}$. Esta formulación requiere la matriz diagonal de pesos R que depende del parámetro de sintonización $r\omega$. Este parámetro da mayor o menor atención a los incrementos de control ΔU al momento de minimizar el error cuadrático. Las expresiones 4.3 y 4.4 muestran la formulación de la matriz R y el cálculo de la matriz Hessiana H para el modelo SISO tratado previamente. Se considera $r\omega = 0.1$ y $Nc = 3$.

$$R = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \quad (4.3)$$

$$H = \begin{bmatrix} 0.1002 & 0.0002 & 0.0001 \\ 0.0002 & 0.1001 & 0.0001 \\ 0.0001 & 0.0001 & 0.1001 \end{bmatrix} \quad (4.4)$$

En el caso de que el controlador no tenga restricciones el proceso de inicialización concluirá con el cálculo de la matriz $Ct = H^{-1}\Phi^T$, la cual interviene en la formulación de la acción de control para el instante de tiempo

k. Cabe indicar que las matrices H , F y Φ heredan los parámetros de sintonía y sirven para el proceso iterativo o fase de ejecución. Las figuras 4.9 a) y 4.9 b) muestran el coste computacional de la fase de inicialización para el ejemplo SISO citado en la tabla 4.1. Para el primer caso se asume un horizonte de predicción variable y se mantiene constante el horizonte de control, para el segundo caso se varía el horizonte de control frente a un horizonte de predicción constante.

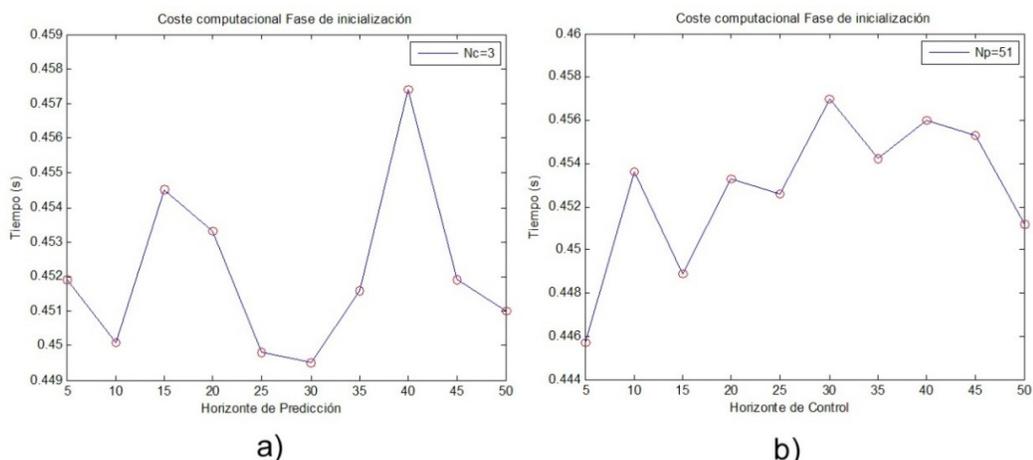


Figura 4.9 Coste computacional de la fase de inicialización para un caso específico.

Los resultados de la figura 4.9 se han obtenido utilizando un procesador Intel CORE(TM) I7 de 2.2 GHz. Aparentemente el coste computacional de la fase de inicialización debería tener una tendencia ascendente cuando los valores de N y M se hacen más grandes ya que el tamaño de las matrices F y Φ aumenta, sin embargo se puede notar en las gráficas de la figura 4.9 que

esto no tiene incidencia alguna en el cálculo realizado en MATLAB® debido a las facultades que posee esta poderosa herramienta para la manipulación de matrices. Otro aspecto que afecta considerablemente para que no se pueda observar una tendencia ascendente y que se manifiesta como inconsistencias, es la característica del sistema operativo Windows 7 que no opera en tiempo real, es decir las operaciones no se realizan a paso constante debido a las interrupciones generadas en el procesador.

En el caso de que el controlador sea con restricciones ($M \cdot x \leq \gamma$) la fase de inicialización puede extenderse con la formulación de la matriz de restricción M y la formulación inicial de la matriz γ . Si se hace referencia a la ecuación 3.109 claramente se puede notar que la matriz γ no es constante durante el proceso iterativo ya que las restricciones de amplitud de la variable de control y variable de salida dependen de la acción de control pasada y los estados presentes respectivamente, obviamente esto hace que el controlador predictivo formule una matriz γ diferente para el problema de optimización durante cada iteración.

La figura 4.10 muestra la formulación de la matriz M y la formulación inicial de la matriz γ para el caso de la planta SISO que se ha manejado durante toda esta subsección. Se considera la aplicación de los tres tipos de restricciones durante todos los puntos del horizonte de control. Además se

considera valores de 5 y 3 para el horizonte de predicción y control respectivamente.

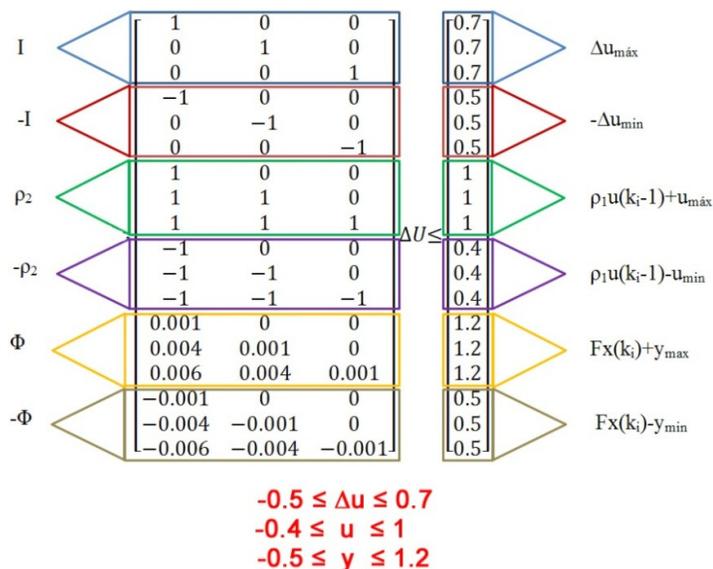


Figura 4.10 Formulación de restricciones para el caso SISO con $N=5$ y $M=3$

Las matrices de restricción M y γ están conformadas por sub-matrices formuladas en base a las restricciones individuales impuestas, estas submatrices pueden ser colocadas en un orden indistinto siempre y cuando los elementos de M correspondan a los elementos de γ . Para el caso de la figura 4.10 se ha asumido que la acción de control anterior $u(k_i - 1)$ y los estados actuales $x(k_i)$ son igual a cero, sin embargo cuando el algoritmo entre en la fase iterativa estos valores serán actualizados para cada iteración dependiendo de la dinámica de la planta.

Para el caso multivariable también existe la posibilidad de formular la matriz de restricciones colocando las sub-matrices individuales en un orden indistinto. La figura 4.11 muestra una alternativa para formular restricciones en un sistema MIMO de tres entradas. Para este caso específico se asume restricciones solamente en la tasa de cambio de la variable manipulada, estas son aplicadas en dos puntos del horizonte de control.

$$\begin{array}{c}
 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \Delta U \leq \begin{bmatrix} 0.5 \\ 1 \\ 2.4 \\ 0.4 \\ 2 \\ 6 \\ 0.5 \\ 1 \\ 2.4 \\ 0.4 \\ 2 \\ 6 \end{bmatrix} \\
 \begin{array}{l} \Delta u_{\text{máx}1} \\ \Delta u_{\text{máx}2} \\ \Delta u_{\text{máx}3} \\ -\Delta u_{\text{mín}1} \\ -\Delta u_{\text{mín}2} \\ -\Delta u_{\text{mín}3} \\ \Delta u_{\text{máx}1} \\ \Delta u_{\text{máx}2} \\ \Delta u_{\text{máx}3} \\ -\Delta u_{\text{mín}1} \\ -\Delta u_{\text{mín}2} \\ -\Delta u_{\text{mín}3} \end{array} \left. \begin{array}{l} \text{Primer punto dentro del} \\ \text{horizonte de predicción} \\ \\ \\ \text{Segundo punto dentro del} \\ \text{horizonte de predicción} \end{array} \right\}
 \end{array}$$

$$-0.4 \leq \Delta u_1 \leq 0.5 \quad -2 \leq \Delta u_2 \leq 1 \quad -6 \leq \Delta u_3 \leq 2.4$$

Figura 4.11 Formulación de restricciones en Δu para el caso MIMO

4.3.3 Proceso iterativo o fase de ejecución.

Esta es la fase donde el algoritmo predictivo interactúa con la planta, por lo tanto el coste computacional con el que se generan los cálculos incidirá directamente en el desempeño del controlador. Este efecto que se hará

presente a través de demoras podrá ser visualizado en la modalidad “en línea” de la plataforma.

En el caso de que el controlador no tenga restricciones el proceso iterativo queda determinado directamente por la ecuación que permite el cálculo de la acción de control: $\Delta U = H \cdot \Phi^T [\bar{R}s.r(k_i) - F \cdot x(k_i)]$. Donde $\bar{R}s.r(k_i)$ es el vector de puntos de ajuste y la existencia de la matriz Hessiana condiciona la posibilidad de aplicar la estrategia de control predictivo bajo los parámetros de sintonía indicados en la fase de inicialización. La figura 4.12 muestra un diagrama de bloques del proceso iterativo que realiza el algoritmo de control predictivo, aquí se puede observar claramente como el algoritmo de ejecución utiliza una retroalimentación de estado.

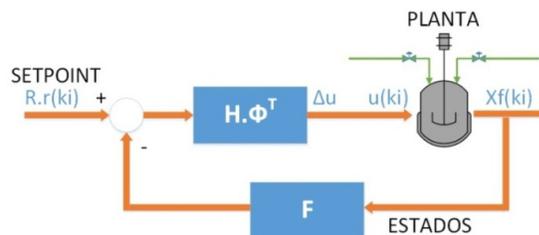


Figura 4.12 Proceso iterativo del algoritmo MPC para el caso sin restricciones.

La longitud del vector ΔU es igual al horizonte de control. El algoritmo predictivo debe encargarse de tomar únicamente el primer elemento del vector y enviarlo a la planta para cada período de iteración. Para el caso

multivariable el vector ΔU tiene longitud igual al horizonte de control multiplicado por el número de entradas n . En este caso el algoritmo predictivo debe encargarse de tomar los primeros n elementos del vector y enviarlos a la planta para cada período de iteración. Las figuras 4.13 a) y b) muestran el coste computacional que demanda el cálculo de la acción de control durante cada período de iteración para un MPC SISO y MIMO (3x3) respectivamente. Para ambos casos se ha considerado horizontes de predicción y control de 7 y 5 respectivamente y se ha ejecutado una simulación para 25 períodos de muestreo.

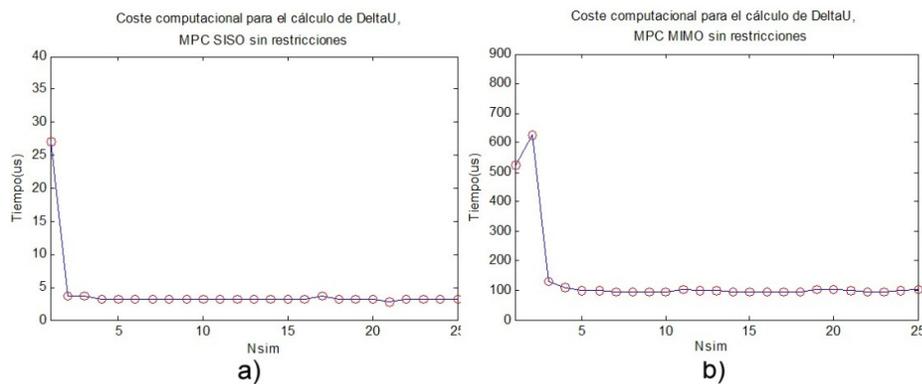


Figura 4.13 Coste computacional del algoritmo MPC sin restricciones

Para la obtención de los resultados mostrados en la figura 4.13 se ha utilizado un procesador Intel CORE(TM) I7 de 2.2 GHz. Cabe indicar que únicamente se está considerando el tiempo de cálculo más no el tiempo que se tarda el controlador en adquirir los estados de la planta. Claramente se puede observar como el tiempo de cálculo entre un MPC SISO y MIMO

difieren considerablemente, esto se debe al incremento del tamaño de las matrices H , Φ y F que se generaron en la fase de inicialización y que son utilizadas por el algoritmo iterativo para el cálculo de Δu .

En el caso de que se impongan restricciones al controlador predictivo el proceso iterativo puede tomar caminos diferentes en cada período de iteración. El algoritmo en esta parte difiere en función de a qué variable se impuso las restricciones y de las características de las restricciones formuladas. Atendiendo a la variable a la que se impuso las restricciones se puede tener los siguientes casos:

- Caso 1: Cuando las restricciones son aplicadas únicamente a la tasa de cambio de la variable de control ΔU . En este caso las matrices de restricción M y γ son constantes durante todo el proceso iterativo.
- Caso 2: Cuando las restricciones son aplicadas a la amplitud de la variable de control y/o amplitud de la variable de salida independientemente de si se impone o no restricciones a la tasa de cambio de la variable de control ΔU . En este caso parte de la matriz de restricción γ es variable por lo tanto para cada iteración se formula una nueva matriz, esto obviamente implica un aumento del coste computacional durante la fase de ejecución.

Atendiendo a las características de las restricciones formuladas se puede tener los siguientes casos:

- Caso 3: Cuando las restricciones son inactivas. En este caso en la iteración k_i se cumple que $M.\Delta U \leq \gamma$, entonces se extrae el primer valor de ΔU y se envía a la planta.
- Caso 4: Cuando las restricciones son activas. En este caso en la iteración k_i no se cumple que $M.\Delta U \leq \gamma$ por lo tanto es necesario calcular un nuevo ΔU utilizando un algoritmo de optimización QP como por ejemplo el propuesto por Hildreth's que se estudió en la sección 3.5.

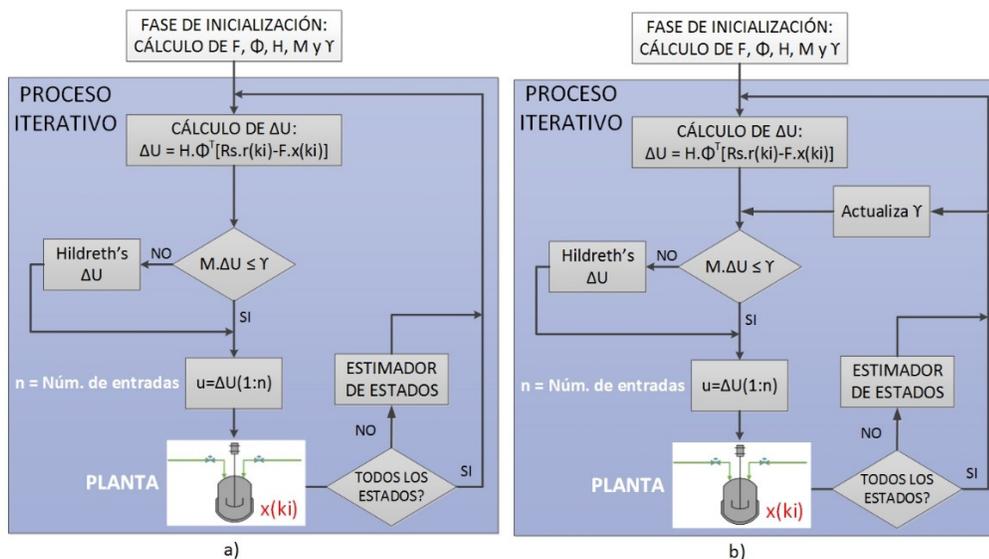


Figura 4.14 Proceso iterativo del algoritmo MPC con restricciones

Los casos 1 y 2 hacen referencia a la formulación de las restricciones y los casos 3 y 4 hacen referencia a la forma en la que se llega al cumplimiento de estas. Por lo tanto para cada iteración puede haber combinaciones del caso uno o dos con el caso tres o cuatro respectivamente. Esto se puede ilustrar de mejor manera en los diagramas de flujo de la figura 4.14. En a) se muestra el caso en el que se aplican restricciones únicamente a la tasa de cambio de la variable de control (caso 1). En b) se muestra el caso en el que se aplican los tres tipos de restricciones (caso 2). En ambos casos se observa claramente como el algoritmo puede incurrir en el caso 3 o 4 dependiendo de las características de las restricciones formuladas.

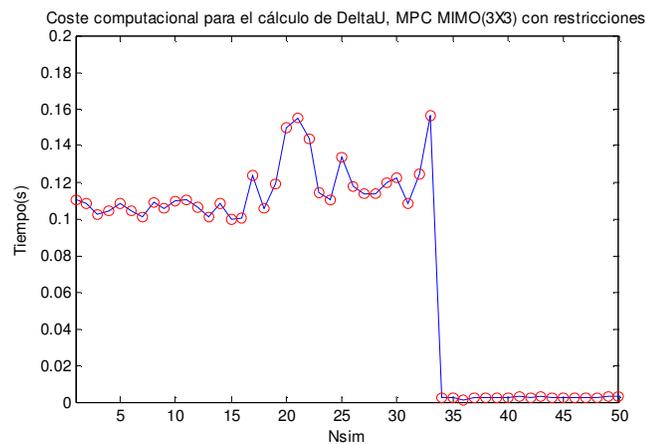


Figura 4.15 Coste computacional para un algoritmo MPC con restricciones.

Un aspecto muy importante en la fase de ejecución es el tiempo de demora que implica el cálculo de la acción de control ΔU . Este tiempo depende de si existe o no restricciones y de si estas son activas o inactivas durante cada

período de iteración. La figura 4.15 muestra el coste computacional para un sistema MIMO (3x3) con los tres tipos de restricciones. Se considera valores de $N = 8$ y $M = 3$, además se ha impuesto las 18 restricciones sobre todos los puntos del horizonte de control. Para la ejecución se ha utilizado un procesador Intel CORE(TM) I7 de 2.2 GHz.

Como se puede observar en la figura 4.15 el tiempo de ejecución de un algoritmo MPC para el cálculo de la acción de control no es constante, esto se debe a dos aspectos fundamentales: el primero hace referencia a la forma en la que se presentan las restricciones para cada iteración. En el caso de que estas sean inactivas el tiempo de cálculo de ΔU es menor, por otra parte si las restricciones son activas el algoritmo recae en un problema de programación cuadrática (Hildreth's), obviamente este cálculo demanda mayor tiempo. El segundo aspecto hace referencia al algoritmo QP, el cual es un algoritmo iterativo finito que depende de algunos parámetros de convergencia.

4.3.4 Ambiente de la plataforma SIMUMPC.

La plataforma SIMUMPC brinda al usuario un ambiente altamente interactivo que permite simular y analizar el comportamiento de sistemas dinámicos sometidos a la acción de un controlador predictivo. En el presente epígrafe se detalla brevemente las principales ventanas y herramientas con las que

cuenta. Mayores detalles se pueden encontrar en el manual de usuario del anexo 1.

El PC controlador consta de tres ventanas principales. La primera ventana (véase la figura 4.16) no es más que una introducción que ilustra al usuario el modo de trabajo del simulador. Al presionar el botón inicio una nueva ventana denominada simMPC se abrirá (véase la figura 4.17). En el ambiente simMPC el usuario puede contar con los siguientes menús desplegables:

- 1. Modelo de planta.** Permite seleccionar la planta a controlar. Estos modelos se encuentran almacenados en una base de datos del programa. El usuario podrá seleccionar algunas alternativas desde un simple motor dc hasta una planta multivariable no lineal.
- 2. Modelo de predicción.** Permite seleccionar el modelo matemático que el controlador utilizará para calcular las predicciones. El usuario podrá seleccionar un modelo idéntico al de la planta a controlar, o podrá ingresar manualmente un modelo diferente. Además existe la opción de identificar un modelo y exportarlo a la plataforma de simulación.
- 3. Modo de funcionamiento.** Permite seleccionar una simulación fuera de línea o en línea. La simulación fuera de línea no requiere de un

segundo computador pues la dinámica de la planta es recreada en el mismo PC.

4. **Comunicación.** Permite configurar las direcciones IP y los puertos de comunicación que requiere el protocolo UDP. Esta configuración deberá hacerse en concordancia con las IPs del PC planta y el PC controlador. Este menú es accesible únicamente cuando se ha seleccionado el modo en línea.

5. **Ayuda.** Brinda al usuario información necesaria para que pueda operar la herramienta de simulación. Aquí existe un acceso directo que abre un manual de usuario, el mismo que se puede encontrar en el apéndice 1.



Figura 4.16 Ventana de inicio de la herramienta SIMUMPC.

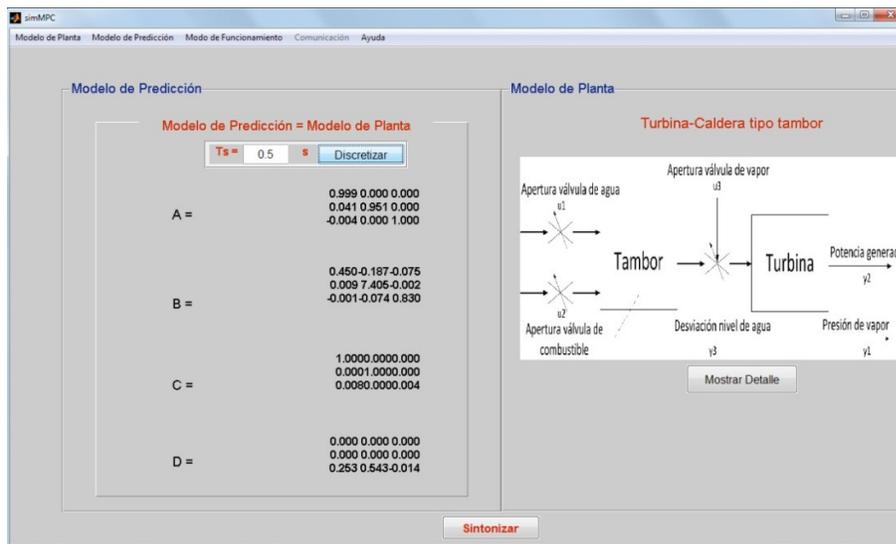


Figura 4.17 Ambiente simMPC.

Cunado se ha seleccionado el modo fuera de línea el botón sintonizar abre una tercera ventana denominada simMPCejec (véase la figura 4.18). En el caso de que se haya seleccionado el modo en línea esta misma venta será abierta siempre y cuando se haya comprobado que exista comunicación entre los dos PCs. Para esto el cable de red debe estar debidamente conectado, las direcciones IP y puertos configurados adecuadamente y el PC - planta colocado en modo de espera. De no ser este el caso la plataforma emitirá un mensaje pidiendo que se verifique que se cumplan estas condiciones.

El ambiente simMPCejec permite configurar los parámetros de sintonía del controlador, incorporar restricciones, ejecutar la simulación y observar los

resultados. Una vez terminada la simulación el botón “Analizar” permite visualizar las gráficas fuera de la plataforma en el formato estándar de MATLAB®.

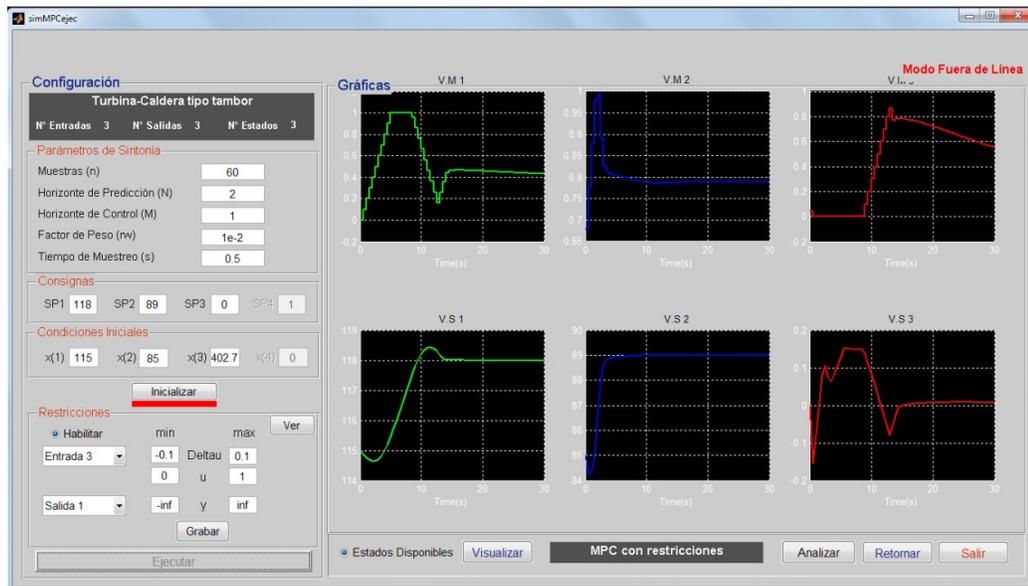


Figura 4.18 Ambiente simMPCejec.

4.4 Pruebas del sistema y análisis de resultados.

En esta sección se describen los resultados gráficos de la implementación del algoritmo. Atendiendo a la estructura de la plataforma SIMUMPC se abordan los dos modos de funcionamiento. En el modo fuera de línea se prueba el desarrollo del algoritmo tanto para el caso SISO como para el caso MIMO. En esta modalidad se muestra el desempeño del controlador en base a los parámetros de sintonía y a las restricciones impuestas, cabe indicar que

en el modo fuera de línea la ejecución se realiza en un mismo PC y los datos que se obtienen se almacenan en vectores y luego de que concluye el tiempo de simulación se construyen las gráficas.

En la modalidad en línea la acción de control es calculada y enviada hacia otro PC. En esta modalidad se incluyen otros parámetros como el tiempo de muestreo, retardos y ruido aleatorio sumado a los estados de la planta, además es posible realizar una estimación de estados. En esta modalidad se trata de llegar a la sintonía del controlador utilizando tres plantas diferentes, dos de estas se utilizan en el modo fuera de línea de manera que se pueda contrastar los resultados.

4.4.1 Modo fuera de línea: MPC SISO sin restricciones.

Para los resultados presentados en esta subsección se ha utilizado una planta que consiste en un servomecanismo de posición (véase la figura 4.19). Esta planta es representada dentro del PC por un conjunto de ecuaciones diferenciales y el modelo de predicción ha sido transformado a espacio de estados discreto. Este modelo de planta se ha utilizado en [84] para la implementación de un controlador PID de orden fraccional. Las expresiones 4.5 y 4.6 muestran las ecuaciones dinámicas del sistema y el modelo discretizado para ser utilizado en el controlador.

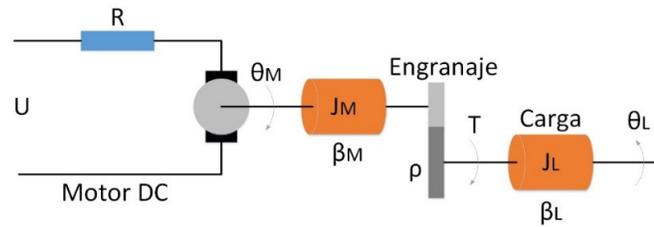


Figura 4.19 MODELO SIS0: Servomecanismo de posición.

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_\theta}{J_L} & -\frac{\beta_L}{J_L} & \frac{k_\theta}{\rho J_L} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_\theta}{\rho J_M} & 0 & -\frac{k_\theta}{\rho^2 J_M} & k_R \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_T}{R J_{LM}} \end{bmatrix} u \quad (4.5)$$

$$\theta_L = [1 \ 0 \ 0 \ 0]x \quad (4.6)$$

En este sistema: $K_\theta=1280.2$, $K_T=10$, $J_M=0.5$, $J_L=25$, $\rho=20$, $B_M=0.1$, $B_L=0.25$, $R=20$ $|U|<220V$ y $K_R = -(\beta_M + K_T^2/R)/J_M$. El modelo de predicción discretizado con $T_s = 0.5s$ se representa a través de las expresiones 4.7, 4.8 y 4.9.

$$Ad = \begin{bmatrix} -0.62 & -0.02 & 0.080 & 0 \\ 2.390 & -0.59 & -0.11 & 0 \\ 0.550 & 0.380 & 0.970 & 0.09 \\ -7.62 & 0.170 & 0.380 & 0.02 \end{bmatrix} \quad (4.7)$$

$$Bd = \begin{bmatrix} 0 \\ 0 \\ 0.03 \\ 0.09 \end{bmatrix} \quad (4.8)$$

$$Cd = [1 \ 0 \ 0 \ 0] \quad (4.9)$$

La entrada del sistema es el voltaje de armadura U y la salida es la posición angular θ_L . La figura 4.20 muestra el resultado de la ejecución del algoritmo utilizando un modelo de planta igual al modelo de predicción. Se considera dos conjuntos de valores de horizonte de predicción y control respectivamente para ver la influencia de estos parámetros de sintonía dentro del desempeño del controlador. La consigna del sistema es $\theta_L = 1\text{rad}$.

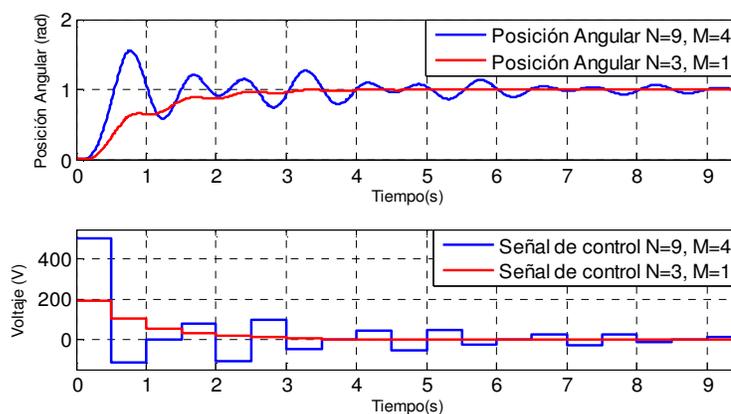


Figura 4.20 Ejecución del algoritmo MPC sobre planta SISO para dos conjuntos diferentes de horizonte de control y predicción.

Mediante las gráficas de la figura 4.20 se puede notar la efectividad del algoritmo para llevar el sistema a la consigna. Para el caso en el que $N=9$ y $M=4$ se logra la sintonía del controlador pero el transitorio es muy prolongado y la amplitud de la señal de control sobrepasa la tensión nominal del motor. Al mover los parámetros de sintonía a $N=3$ y $M=1$ el transitorio mejora notablemente y la amplitud de la señal de control disminuye favorablemente.

Para los dos casos de la figura 4.20 se ha mantenido constante del factor de peso $r\omega$. Este es otro factor preponderante para la sintonía del controlador predictivo, pues da mayor o menor atención a los incrementos de la variable de control ΔU , lo cual obviamente tiene gran incidencia sobre la respuesta del sistema. La figura 4.21 muestra los efectos del parámetro $r\omega$ en la sintonía del controlador predictivo. Se han considerado dos casos con valores diferentes de $r\omega$, para ambos casos $N = 12$ y $M = 9$.

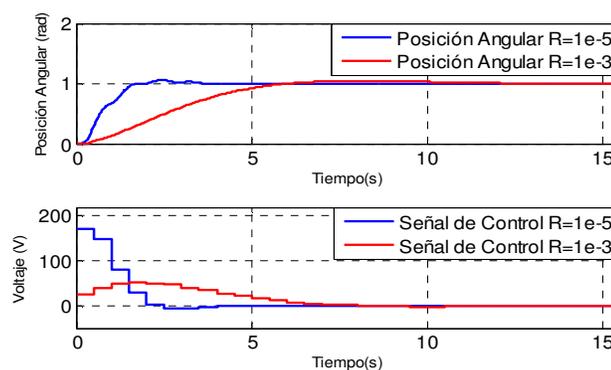


Figura 4.21 Ejecución del algoritmo MPC sobre planta SISO para dos conjuntos diferentes del factor de peso R.

Si se compara los dos casos de la figura 4.21 se puede observar como una disminución del factor de peso da mayor atención a los incrementos de control haciendo que esta señal tenga movimientos más suaves, esto afecta directamente a la respuesta del sistema, lo cual para este caso se ve reflejado en el incremento del tiempo de estabilización.

Atendiendo a otro factor muy importante y que tiene gran incidencia en el desempeño de un controlador predictivo, la figura 4.22 muestra el comportamiento del sistema cuando el modelo de predicción utilizado en el controlador no es igual al modelo de la planta.

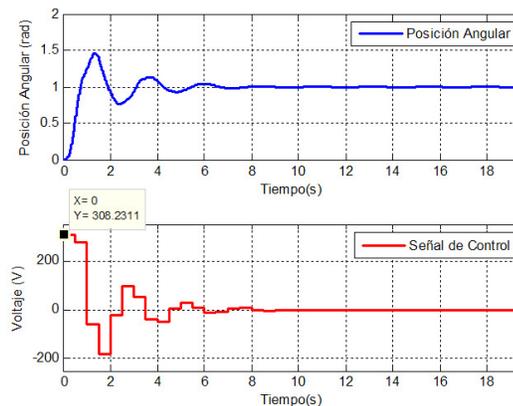


Figura 4.22 Desempeño del controlador predictivo cuando el modelo de predicción difiere del modelo de la planta.

El modelo de predicción distinto al de la planta está descrito por las expresiones 4.10 y 4.11. Los parámetros de sintonía para el experimento son: $N = 9$, $M = 4$ y $r\omega = 4e - 6$.

$$Ad^* = \begin{bmatrix} 0.130 & -0.05 & 0.020 & 0 \\ 6.020 & -0.23 & -0.17 & 0 \\ -0.21 & 0.060 & 1.000 & 0.04 \\ 0.720 & 0.320 & 0.020 & 0 \end{bmatrix} \quad (4.10)$$

$$Bd^* = \begin{bmatrix} 0 \\ 0 \\ 0.03 \\ 0.07 \end{bmatrix} \quad (4.11)$$

Como se puede observar en la figura 4.22 ha desmejorado el desempeño del controlador debido al modelo utilizado en la predicción. Sin embargo la acción del integrador embebido ha logrado que se llegue a la consigna, obviamente sin los parámetros de respuesta temporal ideales, además el esfuerzo de control ha sobrepasado el valor de tensión nominal de la planta.

4.4.2 Modo fuera de línea: MPC SISO con restricciones.

Para el conjunto de pruebas presentadas en esta subsección se continúa utilizando el servomecanismo de posición. Las figuras 4.23 a) y b) muestran el comportamiento del sistema cuando se imponen restricciones a la tasa de cambio de la variable de control. Claramente se puede notar la diferencia en el comportamiento de Δu . Para el primer caso la tasa de cambio de la variable de control no excede el valor de 50 y para el segundo caso el valor de 10, obviamente esto implica cambios en la respuesta del sistema.

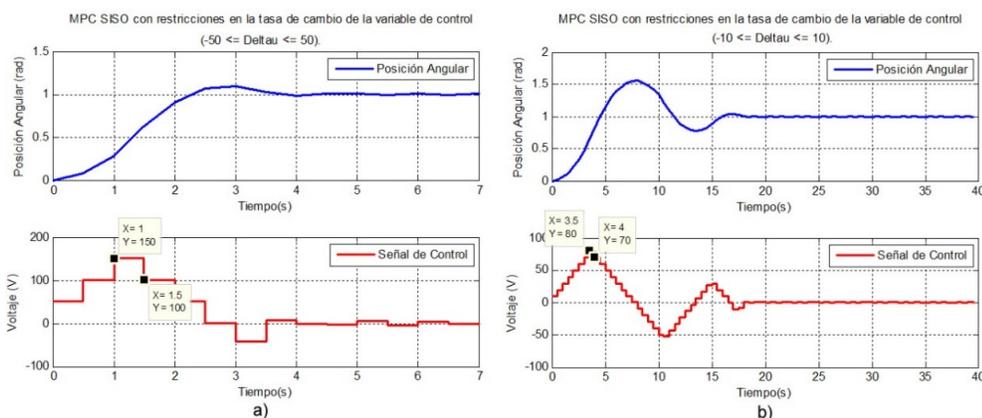


Figura 4.23 MPC SISO con restricciones en Δu .

Como se puede observar en las figuras 4.23 a) y b) el controlador lleva a la salida del sistema al valor de referencia deseado cumpliendo con las restricciones impuestas. Esto demuestra la efectividad del algoritmo de Hildreth's para resolver un problema de optimización con restricciones. Cabe indicar que este es un proceso iterativo por lo tanto el número de iteraciones para resolver el problema QP para cada período de muestreo constituye otro parámetro de sintonía en el controlador predictivo con restricciones. La figura 4.24 muestra un caso en el que se ha seleccionado un valor muy bajo para el número de iteraciones. Claramente se puede notar como el algoritmo de control cumple con la consigna, sin embargo las restricciones impuestas ($-10 \leq \Delta u \leq 10$) han sido violadas en casi todo el tiempo de ejecución.

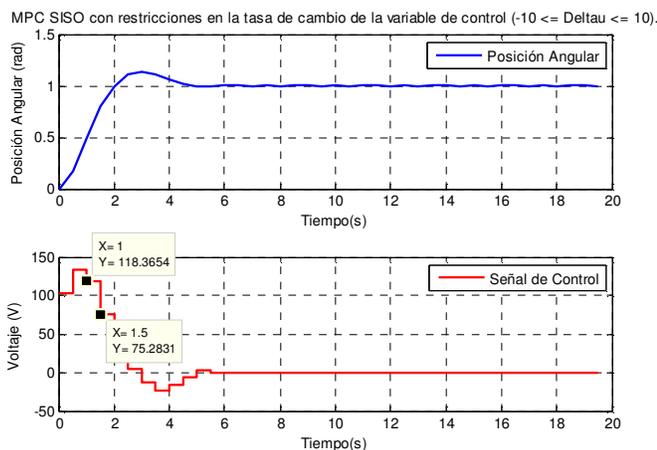


Figura 4.24 Incumplimiento de restricciones en el algoritmo MPC

Para evitar este inconveniente no existe una regla que determine el valor óptimo para el número de iteraciones, pero se debe seleccionar un valor

considerable y experimentar bajo la modalidad prueba – error. Obviamente que un valor alto de iteraciones hará que el cálculo de la señal de control demore más pero se garantizará en cierto modo el cumplimiento de las restricciones.

Continuando con el análisis de la implementación del algoritmo se incorpora un nuevo tipo de restricción a la acción de control. A más de restringir la tasa de cambio se restringe la amplitud de esta variable de manera simultánea. En la figura 4.25 se puede ver como el algoritmo da cumplimiento a lo impuesto: ($0 \leq u \leq 60$) y ($-15 \leq \Delta u \leq 15$).

En el caso SISO no tiene mucho sentido incorporar restricciones en la amplitud de la variable de salida. El efecto de este tipo de restricciones puede ser más contundente en el caso multivariable donde el acoplamiento de variables de salida puede ser controlado.

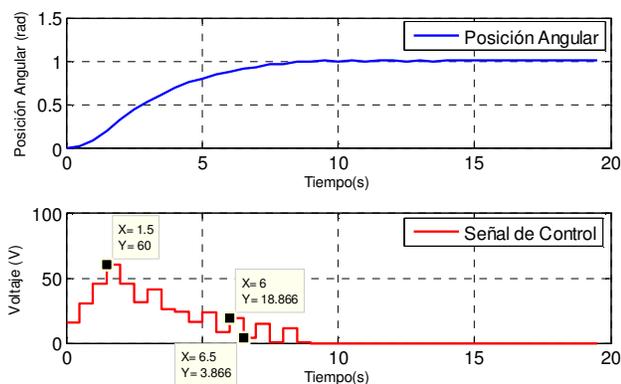


Figura 4.25 Ejecución del algoritmo MPC con restricciones en Δu y u

4.4.3 Modo fuera de línea: MPC MIMO sin restricciones.

Para los resultados presentados en esta subsección se ha utilizado una planta que consiste de un conjunto turbina - caldera tipo tambor (véase la figura 4.26). Esta planta es representada dentro del PC por un sistema no lineal de ecuaciones diferenciales y el modelo de predicción ha sido linealizado y transformado a espacio de estados discreto. Este modelo de planta propuesto por [27] se ha utilizado para la implementación de un controlador predictivo DMC. Las expresiones 4.12 a 4.19 muestran las ecuaciones dinámicas que gobiernan el sistema no lineal, por otra parte las expresiones 4.20, 4.21 y 4.22 muestran las matrices de estado linealizadas, en $y_1 = 115kg/cm^2$, $y_2 = 85MW$ y $y_3 = 0$ para ser utilizadas por el controlador como modelo de predicción.

$$\dot{x}_1 = -0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3 \quad (4.12)$$

$$\dot{x}_2 = \frac{[(0.73u_2 - 0.16)x_1^{9/8} - x_2]}{10} \quad (4.13)$$

$$\dot{x}_3 = \frac{[114u_3 - (1.1u_2 - 0.19)x_1]}{85} \quad (4.14)$$

$$y_1 = x_1 \quad (4.15)$$

$$y_2 = x_2 \quad (4.16)$$

$$y_3 = 0.05 \left(0.13x_3 + 100a_{cs} + \frac{q_e}{9} - 67.97 \right) \quad (4.17)$$

$$a_{cs} = \frac{(1 - 0.001538x_3)(0.81x_1 - 25.6)}{x_3(1.0394 - 0.0012304x_1)} \quad (4.18)$$

$$q_e = (0.854u_2 - 0.147x_1) + 45.59u_1 - 2.51u_3 + 2.096 \quad (4.19)$$

$$Ad^* = \begin{bmatrix} -0.0285 & 0 & 0 \\ 0.083152 & -0.1 & 0 \\ -0.00784 & 0 & 0 \end{bmatrix} \quad (4.20)$$

$$Bd^* = \begin{bmatrix} 0.9 & -0.37 & -0.15 \\ 0 & 15.19 & 0 \\ 0 & -0.14 & 1.658 \end{bmatrix} \quad (4.21)$$

$$Cd^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.22)$$

La figura 4.27 muestra las respuestas del sistema cuando se ha ejecutado el algoritmo de control predictivo sin restricciones para la planta MIMO, considerando las consignas igual a los tres puntos de linealización. Por otra parte la figura 4.28 muestra las señales de control para llevar al sistema a los valores de referencia. Para este experimento se ha utilizado $N = 8$, $M = 4$ y $r\omega = 1$.

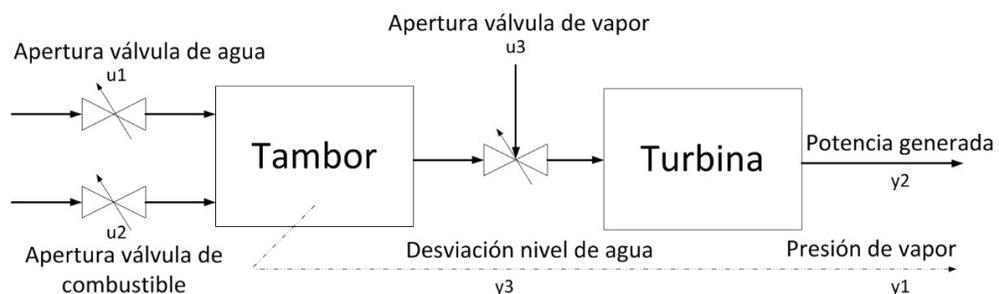


Figura 4.26 Turbina - caldera tipo tambor para pruebas del controlador MPC MIMO.

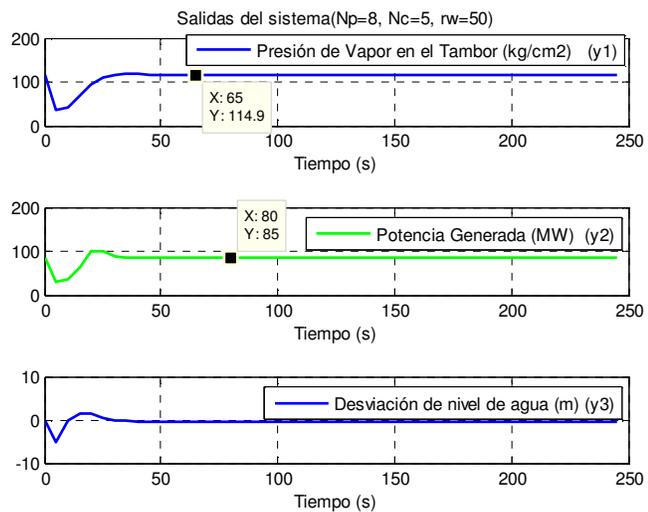


Figura 4.27 Respuestas del sistema turbina - caldera tipo tambor (MPC sin restricciones).

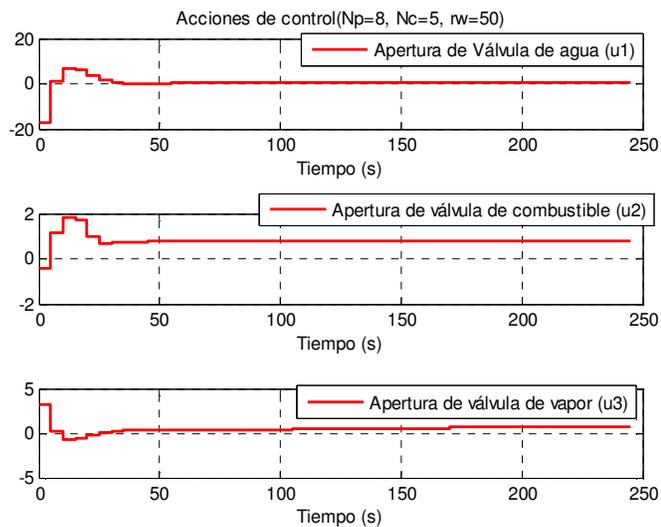


Figura 4.28 Acciones de control para el sistema turbina - caldera tipo tambor (MPC sin restricciones).

En las respuestas del sistema se puede observar claramente la efectividad del controlador predictivo para llevar al sistema a los valores de ajuste

deseados con parámetros del transitorio muy aceptables. Sin embargo las acciones de control violan notablemente la física de las válvulas pues el nivel de apertura máximo es la unidad, esto puede ser más notorio en la acción de control 1 dónde la apertura de esta válvula ha tomado en un instante el valor de -20.

4.4.4 Modo fuera de línea: MPC MIMO con restricciones.

La incorporación de restricciones en el controlador predictivo permite medir la amplitud y/o los incrementos de las variables, lo cual resulta de mucha utilidad para sistemas donde se requiera establecer rangos de operación mínimos y máximos de las variables de salida, o para respetar las especificaciones físicas de los elementos vinculados con la señal de control. Por ejemplo para el caso del sistema turbina - caldera tipo tambor que se muestra en la figura 4.28 las acciones de control toman valores que sobrepasan los límites de cierre y apertura de las válvulas [0 1], esta condición de las acciones de control ha generado que las salidas del sistema lleguen a las consignas con parámetros del transitorio muy aceptables, sin embargo en la vida real esto no ocurriría ya que la válvula 1 por ejemplo no podría abrirse un 315%. Para solventar este inconveniente se debe incorporar restricciones en las señales de control para que estas no sobrepasen los límites de cero y uno respectivamente, de igual manera se puede medir los incrementos y decrementos de las acciones de control.

Las figuras 4.29 y 4.30 muestran las salidas del sistema y las acciones de control cuando se ha ejecutado el algoritmo MPC con restricciones para el sistema turbina - caldera tipo tambor.

Las consignas establecidas y las restricciones impuestas se indican a continuación:

- Punto de ajuste 1 (*Presión de vapor* = 120kg/cm^2).
- Punto de ajuste 2 (*Potencia Generada* = 88MW).
- Punto de ajuste 3 (*Desviación del nivel de agua* = 0).
- Restricciones en el grado de apertura y cierre de las válvulas:
 $0 \leq u_1 \leq 1$; $0 \leq u_2 \leq 1$; $0 \leq u_3 \leq 1$.
- Restricciones en la tasa de cambio de la apertura y cierre de las válvulas: $-0.1 \leq \Delta u_1 \leq 0.1$; $-0.1 \leq \Delta u_2 \leq 0.1$; $-0.1 \leq \Delta u_3 \leq 0.1$.
- Restricciones en las salidas del sistema: $0 \leq y_1 \leq 130$; $0 \leq y_2 \leq 110$; $0 \leq y_3 \leq 10$.

Los parámetros con los que se ha sintonizado el controlador son $N = 3$, $M = 1$, $r\omega = 0.08$. Las gráficas de las acciones de control de la figura 4.30 muestran como la incorporación de restricciones evita que la apertura de las válvulas sobrepase sus límites físicos, además se nota como los incrementos

y decrementos son medidos de acuerdo a las restricciones impuestas en las tasas de cambio de las acciones de control.

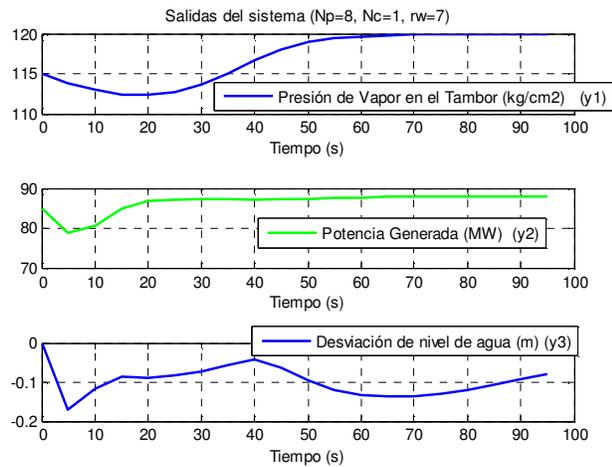


Figura 4.29 Salidas del sistema turbina - caldera tipo tambor (MPC con restricciones).

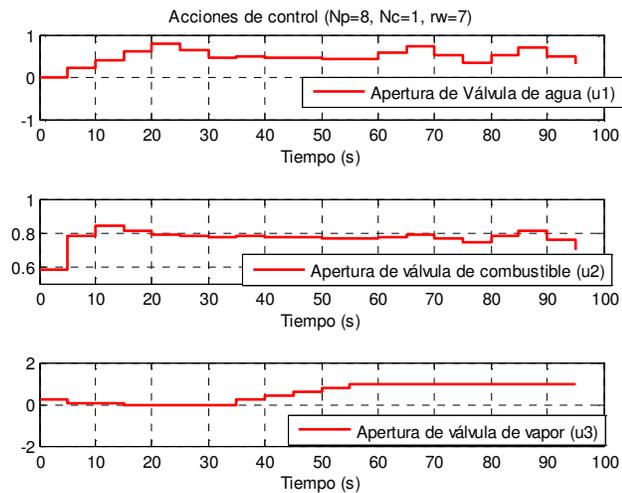


Figura 4.30 Acciones de control para el sistema turbina - caldera tipo tambor (MPC con restricciones).

4.4.5 Modo en Línea: MPC SISO.

Una vez validado el algoritmo MPC con restricciones la modalidad en línea pone en juego la eficacia del controlador sometiéndolo a ciertos parámetros influyentes en su desempeño y que inevitablemente están presentes en una aplicación real. Para esta modalidad se valida la plataforma utilizando 2 plantas diferentes: velocidad de un motor dc y el servomecanismo de posición que fue utilizado en la modalidad fuera de línea, con la intencionalidad de contrastar los resultados obtenidos.

La figura 4.31 muestra el modelo físico del motor dc para el control de velocidad. Los parámetros utilizados son: $J=0.01$; $b=0.1$; $K_e=0.1$; $K_t=0.1$; $R=0.2$; $L=0.5$. El modelo dinámico del sistema se expresa a través de las ecuaciones 4.23 y 4.24.

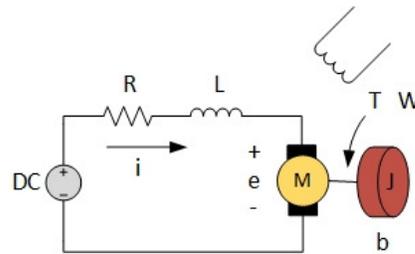


Figura 4.31 Modelo de un motor dc.

$$\frac{d\omega}{dt} = -\frac{b}{J}\omega + \frac{K_t}{J}i \quad (4.23)$$

$$\frac{di}{dt} = -\frac{R}{L}i - \frac{K_e}{L}\omega + \frac{1}{L}V \quad (4.24)$$

El voltaje nominal del motor dc es de 10v y la velocidad nominal de 33 rad/s. Las expresiones 4.25, 4.26 y 4.27 muestran las matrices A, B y C del sistema en tiempo continuo.

$$A = \begin{bmatrix} -10 & 10 \\ -0.2 & -0.4 \end{bmatrix} \quad (4.25)$$

$$B = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \quad (4.26)$$

$$C = [1 \quad 0] \quad (4.27)$$

La figura 4.32 muestra la respuesta del sistema para un punto de ajuste de 30 rad/s. Los parámetros de sintonía que se han utilizado son: $N=4$, $M=1$, $rw=10$ y $Ts=0.5$.

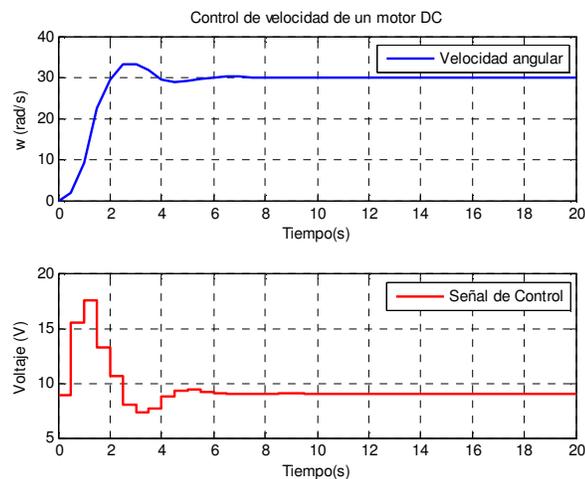


Figura 4.32 Simulación en línea: MPC sin restricciones para control de velocidad motor dc.

La gráfica superior de la figura 4.32 deja ver como el controlador se ha sintonizado logrando el punto de ajuste de 30 rad/s, sin embargo durante el

arranque se ha excedido la señal de control sobrepasando el valor nominal de tensión.

La figura 4.33 muestra el resultado de la simulación cuando se han impuesto restricciones en la tasa de cambio y en la amplitud de la variable de control: $(-2 \leq \Delta u \leq 2)$, $(0 \leq u \leq 10)$. Los parámetros de sintonía son los mismos que se utilizaron para la simulación sin restricciones. Como se puede observar en esta figura las restricciones han hecho que la acción de control no sobrepase el voltaje nominal del motor, sin embargo el tiempo de establecimiento es más grande. Esta situación se podría mejorar encontrando un nuevo juego de parámetros de sintonía.

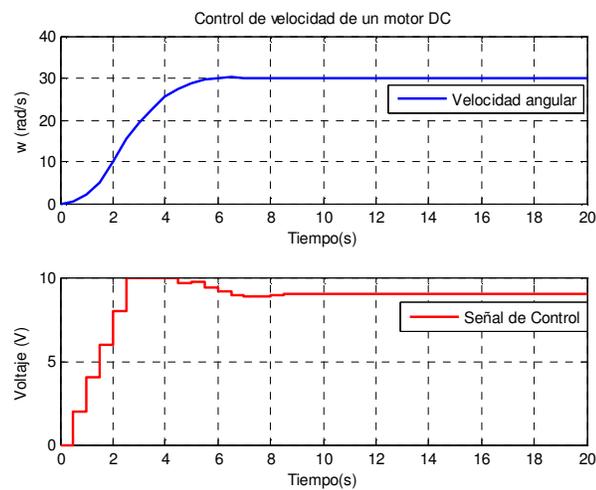


Figura 4.33 Simulación en línea: MPC con restricciones para control de velocidad motor dc.

La figura 4.34 muestra una comparación de la simulación en línea y la simulación fuera de línea para el servomecanismo de posición. Los parámetros de sintonía que se han utilizado son iguales para ambas simulaciones: $N=5$, $M=3$, $r\omega=1E-6$, $T_s=0.5s$, además se han impuesto las siguientes restricciones: $(-10 \leq \Delta u \leq 10)$, $(110 \leq u \leq 110)$.

Como se puede observar en la figura 4.34 pese a configurar los mismos parámetros de sintonía en ambas simulaciones los resultados no son iguales. Como era de esperarse la salida en línea presenta ciertas desmejoras a la sintonía lograda en la modalidad fuera de línea, esto se debe a los retardos generados durante la transmisión de datos.

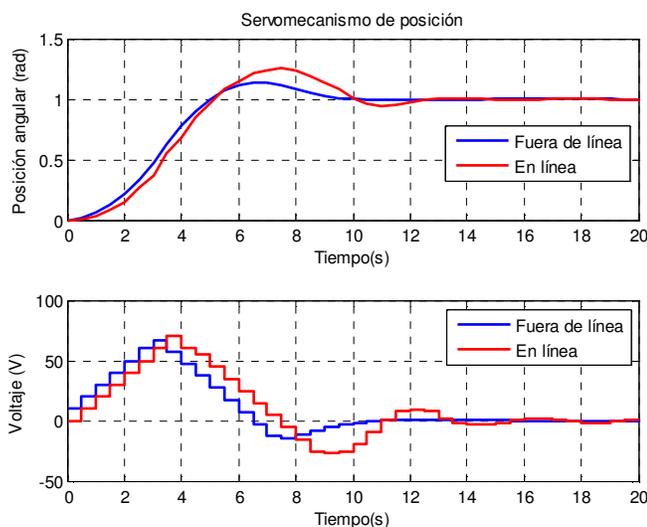


Figura 4.34 Comparación entre modo en línea y fuera de línea para servomecanismo de posición.

4.4.6 Modo en Línea: MPC MIMO.

La figura 4.35 muestra el modelo físico del reactor químico multivariable. Las expresiones 4.28 y 4.29 representan la dinámica no lineal del sistema. Este modelo ha sido tomado de [85, p. 122].

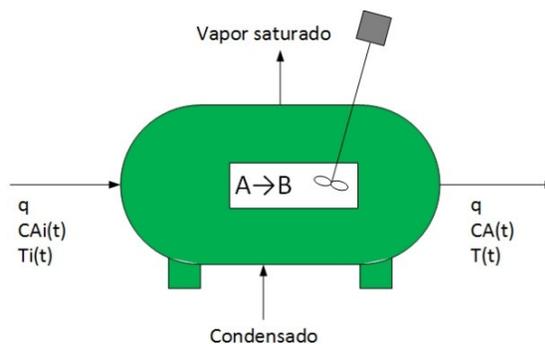


Figura 4.35 Reactor químico multivariable

$$qCA_i(t) - Vk_0 e^{-\frac{E}{RT(t)}} CA(t) = V \frac{dCA(t)}{dt} \quad (4.28)$$

$$q\rho C_p T_i(t) - Vk_0 e^{-\frac{E}{RT(t)}} CA(t) (\Delta H_r) - UA(T(t) - T_s) - q\rho C_p T(t) = V\rho C_v \frac{dT(t)}{dt} \quad (4.29)$$

Donde:

- ρ es la densidad de los productos y los reactivos. Se supone constante
- C_p es la capacidad calorífica a presión constante de los productos y los reactivos. Se supone constante.
- C_v es la capacidad calorífica a volumen constante de los productos y los reactivos. Se supone constante.

- U es el coeficiente global de transferencia de calor. Se supone constante.
- A es el área de transferencia de calor.
- ΔH_r es el calor de reacción. Se supone constante.
- T_s es la temperatura del vapor saturado.

Las expresiones 4.30, 4.31 y 4.32 muestran las matrices A , B y C del sistema linealizado para $CA_i = 0.5$ y $T = 68^\circ\text{C}$. Se asume los siguientes valores de las constantes del sistema: $V=4.8$; $q=0.5$; $\rho = 638$; $C_p=300$; $C_v=148$; $U=4878.4$; $A=1.95$; $\Delta H_r=10008$; $T_s=168$; $E=0.58$; $R=100$; $K_0=0.005$.

$$A = \begin{bmatrix} -0.1100 & 0 \\ -0.0005 & -0.2321 \end{bmatrix} \quad (4.30)$$

$$B = \begin{bmatrix} 0.1042 & 0 \\ 2 & 0.2111 \end{bmatrix} \quad (4.31)$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.32)$$

La figura 4.36 muestra el resultado de la simulación en línea para puntos de ajuste muy cercanos al punto de operación: $SP1 = 0.4$ y $SP2 = 72$. Los parámetros de sintonía utilizados son: $N=3$, $M=1$, $r\omega=0.9$, $T_s=1s$. Claramente se puede observar la eficacia del controlador predictivo bajo la modalidad en línea, pues las consignas han sido alcanzadas satisfactoriamente con valores medidos en la variable de control.

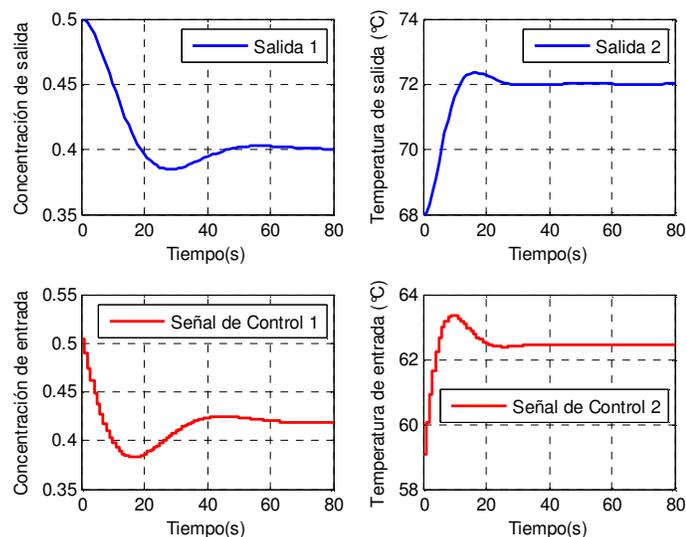


Figura 4.36 Simulación en línea: MPC para reactor químico no lineal.

La figura 4.37 muestra una comparación entre el modo fuera de línea y el modo en línea, cuando se ha ejecutado el algoritmo predictivo sobre el reactor químico no lineal multivariable. Aquí se puede notar claramente una diferencia significativa en el comportamiento del controlador. En la modalidad fuera de línea existe un cambio brusco al inicio de la simulación lo cual indica que la temperatura de entrada debe descender alrededor de 30°C en un segundo para buscar la consigna. Esta es una situación irreal que no ocurre en la modalidad en línea, en todo caso los transitorios son completamente diferentes con lo que se concluye que la simulación en línea puede desacreditar a una simulación fuera de línea ya que esta es la que más se apegar a una situación real.

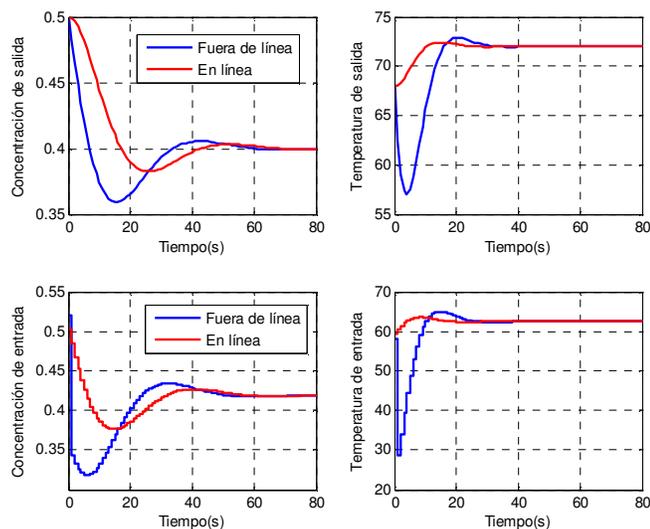


Figura 4.37 Comparación entre modo en línea y fuera de línea para reactor químico no lineal multivariable.

Al final del presente capítulo se puede concluir que el controlador predictivo basado en modelo resulta una técnica muy atractiva a la hora de controlar plantas tanto monovariantes como multivariantes. Claramente se ha podido observar la eficacia del mismo sobre todo para alcanzar las consignas sin error en estado estacionario, esto se debe gracias al uso del integrador embebido.

Por otra parte se puede manifestar que la herramienta en línea fue muy útil pues al comparar los resultados de simulación entre las dos modalidades se pudo notar que las respuestas temporales difieren, esto obviamente obliga a buscar nuevos parámetros de sintonía.

CONCLUSIONES Y RECOMENDACIONES.

El presente trabajo deja ver el procedimiento realizado para la implementación de la plataforma basada en MATLAB® para el entrenamiento en la sintonía de controladores predictivos. Este procedimiento parte de la revisión bibliografía que permitió componer el estado del arte de la estrategia de control. Luego de esto se pasó a revisar el marco teórico y el fundamento matemático que permitió implementar el algoritmo de control.

La implementación y evaluación de desempeño del algoritmo de control predictivo en espacio de estados se lo realizó en varias etapas. Se inició con la modalidad de controlador más sencilla que consiste en el control de una planta lineal mono variable sin restricciones, para esto se utilizaron dos plantas de prueba: control de velocidad de un motor dc y un servomecanismo

de posición. Además se asumió que se conocen todos los estados de la planta. Posteriormente se incorporó restricciones en la amplitud y tasa de cambio de la variable de control y por último en la variable de salida.

Una vez validado el algoritmo sobre plantas mono variables se procedió a extenderlo al caso multivariable. Para esto se utilizaron dos plantas de prueba no lineales: control de un reactor químico y control de una turbina – caldera tipo tambor. Además se asumió que se conocen todos los estados de la planta. Al igual que en el caso monovariable como parte final se incorporó los tres tipos de restricciones posibles.

Todo este procedimiento que se desarrolló para validar el algoritmo de control predictivo con restricciones se lo realizó aislado de la plataforma, es decir utilizando un solo PC tanto para la planta como para el controlador. Paralelo a esto se fueron desarrollando y probando las herramientas adicionales de la plataforma que permitieron la sintonía del controlador utilizando modelos identificados y observadores de estado para lo cual intencionalmente se escondieron ciertos estados de las plantas de prueba.

Los logros desarrollados hasta el momento permitieron crear un interfaz interactivo de usuario bajo GUIDE de MATLAB®, en donde se incorporó plantas adicionales y se creó las facilidades necesarias para que el usuario pueda seleccionar el tipo de planta y pueda cambiar los parámetros de

sintonía del controlador. Todo esto bajo la modalidad fuera de línea es decir utilizando un solo computador.

Para la modalidad en línea se estudió, implementó y probó varias alternativas de comunicación bajo el protocolo UDP, llegando a la conclusión de que el modo más idóneo para el intercambio de datos entre la planta y el controlador es el modo asíncrono con interrupciones temporizadas e independientes para actualizar los estados de la planta y para adquirir los datos de esta.

Todo este procedimiento permitió cumplir con los objetivos planteados y obtener una serie de conclusiones que se detallan a continuación:

1. Actualmente existe abundante literatura sobre control predictivo basado en modelo, lo cual resalta el avance que ha experimentado durante las últimas décadas y la importancia que tiene dentro del control de procesos industriales.
2. De la revisión bibliográfica realizada se pudo observar que existen muchas variantes comerciales de controladores predictivos cuyo nicho radica en Europa. Estas plataformas de hardware y software tratan de dar al usuario las comodidades necesarias para hacer uso de la técnica sin la necesidad de acudir a otras plataformas. Por lo general están

vinculadas con otras herramientas como de adquisición de datos vía OPC, identificación de sistemas, estimación de estados, linealización, reducción de orden, que en mayor o menor grado pueden ser requeridas por el algoritmo de control.

3. La implementación del algoritmo de control predictivo dejó ver que su estructura se basa en operaciones con matrices, las cuales pueden tener dimensiones relativamente grandes, es por esta razón que el nicho de esta estrategia de control radica en la utilización de ordenadores.
4. Con el trabajo realizado se pudo comprobar la fácil adaptación del algoritmo del caso SISO al caso multivariable. El cálculo de las matrices de inicialización es el mismo que se utilizó para el caso mono variable, únicamente se debe prestar atención a las dimensiones de las matrices y al orden en el que se formulan las restricciones.
5. Con las dos modalidades de la plataforma tanto fuera de línea como en línea se pudo notar la efectividad del control predictivo basado en modelo con y sin restricciones. En el caso fuera de línea todas las plantas piloto pudieron ser controladas luego de manipular en menor o mayor grado los parámetros de sintonía del controlador. Para el caso en línea se dio la particularidad de que ciertas plantas no llegaron a ser

sintonizadas, sin embargo esto no descarta la posibilidad de que el usuario de la plataforma pueda hacerlo, ya que no existe una regla específica para manipular los parámetros de sintonía, por lo tanto algún juego de valores especificados puede llegar a controlar dichas plantas. Esto no ocurre cuando se utiliza un controlador PID pues la teoría indica que hacer cuando las especificaciones de tiempo de la respuesta no son las deseadas.

6. Al comparar el comportamiento del controlador predictivo en las modalidades fuera de línea y en línea, bajo los mismos parámetros de sintonía, la diferencia de las respuestas temporales justificó el hecho de haber implementado una plataforma en línea, pues el comportamiento del sistema no fue el mismo, obviamente que luego de haber validado la comunicación entre planta y controlador se debe dar mayor validez al caso en línea pues con este se logra apegar mayormente la simulación al caso real. Esto permite en cierto modo mejorar las prestaciones que presenta mpctool de MATLAB®.
7. El cálculo matricial que requiere el algoritmo predictivo demanda un coste computacional elevado sobre todo en el caso que se requiera incorporar restricciones. Por ejemplo en el caso específico de la turbina – caldera tipo tambor que es una planta no lineal multivariable, el tiempo de cálculo de la acción de control en ciertas iteraciones alcanzó

un valor de 0.4 segundos. Sin embargo en la modalidad fuera de línea esta demora no tiene incidencia alguna en el desempeño del controlador pues la planta permanece estática hasta que le llegue la acción de control. Este inconveniente fue una de las motivaciones principales para desarrollar la plataforma en línea, en la cual la planta está actualizando sus estados continuamente de manera independiente, entonces una demora en el cálculo de la acción de control si afectará el comportamiento del controlador.

8. El tiempo que demora el cálculo matricial debe ser monitoreado durante cada iteración, pues puede ocurrir que este sea mayor al tiempo de muestreo. Esto provocará que colapse el PC controlador debido a que se le está obligando a atender una interrupción sin haber culminado la de la iteración anterior. En el caso particular de MATLAB® esto hace que el programa se cuelgue.

9. Se debe tener en consideración que puede existir restricciones impuestas sobre el sistema que un algoritmo de optimización no pueda satisfacer bajo ciertos parámetros de sintonía. En el caso de que una restricción en la variable de control sea violada es recomendable aumentar el factor de peso para que los movimientos de control sean más suaves.

10. La comunicación entre la planta y el controlador debe ser asíncrona para que la simulación se apegue más a la realidad. Esto quiere decir que ambas PCs deben cumplir su trabajo con total independencia sin que exista un diálogo de “toma y dame” entre los elementos de la comunicación. Por ejemplo la planta actualiza sus estados independientemente de si llega o no la acción de control. En el caso de que no llegue esta está cambiando sus estados con el último valor a disposición, tal cual ocurre en la vida real.

11. Una vez entendido el marco teórico del control predictivo basado en modelo, la implementación del algoritmo formulado en espacio de estados puede ser que resulte más cómoda que otras alternativas de MPC, sin embargo se debe considerar que los estados de las plantas no siempre tienen un significado físico y están disponibles para el controlador. Esto obliga a la estrategia a contar con observadores o estimadores de estado para disponer de los mismos.

APÉNDICE A
MANUAL DE USUARIO DE LA PLATAFORMA SIMUMPC

La plataforma de simulación SIMUMPC está orientada al entrenamiento en la sintonía de controladores predictivos. Esta plataforma presenta dos modos de funcionamiento: “fuera de línea” y “en línea”. En la modalidad fuera de línea el controlador y la planta residen en el mismo PC. Por otra parte en la modalidad en línea el PC controlador realiza las tareas respectivas para adquirir los datos y calcular la acción de control y la envía hacia la planta la cual reside en otro PC. La comunicación entre los dos PCs se la realiza mediante protocolo UDP.

1. Pc – Controlador.

Este Pc es el encargado de la ejecución del algoritmo de control, lo cual incluye adquisición de datos en el caso de que se haya seleccionado la modalidad en línea, cálculo de la acción de control y visualización gráfica de las respuestas del sistema y acciones de control respectivas. Para acceder a la aplicación es necesario escribir “simMPC” en la línea de comando de MATLAB®. Para esto la carpeta respectiva debe estar copiada en la dirección que especifica el navegador de direcciones.

La aplicación inicia con una ventana de presentación (véase la figura A1) en donde se ilustra al usuario el modo de trabajo del simulador. En este ambiente de trabajo únicamente esta disponible la opción de **Ayuda** de la aplicación.



Figura A1 Ventana de inicio de la herramienta SIMUMPC.

Al presionar el botón **Inicio** una nueva ventana se abrirá (véase la figura A2). En este ambiente el usuario puede contar con los siguientes menús desplegables:

1. **Modelo de planta.** Permite seleccionar la planta a controlar. Estos modelos se encuentran almacenados en una base de datos del programa. El usuario podrá seleccionar algunas alternativas desde un simple motor dc hasta una planta multivariable no lineal.
2. **Modelo de predicción.** Permite seleccionar el modelo matemático que el controlador utilizará para calcular las predicciones. El usuario podrá seleccionar un modelo idéntico al de la planta a controlar, o

podrá ingresar manualmente un modelo diferente. Además existe la opción de identificar un modelo y exportarlo a la plataforma de simulación.

- 3. Modo de funcionamiento.** Permite seleccionar una simulación fuera de línea o en línea. La simulación fuera de línea no requiere de un segundo computador pues la dinámica de la planta es recreada en el mismo PC.
- 4. Comunicación.** Permite configurar las direcciones IP y los puertos de comunicación que requiere el protocolo UDP. Esta configuración deberá hacerse en concordancia con las direcciones IP del PC planta y el PC controlador. Este menú es accesible únicamente cuando se ha seleccionado el modo en línea.
- 5. Ayuda.** Brinda al usuario información necesaria para que pueda operar la herramienta de simulación. Aquí existe un acceso directo que abre un manual de usuario, el mismo que se puede encontrar en el apéndice 1.

Para iniciar una sesión de simulación se debe seleccionar el modelo de planta a controlar y se debe especificar el modelo de predicción a utilizar (véase la figura A2).

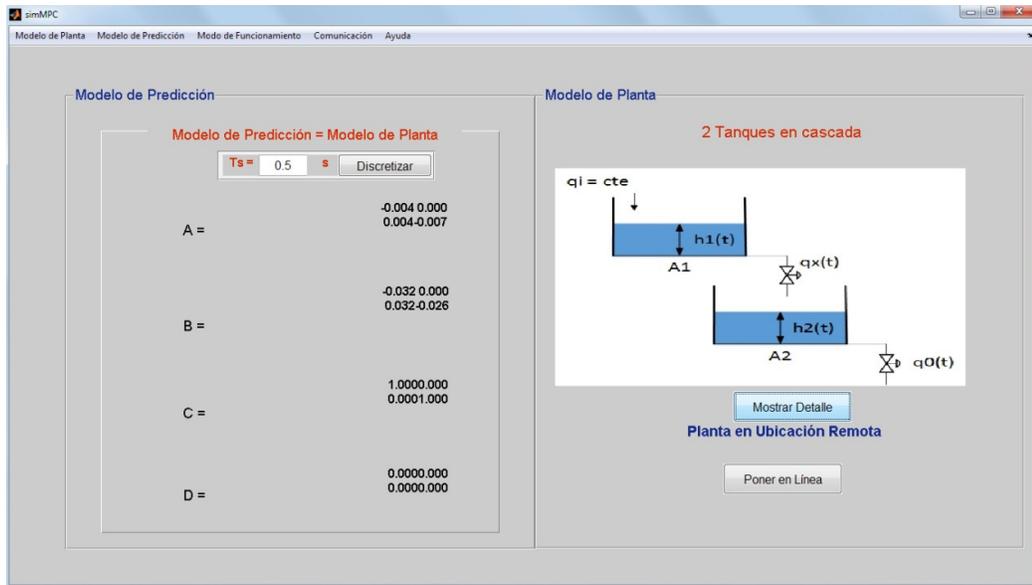


Figura A2 Ventana de inicio de la herramienta SIMUMPC.

Existen tres opciones para especificar el modelo de predicción:

- 1. Modelo de Predicción igual al modelo de planta:** Bajo esta opción la aplicación asigna un modelo de predicción que es una réplica exacta al modelo de la planta. Para iniciar con la sintonía del controlador es necesario discretizar el modelo con un tiempo de muestreo especificado.

2. **Modelo de Predicción ingresado de forma manual:** Bajo esta opción el usuario podrá ingresar manualmente un modelo de predicción que no necesariamente sea igual al modelo de la planta. Esta alternativa resulta altamente atractiva para probar la potencialidad del algoritmo predictivo, pues no siempre se puede contar con un modelo exacto que represente la dinámica de la planta.

The screenshot shows a software window titled "SIMUEEDITMOD" with a sub-window titled "Ingreso modelo de Predicción". Inside, there is a radio button labeled "Modelo Discreto" which is selected. Below this, there are four input fields for matrices:

- A = [0.28 3.43; 1.14 -15.6]
- B = [1.2; 3.69]
- C = [1 0]
- D = [0]

Below the matrices, there is a field for "Tiempo de Muestreo (s)" with the value "0.1". A note below that says "El modelo se ingresa en tiempo discreto". At the bottom, there is a button labeled "Exportar".

Figura A3 Cuadro de diálogo para especificar de forma manual un modelo de predicción.

Para ingresar un modelo de forma manual se debe seleccionar la opción **Ingreso manual** que está dentro del menú **Nuevo modelo**. El cuadro de diálogo que permite el ingreso de un modelo nuevo se muestra en la figura A3. Este modelo puede ser ingresado en formato

continuo o discreto. Una vez que se han especificado las matrices es necesario pulsar el botón **Exportar**. Esta acción guardará en memoria el modelo y estará listo para ser importado desde el ambiente simMPC.

3. Modelo de predicción identificado: Bajo esta opción el usuario podrá identificar un modelo haciendo uso de un algoritmo de sub-espacio de estados. La herramienta de identificación que proporciona la plataforma SIMUMPC se muestra en la figura A4.

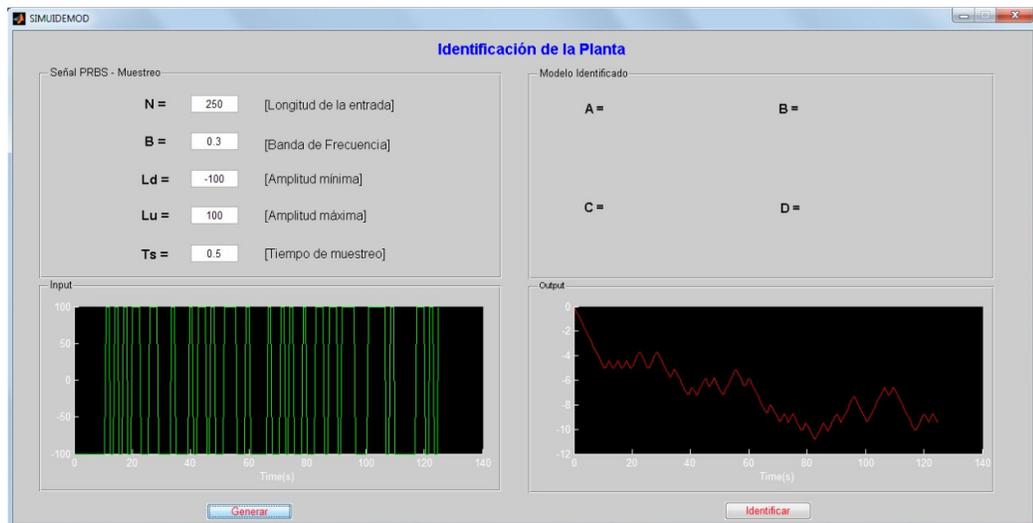


Figura A4 Herramienta para identificación de sistemas dentro de SIMUMPC

Una vez ingresados los parámetros para generar la señal PRBS es necesario pulsar el botón **Generar**. Posteriormente se pulsa el botón **Identificar**, esta acción hará que el algoritmo de identificación devuelva un modelo en espacio

de estados conjuntamente con un gráfica que sirve para validar el experimento. Esto se ilustra de mejor manera en la figura A5.

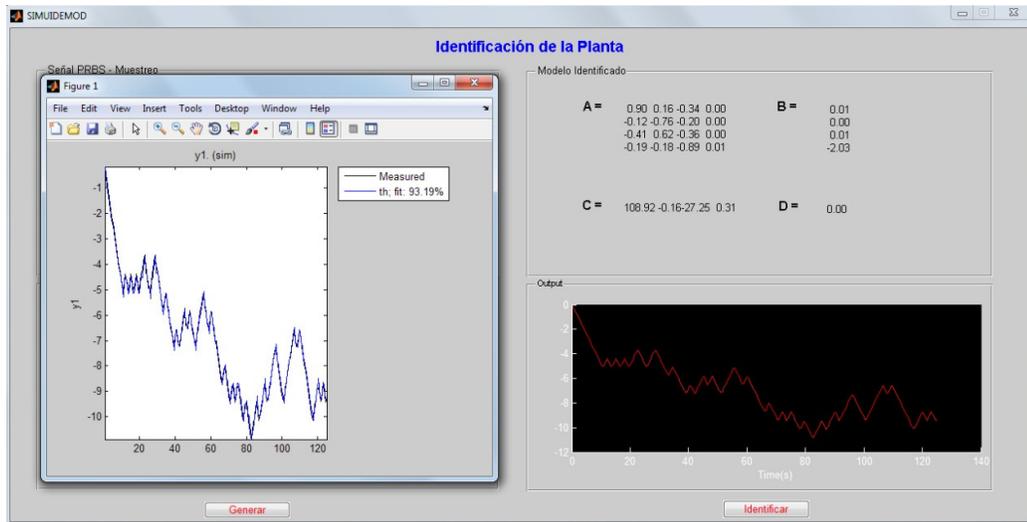


Figura A5 Ejecución del algoritmo de identificación.

Luego de haber ejecutado el proceso de identificación, es necesario pulsar el botón **Importar** del ambiente simMPC para que el modelo identificado sea utilizado por el algoritmo predictivo.

En esta instancia se puede optar por la ejecución del algoritmo ya sea en línea o fuera de línea. Estas dos opciones pueden ser seleccionadas del menú modo de funcionamiento.

En el caso de que se haya seleccionado la opción fuera de línea, una vez que se pulse el botón **Sintonizar** se abrirá un nuevo ambiente de simulación denominado simMPCejec (véase la figura A7). Si se opta por la opción en línea es necesario tener una configuración previa de la comunicación. Esto se lo puede hacer en la opción **Comunicación** del menú principal.

Una vez seleccionada la opción **Comunicación** se abrirá un cuadro de diálogo (véase la figura A6). Aquí se debe especificar el número de puerto local y remoto en concordancia con la PC planta, que en este caso vendría a ser la PC con dirección remota. Las direcciones IP que se especifican no se las puede modificar y estas deben coincidir con la configuración de Windows.



Figura A6 Cuadro de diálogo para configuración de la comunicación.

En el caso se requiera una simulación en línea es necesario configurar la comunicación de manera adecuada y se debe pulsar el botón poner en línea.

Para que exista una comunicación satisfactoria se deben cumplir los siguientes parámetros:

- Configuración adecuada de direcciones IP y puertos de comunicación en los dos PCs.
- El cable de red debe estar debidamente conectado.
- En el PC planta se debe haber seleccionado la misma planta que se seleccionó en el controlador y además se debe habilitar la opción modo de espera.

Si es que alguna de las condiciones no se cumple la aplicación enviará un mensaje solicitando que se revise lo antes indicado y bajo esta circunstancia no se abrirá el siguiente ambiente de simulación. En el caso de que se cumplan todas las condiciones el nuevo ambiente de simulación proporcionará al usuario las herramientas necesarias para configurar los parámetros de sintonía, los puntos de ajuste, las restricciones y además dará la opción de utilizar observadores de estado. En este ambiente de simulación se puede observar las respuestas del sistema y las acciones de control respectivas.

En esta instancia existe una diferencia entre el ambiente de simulación en línea y el fuera de línea. En el caso en línea no existe la posibilidad de

ingresar los estados iniciales de la planta. Esta propiedad deberá ser ingresada desde el PC remoto.

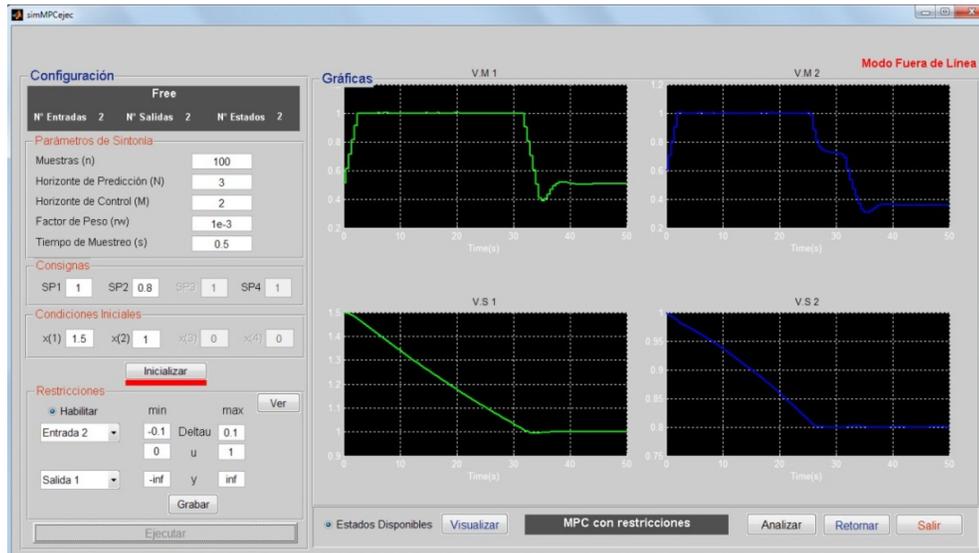


Figura A7 Ambiente SIMUMPC para la sintonía del controlador.

Si es que se ha habilitado la opción de **Restricciones** están deben ser asignadas de acuerdo a los menús desplegables y grabadas utilizando el botón grabar. En el caso de que se quiera tener una vista panorámica de las restricciones impuestas se puede pulsar el botón **Ver** (véase la figura A8). Aquí se puede observar como se han impuesto restricciones a la tasa de cambio y amplitud de las variables manipuladas lo cual está acorde a la respuesta del sistema que se observa en la sección de gráficas. Para este caso demostrativo no se han impuesto restricciones en la variable de salida para esto se debe dejar con los valores asignados por defecto (-inf e inf respectivamente).

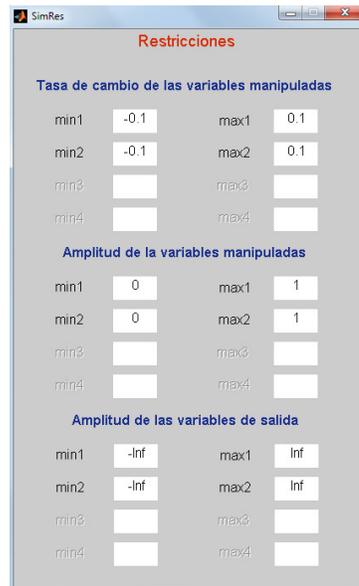


Figura A8 Vista panorámica de las restricciones impuestas en la plataforma SIMUMPC.

En el caso de que se desee se puede pulsar el botón **Analizar**. Entonces las gráficas resultantes de la simulación serán visualizadas en el formato estándar de MATLAB[®] (véase la figura A9). Esto puede resultar útil cuando se desee editar o analizar más detenidamente los resultados.

Para utilizar estimadores de estado existe el botón de selección **Estados disponibles**. Este se encuentra en la parte inferior del ambiente de simulación. Cuando esta opción está desactivada intencionalmente se esconden ciertos estados de la planta y el filtro de Kalman entra en ejecución. En la figura A10 se muestra una comparación de los estados

reales con los estimados cuando se ha seleccionado esta opción. La planta que se ha utilizado es el servomecanismo de posición que se especificó en la sección 4.4.1.

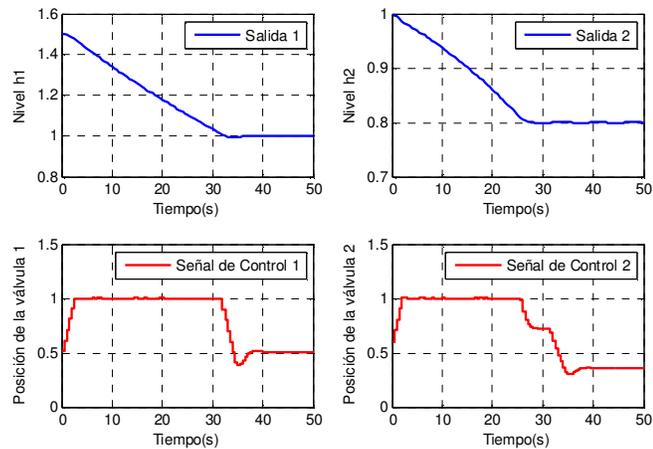


Figura A9 Presentación de resultados utilizando el formato estándar de MATLAB®

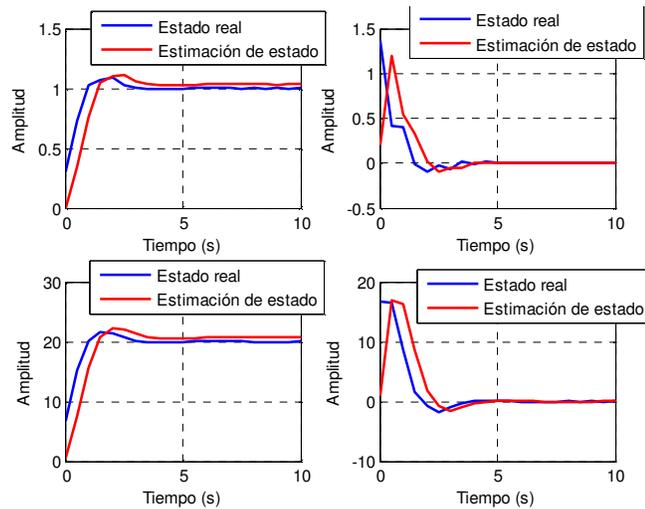


Figura A10 Ejecución del filtro de Kalman y comparación entre los estados reales y estimados.

2. Pc – Planta.

Este Pc es el encargado de recrear la dinámica de las plantas que se encuentran almacenadas en una base de datos de la plataforma. Para abrir la aplicación respectiva se debe escribir “simPLANTA” en la línea de comandos de MATLAB®. Esto hará que se abra un ambiente inicial idéntico al de la PC – controlador que se ilustra en la figura A1.

Al pulsar el botón **Inicio** se abre un nuevo ambiente (véase la figura A11) que permite la configuración necesaria para iniciar una sesión de simulación en línea. Este ambiente tiene dos menús desplegables: **Modelo de planta** y **Comunicación**.

En el menú **Modelo de planta** se debe seleccionar la planta que se va a controlar. Para que exista una comunicación satisfactoria en el PC – controlador se debe haber seleccionado la misma planta, caso contrario el programa emitirá un mensaje de error solicitando se verifique lo antes mencionado.

La opción **Comunicación** por otra parte sirve para configurar los puertos de comunicación UDP. Estos deben estar configurados en concordancia con el PC – controlador. El ambiente que se abre al seleccionar esta opción es similar al mostrado en la figura A6.

Una vez que se ha seleccionado la planta a simular se debe habilitar la opción **Activar modo de espera**. Bajo estas condiciones la PC – planta entrará en un bucle en espera de una señal de confirmación que debe llegar desde el controlador. Esta señal de confirmación es enviada al pulsar el botón **Poner en línea** que se encuentra en el ambiente simMPC del PC – controlador.

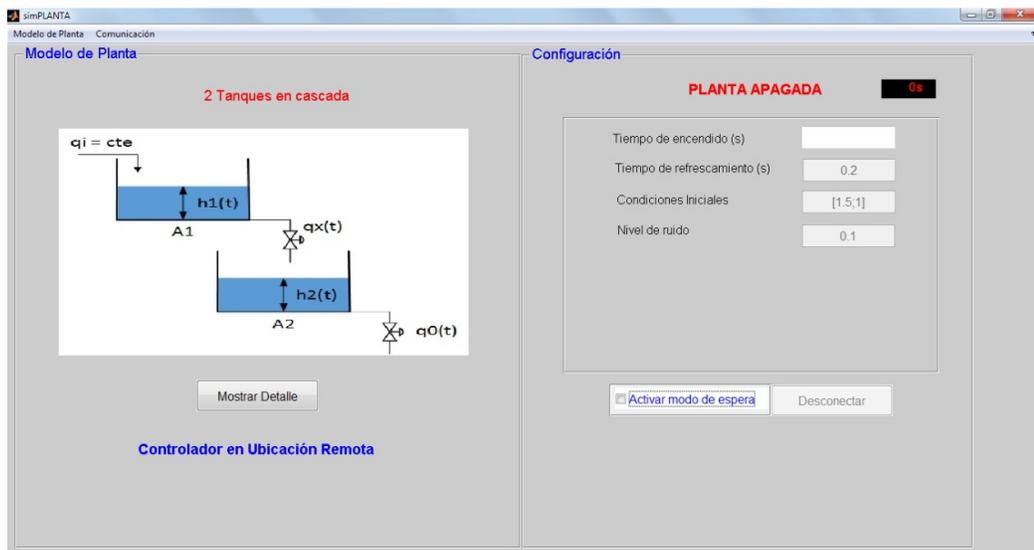


Figura A11 Ambiente de simulación en el PC – planta.

Las condiciones para que haya una comunicación exitosa son las mismas que se detalló en la sección del PC – Controlador. En caso exista una situación favorable la aplicación emitirá un mensaje y entrará en otro bucle esperando se envíe desde el controlador el tiempo de simulación. Este tiempo será enviado al momento que se presione el botón **Inicializar**.

En esta instancia la planta entra en un tercer bucle en la espera de la primera acción de control. Esta llegará desde el controlador el momento que se pulse el botón **Ejecutar**.

Cuando se ha pulsado el botón **Ejecutar** la simulación inicia y la planta va actualizando sus estados conforme al **tiempo de refrescamiento** que se haya configurado. Para que se asemeje más al funcionamiento continuo que tiene una planta se ha seleccionado el tiempo mínimo que permite MATLAB[®] que es de 0.2s. Para tiempos menores **ODE45** puede que cree conflictos y el programa colapse.

En la sección de configuración de la planta se visualiza el tiempo de simulación, se puede cambiar el tiempo de refrescamiento, las condiciones iniciales y además se puede agregar ruido al proceso.

BIBLIOGRAFÍA

- [1] Haruo Takatsu y Toshiaki Itoh, "Future needs for control theory in industry-report of the control technology survey in japanese industry," *Control Systems Technology, IEEE Transactions on.*, vol. 7, no. 3, pp. 298-305, 1999.
- [2] Richalet J., Rault J., Testud A. y Papon J., "Model predictive heuristic control: Applications to industrial processes", 1978.
- [3] Bordons C., "Control Predictivo: Metodología, Tecnología y Nuevas Perspectivas.," Aguadulce, Almería, 2000.
- [4] Quin J. y Badgwell T., "A survey of industrial model predictive control technology.," *PERGAMON Control Engineering Practice.*, pp. 733-764, 2003.
- [5] Quin J., Thomas A. y Badgwell T., "An Overview of Nonlinear Model Predictive Control Applications.," *Progress in Systems and Control Theory.*, vol. 26, nº 3, pp. 369-392, 2000.
- [6] Maciejowski J, *Predictive Control: With Constraints*, Ilustrada ed., P. Hall, Ed., Pearson Education, 2002.
- [7] Roos C., Terlaky T., y Vial J., *Theory and Algorithms for Linear Optimization: An Interior Point Approach.*, 1997.
- [8] Wang Y. y Boyd S, "Fast Model Predictive Control Using Online Optimization.," *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, vol. 18, nº 2, 2010.
- [9] Del Re L., Kaiser O. y Pfinffer R., "Black Box Identification and Predictive Control of High Speed Machine Tools.," *Proceedings of the 1996 IEEE International Conference on Control Applications*, 1996.
- [10] Di Ruscio D., "Model Predictive Control and Identification: A Linear State Space Model Approach.," *Proceedings of the 36th Conference on Decision & Control*, 1997.
- [11] Magni L., Reimondo D. y Allgower F., *Nonlinear Model Predictive Control: Towards New Challenging Applications*, Ilustrada ed., vol. 384 de *Lecture Notes in Control and Information Sciences*, Springer, Ed., 2009, p. 572.

- [12] Grancharova A y Johansen T, *Explicit Nonlinear Model Predictive control: Theory y Applications.*, Ilustrada ed., vol. 429 de *Lecture Notes in Control and Information Sciences*, Springer, Ed., 2012, p. 234.
- [13] Grune L. y Pannek J., *Nonlinear Model Predictive Control: Theory and Algorithms*, Springer, Ed., 2011, p. 372.
- [14] Kouvaritakis B. y Cannon M, *Non-Linear Predictive Control: Theory and Practice*, Ilustrada ed., vol. 61 de *IEE control engineering series: Institution of Electrical Engineers, IET*, Ed., 2008, p. 261.
- [15] Bordons C, "Control Predictivo Generalizado de Procesos Industriales: Formulaciones Aproximadas," 1994.
- [16] Liu X., Niu L. y Liu J., "Nonlinear Multivariable Supervisory Predictive Control.," de *American Control Conference*, 2009.
- [17] Tatjeski P., "Supervisory Predictive Control and on-line set-point and Implementation on a Laboratory Three Tank System.," *Int. J. Appl. Math. Comput. Sci.*, vol. 20, nº 3, pp. 483-495, 2010.
- [18] Zhang zhi sheng, Chen Huai-min, Wu Cheng-fu, y Ma Song-hui., "An effective terminal condition ensuring stability of MPC," *Computer Application and System Modeling*, vol. 7, pp. 202-206, 2010.
- [19] MATHWORKS., "MATHWORKS. Model Predictive Control Toolbox: Working with the Design," 2012. [En línea]. Available: <http://www.mathworks.com/help/mpc/ug/working-with-mpc-designtool.html>.
- [20] Akesson J., "MPCtools 1.0 Reference Manual," 2006.
- [21] Crespo L. y Cárdenas H., *Introducción a TCP/IP: sistemas de transporte de datos.*, U. d. Alicante, Ed., 2007.
- [22] Wikipedia la enciclopedia libre, "Cable categoría 5e," [En línea]. Available: http://es.wikipedia.org/wiki/Cable_de_Categor%C3%ADa_5.
- [23] Holkar K. y Waghmare L., "An Overview of Model Predictive Control.," *International Journal of Control and Automation*, vol. 3, nº 4, pp. 47-64, 2010.
- [24] Cluter C.R y Ramaker B., "Dynamic Matrix Control -A Computer Control Algorithm," de *Automatic Control Conference*, 1980.
- [25] Camacho E. y Bordons C., *Model Predictive Control*, London: Springer, 1999.

- [26] Sun Wenyu y Yuan Ya-Xiang., Optimization Theory and Methods: Nonlinear Programming., Ilustrada ed., vol. Volumen 98 de Applied Optimization Series, Springer, 2006, p. 700.
- [27] Un-Chul Moon, Kwang. Y. Lee, Life Fellow, IEEE, "Step-Response Model Development for Dynamic Matrix Control of a Drum Type Boiler Turbine System.," *IEEE TRANSACTIONS ON ENERGY CONVERSION.*, vol. 24, nº 2, pp. 423-430, 2009.
- [28] Edinzo J., Iglesias M., Sanjuán E. y Smith C., "Tuning equation for dynamic matrix control in siso loops.," vol. 19, 2006.
- [29] Pu Han, Yuhong Li, Hongjun Liu y Dongfeng Wang., "A new dynamic matrix control algorithm with Lyapunov stability.," de *Control, Automation, Robotics and Vision Conference ICARCV 8th*, 2004.
- [30] Bing Wu, "Application of Dynamic Matrix Control Algorithm Based on Real Time Operating System.," de *Software Engineering, 2009. WCSE '09. WRI World Congress on*, 2009.
- [31] Lan Jian, Li Dewey y Xi Yugeng., "Implementation of Dynamic Matrix Control on FPGA.," de *Control Conference (CCC) 29th Chinese*.
- [32] ClarkeD.W , Mohtadi C. y Tuffs P.S, "Properties of Generalized Predictive Control. Parte I. The Basic Algorithm.," *Automática*, vol. 23, nº 2, pp. 859-875, 1987.
- [33] Camacho E., "Constrained Generalized Predictive Control.," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, vol. 38, nº 2, pp. 327-332, 1993.
- [34] Clarke D.W, "Application of Generalized Predictive Control to Industrial Processes.," *IEEE Control Systems Magazine*, 1998.
- [35] Richalet J., "Pratique de la commande predictive," *ADERSA*, 1993.
- [36] Abu el Ata-Doss S. , Fiani P. y Richalet. J, "Handling Input and State Constraints in Predictive Functional Control.," de *Proceedings of the 30th Conference on Decision and Control*, 1991.
- [37] Compas J.M, Decarreau P., Lanquetin G., Estival J.L, Fluget N., Martin R. y Richalete J., "Industrial Application of Predictive Functional Control to Rolling Mill, Fast Robot, River Dam.," *Control Applications, Proceedings of the Third IEEE*, vol. 3, pp. 1643-1655, 1994.

- [38] ABB, "cpmPlus Expert Optimizer: Overview," 2012. [En línea]. Available: www.abb.com/product/db0003db004001/aeb109afc47df071852576170049499e.aspx.
- [39] Welch G. y Bishop G., "An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill, NC 27599-3175," 2001. [En línea]. Available: http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf.
- [40] ABB, "cpmPlus Expert Optimizer Advanced optimization for your industry," 2009. [En línea]. Available: [http://www05.abb.com/global/scot/scot350.nsf/veritydisplay/b81e867927f384e3852576fd0047fd45/\\$file/cpmPlus_Exp_Opt_3BHS%20291766_lr.pdf](http://www05.abb.com/global/scot/scot350.nsf/veritydisplay/b81e867927f384e3852576fd0047fd45/$file/cpmPlus_Exp_Opt_3BHS%20291766_lr.pdf).
- [41] Karaman S., Sanfelice R. y Frazzoli E., "Optimal Control of Mixed Logical Dynamical Systems with Linear Temporal Logic Speci," de *Proceedings of the 47th IEEE Conference on Decision and Control Cancun*, México, 2008.
- [42] Castagnoli D., Kiener M. y Gallestey E., "Rentabilidad del cemento: Hacia la optimización total de las plantas industriales de minerales y cemento," *Revista ABB*, 2006.
- [43] Lundh M., Gaulocher S., Pettersson J., Lindvall H. y Gallestey E., "Model Predictive Control for Flotations Plants," 2008.
- [44] ABB, "Process Improvement Packages Multivariable Control for the Process Industries 3dMPC TM," 1998. [En línea]. Available: [http://www05.abb.com/global/scot/scot296.nsf/veritydisplay/b788e3a9870e9a1ec1256a34007129d6/\\$file/3bus025043r0001_-_en_3dmpc_multivariable_controller.pdf](http://www05.abb.com/global/scot/scot296.nsf/veritydisplay/b788e3a9870e9a1ec1256a34007129d6/$file/3bus025043r0001_-_en_3dmpc_multivariable_controller.pdf).
- [45] ABB Automation Products, *3DMPC Model Predictive Controller Version 1.1. Product Guide*, 2000.
- [46] Hess T., Kroll A., Modén P., y Havre K., "Dynamic Solutions de ABB: La mejor tecnología disponible para la optimización.," pp. 46-54, 2001.
- [47] ASPENTECH, "DMCplus: A powerful, proven multivariable controller for maintaining processes at their optimal operating point.," 2012. [En línea]. Available: <http://www.aspentech.com/products/aspentech-dmcplus.aspx>.

- [48] ASPENTECH, "DMCPlus TM.," 2012. [En línea]. Available: <http://www.aspentech.com/brochures/dmc.pdf>.
- [49] Goodhart S.G, "DMCplus TM model predictive control applications," de *Model Predictive Control: Techniques and Applications - Day 2 (Ref. No. 1999/096), IEEE Two-Day Workshop on*, 1999.
- [50] ASPENTECH y Valero Energy Corporation, "Valero Implements APC Solution to Manage Stock Switches, Reducing Product Downgrades by 400Mper Year," 2000.
- [51] ASPENTECH y Unipetrol., "Leading Czech Refinery and Petrochemical Group Reduces Energy Costs with APC Solution," 2010.
- [52] Honeywell International Inc., "Control Performance Solutions: Profit Controller," 2012. [En línea]. Available: <https://www.honeywellprocess.com/en-US/explore/products/advanced-applications/advanced-control-and-optimization/control-performance-solutions/Pages/profit-controller.aspx>.
- [53] Honeywell International Inc., "Profit Controller Multivariable Control and Optimization Technology. Product Information Note," 2010. [En línea]. Available: <https://www.honeywellprocess.com/library/marketing/notes/ProfitController3292010Updateeop.pdf>.
- [54] Honeywell International Inc., "Profit Controller: Multivariable Control and Local Optimization Technology," [En línea]. Available: <http://es.scribd.com/doc/93925566/Profit-Controller-Rmpct-Overview>.
- [55] Oliveira A., Batista J., Santos J., Ribero M., Charr J. y Lopes R., "Advanced Process Control in Alumina Digestion Unit.," *TMS (The Minerals, Metals and Materials Society)*, 2009.
- [56] Yoo S. C. ,Kim T. G. , Choi B. G., Ahn S. M. , Yoon J. K., Seo S. T. y Kim H. S., "Optimized Model Predictive Control of Commercial BTX Plant using Profit Controller.," de *SICE-ICASE International Joint Conference.*, 2006.
- [57] Moldovanyi N. y Abonyi J., "Control of a Continuous Vacuum Crystalliser in an Industrial Environment: A Feasibility Study - Comparison PID to Model Predictive Control Solution.," de *Hungarian Journal of Industrial Chemistry Vexzoren*, 2007.
- [58] JGC, "Predictive Control for Reactor (PCR) / Predictive Functional Control (PFC)," 1998. [En línea]. Available: http://www.jgc.co.jp/en/04tech/10plant/pcr_pfc.html.

- [59] Laszczyk P. y Richalet J., "Application of Predictive Control to a Heat Exchanger.," *Atomic Energy*, vol. 48, n° 32, 1999.
- [60] MATHWORKS, "Model Predictive Control Toolbox. Key features," 2013. [En línea]. Available: <http://www.mathworks.com/products/mpc/description1.html>.
- [61] Ridvan B., *Methods of Model Based Process Control: [proceedings of the NATO Advanced Study Institute on Methods of Model Based Process Control, Antalya, Turkey, August 7-17, 1994]*, ilustrada ed., vol. 293, Springer, 1995, p. 826.
- [62] Bemporad A., Morani M., y Ricker N., *Model Predictive Control Toolbox. For Use with MATLAB*, 2005.
- [63] MATHWORKS, "Analyze Sensitivity Using the Tuning Advisor. MATLAB R2013b," [En línea]. Available: <http://www.mathworks.com/help/mpc/ug/sensitivity-analysis-using-thetuning-advisor.html>.
- [64] Guzmán J., Berenguel M. y Dormido S., "MIMO-GPCIT: Generalized Interactive Tool for MIMO Systems," UNIVERSIDAD DE ALMERÍA, 2010. [En línea]. Available: <http://aer.ual.es/mimo-gpcit/>.
- [65] Guzmán J., Berenguel M. y Dormido S., "MIMO-GPCIT. Herramienta interactiva de control predictivo generalizado para sistemas multivariables con restricciones," *Revista Iberoamericana de Automática e Informática Industrial*, vol. 1, n° 1, pp. 57-68, 2004.
- [66] Lensen R., *Modular and multiplatform MCPC development. A modular and multiplatform software tool for Multivariable Constrained Predictive Control*.
- [67] Wang L., *Model Predictive Control System Design and Implementation Using MATLAB®*. Advances in Industrial Control, ISSN 1430-9491, ilustrada ed., Springer, Ed., 2009, p. 376.
- [68] Kalman R.E., "A New Approach to Linear Filtering and Prediction Problems," de *Transaction of the ASME. Journal of Basic Engineering*, 1960.
- [69] Girija G. Raol J.R. y Singh J. , *Modelling and Parameter Estimation of Dynamic Systems*, 2004.
- [70] Ollero Baturone A., *Control por Computador: Descripción Interna y Diseño Óptimo*, Barcelona. España, 1991.
- [71] Kwon W.H y Pearson A.E, "On feedback stabilization of time varying discrete systems," de *IEEE Transactions in Automatic Control*, 1979.

- [72] Weiland S., Stoorvogel A., y Tiagounov A., "End point Parametrization and Guaranteed Stability for a Model Predictive Control Scheme.," de *Proceedings of the 40th IEEE Conference on Decision and Control Orlando, Florida, 2001*.
- [73] Zdeněk Dostál, *Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities*, Ilustrada ed., Springer, 2009, p. 302.
- [74] Lee G.M., Tam N.N. y Yen N.D, *Quadratic Programming and Affine Variational Inequalities: A Qualitative Study*, vol. 78 de *Nonconvex optimization and its applications*, Springer, 2006, p. 359.
- [75] Rodríguez D. y Bordons C., "Apuntes de Ingeniería de Control: Análisis y Control de Sistemas en Espacio de Estado, Identificación de Sistemas, Control Adaptivo, Control Predictivo," 2005.
- [76] Christofides P., Liu J. y Muñoz de la Peña D., *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*, Springer, 2011, p. 260.
- [77] Bertsekas D. P., *Nonlinear programming*, Segunda ed., U. d. Michigan, Ed., Athena Scientific, 2011, p. 777.
- [78] Nocedal J. y Wright S., *Numerical Optimization*, Springer series in operations research and financial engineering., 2006, p. 686.
- [79] Axehill D., *Applications of Integer Quadratic Programming in Control and Communication*, Linköping, Sweden, 2005.
- [80] Zerpa L., "Optimización para Ingenieros. Optimización con Restricciones (Notas de Clase)," 2006.
- [81] MATHWORKS, "Instrument Control Toolbox: Control and communicate with test and measurement instruments. User's Guide," 2007. [En línea]. Available: <https://dspace.ist.utl.pt/bitstream/2295/137643/1/MATLAB%20Instrument%20Control%20Toolbox.pdf>.
- [82] Qin S.J, "An overview of subspace identification," *Computers and Chemical Engineering*, vol. 1513, nº 30, 2006.
- [83] MATHWORKS, "Ordinary Differential Equations," 2013. [En línea]. Available: <http://www.mathworks.com/help/matlab/math/ordinary-differential-equations.html>.

- [84] Xue D., Zhao C. y Chen Y., "Fractional Order PID Control of A DC - Motor with Elastic Shaft: A Case Study.," de *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, 2006.
- [85] Smith C. and Corripio A., *Control Automático de Procesos. Teoría y Práctica*, Noriega Limusa.
- [86] Houpis C. y Lamont G., *Digital control systems: theory, hardware, software*, Segunda ed., Michigan: McGraw-Hill, 1992, p. 752.
- [87] Liu J., Chen X., Muñoz D. y Panagiotis D., "Sequential and Iterative Architectures for Distributed Model Predictive Control of Nonlinear Process Systems. Part II: Application to a Catalytic Alkylation of Benzene Process," de *American Control Conference Marriott Waterfront*, Baltimore, MD, USA, 2010.
- [88] Cranmer A., Shahbkhiti M. y Hedick J., "Grey-Box Modeling Architectures for Rotational Dynamic Control in Automotive Engines," de *American Control Conference*, Fairmont Queen Elizabeth Montreal, Canada, 2012.
- [89] Rivera R. y Velez M., "Gray-Box Modeling of Electric Drive Systems Using Neural Networks.," de *Proceedings of the 2001 IEEE International Conference on Control Applications*, México, 2001.
- [90] MATHWORKS, "System Identification Toolbox. Create linear and nonlinear dynamic system models from measured input-output data," 2013. [En línea]. Available: <http://www.mathworks.com/products/sysid/>.
- [91] Welch G. y Bishop G., *An Introduction to the Kalman Filter*. University of North Carolina at Chapel Hill, 2001.
- [92] Crespo L. y Candelas F, *Introducción a TCP/IP. Sistemas de transporte de datos*, Alicante: Publicaciones de la Universidad de Alicante. Espagráfico, 1998.