

**DAS** Departamento de Automação e Sistemas  
**CTC** **Centro Tecnológico**  
**UFSC** Universidade Federal de Santa Catarina

# **Análisis, Implementación y Validación de un Algoritmo de Control Predictivo Repetitivo Basado en Modelo.**

*Informe sometido a la Universidad Federal de Santa Catarina  
como un requisito para la aprobación de la disciplina:*

***DAS 5511: Proyecto de Fin de Carrera***

***Daniel Miranda Cruz***

*Barcelona, julio 2012*

---

**Análisis, Implementación y Validación de un Algoritmo de Control  
Predictivo Repetitivo Basado en Modelo.**

***Daniel Miranda Cruz***

***Tutores:***

***Ramon Costa Castelló, Dr.***

---

Firma del Tutor

***Julio Elias Normey Rico, Dr.***

---

Firma del Tutor

Este informe ha sido juzgado en el contexto de la disciplina

**DAS 5511: Proyecto de Fin de Carrera**

y aprobado en su forma final por el

**Curso de Ingeniería de Control y Automatización**

# Agradecimientos

Me gustaría agradecer mi familia y amigos, que me han dado el soporte para hacer este estudio.

Un grand agradecimiento a mi novia, Luisa, que sin su cariño y objetividad no estaría haciendo este trabajo.

También me gustaría agradecer a los tutores, Ramón Costa y Julio Normey Rico, que fueron el medio de conducción y base del conocimiento con toda la orientación para el objetivo final de este trabajo.

# Resumen

Para el tratamiento de señales con características periódicas, el control repetitivo (RC), basado en el modelo interno, se muestra como una técnica de control muy efectiva. Sin embargo, en su forma clásica no tiene la capacidad de tratar restricciones en las variables, cosa que el control predictivo tiene muy bien consolidada. Así, este proyecto objetiva la implementación y validación de una técnica de control que una el control repetitivo con el control predictivo basado en modelo, el MPC repetitivo.

**Palavras Clave:** Señales periódicas, Control Repetitivo, Control Predictivo, Validación, Implementación.

# Abstract

Repetitive control (RC), based on internal model, is a very effective methodology to treat periodic signals. Although its classical form can't handle restriction, which the predictive control has consolidated. Thus, this project aims to implement and validate a control technique that brings together repetitive control and predictive control, the Repetitive MPC.

**Key-words:** Periodic Signals, Repetitive Control, Predictive Control, Validation, Implementation.

# Índice de figuras

2.1	Estructura básica de un lazo repetitivo en tiempo continuo reproducida de [1]. . . . .	4
2.2	Diagrama de bloques del sistema de control de lazo cerrado con el controlador repetitivo “plug-in”. . . . .	5
2.3	Respuesta frecuencial del sistema en lazo abierto sin y con controlador repetitivo ( $K_r = 0,7$ , $H(z) = 1$ ). . . . .	8
2.4	Respuesta del sistema y acción de control de la estructura <i>plug-in</i> , para $K_r = 0,3$ (izquierda) y $K_r = 0,7$ (derecha). . . . .	9
3.1	Sistema de control con realimentación de estados observados. . . . .	11
3.2	Diagrama de bloques del problema de seguimiento de referencias con el lazo repetitivo insertado . . . . .	13
3.3	Respuesta del sistema y acción de control de la estructura con realimentación de estados observados. . . . .	16
3.4	Respuesta en frecuencia de la función de sensibilidad complementaria. . . . .	16
3.5	Respuesta del sistema y acción de control con polos ajustados del lazo cerrado de la estructura <i>plug-in</i> ( $K_r = 0,7$ ). . . . .	17
4.1	Estrategia del control predictivo utilizando el horizonte deslizante, donde apenas la primera señal de control calculada es implementada. Reproducido de [17]. . . . .	20
4.2	Esquema utilizado en algunos algoritmos MPC con acción de incremento por un periodo de muestreo y adaptación para insertar la acción repetitiva. . . . .	22
4.3	Respuesta del algoritmo MPC repetitivo sin restricciones aplicado en la planta $G_p$ . . . . .	29

5.1	Respuesta del algoritmo MPC Repetitivo con restricciones aplicado en la planta $G_p$ . Influencia de la restricción en $U$ (izquierda) y de la restricción en $\Delta U$ (derecha). . . . .	36
6.1	Estructura del roto-magnet con motor DC, imanes y <i>encoder</i> (izquierda) y esquema de actuación del aparato, donde $\omega$ es la velocidad angular y $\Gamma_p$ es la perturbación de torque (derecha). . . . .	39
6.2	Esquema de funcionamiento del <i>xPC Target</i> . . . . .	40
6.3	Simulación en Tiempo Real del Proceso en Lazo Abierto, con entrada de $0,35V$ . . . . .	41
6.4	Simulación en Tiempo Real con controlador PI. . . . .	43
6.5	Configuración de los equipos en el lab. . . . .	44
6.6	Simulación en Tiempo Real con Estructura <i>Plug-in</i> . . . . .	45
6.7	Simulación en Tiempo Real con Realimentación de Estados Observados. . . . .	46
6.8	Simulación en Tiempo Real con MPC Repetitivo . . . . .	46
6.9	Simulación en Tiempo Real con MPC Repetitivo con Restricciones. . . . .	47

# Índice general

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación . . . . .	1
1.2	Esbozo del Proyecto . . . . .	2
<b>2</b>	<b>Control Repetitivo</b>	<b>3</b>
2.1	Formulación del Modelo Interno . . . . .	3
2.2	Estructura “ <i>Plug-in</i> ” . . . . .	5
2.2.1	Un Ejemplo de Diseño del Controlador . . . . .	6
2.2.2	Simulación . . . . .	7
2.3	Conclusiones . . . . .	8
<b>3</b>	<b>Realimentación de Estados Observados</b>	<b>10</b>
3.1	Principio de la Separación . . . . .	10
3.2	Diseño del Observador . . . . .	12
3.3	Ajuste de la Representación del Sistema . . . . .	12
3.4	Diseño de las Ganancias . . . . .	14
3.5	Simulación . . . . .	14
3.5.1	Notación Simplificada . . . . .	15
3.5.2	Comparación con la Estructura “ <i>Plug-in</i> ” . . . . .	15
3.6	Conclusiones . . . . .	18
<b>4</b>	<b>MPC Repetitivo</b>	<b>19</b>
4.1	Control Predictivo Basado en Modelo (MPC) . . . . .	19

4.2	Modelo de Predicción . . . . .	20
4.3	Modelo en Espacio de Estados con Acción Repetitiva Integrada . . . . .	21
4.3.1	Cálculo de las Predicciones . . . . .	23
4.4	Respuesta Libre y Forzada . . . . .	24
4.5	Función Objetivo . . . . .	25
4.6	Obtención de la Ley de Control . . . . .	26
4.7	Modelo Sin Restricciones . . . . .	28
4.7.1	Ajustes . . . . .	28
4.7.2	Simulación . . . . .	28
4.8	Conclusiones . . . . .	30
<b>5</b>	<b>Tratamiento de Restricciones</b>	<b>31</b>
5.1	Restricción en la Variación de la Entrada . . . . .	32
5.1.1	Variación de un Periodo $T_p$ ( $NT_s$ ) . . . . .	32
5.2	Restricción en la Amplitud de la entrada . . . . .	32
5.3	Restricción en la salida . . . . .	34
5.4	Insertando las Restricciones en el Problema de Optimización . . . . .	34
5.5	Modelo Con Restricciones . . . . .	35
5.5.1	Ajustes . . . . .	35
5.5.2	Simulación . . . . .	35
5.6	Conclusiones . . . . .	37
<b>6</b>	<b>Aplicación Práctica</b>	<b>38</b>
6.1	Proceso Real . . . . .	38
6.2	<i>xPC Target</i> . . . . .	38
6.2.1	Funcionamiento . . . . .	39
6.3	Sistema en Lazo Abierto . . . . .	40

---

6.4	Ajustes de los Algoritmos . . . . .	41
6.5	Simulación - PI . . . . .	42
6.6	Simulaciones Experimentales . . . . .	43
6.6.1	<i>Plug-in</i> . . . . .	44
6.6.2	Realimentación de Estados Observados . . . . .	44
6.6.3	MPC Repetitivo . . . . .	45
6.6.4	MPC Repetitivo con Restricciones . . . . .	47
6.7	Análisis Comparativa . . . . .	48
<b>7</b>	<b>Conclusiones</b>	<b>49</b>
7.1	Trabajos Futuros . . . . .	49
	<b>Referencias</b>	<b>51</b>
	<b>Apéndice A - Códigos <i>Matlab</i> de las Simulaciones</b>	<b>53</b>
A.1	Estructura <i>Plug-in</i> . . . . .	53
A.2	Estructura con Realimentación de Estados . . . . .	54
A.3	MPC Repetitivo . . . . .	56
A.4	MPC Repetitivo con Restricciones . . . . .	58

# 1 Introducción

## 1.1: Motivación

A causa del crecimiento tecnológico las aplicaciones de sistemas de control presentan una gran complejidad. Para cada problema de control la complejidad se muestra de varias maneras, siendo una de ellas la característica de las señales que excitan el sistema en forma de perturbaciones a rechazar o consignas a seguir.

Cuando estas señales tienen características periódicas controladores clásicos tipo PI o PID no obtienen una respuesta satisfactoria. Sin embargo el control repetitivo posee características de tratamiento de señales periódicas que posibilitan alcanzar el desempeño deseado.

Señales con características periódicas están presentes en muchas aplicaciones reales que tienen sistemas mecánicos rotatorios o sistemas electro-electrónicos de potencia [1, 2, 20].

Para el ajuste de un controlador repetitivo utilizando la forma clásica existen algunos problemas, como la garantía de estabilidad cuando la planta posee determinadas características. Además, no tiene tratamiento de restricciones en las variables lo que puede provocar el mal desempeño del controlador o llevar el sistema a trabajar en condiciones límites. Pero el control predictivo basado en modelo tiene varias características que el control repetitivo clásico no tiene.

De esta forma, el objetivo es añadir el modelo interno del control repetitivo en un algoritmo de control MPC desarrollando un controlador que posee las características en conjunto. En la literatura existen estudios con este objetivo pero utilizando sistemas complejos. En [5] el modelo del sistema es un modelo discreto en espacio de estados variante en el tiempo. En [6] es utilizada una descomposición en frecuencia de la señal de referencia y esta es añadida a un modelo en espacio de estados. Y en [7] que considera el sistema repetitivo con dos dimensiones, donde una es el índice de cada período y la otra es el índice en el tiempo utilizando desigualdades de matrices lineales (LMI) para el cálculo de la solución.

Así, en este proyecto será desarrollada una forma sencilla de insertar el modelo interno repetitivo en el MPC.

## 1.2: Esbozo del Proyecto

Este proyecto empieza en el capítulo 1 que hace una introducción del control repetitivo, cuales condiciones trabaja y como puede ser mejorado con características del control predictivo.

El capítulo 2 presenta el control repetitivo en su estructura clásica *plug-in* con análisis de estabilidad y simulaciones.

En el capítulo 3 el modelo interno es insertado en una estructura de control por realimentación de estados observados, para una verificación de la necesidad del modelo interno para que el objetivo del control repetitivo sea cumplido.

Para el capítulo 4, está una introducción del control predictivo basado en modelo sus características y el desarrollo de una manera de insertar el modelo interno en el MPC.

El capítulo 5 muestra como es posible tratar las restricciones en el algoritmo MPC repetitivo desarrollado.

Para verificar el desempeño de los controladores, en el capítulo 6 son realizadas pruebas experimentales utilizando el *xPC Target* de todos los algoritmos, aplicados en el roto-magnet (que es un planta didáctica).

Las conclusiones de los resultados y futuros trabajos son explicados en el capítulo 7.

Por fin, se encuentran las referencias utilizadas en este trabajo y los anexos.

## 2 Control Repetitivo

### 2.1: Formulación del Modelo Interno

Tanto los controladores PID's como los repetitivos, son basados en el Principio del Modelo Interno (PMI) [3]. Este principio dice que es necesario un modelo interno de la señal a ser seguida o rechazada dentro del lazo de control, sea en el controlador o en proceso. Para los PID's, por ejemplo, hay el término integral ( $\frac{1}{s}$ ) que le representa, en el dominio de Laplace, la señal escalón.

Puesto que el control repetitivo se basa en el principio del modelo interno, es necesario introducir en el lazo de control un modelo generador de las señales que se desean seguir o rechazar. Así, una señal periódica de período  $T_0$  puede ser descrita como la suma infinita de funciones senoidales de frecuencia  $f_0$  ( $f_0 = 1/T_0$ ) y todas sus armónicas, conocido como serie de Fourier. Su representación, en la forma compleja, es mostrada en 2.1.

$$r(t) = \sum_{n=-\infty}^{\infty} a_n \cdot e^{j \frac{2\pi n t}{T_p}} \quad (2.1)$$

Donde  $T_p = T_0$  es el periodo de la señal,  $a_n$  los coeficientes relacionados con cada armónica.

Con esto, aplicando la transformada de Laplace en la ecuación anterior, se llega a la forma 2.2, la cual no es realizable de forma práctica, por tener el producto de infinitas funciones de transferencia. De esta manera es necesario reescribir la ecuación 2.2 de una forma mas compacta e implementable, obteniendo 2.3.

$$R(s) = \frac{1}{s} \prod_{n=1}^{\infty} \frac{\left(\frac{2\pi n}{T_p}\right)^2}{s^2 + \left(\left(\frac{2\pi n}{T_p}\right)^2\right)} \quad (2.2)$$

$$R(s) = \frac{T_p \cdot e^{-\frac{s T_p}{2}}}{1 - e^{-s T_p}} \quad (2.3)$$

Analizando la ecuación se puede interpretar que en el numerador  $T_p \cdot e^{-\frac{sT_p}{2}}$  hay apenas una ganancia  $T_p$  y un retardo puro. Así, para reproducirse una señal periódica es suficiente agregar la siguiente función de transferencia  $(1)/(1 - e^{-sT_p})$  en el lazo de control [1, 2]. Este sistema puede ser implementado como un lazo de realimentación positiva con o termino  $e^{-sT_p}$  en el camino de realimentación como en la figura 2.1.

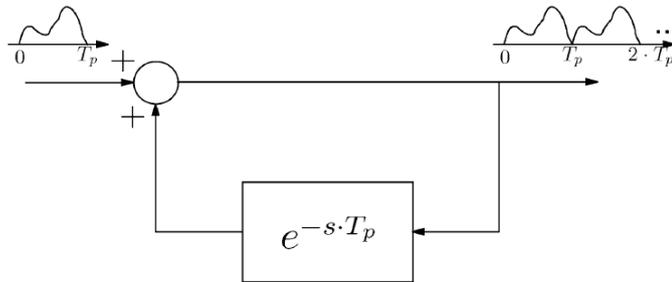


Figura 2.1: Estructura básica de un lazo repetitivo en tiempo continuo reproducida de [1].

En cierto modo se puede comparar el control repetitivo con algunas técnicas de control por aprendizaje. Su estructura básica puede aprender una señal de tamaño  $T_p$  y repetirla como una señal periódica de período  $T_p$ , si la entrada aplicada es diferente de cero durante  $T_p$  y cero después de  $T_p$ . El comportamiento del lazo repetitivo en el dominio temporal, puede ser analizado como una acción de integración de período  $T_p$  por período  $T_p$ . Esto puede ser comparado de la misma forma como que para seguir (o rechazar) una señal de tipo escalón, donde el controlador necesita la acción integral en su estructura (como los PID's por ejemplo).

El gran problema de la estructura básica es la implementación práctica de un retardo puro en tiempo continuo. Sin embargo, en tiempo discreto esta tarea es simple. Por lo tanto, sólo hay que añadir la ecuación 2.4 al lazo de control. Donde  $N = T_p/T_s$  representa el número de muestras que se almacenan. El valor depende del período de muestreo  $T_s$  y del período  $T_p$  de la señal a seguir o rechazar. Con esta restricción el sistema de control sólo conseguirá reproducir las componentes armónicas que se encuentran abajo de la frecuencia de Nyquist ( $\omega_s/2 = \pi/T_s$ ).

$$Gr(z) = \frac{z^{-N}}{1 - z^{-N}} = \frac{1}{z^N - 1} \quad (2.4)$$

## 2.2: Estructura "Plug-in"

El concepto de control repetitivo puede ser aplicado de muchas maneras diferentes. Estas formas siempre se basan en un sistema como la ecuación 2.4 insertada en el lazo de control.

Una de las formas es la técnica "plug-in", donde el controlador repetitivo es insertado en el lazo de control para mejorar un controlador nominal existente ( $G_c(z)$ ) figura 2.2.

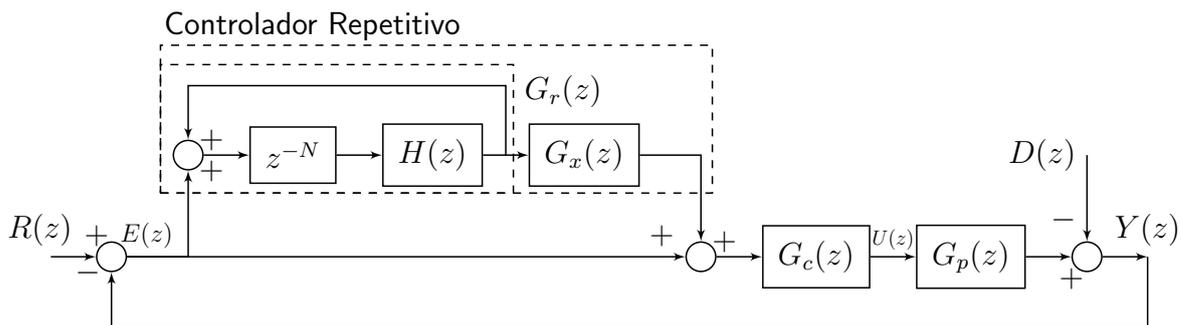


Figura 2.2: Diagrama de bloques del sistema de control de lazo cerrado con el controlador repetitivo "plug-in".

Donde el controlador nominal  $G_c(z)$  tiene la función de estabilizar la planta  $G_p(z)$  y rechazar perturbaciones  $D(z)$  en un determinado espectro de frecuencias.

El bloque  $H(z)$  es generalmente un filtro FIR (*Finite-Impulse Response*) pasa bajo de fase nula, haciendo que el sistema de lazo cerrado no tenga una alta ganancia en las frecuencias donde hay incertidumbres en el modelo. Lo que resulta en una mayor robustez del sistema. Tampoco hay problemas de causalidad, ya que el filtro está en serie con  $N$  retrasos.

El bloque  $G_x(z)$  tiene la función de garantizar la estabilidad del sistema de lazo cerrado, lo que depende de las tres condiciones siguientes según [2]:

- **Primera condición de estabilidad:** El sistema de lazo cerrado sin el controlador repetitivo debe ser estable. Esto se consigue con el diseño del controlador nominal  $G_c(z)$ .
- **Segunda condición de estabilidad:** El filtro  $H(z)$  debe tener módulo inferior

a 1 ( $\|H(z)\|_\infty < 1$ ). Como se trata de un filtro diseñado, esta condición implica algunas restricciones en su diseño.

- **Tercera condición de estabilidad:** El bloque  $G_x(z)$  tiene papel clave para que las condiciones de estabilidad sean cumplidas. De forma que, las características del sistema de lazo cerrado sin el controlador repetitivo tienen una gran influencia en esta condición, como muestra la ecuación 2.5.

$$\|1 - \frac{G_c(z)G_p(z)}{1 + G_c(z)G_p(z)}G_x(z)\|_\infty < 1 \quad (2.5)$$

Una forma común de seleccionar  $G_x(z)$  garantizando las condiciones, es  $G_x(z) = K_r(G_c(z)G_p(z)/(1 + (G_c(z)G_p(z))))^{-1}$ . Pero, si utilizada, puede causar varios problemas como, por ejemplo, cuando la planta tiene fase no mínima, pues algunas cancelaciones prohibidas pueden aparecer en la construcción de la función de transferencia de lazo abierto, lo que hace que el sistema no sea internamente estable.

En [2] surge este problema, y para solucionarlo es utilizado un enfoque donde los ceros de fase mínima son cancelados y las fases de los ceros de fase no-mínima son compensadas [4].

### 2.2.1: Un Ejemplo de Diseño del Controlador

Para analizar esta estructura de control, serán implementadas algunas simulaciones con el modelo de proceso:

$$G_p(s) = \frac{K_p e^{-T_d s}}{\tau s + 1}, \quad K_p = 0,3, T_d = 0,02, \tau = 0,1 \quad (2.6)$$

En el diseño, es necesario como primero paso fijar el período de muestreo  $T_s$ , y para esto algunas variables ayudan imponiendo restricciones como: el valor del retardo puro  $T_d$ , el período de la señal a seguir o rechazar  $T_p$  y el valor de muestras a almacenar  $N$ . Donde  $T_s$  necesita ser submúltiplo entero del  $T_d$  y del  $T_p$  ( $T_p = NT_s$ ) y también:

- El periodo de muestreo debe identificar la respuesta transitoria de la planta. Utilizando el periodo de muestreo menor que diez veces la menor constante de tiempo ( $10T_s < \tau$ ) es una buena elección.
- $T_s$  también debe ser un submúltiplo entero tanto del periodo fundamental ( $T_p$ ) de la señal a rechazar o seguir ( $T_p = NT_s$ ) cuanto del retardo puro  $T_d$ .

- la cantidad de muestras a almacenar ( $N$ ) debe ser de tamaño suficiente para que la señal a seguir o rechazar en forma de muestras sea lo más parecida con su forma continua.

La señal que el controlador tendrá que reproducir tiene el periodo  $T_p = 0,4$ , que es múltiplo del retardo puro. Así, los parámetros son definidos como:

- $T_p = 0,4$  ,  $N = 20$  ,  $T_s = T_p/N = 0,02$ .

Entonces los bloques  $G_c(z)$ ,  $H(z)$  y  $G_x(z)$  son diseñados para garantizar la estabilidad del sistema. De esta forma,  $G_c(z)$  debe garantizar la primera condición de estabilidad haciendo con que el sistema en lazo cerrado sin el controlador repetitivo sea estable. Como la planta es estable tanto en lazo abierto como en lazo cerrado escogiendo  $G_c(z) = 1,5$  de forma simplificada la primera condición es garantizada.

El bloque  $H(z)$  debe cumplir la segunda condición. Puede ser escogido como  $H(z) = 1$ , de forma que no existirá un filtrado en las altas frecuencias. Esto puede ser hecho insertando un filtro FIR pasa bajo de fase nula mejorando la robustez.

Como la planta no tiene ceros de fase no mínima,  $G_x(z)$  es diseñado con la estructura  $G_x(z) = K_r(G_c(z)G_p(z)/(1 + (G_c(z)G_p(z))))^{-1}$ .

Es posible observar la acción del controlador repetitivo en la figura 2.3, donde es trazado el diagrama de Bode del sistema en lazo abierto sin y con el repetitivo. De forma que, en el caso con repetitivo existen picos de resonancia en la frecuencia fundamental de la señal ( $f_p = 2\pi/T_p = 15,7rad/s$ ) y sus armónicas. Esto garantiza el rechazo o seguimiento de la señal periódica, pero trae elevadas ganancias en un rango de altas frecuencias donde hay errores de modelado y ruidos. Con el bloque  $H(z)$  diseñado como un filtro FIR estas ganancias en alta frecuencias son filtradas obteniendo una mejor robustez.

### 2.2.2: Simulación

Para verificar el funcionamiento del controlador, se realizará una simulación donde al principio se aplica una señal escalón como referencia y entonces, una perturbación periódica actúa en el sistema por un periodo de tiempo.

$$Ref = 1 , Perturbacion = 0,5\sin\left(\frac{2\pi}{T_p}\right) + 0,25\sin\left(\frac{4\pi}{T_p}\right)$$

La figura 2.4 presenta la influencia del parámetro  $K_r$  en el lazo de control con la estructura "plug-in". De forma que, observando el desenvolvimiento del sistema, con

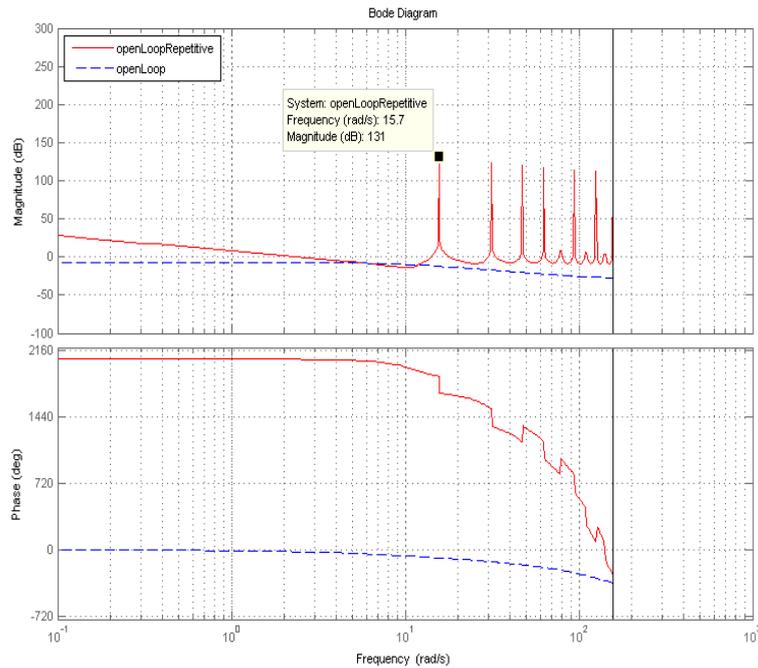


Figura 2.3: Respuesta frecuencial del sistema en lazo abierto sin y con controlador repetitivo ( $K_r = 0,7$ ,  $H(z) = 1$ ).

valores de  $K_r$  próximos de 0 la acción de control es más conservadora y el seguimiento de la referencia tarda más. Siendo lo contrario cuando  $K_r$  asume valores próximos a 1. En los dos casos la perturbación es rechazada completamente.

## 2.3: Conclusiones

Este capítulo tuvo como tema la explicación de la estructura *plug-in* del control repetitivo, sus condiciones de estabilidad y parámetros de diseño con algunas simulaciones.

Enseñando las características necesarias para que el sistema tenga la capacidad de rechazar (o seguir) las perturbaciones (o referencias) periódicas, como los polos del lazo abierto y los picos de ganancias en las frecuencias fundamentales de la perturbación. Esta estructura representa la forma clásica del control repetitivo y sirve como una base para se empezar cualquier estudio y implementación de un algoritmo de control con el modelo interno repetitivo.

Con esto, como un próximo paso será insertar este modelo interno en una estructura de control diferente, como la realimentación de estados.

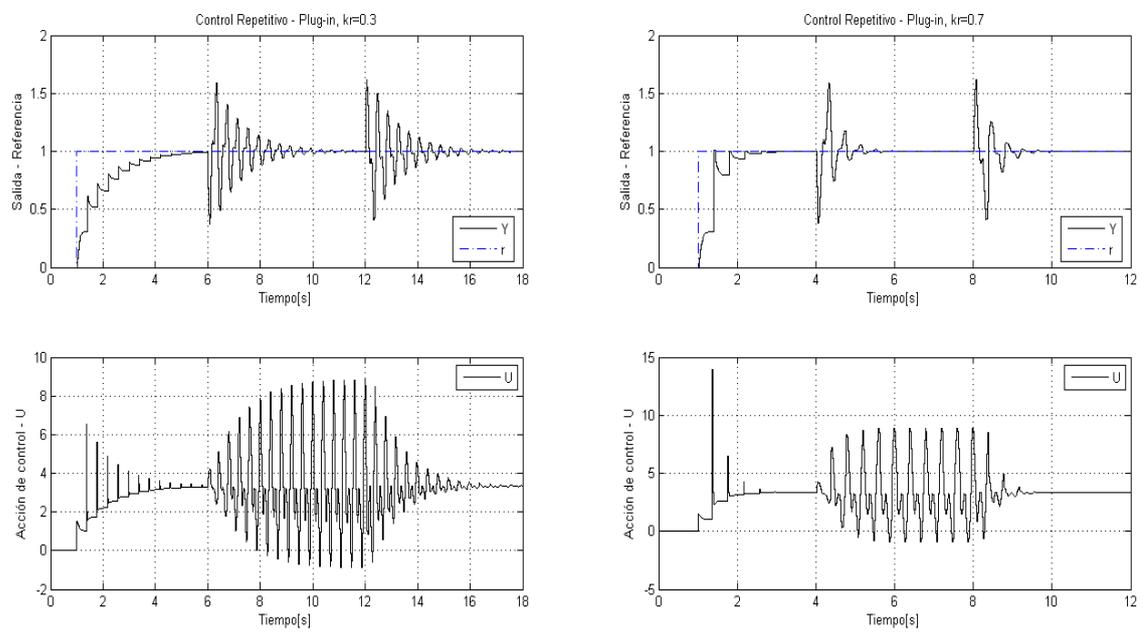


Figura 2.4: Respuesta del sistema y acción de control de la estructura *plug-in*, para  $K_r = 0,3$  (izquierda) y  $K_r = 0,7$  (derecha).

# 3 Realimentación de Estados Observados

Otra forma de aplicar el concepto del control repetitivo es utilizando la idea del control por realimentación de estados observados. Después de insertar el lazo repetitivo dentro de este esquema de control con las modificaciones necesarias, el nuevo esquema de control tendrá la capacidad de reproducir señales con características periódicas.

De forma más específica, el lazo repetitivo será insertado dentro del problema de seguimiento de referencias utilizando integradores. De acuerdo con los siguientes pasos:

1. Diseño del observador;
2. Ajuste de la representación del sistema;
3. Diseño de las ganancias de realimentación de estados;

## 3.1: Principio de la Separación

El desarrollo de este aparato utiliza como base un sistema de control como en la figura 3.1, que muestra la forma clásica en tiempo discreto del modelo de control por realimentación de estados observados.

Este modelo puede ser representado como el siguiente conjunto de ecuaciones:

$$x(k+1) = Ax(k) + Bu(k) \tag{3.1}$$

$$y(k) = Cx(k) \tag{3.2}$$

$$\hat{x}(k+1) = (A - K_f C)\hat{x}(k) + Bu(k) + K_f y(k) \tag{3.3}$$

$$u(k) = K\hat{x}(k) + r(k) \tag{3.4}$$

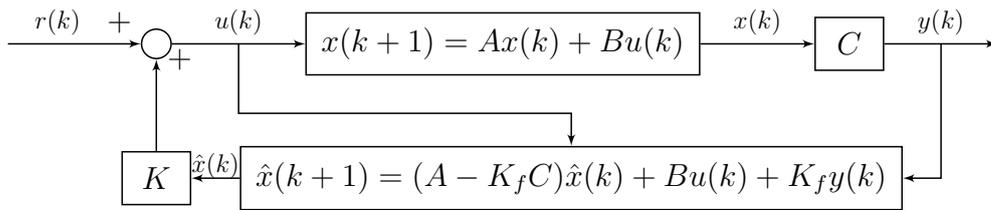


Figura 3.1: Sistema de control con realimentación de estados observados.

Donde  $x$  son los estados del sistema a ser controlado,  $y$  la salida de medidas,  $u$  el control,  $\hat{x}$  los estados estimados,  $r$  la referencia y  $A, B, C, K, K_f$  son matrices constantes con dimensiones adecuadas.

Utilizando las variables de estado  $x(k)$  y el error de estimación  $e(k) = x(k) - \hat{x}(k)$ , este sistema tiene la siguiente representación en lazo cerrado:

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ e(k+1) \end{bmatrix} &= \begin{bmatrix} A + BK & -BK \\ 0 & A - K_f C \end{bmatrix} \begin{bmatrix} x(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} r(k) \\ y(k) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ e(k) \end{bmatrix} \end{aligned} \quad (3.5)$$

Cuyos valores propios son obtenidos de la siguiente ecuación característica:

$$\begin{aligned} \det \left( \begin{bmatrix} sI_n - (A + BK) & BK \\ 0 & sI_n - (A - K_f C) \end{bmatrix} \right) &= 0 \\ \det(sI_n - (A + BK)) \det(sI_n - (A - K_f C)) &= 0 \end{aligned} \quad (3.6)$$

Los valores propios de las matrices  $A + BK$  e  $A - K_f C$  definen la dinámica del sistema en lazo cerrado. Así, como la dinámica del error de estimación depende de la ganancia del observador, normalmente esta es diseñada para que sea más rápida que la dinámica del sistema de lazo cerrado sin comprometer la estabilidad y robustez.

Con esto, es posible verificar que la dinámica del sistema  $A + BK$  es dominante, porque son los valores propios del sistema de lazo cerrado cuando la dinámica del error de estimación de los estados es nula.

De este modo, el “Principio de la Separación” dice que es posible realizar el diseño de las ganancias del observador de forma independiente del proyecto de las ganancias de realimentación de estados [8, 9].

## 3.2: Diseño del Observador

Como estimador de los estados será utilizado el filtro de Kalman discreto <sup>1</sup>, que es un observador de tipo Luenberger de orden completa con la forma:

$$\hat{x}(k+1) = (A - LC)\hat{x}(k) + Bu(k) + Ly(k) \quad (3.7)$$

Este filtro de tiene una particularidad; el calculo de su ganancia lleva el error de estimación a tener covarianza mínima.

Para que esto sea posible, el conjunto  $(A, C)$  tiene que ser detectable, porque así se puede estabilizar la dinámica del error de estimación [9].

## 3.3: Ajuste de la Representación del Sistema

Para realizar el ajuste de la representación del sistema, el lazo repetitivo será insertado en el problema de seguimiento de referencias con integradores.

Primeramente, el lazo repetitivo en forma de función de transferencia  $1/(z^N - 1)$  es puesto en representación de espacio de estados con la forma:

$$x_r(k+1) = A_r x_r(k) + B_r u_r(k) \quad (3.8)$$

Donde las matrices  $A_r, B_r$  son resultantes de la realización de estados de la función de transferencia del lazo repetitivo, que puede representarse de varias formas, ya que la función de transferencia del lazo repetitivo es propia <sup>2</sup>, por ejemplo:

- Forma canónica compañera controlable;
- Forma canónica compañera observable;
- Forma canónica modal ( $A_r$  esta en su forma de Jordan);
- Forma mínima ( $A_r, B_r$  es controlable y  $A_r, C_r$  es observable);

Las formas compañeras no son muy bien condicionadas y pueden traer algunos errores numéricos, de modo que será escogida la forma modal para representar el lazo repetitivo. Esta representación tiene la característica de que la matriz  $A_r$  sea

<sup>1</sup>En *matlab* la función *kalman* calcula el observador basada en el modelo de proceso.

<sup>2</sup>Una función de transferencia propia tiene grado del numerador menor o igual al grado del denominador ( $\text{gradoNum} \leq \text{gradoDen}$ ).

diagonal en bloques que son los valores propios del sistema o los polos de la ecuación característica del sistema, es decir, se el sistema tiene valores propios en  $(\lambda_1, \sigma \pm j\omega, \lambda_2)$ , la matriz  $A_R$  en la forma modal tendrá la forma:

$$A_r = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \sigma & \omega & 0 \\ 0 & -\omega & \sigma & 0 \\ 0 & 0 & 0 & \lambda_2 \end{bmatrix}$$

Definida la representación a utilizar, entonces el integrador es cambiado por el lazo repetitivo en espacio de estados y el sistema se queda como en la figura 3.2.

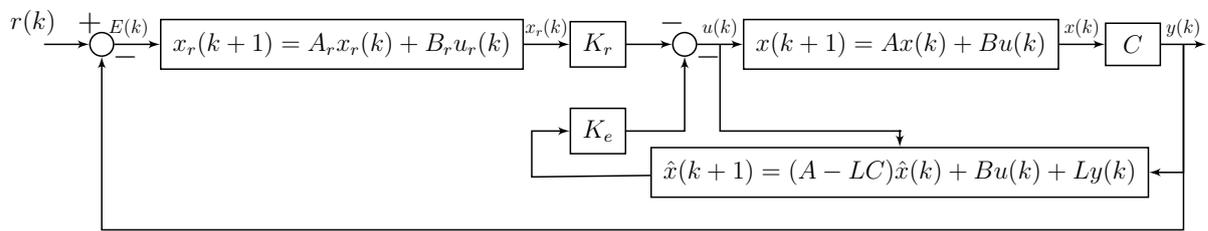


Figura 3.2: Diagrama de bloques del problema de seguimiento de referencias con el lazo repetitivo insertado

Como  $E(k) = u_r(k)$ , y con  $x(k)$  siendo los estados de la planta y  $x_r(k)$  los estados del lazo repetitivo el sistema tiene la siguiente representación cuando la dinámica del error de estimación es nula:

$$\begin{aligned} \underbrace{\begin{bmatrix} x(k+1) \\ x_r(k+1) \end{bmatrix}}_{x_a(k+1)} &= \underbrace{\begin{bmatrix} A & 0 \\ -B_r C & A_r \end{bmatrix}}_{A_a} \underbrace{\begin{bmatrix} x(k) \\ x_r(k) \end{bmatrix}}_{x_a(k)} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_a} u(k) + \underbrace{\begin{bmatrix} 0 \\ B_r \end{bmatrix}}_{E_a} r(k) \\ y(k) &= \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C_a} \underbrace{\begin{bmatrix} x(k) \\ x_r(k) \end{bmatrix}}_{x_a(k)}, \quad u(k) = - \underbrace{\begin{bmatrix} K_e & K_r \end{bmatrix}}_{K_a} \underbrace{\begin{bmatrix} x(k) \\ x_r(k) \end{bmatrix}}_{x_a(k)} \end{aligned} \quad (3.9)$$

El sistema de control tiene las ecuaciones:

$$\begin{aligned} \begin{bmatrix} x_a(k+1) \\ e(k+1) \end{bmatrix} &= \begin{bmatrix} A_a - B_a K_a & B_a K_e \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x_a(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} E_a \\ 0 \end{bmatrix} r(k) \\ y(k) &= \begin{bmatrix} C_a & 0 \end{bmatrix} \begin{bmatrix} x_a(k) \\ e(k) \end{bmatrix} \end{aligned} \quad (3.10)$$

Donde  $e(k) = x(k) - \hat{x}(k)$  es el error de estimación de los estados,  $L$  la ganancia del observador y  $A_a, B_a, C_a, E_a, x_a, K_a$  son matrices de la representación 3.9.

El lazo repetitivo es insertado en el problema de seguimiento de referencias y es posible realizar el proyecto de las ganancias de realimentación de estados.

### 3.4: Diseño de las Ganancias

Dada la nueva representación del sistema, el problema es determinar  $L$  y  $K_a$  para que los valores propios de las matrices  $A_a - B_a K_a$  y  $A - LC$  estén dentro del círculo unitario. Con el principio de la separación es posible proyectar  $L$  como en el filtro de Kalman y, de forma independiente, proyectar la matriz de ganancias  $K_a$  por asignación de valores propios <sup>3</sup>.

Entonces, para el diseño de las ganancias de realimentación de estados es necesario verificar cuantos valores propios están presentes en el sistema de lazo cerrado. Porque esta cantidad tiene una relación con el valor  $N$  y también con el período de muestreo  $T_s$ , ya que los valores propios de la matriz  $A_r$  en espacio de estados son los polos del lazo repetitivo  $1/(z^N - 1)$ .

De esta forma, para ajustar el esquema de control será utilizada la misma planta 2.6 de las pruebas aplicadas en la estructura *Plug-in*, con las mismas condiciones.

Pero, utilizando la representación en espacio de estados del repetitivo surge una nueva restricción en  $N$ , porque su selección va a influenciar en la cantidad de valores propios que serán asignados en el sistema de lazo cerrado. Mientras este valor sea pequeño, la señal no será reconocida completamente (la señal en forma de muestras será muy diferente que en su forma continua) trayendo problemas de estabilidad y desempeño. Si tiene un valor grande, el sistema resultará de orden elevada, y algunos problemas numéricos pueden ocurrir en los cálculos de la representación de estados o en la asignación de valores propios <sup>4</sup>.

### 3.5: Simulación

En la simulación serán consideradas dos formas para la asignación de polos, una es utilizando una notación simplificada y la otra es utilizando los mismos polos de

<sup>3</sup>Con la rutina *place* del *matlab* es posible encontrar la matriz de ganancias  $K$  de realimentación de estados para que los valores propios de  $A - BK$  estén en un lugar deseado.

<sup>4</sup>La rutina *place* del *matlab*, genera un error cuando se intenta asignar valores propios muy próximos.

la estructura *plug-in*, para compararlas.

También se utilizarán las mismas señales que están presentes en la simulación de la estructura *plug-in*.

De este modo, los parámetros de la simulación son:

- $T_p = 0,4$  ,  $N = 20$  ,  $T_s = T_p/N = 0,02$ .
- $Ref = 1$  ,  $Perturbacion = 0,5\sin(\frac{2\pi}{T_p}) + 0,25\sin(\frac{4\pi}{T_p})$

Con esto, existirán  $N + x$  valores propios en el sistema para que sean asignados, donde  $x$  son los valores propios de la representación en espacio de estados del proceso a ser controlado.

### 3.5.1: Notación Simplificada

Para ajustar la posición  $P$  de los valores propios del sistema en lazo cerrado, puede utilizarse la siguiente notación

$$P = [vp_{lc} \quad \alpha \cdot vp_{Ar}] , \quad \alpha \in (0, 1)$$

De esta manera,  $vp_{lc}$  son los valores propios que serán asignados y son referentes a los valores propios de la representación en espacio de estados del proceso. El parámetro  $\alpha$  garantiza que los valores propios del lazo repetitivo ( $vp_{Ar}$ ) se encuentren dentro del círculo unitario.

Es posible observar en las figuras 3.3 y 3.4, que el sistema es capaz de rechazar completamente la perturbación periódica visualizando tanto su respuesta temporal como su respuesta en frecuencia. Se observan los picos de resonancia en la frecuencia fundamental de la señal ( $f_p = 1/T_p = 15,7\text{rad/s}$ ) y en sus armónicas. En esta simulación los polos de lazo cerrado son ajustados con  $vp_{lc} = [0,8 \ 0,1]$  y  $\alpha = 0,9$ .

### 3.5.2: Comparación con la Estructura “Plug-in”

Para hacer esta comparación, fueran cogidos los polos del sistema en lazo cerrado con el controlador repetitivo (ajustado con  $Kr = 0,7$ ) y puestos como objetivo de la asignación de polos de la estructura con realimentación de estados observados. En la figura 3.5 está el resultado de esta simulación que puede ser comparada con la figura 2.4. Ambos sistemas tienen una respuesta semejante para seguir un escalón y

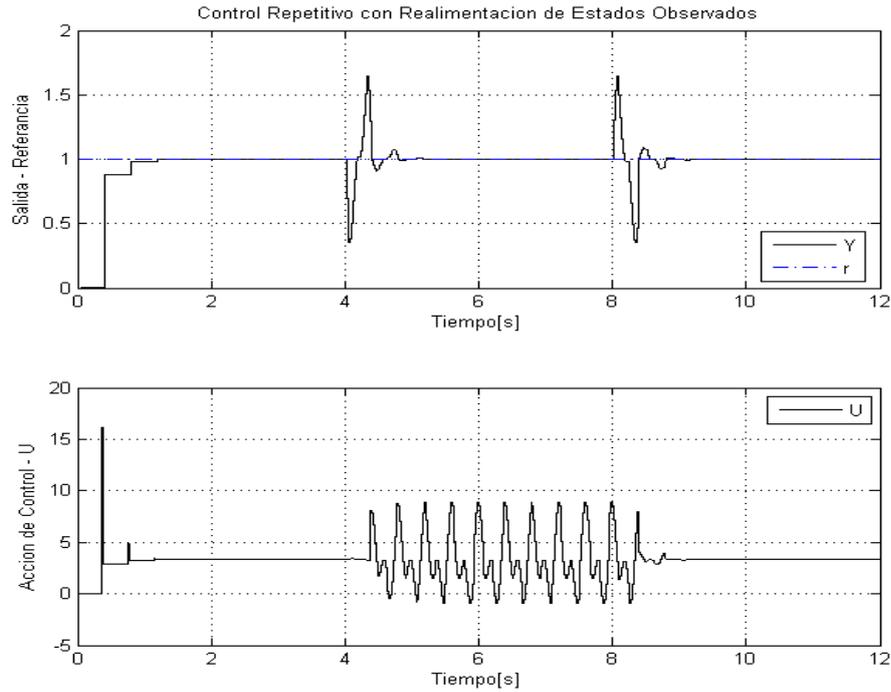


Figura 3.3: Respuesta del sistema y acción de control de la estructura con realimentación de estados observados.

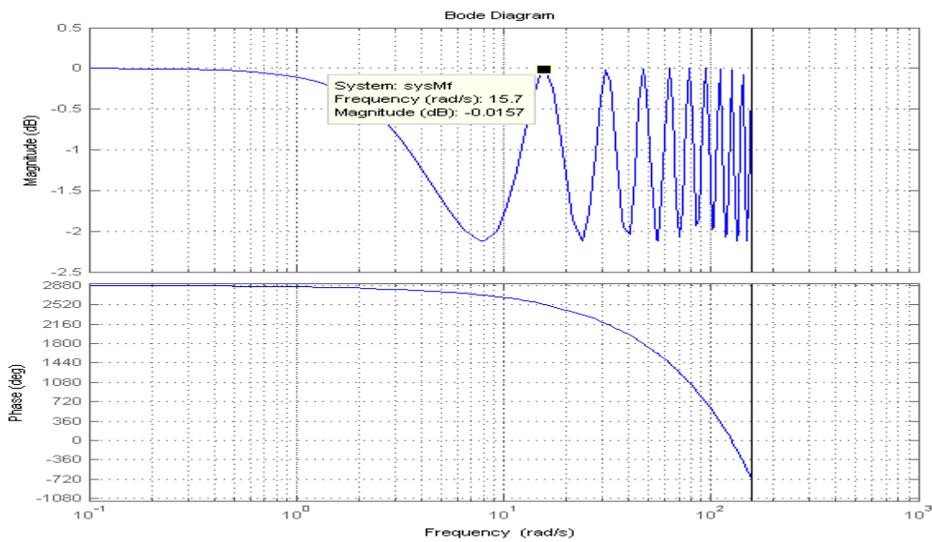


Figura 3.4: Respuesta en frecuencia de la función de sensibilidad complementaria.

rechazar la perturbación sin embargo difieren en la forma de la salida, que en el caso *plug-in* tiene unos saltos, que no pasa en el caso en estudio.

Las diferencias son muy grandes si se comparan las dos estructuras de control.

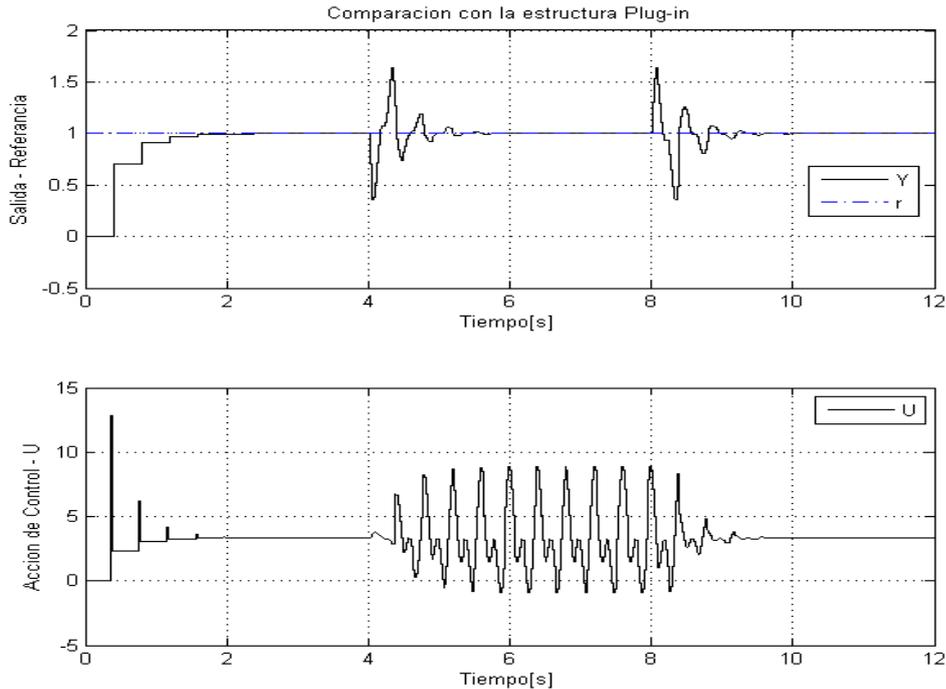


Figura 3.5: Respuesta del sistema y acción de control con polos ajustados del lazo cerrado de la estructura *plug-in* ( $K_r = 0,7$ ).

Las restricciones en el periodo de muestreo  $T_s$  son las mismas, excepto que en el caso con realimentación de estados  $N$  influencia el tamaño de la matriz de estados del repetitivo ( $A_r$ ). Pero en los dos casos, el diseño empieza con la selección del periodo de muestreo y esto está relacionado con:

- La dinámica y el retardo puro del sistema a ser controlado;
- El período fundamental  $T_p$  de la señal a seguir o rechazar;
- La cantidad de muestras a almacenar  $N$  dentro de cada período.

En la estructura *Plug-in* los parámetros de diseño son: el controlador nominal  $G_c(z)$ , el filtro repetitivo  $H(z)$  y la función  $G_x(z)$  que trabajan en conjunto para garantizar la estabilidad del sistema en lazo cerrado. Fijados los parámetros, es posible cambiar la dinámica del sistema por medio de la variable  $K_r$  con un grado de libertad, y tener respuestas con acciones de control más agresivas ( $K_r p \rightarrow 1$ ) o más conservadoras ( $K_r \rightarrow 0$ ).

Con la realimentación de estados observados son necesarios: el modelo en espacio de estados del lazo repetitivo 3.8 (puede tener o no, el filtro  $H$  insertado)

y el modelo en espacio de estados del proceso para diseño del observador. Así, la dinámica del sistema es definida por las ganancias  $K_a$  y  $L$ , que por el principio de la separación pueden ser diseñadas de forma separadas.

Sin embargo, para realizar un cambio en la dinámica del sistema no es de forma tan fácil como en la estructura *Plug-in* variándose el  $K_r$ , porque la ganancia de realimentación  $K_a$  para la asignación de los valores propios tiene  $N + x$  grados de libertad que dependen de los estados del repetitivo  $K_r$  y de los estados del observador  $K_e$ . Esta ganancia es calculada utilizando la función *place* que no es un planteamiento tan sencillo, pues los elementos de  $K_a \in \mathfrak{R}$  pueden asumir mucho más valores que  $K_r$ .

### 3.6: Conclusiones

En este capítulo fue planteada una manera de insertar el modelo interno en una estructura de control diferente, en el caso, la realimentación de estados observados que tiene diferentes características y parámetros de diseño. En esta representación el sistema necesita de ajustes diferentes y tiene más grados de libertad que pueden influenciar la dinámica del sistema en lazo cerrado que viene junto con un diseño más complejo (ganancias del observador y de la realimentación de estados).

Con los polos del sistema en lazo cerrado de la estructura *plug-in*, fue posible hacer una comparación del desempeño de ambos controladores como se tuviesen el mismo diseño, ajustando la posición deseada de los polos para la estructura con realimentación de estados observados.

Por consiguiente, será desarrollado el método para la inserción del modelo interno en el MPC.

## 4 MPC Repetitivo

Para que sea posible utilizar las ventajas de un controlador predictivo conjuntamente con las del control repetitivo, es necesario añadir la estructura básica del repetitivo en el sistema de control predictivo, como dice el PMI.

Así, será implementado un algoritmo de control MPC de tal forma que sea posible la inserción de la estructura básica repetitiva en el sistema de control. Para este estudio primeramente se realiza una revisión del MPC.

### 4.1: Control Predictivo Basado en Modelo (MPC)

El control predictivo se refiere a una clase de algoritmos de control, que usan como base un modelo explícito del proceso a ser controlado para predecir el comportamiento futuro de la salida. Esta capacidad de predicción posibilita calcular la señal de control óptima minimizando una función objetivo dentro de un horizonte futuro <sup>1</sup>, como en la figura 4.1. Siendo posible insertar en este problema las restricciones que las variables manipuladas y de proceso están sujetas. Con estas características, el control predictivo tiene algunas ventajas y desventajas como:

- Puede ser utilizado para controlar procesos con diferentes características como: con dinámicas complejas, simples, grandes retrasos, fase no mínima o inestables.
- Hace la compensación del tiempo muerto del proceso de forma intrínseca.
- Encapsula el comportamiento de múltiples controladores SISO y las acciones de realimentación y anticipación pueden ser incluidas de forma directa.

---

<sup>1</sup>La idea del horizonte deslizante es que en cada instante, el horizonte es dislocado en el futuro y apenas la primera señal de control calculada es aplicada.

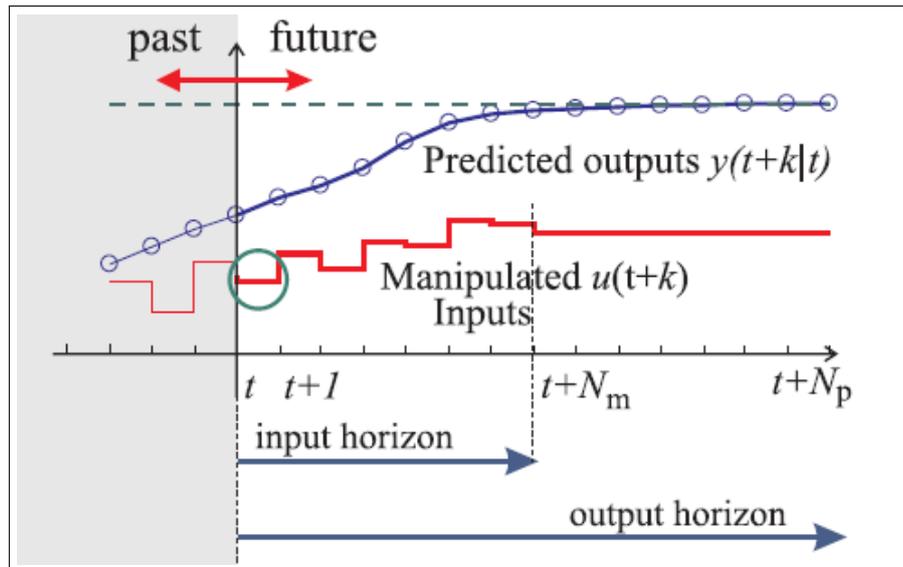


Figura 4.1: Estrategia del control predictivo utilizando el horizonte deslizante, donde apenas la primera señal de control calculada es implementada. Reproducido de [17].

- Puede ser utilizado tanto para sistemas SISO cuanto para MIMO con interacciones entre los procesos.

Los siguientes elementos básicos están presentes [10, 11]:

- El modelo de predicción;
- La función objetivo;
- El método para obtención de la ley de control;

Así, los elementos básicos de esta formulación del MPC repetitivo son descritos en la secuencia.

## 4.2: Modelo de Predicción

El modelo de predicción de la planta es muy importante para el controlador, porque es necesario saber el valor de la salida en instantes de tiempo futuros para que sea posible calcular la señal de control futura. Dicho modelo es una parte del problema de optimización a resolver [16].

Para esto, es necesario describir el comportamiento dinámico que relaciona las entradas con las salidas del proceso, de forma que esta descripción sea similar al

comportamiento del proceso real.

Existen varios métodos de modelar el proceso [13], como:

- Respuesta Impulsiva;
- Respuesta al Escalón;
- Función de Transferencia;
- Espacio de Estados;
- Otros modelos cuando no es posible usar modelos lineales:
  - Modelos No Lineales;
  - Redes Neuronales;
  - Lógica Nebulosa (fuzzy);

### 4.3: Modelo en Espacio de Estados con Acción Repetitiva Integrada

Teniendo el modelo en espacio de estados del proceso en lazo abierto como:

$$x_p(k+1) = A_p x_p(k) + B_p u(k) \quad (4.1)$$

$$y(k) = C_p x_p(k) \quad (4.2)$$

es necesario hacer algunos cambios en la representación para que el modelo tenga la acción repetitiva integrada.

Diferentes algoritmos de control MPC son basados en la variación de la señal de control ( $u(k) = u(k-1) + \Delta u(k)$ ), es decir, el controlador calcula el incremento de la acción de control. Con esto, el sistema de control tiene una acción integral embebida ( $1/1 - z^{-1}$ ).

Para el caso de tener una perturbación repetitiva, considerando el PMI, el modelo usado tendrá el lazo repetitivo embebido, como se muestra en la figura 4.2.

Para el desarrollo del modelo en variables de estado se definen las variables:

$$\Delta x_p(k) = x_p(k) - x_p(k-N) \quad (4.3)$$

$$\Delta u(k) = u(k) - u(k-N) \quad (4.4)$$

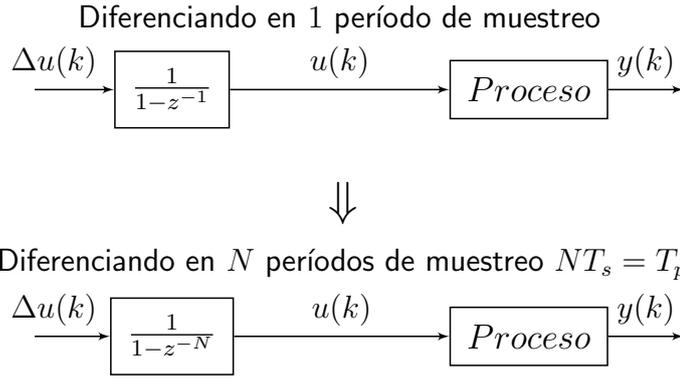


Figura 4.2: Esquema utilizado en algunos algoritmos MPC con acción de incremento por un periodo de muestreo y adaptación para insertar la acción repetitiva.

que son los incrementos del estado  $x_p(k)$  y del control  $u(k)$  a cada período  $T_p = NT_s$  y aplicándolas en la ecuación de espacio de estados del proceso 4.1, llegase al siguiente sistema

$$\Delta x_p(k+1) = A_p \Delta x_p(k) + B_p \Delta u(k) \quad (4.5)$$

Donde la entrada de este sistema es la variación de la señal de control  $\Delta u(k)$  de un período al otro. Con esto, ahora es necesario insertar la salida de la planta  $y(k)$  en el modelo para que sea posible hacer las predicciones de la salida en función de la entrada  $\Delta u(k)$ , es decir, que para calcular las predicciones debe haber una relación entre  $y(k)$  y  $\Delta u(k)$ .

Así, con la expresión

$$\begin{aligned} y(k+1) - y(k-N+1) &= C_p(x_p(k+1) - x_p(k-N+1)) = C_p \Delta x_p(k+1) \\ y(k+1) &= C_p A_p \Delta x_p(k) + C_p B_p \Delta u(k) + y(k-N+1) \end{aligned} \quad (4.6)$$

se puede observar que en esta expresión el valor de  $y(k+1)$  en un período está relacionada con el valor de  $y(k-N+1)$  del período anterior. De esta manera, introduciendo el vector

$$x(k) = [\Delta x_p(k)^T \ y(k) \ y(k-1) \ y(k-2) \ \dots \ y(k-N+1) \ y(k-N)]^T$$

representando el vector de estados de la nueva representación del sistema, que tiene las ecuaciones

$$\begin{aligned}
 \underbrace{\begin{bmatrix} \Delta x_p(k+1) \\ y(k+1) \\ y(k) \\ \vdots \\ y(k-N+2) \\ y(k-N+1) \end{bmatrix}}_{x(k+1)} &= \underbrace{\begin{bmatrix} A_p & 0 & 0 & \dots & 0 & 0 \\ C_p A_p & 0 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \Delta x_p(k) \\ y(k) \\ y(k-1) \\ \vdots \\ y(k-N+1) \\ y(k-N) \end{bmatrix}}_{x(k)} + \underbrace{\begin{bmatrix} B_p \\ C_p B_p \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_B \Delta u(k) \\
 y(k) &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \end{bmatrix}}_C \underbrace{\begin{bmatrix} \Delta x_p(k) \\ y(k) \\ y(k-1) \\ \vdots \\ y(k-N+1) \\ y(k-N) \end{bmatrix}}_{x(k)}
 \end{aligned} \tag{4.7}$$

Con este nuevo modelo, es posible hacer las predicciones de la planta para el diseño del MPC y con la ventaja de tener embebido en su estructura la acción repetitiva. Así, el sistema tendrá la capacidad de rechazar (o seguir) señales con período  $T_p$  con un controlador que tiene las características del MPC y del repetitivo juntas.

### 4.3.1: Cálculo de las Predicciones

Asumiendo que la trayectoria de control futura y la variable de salida futura del modelo son definidas por los vectores:

$$\begin{aligned}
 \Delta U &= [\Delta u(k) \ \Delta u(k+1) \ \Delta u(k+2) \ \dots \ \Delta u(k+M-1)]^T \\
 Y &= [y(k+1|k) \ y(k+2|k) \ y(k+3|k) \ \dots \ y(k+P|k)]^T
 \end{aligned}$$

Donde  $M$  representa el horizonte de control, que indica la cantidad de parámetros para capturar en la trayectoria de control futura.  $P$  es el horizonte de predicción (más grande o igual a  $M$ ) que a partir de una muestra de la salida de la planta  $y(k)$ , las salidas futuras son calculadas  $P$  números de muestras. El valor de  $P$  es también el tamaño de la ventana de optimización.

Entonces, para un modelo en espacio de estados en tiempo discreto como en

la ecuación 4.7, las salidas futuras son calculadas de forma recursiva utilizando los datos del vector de control [15]:

$$\begin{aligned}
y(k+1|k) &= CAx(k) + CB\Delta u(k|k) \\
y(k+2|k) &= CAx(k+1|k) + CB\Delta u(k+1|k) \\
&= CA^2x(k) + CAB\Delta u(k|k) + CB\Delta u(k+1|k) \\
y(k+3|k) &= CAx(k+2|k) + CB\Delta u(k+2|k) \\
&= CA^3x(k) + CA^2B\Delta u(k|k) + CAB\Delta u(k+1|k) + CB\Delta u(k+2|k) \\
&\vdots \\
y(k+P|k) &= CA^Px(k) + CA^{P-1}B\Delta u(k|k) + CA^{P-2}B\Delta u(k+1|k) \\
&+ \dots + CA^{P-M}B\Delta u(k+P-1|k).
\end{aligned} \tag{4.8}$$

Utilizando los vectores 4.3.1 y las ecuaciones de predicción 4.8 es posible obtener la siguiente representación compacta en forma de matrices:

$$Y = G\Delta U + fx(k) \tag{4.9}$$

donde:

$$f = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^P \end{bmatrix}; G = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{P-1}B & CA^{P-2}B & CA^{P-3}B & \dots & CA^{P-M}B \end{bmatrix}$$

## 4.4: Respuesta Libre y Forzada

Normalmente en los algoritmos MPC la señal de control futura es calculada teniendo dos componentes:

$$u(k) = u_l(k) + u_f(k) \tag{4.10}$$

Donde,  $u_l(k)$  es considerando que la última señal de control es mantenida en los instantes de tiempo futuros y  $u_f(k)$  es la próxima señal de control calculada [11, 10].

De esta forma, las predicciones futuras también son divididas en:  $y_l(k)$ , cuando la entrada es igual a  $u_l(k)$  y  $y_f(k)$ , cuando la entrada es  $u_f(k)$ . Esto es posible observar analizando la ecuación 4.9, donde el término  $G\Delta U$  representa la evolución del proceso debido las acciones de control futuras con condiciones iniciales nulas, mientras el término  $fx(k)$  es debido su estado actual.

## 4.5: Función Objetivo

El objetivo de control  $J(k)$  (o función de coste) es un criterio escalar que es formulado de formas diferentes para cada algoritmo MPC, pero normalmente tienen el propósito de que la salida futura de la planta  $y(k)$  siga, dentro del horizonte de predicción definido, una determinada señal  $r(k)$  y considerando que el esfuerzo de control  $\Delta u(k)$  necesario para hacerlo debe ser penalizado.

Así, la función costo tiene la siguiente representación:

$$J(k) = \sum_{i=1}^P (y(k+i|k) - r(k+i|k))^T Q (y(k+i|k) - r(k+i|k)) + \sum_{i=0}^{M-1} \Delta u^T(k+i|k) R \Delta u(k+i|k) \quad (4.11)$$

Donde  $Q, R$  son matrices de ponderación simétricas y positivas definidas que son especificadas. De forma general, las matrices son escogidas como  $Q = qI_P, R = rI_M$  y apenas los parámetros  $q, r$  son definidos o también pueden ser compuestas de secuencias exponenciales.

La función 4.11 también puede ser representada en forma de matrices:

$$J = (Y - r)^T Q (Y - r) + \Delta U^T R \Delta U \quad (4.12)$$

Donde  $Q, R$  son matrices diagonales en bloques. Esta forma compacta simplifica el análisis.

Los valores escogidos para  $P, M, Q, R$  tienen una grand influencia en la dinámica del sistema [13, 14]:

- Se  $P \uparrow$ , entonces el sistema de control irá enfocar más en el régimen permanente y menos en el transitorio. Se  $P \downarrow$ , el sistema enfocará mas en el transitorio pero sin información suficiente de como el proceso está desarrollando y para donde irá parar en régimen permanente.

El valor de  $P$  también determina el tamaño de la matriz de predicción  $G$ .

- Se  $M \uparrow$ , el sistema penaliza más el futuro con una acción de control más suave dejando el problema para ser resuelto en el futuro, escogido para sistemas complejos donde la solución optima depende de más cambios en la acción de control. Con  $M \downarrow$  el sistema tendrá que calcular una solución inmediata.

El valor de  $M$  determina el tamaño de la matriz  $H$ , para el proceso de optimización.

- Para  $Q \uparrow$  el error será más penalizado y la acción de control más suave. Y con  $Q \downarrow$  el error no será tan penalizado con una acción de control más brusca.
- Con  $R \uparrow$  la optimización tendrá soluciones con valores pequeños, con una acción de control más suave. Se  $R \downarrow$  la acción de control tiene valores más grandes y un esfuerzo más grande.

## 4.6: Obtención de la Ley de Control

Con el objetivo de obtener los valores del vector de controles  $U$  es necesario minimizar la función  $J$ . Para hacerlo, los valores de la salida futura son calculados utilizando: los valores pasados de las variables de entrada y salida, las señales de control futuras y el modelo de predicción escogido, como explicado en [11].

Con esto, es posible sustituir dichos parámetros en la función de coste sacando una expresión que después de minimizada lleva a los valores deseados. Es decir, el problema de control es:

$$\Delta u^*(k) = \arg \min_{\Delta u(k)} J(k) \quad (4.13)$$

Sujeto al modelo de predicción y a restricciones en las variables de proceso cuando especificadas. Cuando el criterio de la función de coste es cuadrática, el modelo de predicción es lineal y no son consideradas las restricciones, es posible obtener una solución analítica del problema de optimización. De lo contrario, un método de optimización iterativo debe ser utilizado como por ejemplo, métodos de programación cuadrática.

Así, para obtener la solución básica sin restricciones del MPC es necesario insertar el modelo de predicción en espacio de estados con la forma de la ecuación

4.9 en la función objetivo 4.12

$$\begin{aligned}
J &= (G\Delta U + fx(k) - r)^T Q(G\Delta U + fx(k) - r) + \Delta U^T R \Delta U, \\
&= (\Delta U^T G^T + x^T(k) f^T - r^T) Q(G\Delta U + fx(k) - r) + \Delta U^T R \Delta U, \\
&= \underbrace{\Delta U^T G^T Q G \Delta U}_1 + \underbrace{\Delta U^T G^T Q fx(k) - \Delta U^T G^T Q r + x^T(k) f^T Q G \Delta U + x^T(k) f^T Q fx(k)}_2 \\
&\dots - \underbrace{x^T(k) f^T Q r}_3 - \underbrace{r^T Q G \Delta U}_2 - \underbrace{r^T Q fx(k) + r^T Q r}_3 + \underbrace{\Delta U^T R \Delta U}_1, \\
&= \Delta U^T \underbrace{[G^T Q G + R]}_1 \Delta U + 2 \underbrace{[G^T Q (fx(k) - r)]^T}_2 \Delta U + \underbrace{(fx(k) - r)^T Q (fx(k) - r)}_3,
\end{aligned} \tag{4.14}$$

que tiene la forma cuadrática:

$$J = \Delta U^T H \Delta U + 2F^T \Delta U + J_0 \tag{4.15}$$

donde:

$$\begin{aligned}
H &= G^T Q G + R, \\
F &= G^T Q (fx(k) - r), \\
J_0 &= (fx(k) - r)^T Q (fx(k) - r).
\end{aligned} \tag{4.16}$$

Derivando la función  $J$  en relación a  $\Delta U$  se llega a la ecuación

$$\frac{\partial J}{\partial \Delta U} = 2H \Delta U + 2F \tag{4.17}$$

Como el criterio tiene forma cuadrática, entonces el punto de mínimo  $\Delta u^*$  de la curva  $J$  se encuentra donde la derivada tiene valor igual a 0, es decir,  $\partial J / \partial \Delta U = 0$  con la solución

$$u^* = -H^{-1} F = (G^T Q G + R)^{-1} G^T Q (r - fx(k)) \tag{4.18}$$

Aquí,  $H$  es la matriz Hessiana (teoría de control óptimo) obtenida de la segunda derivada de la función costo en relación a  $\Delta U$ . Esta matriz tiene relación con las matrices  $Q, R$  para que sea positiva definida y garantizar una solución mínima única.

En la práctica, no se calcula la inversa de la matriz  $H$  porque exige tanto más tiempo para cálculo como de más poder de procesamiento. Así, lo que se hace es resolver como un conjunto de ecuaciones simultáneas <sup>2</sup>.

<sup>2</sup>En *matlab* se puede utilizar el operador \ que calcula el conjunto de ecuaciones simultáneas.

## 4.7: Modelo Sin Restricciones

Como dicho al principio del capítulo, el modelo para diseño del algoritmo MPC es el modelo en espacio de estados del proceso, con el lazo repetitivo embebido. Con esto algunos ajustes son necesarios para que el controlador tenga una buena respuesta.

### 4.7.1: Ajustes

Para que sea posible hacer una comparación entre las diferentes estructuras, será escogido el mismo modelo para el proceso, la planta 2.6 y con las mismas condiciones.

- $G_p(s) = \frac{K_p e^{-T_d s}}{\tau s + 1}$ ,  $K_p = 0,3$ ,  $T_d = 0,02$ ,  $\tau = 0,1$
- $T_p = 0,4s$ ,  $N = 20$ ,  $T_s = T_p/N = 0,02s$

Con estas variables fijadas, es posible ajustar los parámetros del controlador que son:

- tamaño de los horizontes de predicción y control  $P, M$ ;
- matrices de ponderación  $Q, R$ ;
- modelo en espacio de estados del proceso discretizado con período de muestreo  $T_s$  y con lazo repetitivo  $\frac{1}{(1-z^{-N})}$  embebido.

Un punto importante es que el valor del horizonte de predicción  $P$  sea mas grande que el valor de  $N$ , es decir, para que el controlador pueda actuar de forma a corregir el desenvolvimiento del sistema, como mínimo debe predecir el cambio de la dinámica dentro de un período  $T_p$  de la señal periódica. Así, la relación  $P > N$  debe estar presente para el ajuste de los parámetros del controlador.

### 4.7.2: Simulación

Para verificar el funcionamiento del controlador, este pasará por la misma simulación que fue aplicada en las otras estructuras, con las señales:

$$Ref = 1, Perturbacion = 0,5 \sin\left(\frac{2\pi}{T_p}\right) + 0,25 \sin\left(\frac{4\pi}{T_p}\right),$$

En la simulación, el controlador tiene las siguientes configuraciones:

$$P = 40 = 2N, \quad M = 10, \quad Q = 1I_P, \quad R = 10^{-2}I_M.$$

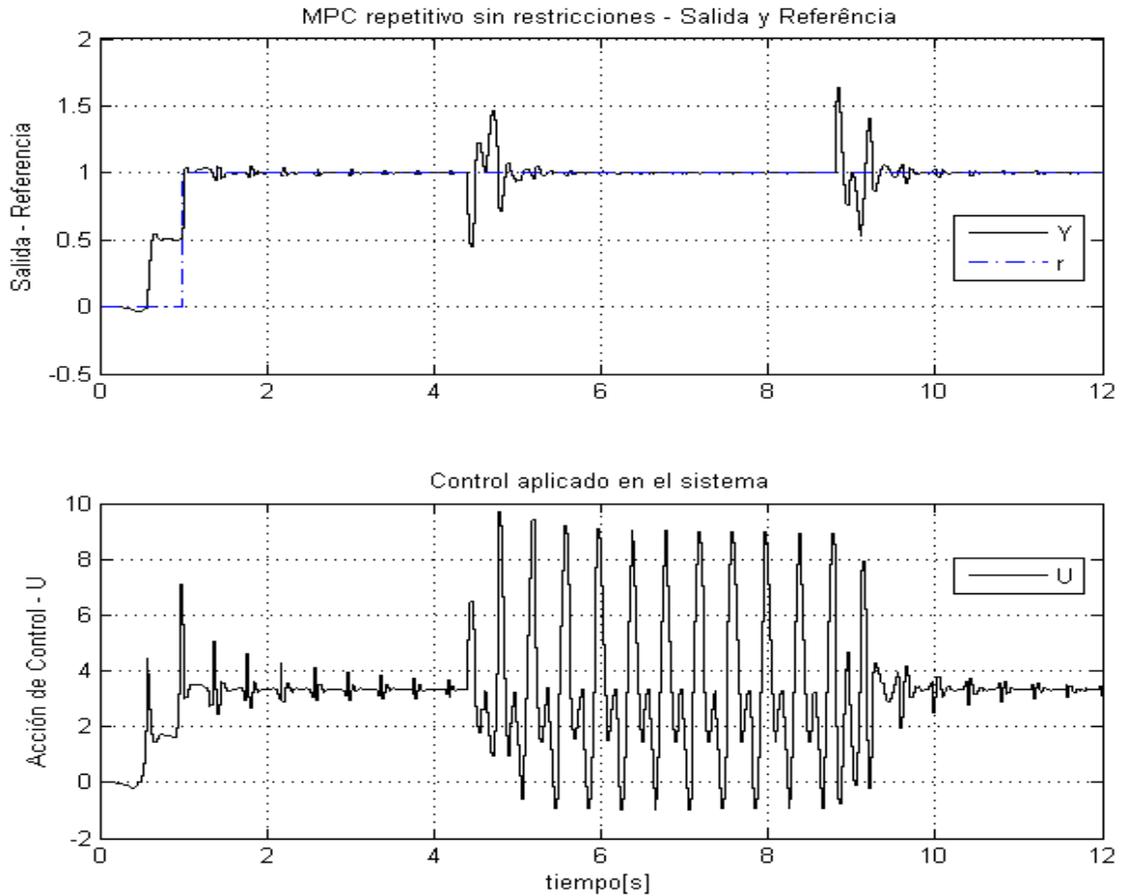


Figura 4.3: Respuesta del algoritmo MPC repetitivo sin restricciones aplicado en la planta  $G_p$ .

El desempeño del algoritmo de control MPC repetitivo sin restricciones con las configuraciones presentadas puede ser observado en la figura 4.3. Ahi, es posible identificar una de las principales características del MPC que es la capacidad de actuar de forma anticipada en el sistema ante los cambios en la referencia. Para la perturbación, el sistema consigue rechazarla completamente, ya que como tiene la acción repetitiva la señal de control puede moverse con el objetivo de mantener la salida siguiendo la referencia.

## 4.8: Conclusiones

Con el objetivo principal del proyecto, este capítulo muestra las principales características del control predictivo basado en modelo y como es posible hacer la inserción del lazo repetitivo en un algoritmo de control MPC de forma sencilla, con un modelo en espacio de estados que tiene la acción repetitiva embebida.

En algunas simulaciones, es posible observar que el sistema de control tiene la capacidad de rechazar perturbaciones periódicas.

Ahora, es necesario verificar se el sistema desarrollado tiene la capacidad de tratar restricciones en las variables.

## 5 Tratamiento de Restricciones

En general, los procesos están sujetos a restricciones de varios tipos. Los actuadores tienen los límites de acción y una determinada velocidad de cambio, como por ejemplo válvulas, bombas o actuadores hidráulicos.

De esta forma, varias características de construcción, de seguridad o medioambientales pueden traer restricciones en las variables de proceso como temperaturas y presiones máximas. Otra situación, es que los puntos de operación son definidas por la intersección de algunas restricciones del sistema con restricciones económicas, haciendo el sistema de control operar cerca de los límites. Así, todo esto puede ser insertado en la función de coste y considerado en el proceso de optimización [10, 11, 13].

Los tipos mas comunes de restricciones son:

- Restricción en la amplitud de la entrada (saturación);
- Restricción en la variación de la entrada (*slew-rate* máximo);
- Restricción en la salida (max/min).

Existen también otros tipos de restricciones como [12]:

- Restricciones de la salida en banda, donde la salida se debe mantener dentro de unas bandas;
- Restricciones de sobreoscilación, para que las salida no supere el valor de la referencia en un cierto intervalo ed tiempo;
- Restricciones de monotonidad, para evitar el "kick-back."º vuelta atrás;
- Restricciones de fase no mínima, para limitar los picos de fase no mínima.

Estas restricciones pueden ser escritas como una desigualdad lineal:

$$Ax \leq b. \quad (5.1)$$

y así, pueden ser insertadas en el problema de optimización.

Sin embargo, algunas de estas restricciones consideran que la variable de control esta diferenciada en un periodo de muestreo

$$\Delta u(k) = u(k) - u(k - 1),$$

así, que al hacer el modelo con la acción repetitiva embebida esta variable toma una nueva forma

$$\Delta u(k) = u(k) - u(k - N),$$

y las restricciones ahora son interpretadas de otra manera. La restricción en la variación de la entrada, por ejemplo, se torna la restricción en la variación de la entrada al largo de un periodo de la perturbación.

Las diferencias entre ellas y sus relaciones con el controlador serán explicadas en este capítulo.

## 5.1: Restricción en la Variación de la Entrada

### 5.1.1: Variación de un Periodo $T_p$ ( $NT_s$ )

Esta restricción actúa en cuanto la señal de entrada varia de un periodo  $T_p$  al próximo. Toman la forma:

$$\Delta u_{min} \leq \Delta u(k) \leq \Delta u_{max} \quad (5.2)$$

con  $\Delta u(k) = u(k) - u(k - N)$ . Esta desigualdad puede ser expresada en forma matricial:

$$\begin{bmatrix} I \\ -I \end{bmatrix} \Delta U \leq \begin{bmatrix} \Delta U_{max} \\ \Delta U_{min} \end{bmatrix} \quad (5.3)$$

## 5.2: Restricción en la Amplitud de la entrada

En cualquier problema de control práctico la señal de control está limitada por un valor mínimo y un valor máximo, sea por limitaciones físicas o por cuestiones de

seguridad. Estas restricción es especificada como:

$$u_{min} \leq u(k) \leq u_{max} \quad (5.4)$$

En la forma clásica con integrador embebido la variable de diferenciación  $\Delta u(k)$  es relacionada con  $u(k)$  y  $u(k-1)$  ( $\Delta u(k) = u(k) - u(k-1)$ ), utilizando la relación:

$$\begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+M-1) \end{bmatrix} = \underbrace{\begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \end{bmatrix}}_c u(k-1) + \underbrace{\begin{bmatrix} I & 0 & 0 & \dots & 0 \\ I & I & 0 & \dots & 0 \\ I & I & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & I & I & \dots & I \end{bmatrix}}_s \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+M-1) \end{bmatrix} \quad (5.5)$$

cuya forma matricial compacta es

$$U = cu(k-1) + S\Delta U \quad (5.6)$$

Con esto, es posible hacer una relación entre  $U(k)$  y  $\Delta U(k)$  y insertarla en el problema de optimización.

Sin embargo, con la nueva forma que tiene la acción repetitiva embebida, la variable de diferenciación  $\Delta u(k)$  depende de los valores de  $u(k)$  y  $u(k-N)$  haciendo con que la relación sea planteada de forma mas sencilla con la forma:

$$\begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+M-1) \end{bmatrix} = \begin{bmatrix} u(k-N) \\ u(k-N+1) \\ u(k-N+2) \\ \vdots \\ u(k-N+M-1) \end{bmatrix} + \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+M-1) \end{bmatrix} \quad (5.7)$$

que en matrices, es:

$$U = U(k-N) + \Delta U \quad (5.8)$$

De esta forma es posible representar la desigualdad:

$$\begin{bmatrix} I \\ -I \end{bmatrix} \Delta U \leq \begin{bmatrix} U_{max} - U(k-N) \\ -U_{min} + U(k-N) \end{bmatrix} \quad (5.9)$$

Donde  $U(k-N)$  es el vector con las  $M$  primeras acciones de control calculadas del último periodo  $T_p$ .

### 5.3: Restricción en la salida

Con la forma:

$$y_{min} \leq y(k) \leq y_{max} \quad (5.10)$$

Utilizando el modelo de predicción 4.9 es escrita como:

$$Y_{min} \leq G\Delta U + fx(k) \leq Y_{max} \quad (5.11)$$

que, en forma matricial es:

$$\begin{bmatrix} G \\ -G \end{bmatrix} \Delta U \leq \begin{bmatrix} Y_{max} - fx(k) \\ -Y_{min} + fx(k) \end{bmatrix} \quad (5.12)$$

### 5.4: Insertando las Restricciones en el Problema de Optimización

Cogiendo los tres tipos de restricciones representadas en 5.3, 5.9 y 5.12 es posible combinarlas como una desigualdad lineal do tipo 5.1

$$\underbrace{\begin{bmatrix} I \\ -I \\ I \\ -I \\ G \\ -G \end{bmatrix}}_A \Delta U \leq \underbrace{\begin{bmatrix} \Delta U_{max} \\ -\Delta U_{min} \\ U_{max} - U(k - N) \\ -U_{min} + U(k - N) \\ Y_{max} - fx(k) \\ -Y_{min} + fx(k) \end{bmatrix}}_b \quad (5.13)$$

Con esto, el problema de control es encontrar el vector de acciones de control futuras  $U$  que minimiza la función costo 4.15 sujeta a las desigualdades 5.13.

Como la función costo es cuadrática, y las restricciones son desigualdades lineales, el problema es encontrar una solución óptima de un problema de programación cuadrática padrón.

$$\min_U (U^T H U + 2F^T U + J_0) \quad (5.14)$$

$$\text{Sujeto a: } AU \leq b \quad (5.15)$$

que en *MATLAB* es calculado por la función *quadprog*

$$U = \text{quadprog}(H, F, A, b) \quad (5.16)$$

Para esto, el vector con acciones de control anteriores debe ser conocido y a cada instante de tiempo  $k$  el vector  $F$  es calculado en términos de las variables de proceso.

## 5.5: Modelo Con Restricciones

Este modelo tiene las mismas características que el modelo sin restricciones, difiriendo apenas en la forma de hacer los cálculos de la acción de control que ahora necesitan un algoritmo de solución de problemas cuadráticos. Es decir, que cuando las restricciones tienen valores elevados, el MPC con restricciones tiene el mismo comportamiento que el MPC sin restricciones.

### 5.5.1: Ajustes

Para los ajustes, se escoge el modelo de la planta  $G_p$  2.6, con los mismos valores de  $T_p$ ,  $N$  y  $T_s$ . Así, el modelo con restricciones del MPC repetitivo necesita de los mismos parámetros del modelo sin restricciones:  $P$ ,  $M$ ,  $Q$ ,  $R$  y el modelo en espacio de estados del proceso con lazo repetitivo embebido.

Pero, el modelo con restricciones evidentemente necesita también de las matrices de restricciones combinadas en forma de desigualdades lineales 5.13 en conjunto con los valores máximos y mínimos de estas restricciones.

### 5.5.2: Simulación

En esta simulación, el controlador será sujeto a las mismas condiciones (señales de referencia y perturbación) que las otras estructuras y para verificar la influencia de las restricciones, el controlador tendrá las mismas configuraciones del modelo sin restricciones:

$$P = 40 = 2N, \quad M = 10, \quad Q = 1I_P, \quad R = 10^{-2}I_M.$$

con diferentes tipos de restricciones.

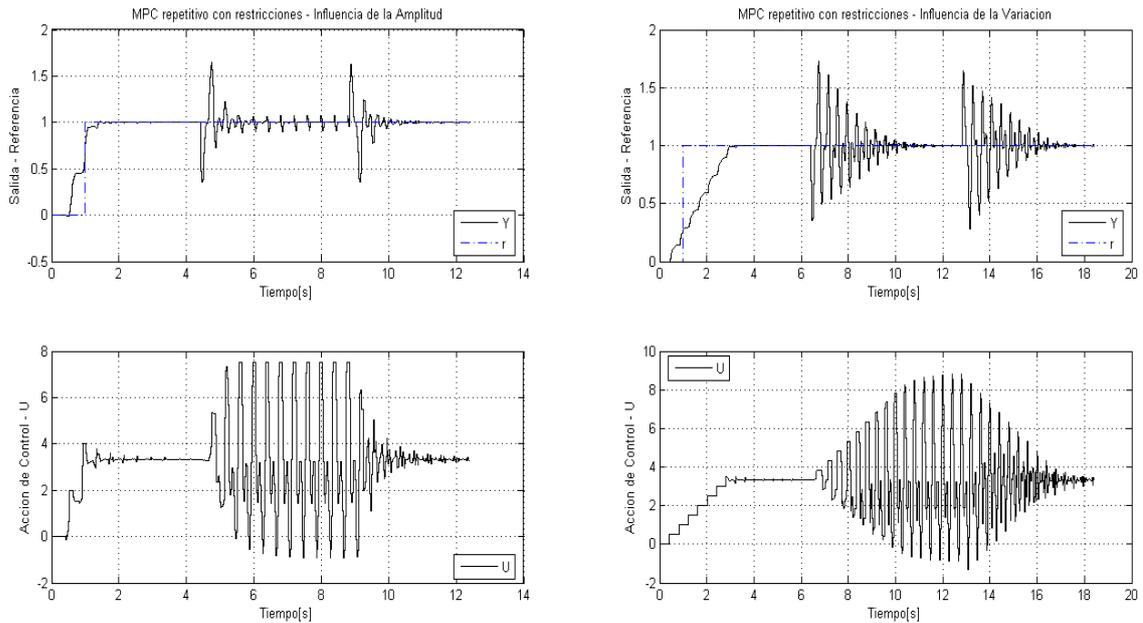


Figura 5.1: Respuesta del algoritmo MPC Repetitivo con restricciones aplicado en la planta  $G_p$ . Influencia de la restricción en  $U$  (izquierda) y de la restricción en  $\Delta U$  (derecha).

Con esto, la respuesta del sistema cambia para las diferentes restricciones. En la figura 5.1 el sistema tiene las siguientes restricciones:

- Izquierda:  $U_{max} = -U_{min} = 7,5$  ,  $\Delta U_{max} = -\Delta U_{min} = 0,5$
- Derecha:  $U_{max} = -U_{min} = 10$  ,  $\Delta U_{max} = -\Delta U_{min} = 2$

Así, que en la figura de la derecha el sistema tiene una respuesta lenta tanto para seguir la referencia como para rechazar la perturbación. Pero al final el objetivo en llegar al error nulo es cumplido. Es posible observar que en ningún momento la variación de la señal de control de un periodo  $T_p$  al otro, es mas grande que el valor de la restricción ( $\Delta U_{min} \leq \Delta U \leq \Delta U_{max}$ ) mostrando que el optimizador está haciendo su trabajo bien.

En la figura de la izquierda, el sistema tiene una respuesta mas rápida pero, para el rechazo de la perturbación no llega al error nulo. Esto porque, como existe la restricción en la amplitud de la señal de control en 7,5 no es suficiente para que el sistema pueda actuar de forma correcta. Entonces, para las restricciones en la amplitud de la señal de control, es necesario verificar cual es la amplitud mínima de la acción de control para actuar en una perturbación periódica ya que para sistemas lineales, dada una entrada con frecuencia y amplitud fija, la salida también será una señal periódica de misma frecuencia pero con diferente amplitud y desfase.

## 5.6: Conclusiones

Este capítulo demuestra los tipos de restricciones más comunes existentes en los sistemas de control y los ajustes necesarios para hacer con que el MPC repetitivo pueda tratar restricciones. Fueran consideradas en las simulaciones apenas restricciones en la amplitud de la entrada ( $u(k)$ ) y en la variación de la entrada a cada periodo  $T_p$  ( $\Delta u(k)$ ).

De esta forma el MPC repetitivo tiene un buen desempeño, sin superar las restricciones y consiguiendo rechazar la perturbación periódica.

Con este planteamiento es posible hacer pruebas experimentales de los algoritmos en una planta real para validación de los estudios.

# 6 Aplicación Práctica

En este capítulo, serán descritos los procedimientos y ajustes necesarios para que sea posible hacer la implementación práctica de los algoritmos desarrollados, con el objetivo de controlar una planta real didáctica por medio de un ambiente que posibilita el prototipado rápido de los sistemas de control. Así, será posible hacer una comparación entre las diferentes estructuras aplicadas a un mismo proceso.

## 6.1: Proceso Real

El proceso a ser controlado es un equipo desarrollado para el estudio de sistemas que tienen perturbaciones (o referencias) periódicas. Se trata de un motor DC con un aparato magnético para reproducir las perturbaciones, aplicadas como torque de carga pulsante.

Este aparato, es compuesto de una barra fija al eje del motor con imanes permanentes en su punta, haciendo como si toda la barra fuera un imán. También están fijados dos electroimanes que interactúan con la barra. En la figura 6.1 está el equipo y un esquema de la actuación entre los imanes para generación de la perturbación.

De esta forma, cuando el eje del motor gira a una velocidad constante, la perturbación producida tiene periodo fundamental relacionado con el inverso de la velocidad ( $T_p = \omega^{-1}$ ). Con esto, el objetivo de control es hacer con que el motor tenga una velocidad angular constante y deseada.

## 6.2: xPC Target

El *xPC Target* en conjunto con *Real-Time Windows Target* y otros productos, posibilitan la creación de un ambiente de testes en tiempo real para la creación rápida de prototipos funcionales y simulaciones *hardware in the loop* (HIL) [18].

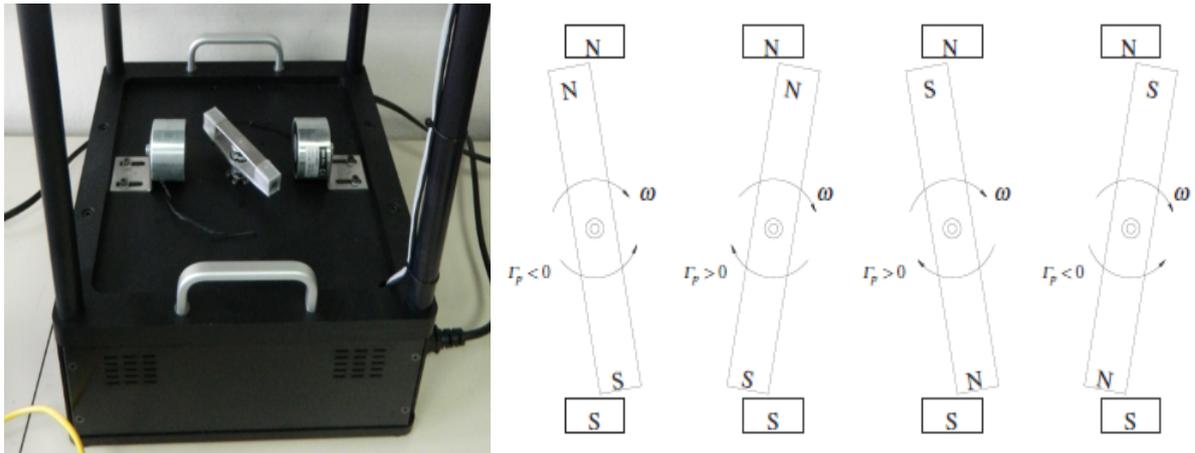


Figura 6.1: Estructura del roto-magnet con motor DC, imanes y *encoder* (izquierda) y esquema de actuación del aparato, donde  $\omega$  es la velocidad angular y  $\Gamma_p$  es la perturbación de torque (derecha).

- **Creación Rápida de Prototipos Funcionales**

En este escenario, el objetivo principal es probar la funcionalidad de los algoritmos de control en un ambiente que sea lo más próximo del ambiente real, donde están presentes los actuadores y sensores reales, con la posibilidad de ajustar y optimizar parámetros en una simulación en tiempo real.

Con esto es posible hacer diferentes pruebas y correcciones en el laboratorio antes de hacerlas en el campo.

- **Simulaciones *Hardware in the Loop* (HIL)**

Es similar a la creación rápida de prototipos pero, en lugar de simular el sistema de control en tiempo real, el modelo de la planta es simulado en tiempo real. Así, un controlador real es conectado a un sistema de simulación de forma que el controlador piense estar controlando la planta real.

Así, es posible verificar la funcionalidad del controlador antes de insertarlo en el sistema real y hacer pruebas sin la necesidad de prototipos caros.

De esta forma, como el objetivo de estudio es la implementación de los algoritmos de control, el escenario será ajustado para la creación rápida de prototipos funcionales.

### 6.2.1: Funcionamiento

El *xPC target* tiene un ordenador que es el *Host* y otro que es el *target*. Para el funcionamiento, en el *host* son desarrollados los modelos *SIMULINK* del sistema

de control y ahí el modelo de la planta es cambiado por los *drivers* de las tarjetas de conversión de datos que son enviados a los actuadores y recibidos de los sensores.

Con el modelo listo, es necesario hacer la construcción del modelo (*Build Model*), donde el *Real-Time Workshop* hace la conversión del modelo en *SIMULINK* para código *C* y con la utilización de un compilador *C/C++*, el código es compilado en un archivo ejecutable en tiempo real. Este archivo es descargado via redde TCP/IP en el *target* que ya está arrancado con el núcleo en tiempo real [19].

Con esto es posible comunicarse con el *target* para arrancar o parar aplicación, cambiar parámetros, guardar datos y muchas otras utilidades.

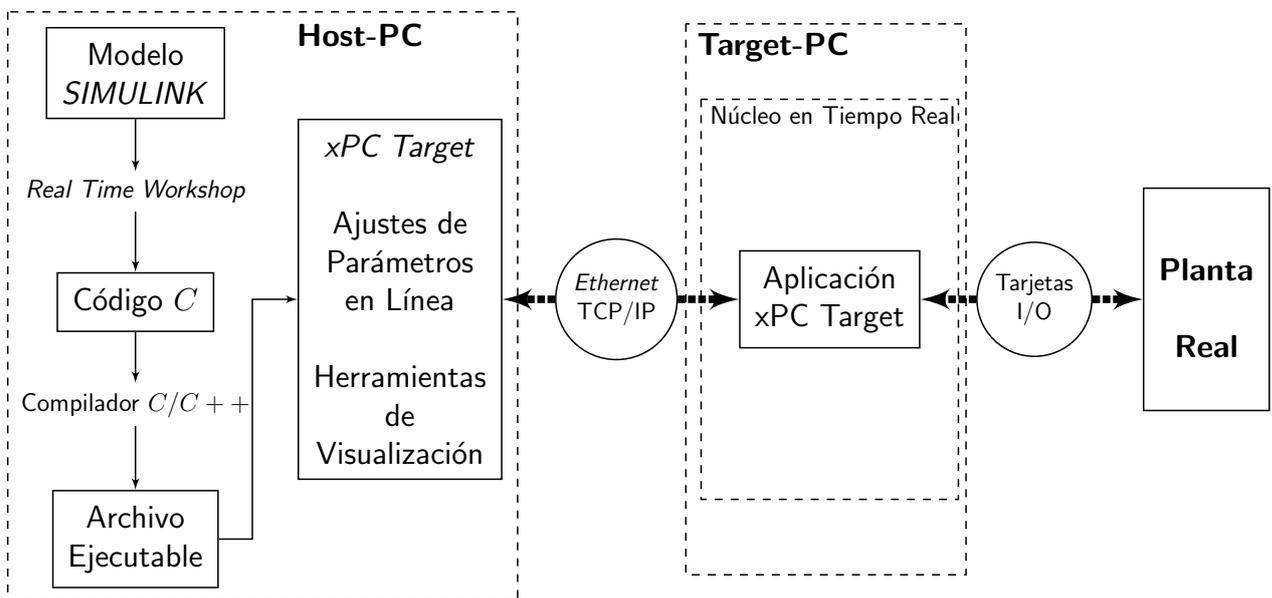


Figura 6.2: Esquema de funcionamiento del *xPC Target*

De esta forma, el *xPC Target* hace con que el camino entre el proceso de simulación hasta la implementación sea acortado. Esto porque para hacer un cambio en el modelo en *SIMULINK* es mas sencillo que hacer el mismo cambio en el código *C* del sistema.

### 6.3: Sistema en Lazo Abierto

Para verificar como el aparato generador de perturbaciones funciona y la dinámica del sistema, fue realizada una simulación en tiempo real del sistema en lazo abierto.

En la figura 6.3 se encuentra el resultado de la simulación con una entrada de

0,35V de tensión el sistema llega a aproximadamente 6rev/s, siendo visible la influencia de la perturbación en la salida del sistema de forma periódica.

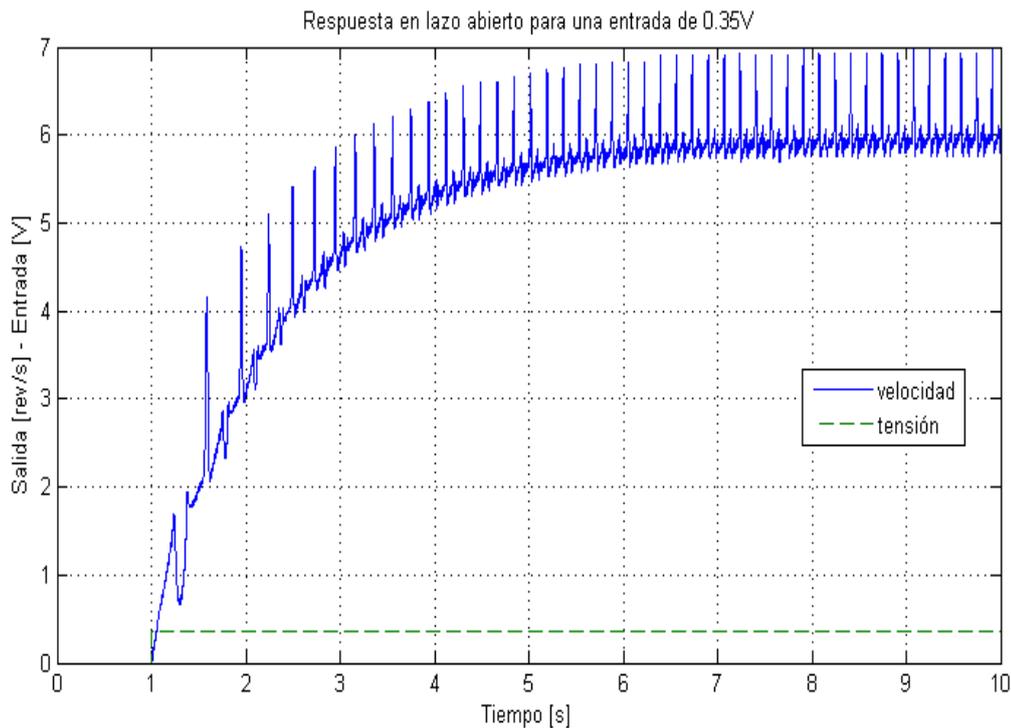


Figura 6.3: Simulación en Tiempo Real del Proceso en Lazo Abierto, con entrada de 0,35V

## 6.4: Ajustes de los Algoritmos

Para el ajuste de los algoritmos estudiados y posibilitar una comparación entre ellos, es necesario tener un mismo ambiente de prueba. De esta forma, la definición de algunos parámetros comunes para todos los algoritmos, como el periodo de la perturbación  $T_p$ , el numero de muestras a almacenar  $N$ , el periodo de muestreo  $T_s$ , el filtro FIR  $H(z)$  y el modelo de la planta  $G_p(s)$  será basada en el estudio hecho en [20].

- **Modelo de la Planta -  $G(s)$**

A causa del sistema en lazo abierto tener una ganancia muy alta, la identificación del modelo es hecha en lazo cerrado con un controlador de ganancia proporcional ( $K_{id} = 1$ ). Con esto, el modelo en lazo abierto del sistema identificado sin los electroimanes es:

$$G(s) = \frac{K}{\tau s + 1} = \frac{16,152}{0,457\tau s + 1} \frac{rev/s}{V} \tag{6.1}$$

- **Periodo Fundamental de la Perturbación -  $T_p$**

Como el periodo de la perturbación está directamente relacionado con la velocidad del sistema que es el objetivo de control, entonces se escoge el punto de operación objetivo como:  $\omega_{ref} = 4 \text{ rev/s}$  y con esto la perturbación periódica es  $T_p = \omega_{ref}^{-1} = 0,25 \text{ s}$ .

- **Numero de Muestras a Almacenar -  $N$**

Con  $N = 250$  suficientes muestras por periodo  $T_p$  son almacenadas para que la señal en forma de muestras sea lo mas próxima de la señal en forma continua.

- **Periodo de Muestreo -  $T_s$**

El periodo de muestreo es calculado con:  $T_s = T_p/N = 1/(\omega_{ref}N) = 0,001 \text{ s}$

- **Filtro FIR -  $H(z)$**

Con  $H(z) = 0,25z + 0,5 + 0,25z^{-1}$  tiene un buen desempeño para este caso.

- **Controlador Nominal -  $G_c(z)$**

Como un controlador nominal para el ajuste de un lazo interno de la estructura *plug-in* fue escogido un PI, con

$$G_c(z) = \frac{1,8z - 1,796}{z - 1} \quad (6.2)$$

el lazo interno tiene una buena robustez.

Todas estas variables definidas sirven como base para la configuración y diseño de todas las estructuras de control en estudio para que sean implementadas y validadas en la planta. En algunas estructuras, necesitan además del modelo en espacio de estados del lazo repetitivo y de la planta.

## 6.5: Simulación - PI

Esta simulación objetiva enseñar la necesidad del modelo interno para el rechazo de perturbaciones periódicas. Para esto el sistema fue ajustado teniendo solamente el controlador nominal  $G_c(z)$ , que tiene la forma de un PI clásico. Es posible observar en la figura 6.4 que el sistema tiene una buena respuesta y llega al punto de operación de forma rápida. Sin embargo no consigue rechazar la perturbación presente en la salida, lo que es esperado ya que este tipo de controlador no tiene el modelo interno generador de la señal.

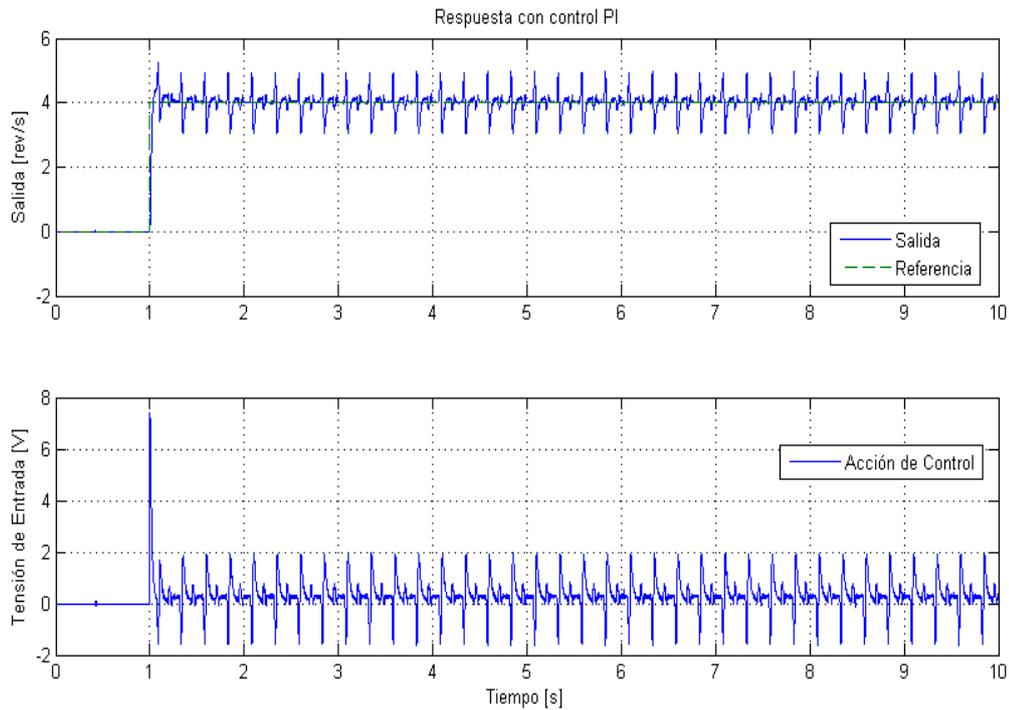


Figura 6.4: Simulación en Tiempo Real con controlador PI.

## 6.6: Simulaciones Experimentales

Para las simulaciones con *xPC Target* los equipos fueran colocados con la configuración en la figura 6.5

Por consiguiente, serán explicadas las simulaciones experimentales hechas en la planta con las diferentes estructuras estudiadas y los cambios y ajustes necesarios para el funcionamiento de los algoritmos.

Para las implementaciones fue necesario añadir el filtro FIR  $H(z)$  que no fue considerado en las simulaciones teóricas.

Otro punto a ser destacado, es que en la implementación en *SIMULINK* de la planta con los *drivers* de entrada y salida existe una saturación en la señal de entrada por cuestiones de seguridad del equipo. Así, la señal de control está limitada entre  $-10 V$  y  $10 V$  y en las simulaciones el valor de la señal de control mostrada es la señal calculada sin la saturación.

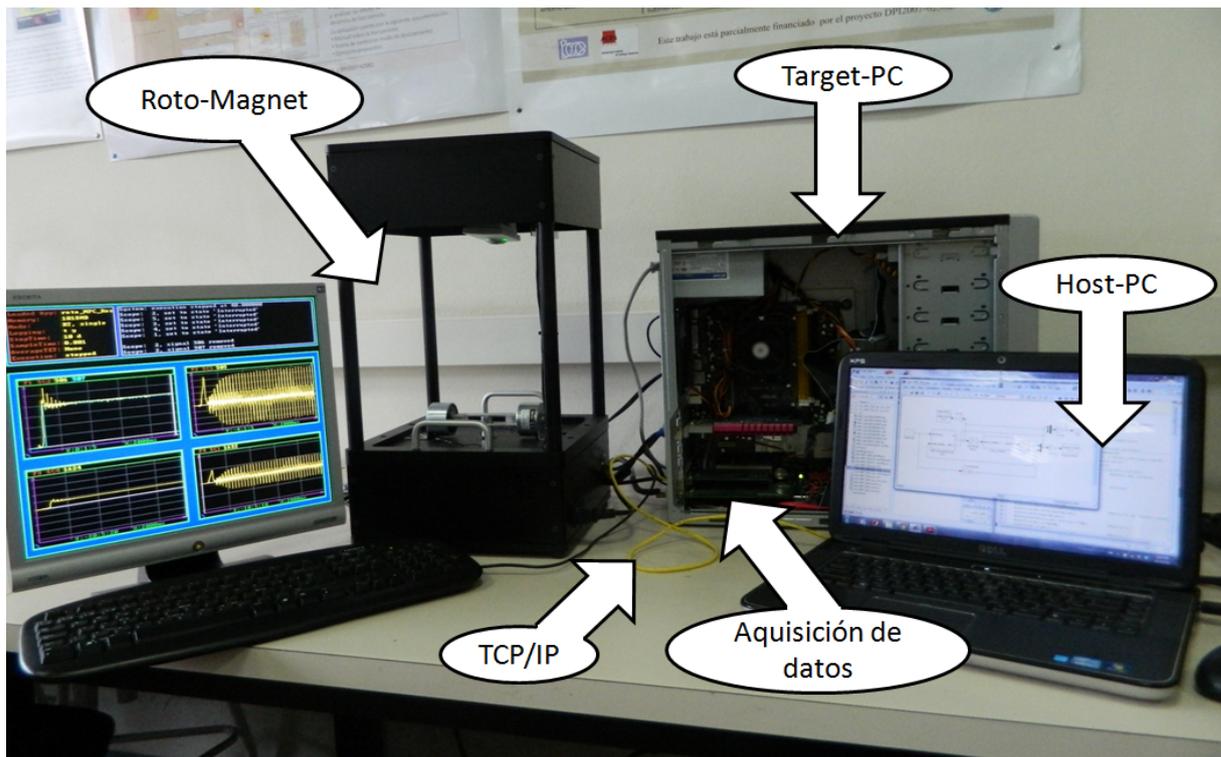


Figura 6.5: Configuración de los equipos en el lab.

### 6.6.1: Plug-in

En esta estructura el controlador repetitivo fue ajustado con  $K_r = 0,7$ .

En el resultado de la simulación, figura 6.6, la salida del sistema tiene una sobre señal que llega a las  $10 \text{ rev/s}$ , esto debido a un pico en la señal de control calculada. El sistema consigue rechazar la perturbación periódica en un tiempo de aproximadamente 3 segundos.

### 6.6.2: Realimentación de Estados Observados

Para esta estructura son necesarios los modelos en espacio de estados del lazo repetitivo y de la planta. En el ajuste fueran seleccionados:

- Diseño del Observador

Con  $Q_w = 0,9$ ,  $R_v = 0,01$  y el modelo en espacio de estados de la planta es posible hacer el diseño de la ganancia del filtro.

- Diseño de la Realimentación de Estados

La posición deseada de los polos en lazo cerrado es ajustada utilizando la nota-

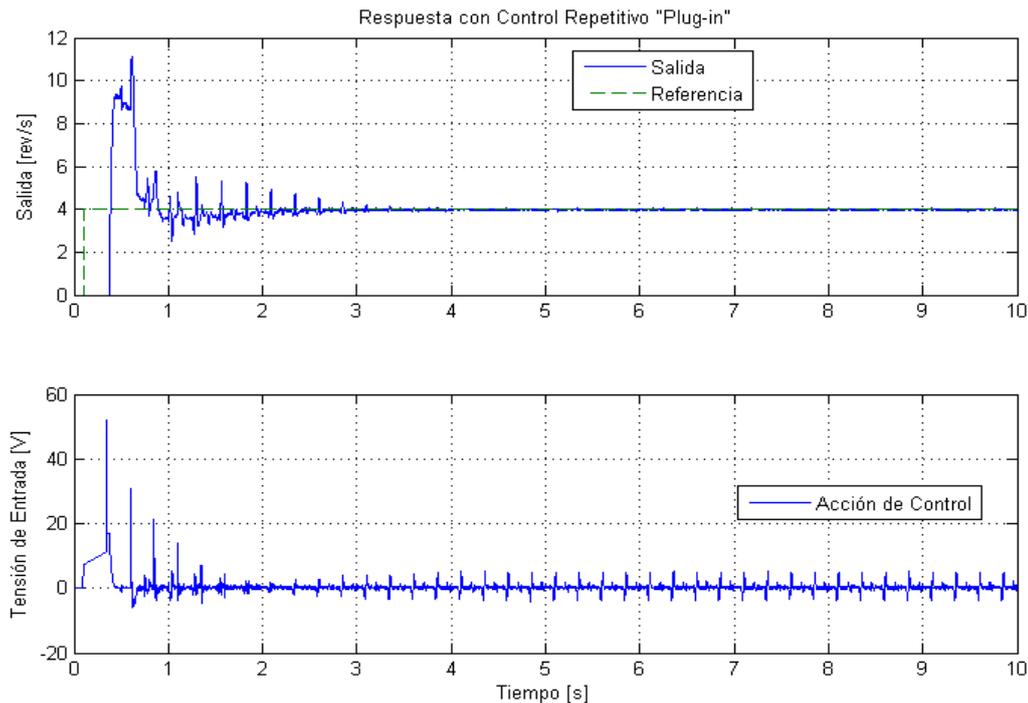


Figura 6.6: Simulación en Tiempo Real con Estructura *Plug-in*

ción simplificada  $P = [v_{plc} \ \alpha \cdot v_{pAr}]$ , con  $v_{plc} = 0,85$  y  $\alpha = 0,98$ .

La respuesta del sistema está en la figura 6.7 y tiene características mejores, sin la presencia de sobre señal con algunos picos debido la perturbación, pero tiene también un pico en la señal de control. La perturbación es rechazada después de 1,5 s.

### 6.6.3: MPC Repetitivo

Para el ajuste:

$$P = 2N, \quad M = 10, \quad Q = 1, \quad R = 0,1$$

y es necesario también, el modelo en espacio de estados de la planta con acción repetitiva embebida.

La figura 6.8 muestra el desempeño del algoritmo. Una característica que puede ser destacada es la capacidad de predicción y actuación del controlador anticipando el cambio de la referencia. Hace con que la perturbación sea rechazada en 1 s después de la referencia ser aplicada. Otra diferencia es que la señal de control no tiene picos.

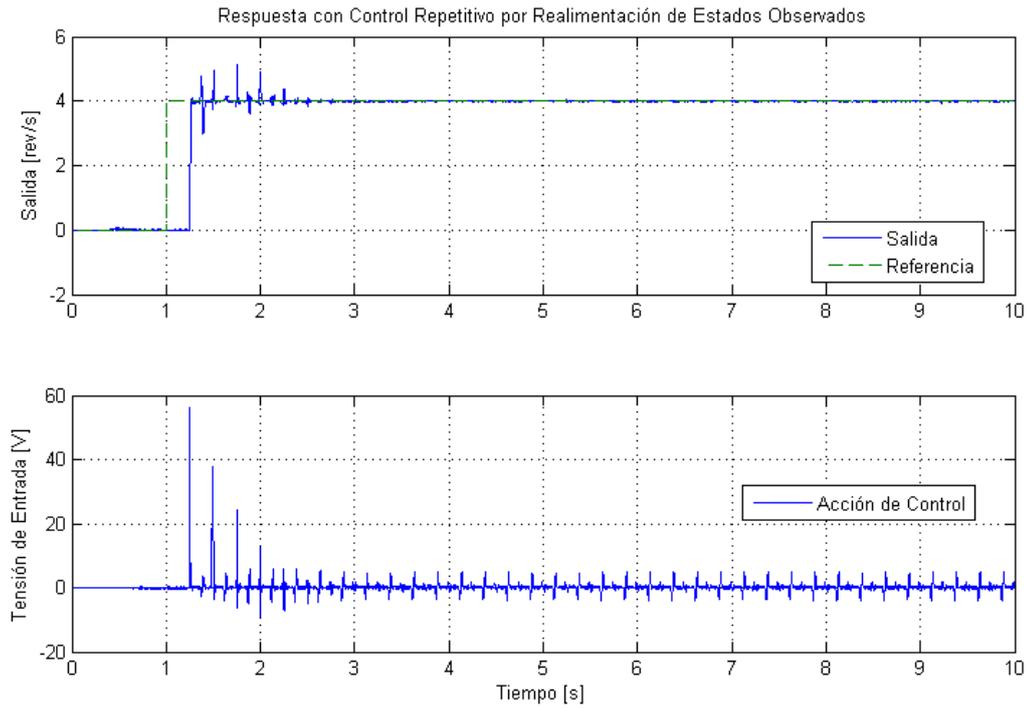


Figura 6.7: Simulación en Tiempo Real con Realimentación de Estados Observados.

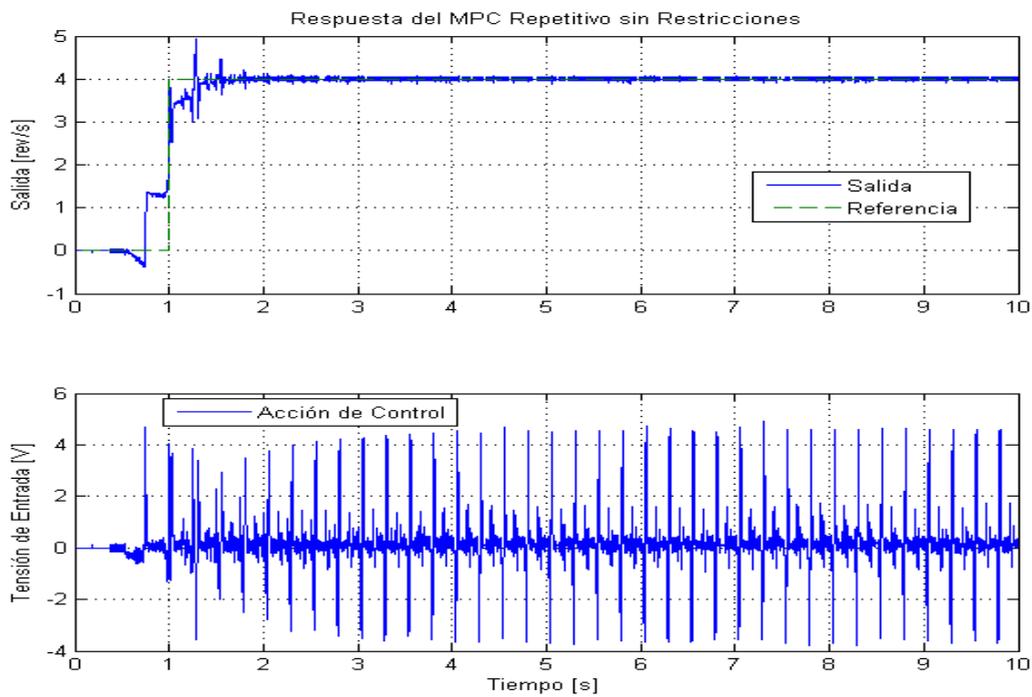


Figura 6.8: Simulación en Tiempo Real con MPC Repetitivo

### 6.6.4: MPC Repetitivo con Restricciones

Como para el sistema rechazar la perturbación es necesario una amplitud mínima de la acción de control, entonces la restricción en la amplitud no puede pasar el límite de la amplitud mínima. Entonces para una buena observación de como el MPC repetitivo puede trabajar con las restricciones será aplicada una restricción en la variación de la entrada a cada periodo  $T_p$ .

De esta forma el controlador tiene los mismos ajustes:

$$P = 2N, M = 10, Q = 1, R = 0,1$$

con las siguientes restricciones:

$$U_{max} = -U_{min} = 8 V$$

$$\Delta U_{max} = -\Delta U_{min} = 0,1 V/T_p$$

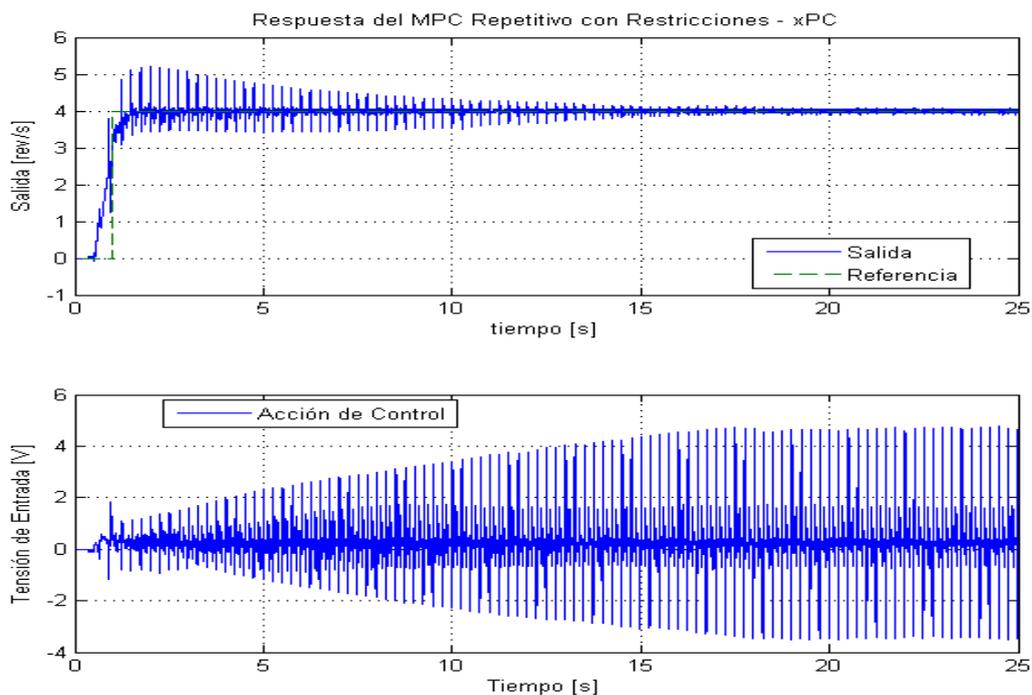


Figura 6.9: Simulación en Tiempo Real con MPC Repetitivo con Restricciones.

Con esto, el experimento realizado con la acción de las restricciones puede ser observado en la figura 6.9. Ahí, la acción de control en un determinado periodo no varía más que  $0,1 V$  del periodo anterior aumentando el tiempo necesario para rechazar la perturbación para aproximadamente  $18 s$ .

## 6.7: Análisis Comparativa

Como variables básicas de ajuste, todos necesitan de las mismas y algunos de pequeñas variaciones en estas variables. Pero para los parámetros de diseño cada algoritmo tiene una forma de ajuste diferente que van desde una forma mas simples como el *plug-in*, que necesita del valor  $K_r$  hasta una forma complexa, como el ajuste del MPC repetitivo que necesita de la combinación de valores de  $P, M, Q, R$  y una solución optima para el valor de la señal de control.

Todos los algoritmos consiguen rechazar la perturbación periódica y cada uno tiene sus características y parámetros de diseño. Sin embargo, el MPC repetitivo tiene algunas que los otros algoritmos no consiguen alcanzar, como la capacidad de predicción y el tratamiento de restricciones haciendo de este sistema de control un sistema más completo.

# 7 Conclusiones

El control repetitivo tiene muchas maneras de ser implementado y con el modelo interno insertado en el sistema de control es posible garantizar el rechazo (o seguimiento) de perturbaciones (o referencias) con características periódicas. Por otro lado, el control predictivo basado en modelo tiene características que definen el sistema de control con muchas cualidades que no están presentes en la forma clásica del control repetitivo. De esta forma este proyecto tuvo como objetivo principal la inserción del modelo interno repetitivo en un algoritmo de control predictivo.

Esta inserción fue implementada utilizando un modelo en espacio de estados con acción repetitiva embebida, posibilitando la combinación de ambas estrategias de control en un algoritmo de control que consigue operar con señales que poseen características periódicas.

El algoritmo desarrollado, MPC repetitivo, fue validado de forma experimental en una aplicación práctica real usando el *xPC Target* y así, este sistema de control puede ser aplicado en diferentes aplicaciones en la ingeniería de control que tratan de señales con características periódicas, como sistemas mecánicos rotatorios, robóticos, en electrónica de potencia y otros.

## 7.1: Trabajos Futuros

Es común los sistemas que tratan de señales periódicas pasaren por cambios en la referencia deseada, y con esto ocurre también el cambio del periodo fundamental de la perturbación. Esto trae una degradación del rendimiento del sistema de control repetitivo que no es considerada en este proyecto. De esta forma, un tema de trabajo futuro sería el estudio del MPC repetitivo para señales con periodo que pueden variar en el tiempo o son inciertas.

Otro tema de trabajo futuro sería la análisis de como los parámetros de ajuste del MPC repetitivo pueden ser seleccionados, con base en criterios deseados de ren-

dimiento y utilizando, por ejemplo, técnicas de inteligencia artificial.

Un otro trabajo posible sería hacer la inserción del lazo repetitivo utilizando una estructura de control, que sería la interpretación en espacio de estados del GPC (Control Predictivo Generalizado), y hacer una comparación con el MPC repetitivo implementado.

# Referencias

- [1] R. Costa-Castelló, R. Griñó, y Sudrià A., "Control repetitivo Una solución para numerosas aplicaciones", *Automática e Instrumentación SOLUCIONES*, n.º 328, Abril 2002.
- [2] R. Costa-Castalló, y J. Nebot, y R. Griñó, "Demonstration of the Internal Model Principle by Digital Repetitive Control of an Educational Laboratory Plant", *IEEE Transactions on Education*, vol. 48, no. 1, Febrero 2005.
- [3] B. Francis, W.Wonham, "Internal Model Principle in control theory," *Automatica*, vol. 12, pp. 457–465, 1976.
- [4] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *J. OD Dynamic Systems, Measurements Control*, vol. 109, pp. 65–68, Mar, 1987.
- [5] J.H. Lee, S. Natarajan, K.S. Lee, "A model-based predictive control approach to repetitive control of continuous processes with periodic operations", *Journal of Process Control* 11 (2001) 195–207.
- [6] L. Wang, S.Chai, E. Rogers, C.T. Freeman, "Multivariable Repetitive-predictive Controllers using Frequency Decomposition", *IEEE Transactions*, November 2011.
- [7] P.Y. Tiwari, M.V. Kothare, "Repetitive Model Predictive Control using Linear Matrix Inequalities", *American Control Conference*, June 2008.
- [8] P. Albertos, A. Sala, "Multivariable Control Systems", Springer, 2004.
- [9] E. Castelan, "Control Multivariável - DAS 5131", Libro de Clase, Mayo 2002.
- [10] J. E. Normey-Rico, E. F. Camacho, "Control of dead-time processes", Springer, 2007.
- [11] E. F. Camacho, C. Bordons, "Model Predictive Control", Springer, 2004.

- [12] C. Bordóns, “Capítulo 2: Control Predictivo con Restricciones”, Curso: Control Predictivo, Nuevas Tendencias y Aplicaciones, Universidad Politécnica de Valencia, 2007.
- [13] J. E. Normey-Rico, “Control Predictivo Basado en Modelo”, Slides de Clase, Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina.
- [14] A. Plucenio, “DAS 5945 - Técnicas de Controle Avançado aplicadas à Indústria do Petróleo e Gás.”, Slides de Clase, PRH34, Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina.
- [15] L. Wang, “Model Predictive Control System Design and Implementation Using MATLAB®”, Springer, 2009.
- [16] D. Di Ruscio, “Model Predictive Control and Optimization”, Lecture Notes, System and Control Engineering, Department of Technology, Telemark University College, Mars 2001.
- [17] A. Bemporad, M. Morari, “Robust Model Predictive Control: A Survey”, Paper, Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH).
- [18] XPC ”Mathworks - <http://www.mathworks.es/products/xpctarget/> ”.
- [19] Mathworks - XPC Getting Started Guide and User’s Guide.
- [20] G. A. Ramos, “Digital Repetitive Control Under Varying Frequency Conditions”, Tesi Doctoral, Universitat Politècnica de Catalunya, Maig 2012.

# APÉNDICE A – Códigos Matlab de las Simulaciones

## A.1: Estructura Plug-in

```
1 %% RC - Plug-in
2 %Daniel Miranda Cruz - PFC - 2012/1
3 clear all; close all; clc;
4 %% Parametros de Inicializacion
5 tsim = 12;
6 Kp = 0.3; tau = 0.1; Td = 0.02;
7 Tp = 0.4; N= 20; Ts = Tp/N;
8 K = 1.5; Kr = 0.7;
9 s = tf('s'); z = tf('z',Ts);
10 Gp = (K*exp(-Td*s))/(tau*s+1); %Proceso
11 Gd = c2d(Gp,Ts,'zoh'); [Np,Dp] = tfdata(Gd,'v');
12 R = (1)/(z^N-1); %Lazo Repetitivo
13 %% Lazo de realimentacion
14 Gcl = feedback((K*Gd),1,-1);
15 %% Controle para estabilizacion
16 [N0,D0] = tfdata(Gcl,'v');
17 Gx = Kr*tf(D0,N0,Ts);
18 Gr = (R*Gx);
19 %% Sistema en lazo abierto con y sin repetitivo
20 op = K*Gd; opRep = (1+Gr)*K*Gd;
21 set(op,'name','openLoop'); set(opRep,'name','openLoopRepetitive');
22 %% Sistema total en lazo cerrado
23 Gclt = feedback(((1+Gr)*K*Gd),1,-1);
24 step(Gclt); grid on; figure;
25 bode(opRep,'r',op,'—'); grid on;
26 %% Simulacion y Resultados
27 sim('RC-plugin');
```

```

28 figure; subplot(2,1,1);
29 stairs(t,y(:,2),'k');hold on; stairs(t,y(:,3),'-');
30 title('Repetitive Control - Plug-in, kr=0.7');
31 ylabel('Salida - Referencia');
32 legend('Y','r','Location','best'); grid on;
33 subplot(2,1,2);
34 stairs(t,u(:,2),'k');
35 xlabel('Tiempo[s]'); ylabel('Accion de control - U');
36 legend('U','Best'); grid on;

```

## A.2: Estructura con Realimentación de Estados

```

1 %% RC - Realimetnacion de Estados Observados
2 %Daniel Miranda Cruz - PFC - 2012/1
3 clear all; close all; clc;
4 %% Parametros de Inicializacion
5 tsim = 12;
6 Kp = 0.3; tau = 0.1; Td = 0.02;
7 Tp = 0.4; N= 20; Ts = Tp/N;
8 K = 1.5; Krep = 0.7;
9 s = tf('s'); z = tf('z',Ts);
10 Rep = ((1)/(z^N-1)); %Lazo Repetitivo
11 Gp = (Kp*exp(-Td*s))/(tau*s+1);
12 Gd = c2d(Gp,Ts,'zoh'); %Proceso
13 %% Proceso y Lazo Repetitivo en Espacio de Estados
14 sysRep = canon(Rep,'modal');
15 [Ar,Br,Cr,Dr] = ssdata(sysRep);
16 numAr = length(Ar*Ar'); numCr = length(Cr*Cr');
17 sysPlant = ss(Gd);
18 sysnd = delay2z(sysPlant);
19 sysP = canon(sysnd,'modal');
20 [Ap,Bp,Cp,Dp] = ssdata(sysP);
21 numAp = length(Ap*Ap'); numCp = length(Cp*Cp');
22 %% Observador de Estados - Filtro de Kalman
23 Qw = 0.9; Rv = 0.01;
24 [Kest,L,P] = kalman(sysP,Qw,Rv);
25 %% Representacion Compacta
26 Aa = [Ap zeros(numAp,numAr);(-Br*Cp) Ar];
27 Ba = [Bp;zeros(numAr,numCr)];
28 Ca = [Cp zeros(numCr,numAr)];

```

```

29 Ea = [zeros(numAp,numCp);Br];
30 %% Posicion Deseada de los Valores Propios en Lazo Cerrado
31 avalAp = eig(Ap); avalAr = eig(Ar); %Usando notacion simplificada
32 P = [0.8 0.1 0.9*avalAr'];
33 Gcl = feedback((K*Gd),1,-1); %Usando polos del sistema Plug-in
34 [N0,D0] = tfdata(Gcl,'v'); Gx = Krep*tf(D0,N0,Ts);
35 Gr = (Rep*Gx); Gclt = feedback(((1+Gr)*K*Gd),1,-1);
36 P_plug = pole(Gclt)';
37 %% Ganancias de Realimentacion
38 Ka = place(Aa,Ba,P); Ka_plug = place(Aa,Ba,P_plug);
39 Ke = Ka(1,1:2); Kr = Ka(1,3:length(Aa));
40 Ke_plug = Ka_plug(1,1:2); Kr_plug = Ka_plug(1,3:length(Aa));
41 Aval = eig(Aa-Ba*Ka); Aval_plug = eig(Aa-Ba*Ka_plug);
42 %% Closed Loop matrix
43 Amf = [Aa-Ba*Ka Ba*Ke;zeros(numAp,length(Aa)) Ap-L*Cp];
44 Bmf = [Ea;zeros(numAp,numCp)];
45 Cmf = [Ca zeros(numCp,numAp)];
46 sysMf = ss(Amf,Bmf,Cmf,0,Ts); bode(sysMf);
47 %% Simulation and plotting
48 %Notacion Simplificada
49 sim('RC.RealEstados'); figure;
50 subplot(2,1,1);
51 stairs(t,y(:,2),'k');hold on; stairs(t,y(:,3),'-');
52 title('Control Repetitivo con Realimentacion de Estados Observados');
53 xlabel('Tiempo[s]'); ylabel('Salida - Referencia');
54 legend('Y','r','location','Best'); grid on;
55 subplot(2,1,2);
56 stairs(t,u(:,2),'k');
57 xlabel('Tiempo[s]'); ylabel('Accion de Control - U');
58 legend('U','Best'); grid
59 %Comparacion con Estructura Plug-in
60 sim('RC.RealEstados.Plugin'); figure;
61 subplot(2,1,1);
62 stairs(t,y(:,2),'k');hold on; stairs(t,y(:,3),'-');
63 title('Comparacion con la estructura Plug-in');
64 xlabel('Tiempo[s]'); ylabel('Salida - Referencia');
65 legend('Y','r','location','Best'); grid on;
66 subplot(2,1,2);
67 stairs(t,u(:,2),'k');
68 xlabel('Tiempo[s]'); ylabel('Accion de Control - U');
69 legend('U','Best'); grid

```

## A.3: MPC Repetitivo

```

1 %% MPC Repetitivo – Sin restricciones
2 %Daniel Miranda Cruz – PFC – 2012/1
3 clear all; close all; clc;
4 %% Parametros de Inicializacion
5 tsim = 12; %Tiempo de simulacion
6 Kp = 0.3; tau = 0.1; Td = 0.02; %Parametros del modelo
7 Tp = 0.4; N= 20; Ts = Tp/N;
8 s = tf('s'); z = tf('z',Ts);
9 Rep = (1)/(1-z^(-N)); %Lazo repetitivo
10 Gp = (Kp*exp(-Td*s))/(tau*s+1);
11 Gd = c2d(Gp,Ts,'zoh'); %Proceso discreto
12 %% Modelo Extendido con Lazo Repetitivo Embebido
13 %Con las variables: Delta_x(k)=x(k)-x(k-N) y Delta_u(k)=u(k)-u(k-N)
14 %el modelo tiene la forma: x(k+1)=Ax(k)+Bu(k)
15 % y(k) = Cx(k)
16 %Donde x(k) = [Delta_x(k);y(k);y(k-1);...;y(k-N+1);y(k-N)]
17 ret = ss(z^(-N)); %Modelo del retardo
18 sysnd = delay2z(Gd);
19 sysP = canon(sysnd,'modal');
20 [Ap,Bp,Cp,Dp] = ssdata(sysP); %Proceso
21 numAp = length(Ap*Ap');
22 [m1,n1]=size(Cp);
23 [n1,n_in]=size(Bp);
24 A=zeros(n1+N+1,n1+N+1); A(1:n1,1:n1)=Ap;
25 A(n1+1:n1+m1,1:n1)=Cp*Ap;
26 A(n1+2:n1+N+1,n1+2:n1+N+1)=ret.A; %Matriz A del modelo extendido
27 A(n1+1:n1+m1,n1+N)=eye(m1,m1);
28 A(n1+2,n1+1)=eye(m1,m1);
29 B=zeros(n1+N+1,n_in); B(1:n1,:)=Bp; %Matriz B
30 B(n1+1:n1+m1,:)=Cp*Bp;
31 C=zeros(m1,n1+N+1); %Matriz C
32 C(:,n1+1:n1+m1)=eye(m1,m1);
33 numA = length(A*A');
34 sys = ss(A,B,C,0,Ts); %Sistema del Modelo Esxtendido
35 %% Parametros del MPC
36 P = 2*N; %Horizonte de Prediccion
37 M = 10; %Horizonte de Control
38 Q = 1*eye(P); R = 0.01*eye(M); %Matrices de Ponderacion
39 %% Matrices para los calculos de las predicciones

```

```

40 h(1,:)=C;
41 F(1,:)=C*A;           %Matriz F
42 for k=2:P
43 h(k,:)=h(k-1,:)*A; F(k,:)= F(k-1,:)*A;
44 end
45 v=h*B; G=zeros(P,M); G(:,1)=v;   %Matriz G
46 for i=2:M
47     G(:,i)=[zeros(i-1,1);v(1:P-i+1,1)]; %Toeplitz matrix
48 end
49 H = G'*Q*G+R;
50 %% Definicion de la referencia y de la perturbacion
51 N_steps = (tsim/Ts)+P+1+N;   %Cantidad de pasos de integracion
52 r = zeros(N_steps,2); r(1,1) = 0;
53 r(1/Ts:N_steps,2)=1;
54 r(1,2:P+1)=r(1:P,2);
55 for t=2:N_steps;
56     r(t,1)=r((t-1),1)+Ts;
57 end
58 for t2=2:N_steps-P-1;           %Composicion de la
59     r(t2,2:P+1) = r(t2:t2+P-1,2); %referencia em tamanos P, para ...
    prediccion
60 end                             %Perturbacion en forma sinusoidal
61 pert = zeros(N_steps,2);       %actuando por un intervalo de tiempo
62 pert(:,1) = r(:,1);           %con frec fundamentasl y 1 harmonica
63 for kkp=round(N_steps/3):round(N_steps*2/3);
64     pert(kkp,2) = 0.5*sin(2*pi/N*kkp)+0.25*sin(4*pi/N*kkp);
65 end
66 %% Parametros de la simulacion
67 vet_ref = ones(P,1);
68 U1 = zeros(1,M); U1(1,1)=1;
69 R1 = zeros(1,P); R1(1,1)=1;
70 %% Simulacion y Resultados
71 %utiliza modelo en simulink
72 sim('MPC_FT_repEMB');
73 figure; subplot(211);
74 stairs(yr(:,1),yr(:,2),'k'); hold on; stairs(yr(:,1),yr(:,3),'-');
75 ylabel('Y - r');
76 legend('Salida','Referencia','Location','best'); grid on;
77 title('MPC repetitivo sin restricciones - Salida y Referencia');
78 subplot(212);
79 stairs(u(:,1),u(:,2),'k');
80 xlabel('tiempo'); ylabel('U');
81 legend('Accion de Control - U','Location','best'); grid on;

```

```
82 title('Control aplicado en el sistema');
```

## A.4: MPC Repetitivo con Restricciones

```

1 %% MPC Repetitivo - Con restricciones
2 %Daniel Miranda Cruz - PFC - 2012/1
3 clear all; close all; clc;
4 %% Parametros de Inicializacion
5 tsim = 12; %Tiempo de simulacion
6 Kp = 0.3; tau = 0.1; Td = 0.02; %Parametros del modelo
7 Tp = 0.4; N= 20; Ts = Tp/N;
8 s = tf('s'); z = tf('z',Ts);
9 Rep = (1)/(1-z^(-N)); %Lazo repetitivo
10 G1 = (Kp*exp(-Td*s))/(tau*s+1);
11 Gd = c2d(G1,Ts,'zoh'); %Proceso discreto
12 %% Modelo Extendido con Lazo Repetitivo Embebido
13 %Con las variables: Delta_x(k)=x(k)-x(k-N) y Delta_u(k)=u(k)-u(k-N)
14 %el modelo tiene la forma: x(k+1)=Ax(k)+Bu(k)
15 % y(k) = Cx(k)
16 %Donde x(k) = [Delta_x(k);y(k);y(k-1);...;y(k-N+1);y(k-N)]
17 ret = ss(z^(-N)); %Modelo del retardo
18 sysnd = delay2z(Gd);
19 sysP = canon(sysnd,'modal');
20 [Ap,Bp,Cp,Dp] = ssdata(sysP); %Proceso
21 numAp = length(Ap*Ap');
22 [m1,n1]=size(Cp);
23 [n1,n_in]=size(Bp);
24 A=zeros(n1+N+1,n1+N+1); A(1:n1,1:n1)=Ap;
25 A(n1+1:n1+m1,1:n1)=Cp*Ap;
26 A(n1+2:n1+N+1,n1+2:n1+N+1)=ret.A; %Matriz A del modelo extendido
27 A(n1+1:n1+m1,n1+N)=eye(m1,m1);
28 A(n1+2,n1+1)=eye(m1,m1);
29 B=zeros(n1+N+1,n_in); B(1:n1,:)=Bp; %Matriz B
30 B(n1+1:n1+m1,:)=Cp*Bp;
31 C=zeros(m1,n1+N+1); %Matriz C
32 C(:,n1+1:n1+m1)=eye(m1,m1);
33 numA = length(A*A');
34 sys = ss(A,B,C,0,Ts); %Sistema del MOdelo Esxtendido
35 %% Parametros del MPC
36 P = 2*N; %Horizonte de Prediccion

```

```

37 M = 10; %Horizonte de Control
38 Q = 1*eye(P); R = 0.01*eye(M); %Matrices de Ponderacion
39 %% Matrices para los calculos de las predicciones
40 h(1,:) = C;
41 F(1,:) = C*A; %Matriz F
42 for k=2:P
43 h(k,:) = h(k-1, :)*A; F(k, :) = F(k-1, :)*A;
44 end
45 v=h*B; G=zeros(P,M); G(:,1)=v; %Matriz G
46 for i=2:M
47 G(:,i)=[zeros(i-1,1);v(1:P-i+1,1)]; %Matriz Toeplitz
48 end
49 H = G'*Q*G+R;
50 %% Definicion de la referencia y de la perturbacion
51 N_steps = (tsim/Ts)+P+1+N; %Cantidad de pasos de integracion
52 r = zeros(N_steps,2); r(1,1) = 0;
53 r(1/Ts:N_steps,2)=1;
54 r(1,2:P+1)=r(1:P,2);
55 for t=2:N_steps; %Composicion de la referencia
56 r(t,1)=r((t-1),1)+Ts; %em tamanos P, para prediccion
57 end
58 for t2=2:N_steps-P-1;
59 r(t2,2:P+1) = r(t2:t2+P-1,2);
60 end
61 pert = zeros(N_steps,2);
62 pert(:,1) = r(:,1);
63 %Perturbacion en forma sinusoidal actuando por un intervalo de tiempo
64 %con frec fundamental(2pi/Tp) y 1 harmonica(4pi/Tp)
65 for kkp=round(N_steps/3):round(N_steps*2/3);
66 pert(kkp,2) = 0.5*sin(2*pi/N*kkp)+0.25*sin(4*pi/N*kkp);
67 end
68 %% Parametros de la simulacion
69 vet_ref = ones(P,1);
70 IN = eye(N); Um = IN(1:M, :);
71 U1 = zeros(1,M); U1(1,1)=1;
72 R1 = zeros(1,P); R1(1,1)=1;
73 %% Parametros de las Restricciones
74 Umax = 10; Umin = -10;
75 dUmax = 0.5; dUmin = -0.5;
76 c=ones(M,1);
77 AA=[]; bb=[];
78 AdU = [eye(M);-eye(M)]; %Restriccion en la variacion de la entrada
79 bdU=[dUmax*c;-dUmin*c]; %en un period Tp

```

```

80 AU = AdU; %Restriccion en la Amplitud de la entrada
81 AA = [AU;AdU];
82 %% Simulacion
83 Xp = zeros(numA,1); xp = zeros(numAp,1); y=0;
84 Δ_u(1:N_steps-P-1+N)=0;
85 X(numA,1:N_steps-P-1+N)=0;
86 U(1:N_steps-P-1+N)=0;
87 x(numAp,1:N_steps-P-1+N)=0;
88 Y(1:N_steps-P-1+N)=0;
89 for k=(N+2):(N_steps-P-1);
90     %Matrices de Restricciones
91     bU=[c*Umax-U((k-N):(k-N+M-1))'; -c*Umin+U((k-N):(k-N+M-1))'];
92     bb = [bU;bdU];
93     %Solucion del problema de optimizacion
94     f = (G'*Q*(F*Xp-r(k:(k+(P-1)),2)));
95     vet_u = quadprog(H,f,AA,bb); %-(H)\f;
96     Δ_u(k) = vet_u(1,1);
97     U(k) = U(k-N)+Δ_u(k);
98     %U(k) = Δ_u(k)+q1*U(k-N+1)+q0*U(k-N)+q1*U(k-N-1);
99     Y(k) = y;
100    x(:,k) = xp;
101    X(:,k) = Xp; %Actualizacion de las variables
102    xp = Ap*xp+Bp*U(k);
103    y = Cp*xp-pert(k,2);
104    Xp = [xp-x(:,k-N+1);y;flipud(Y(1,k-N+1:k))];
105 end
106 %utilizando modelo en simulink
107 sim('MPC_FT_Rest_repEMB');
108 %% Resultados
109 t=(0:N_steps-P-2)*Ts;
110 figure;
111 subplot(211);
112 stairs(t,Y(1:N_steps-P-1),'k'); hold on; ...
    stairs(t,r(1:N_steps-P-1,2),'-.');
113 xlabel('Tiempo[s]'); ylabel('Salida - Referencia');
114 legend('Y','r','Location','best'); grid on;
115 title('MPC repetitivo con restricciones');
116 subplot(212);
117 stairs(t,U(1:N_steps-P-1),'k');
118 xlabel('Tiempo[s]'); ylabel('Accion de Control - U');
119 legend('U','Location','best'); grid on;

```