

CONTROL ÓPTIMO ADAPTATIVO DE UN SISTEMA MULTIVARIABLE DE PÉNDULO INVERTIDO

DIEGO ALEJANDRO RAMÍREZ GAVIRIA

Cód: 1093222999

ANDRÉS FELIPE RAMÍREZ GIRALDO

Cód: 1088018655

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
INGENIERÍA MECATRÓNICA POR CICLOS
FACULTAD DE TECNOLOGÍA
PEREIRA
2015**

**CONTROL ÓPTIMO ADAPTATIVO DE UN SISTEMA MULTIVARIABLE DE
PÉNDULO INVERTIDO**

PROYECTO DE GRADO

PARA OPTAR POR EL TÍTULO DE INGENIERO EN MECATRÓNICA

Autores:

DIEGO ALEJANDRO RAMÍREZ GAVIRIA

ANDRÉS FELIPE RAMÍREZ GIRALDO

Director:

Ph.D. EDUARDO GIRALDO

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
INGENIERÍA MECATRÓNICA POR CICLOS
FACULTAD DE TECNOLOGÍA
PEREIRA
2015**

RESUMEN

El control es uno de los pilares en la Ingeniería Mecatrónica, ya que gracias a este y a sus grandes avances, es posible controlar desde la velocidad de giro de un motor hasta los grandes transbordadores espaciales, estos últimos siendo sistemas complejos que cambian en el tiempo y es necesario utilizar técnicas de control moderno.

Con base en lo anterior, se decidió controlar un péndulo invertido, el cual es un sistema inestable y variante en el tiempo, este último controlado mediante el control adaptativo ya que al cambiar en el tiempo un controlador fijo en algún momento no llevaría la variable de control a su punto deseado.

Por otra parte se reduce el error a su mínimo posible y es allí donde es de gran relevancia el control óptimo aplicado al sistema. Por último estas dos técnicas de control moderno se aplican en un modelo simulado y además se prueban en una planta que es construida con el kit de LEGO MINDSTORMS NXT y software de control como LabVIEW y MATLAB.

TABLA DE CONTENIDO

1. DISEÑO DEL MODELO	8
2. IDENTIFICACIÓN DE LOS PARÁMETROS DEL SISTEMA	11
2.1 ALGORITMO DE PROYECCIÓN	13
2.2 ALGORITMO DE MÍNIMOS CUADRADOS	14
3. DISEÑO DEL CONTROLADOR ADAPTATIVO	15
4. DISEÑO DEL CONTROLADOR ÓPTIMO	18
4.1 CONTROLABILIDAD	19
4.2 OBSERVABILIDAD	20
4.3 CONTROL EN ESPACIO DE ESTADOS	20
4.3.1 OBSERVADOR DE ESTADOS	21
4.3.2 FÓRMULA DE ACKERMAN	22
4.4 CONTROL ÓPTIMO CUADRÁTICO	24
4.4.1 DETERMINACIÓN DE LAS MATRICES Q Y R	25
4.4.2 CONTROL EN ESPACIO DE ESTADOS CON GANANCIA EN LAZO DIRECTO	27
5. IMPLEMENTACIÓN Y RESULTADOS	28
5.1 ALGORITMO DE PROYECCIÓN	28
5.2 ALGORITMO DE MÍNIMOS CUADRADOS	30
5.3 CONTROL PID ADAPTATIVO	31
5.4 CONTROL ÓPTIMO CUADRÁTICO	33
5.5 IMPLEMENTACIÓN DEL CONTROL ÓPTIMO SOBRE UNA PLANTA FÍSICA	36
6. CONCLUSIONES	39
7. RECOMENDACIONES	40
8. ANEXOS	41
8.1 CÓDIGO DE MÍNIMOS CUADRADOS EN MATLAB FUNCTION	41
8.2 CÓDIGO DE CONTROL PID ADAPTATIVO	42
8.3 CÓDIGO DE CONTROL ÓPTIMO ADAPTATIVO	43
9. BIBLIOGRAFÍA	45

TABLA DE FIGURAS

1.1	Diseño de péndulo invertido en SolidWorks	8
1.2	Exportación del diseño	9
1.3	Diagrama de bloques del modelo en SimuLink	9
1.4	Diagrama de bloques del sistema completo	10
1.5	Sistema importado en Matlab	10
2.1	Sistema de péndulo invertido [2]	11
2.2	Explicación geométrica del algoritmo de proyección.	13
3.1	Esquema general de un sistema de control adaptativo [5]	15
3.2	Lazo de control para PID tipo 2	16
4.1	Sistema de control en lazo cerrado con $u(k) = -Kx(k)$	21
4.2	Diagrama esquemático del observador de estados	21
4.3	Sistema de control regulado mediante la realimentación del estado observado	23
4.4	Sistema de control en lazo directo mediante la realimentación del estado observado	27
5.1	Obtención de entradas y salidas del sistema	28
5.2	Parámetros del sistema mediante proyección	29
5.3	Convergencia entre las salidas reales y las estimadas	29
5.4	Convergencia entre salidas estimadas y reales	30
5.5	Identificación de parámetros mediante mínimos cuadrados	31
5.6	Implementación de control PID adaptativo.	31
5.7	Respuesta del sistema	32
5.8	Respuesta del sistema simulado	32
5.9	Diagrama de bloques control óptimo	33
5.10	Respuesta del sistema con matrices de pesos identidad	34

5.11	Respuesta del sistema con matrices Q y R seleccionadas	35
5.12	Planta para control de ángulo construida con NXT	36
5.13	Visualización de la planta en el software LabView	37
5.14	Diseño del controlador en LabVIEW	37
5.15	Panel frontal en LabVIEW	38

INTRODUCCIÓN

El presente documento tiene como principal objetivo, dar a conocer el trabajo que se realizó con un sistema multivariable de péndulo invertido, donde las variables del sistema que se intervinieron fueron la posición angular y traslacional.

La finalidad del trabajo es realizar un control óptimo adaptativo del sistema. Para dicho desarrollo se hace fundamental una continua identificación de los parámetros de la planta, debido a que ésta los varía constantemente con el tiempo. Adicionalmente, se presentará un desarrollo con ecuaciones de estados en el capítulo de control óptimo, con la finalidad de determinar la señal de control para el sistema; dichas ecuaciones son requeridas para poder hacer uso adecuado de las funciones DLQR y Ackerman, que se hacen indispensables en el desarrollo del trabajo.

Además, se hace uso del software MATLAB para poder realizar la respectiva programación del proceso, que incluye códigos de identificación y códigos de control. Adicionalmente, se realiza un modelo de la planta en SolidWorks y posteriormente se exporta a MATLAB para realizar la simulación desde allí. Por medio de Simulink se obtienen los bloques que representan la planta importada y con los cuales se realizan las operaciones anteriormente mencionadas.

Es importante mencionar, que con ayuda del software LabVIEW se realiza una importante prueba en un LEGO MINDSTORMS NXT. La prueba consiste en corroborar la funcionalidad del código de identificación y control óptimo para la variable de posición angular, usando como entrada para la obtención de datos un giroscopio.

Finalmente, en el documento se encontrarán los resultados obtenidos frente a técnicas de control clásico.

1. DISEÑO DEL MODELO

Gracias a la variedad de herramientas con que se cuenta hoy en día en el campo del control, se decide construir la planta en un software de diseño asistido por computadora (CAD) y mediante el software MATLAB se desea validar y probar todas las etapas del proceso.

Con base en lo anterior, el software CAD que se desea utilizar es SolidWorks 2010, el cual es una potente herramienta 3D que mediante un toolbox llamado SimMechanics Link es posible importar el diseño y de esta forma poder controlarlo en MATLAB. En la figura 1.1 se muestra el diseño de un péndulo invertido simple, para el cual sus movimientos se encuentran restringidos únicamente en el plano horizontal y en el pivote del péndulo, es decir, las posiciones que se desean controlar (Angular y Traslacional).

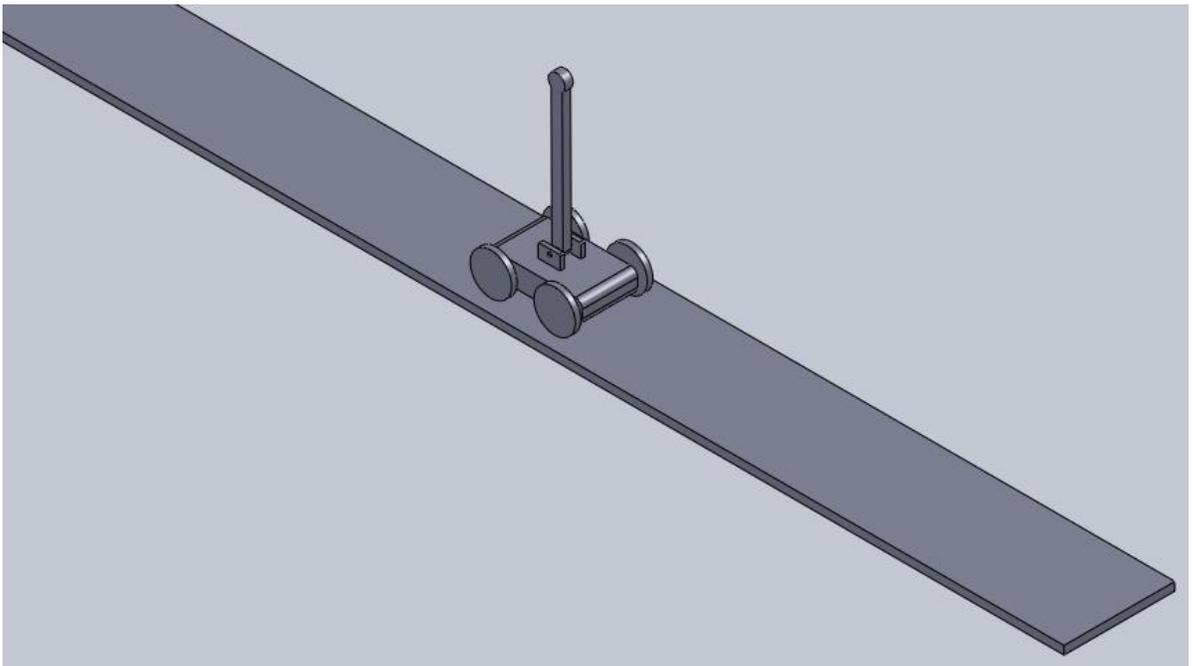


Figura 1.1 Diseño de péndulo invertido en SolidWorks. Los autores

Una vez diseñado el modelo se procede a su exportación, para lo cual es necesario que esté activa la herramienta SimMechanics en ambos software, en la figura 1.2 se muestra la opción para exportar el diseño.

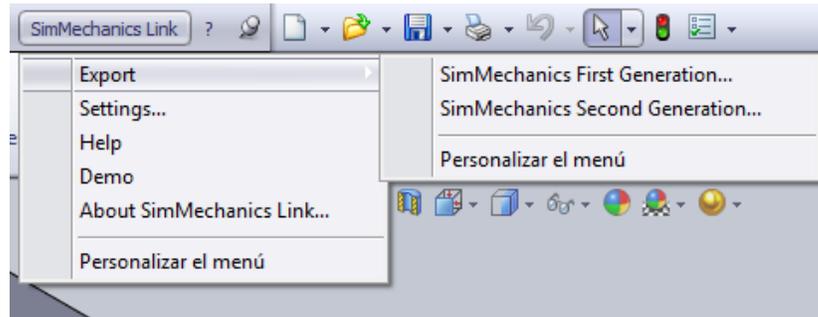


Figura 1.2 Exportación del diseño

Finalmente, el archivo queda con extensión *.xml* y en la ventana de comandos de MATLAB se escribe el siguiente comando para importar el diseño:

mech_import("Nombredelarchivo.xml")

De la anterior sentencia se obtiene como resultado el diagrama de bloques en SimuLink, del cual se dispone para hacer el control.

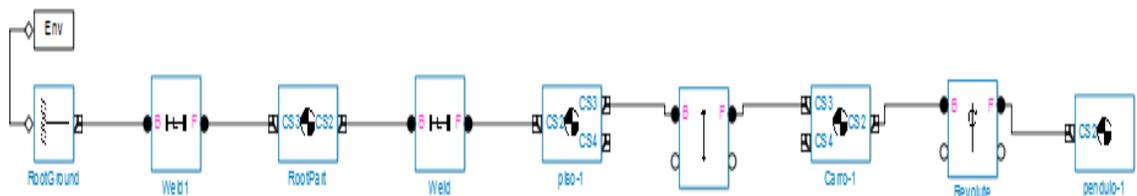


Figura 1.3 Diagrama de bloques del modelo en SimuLink

Teniendo en cuenta el diagrama de bloques generados, y para tener un acercamiento a la realidad, se optó por adicionar al sistema sus respectivas fricciones, ya que esto permite tener un mejor control del sistema. A continuación, en la figura 1.4 se puede observar el sistema final con sus respectivas fricciones.

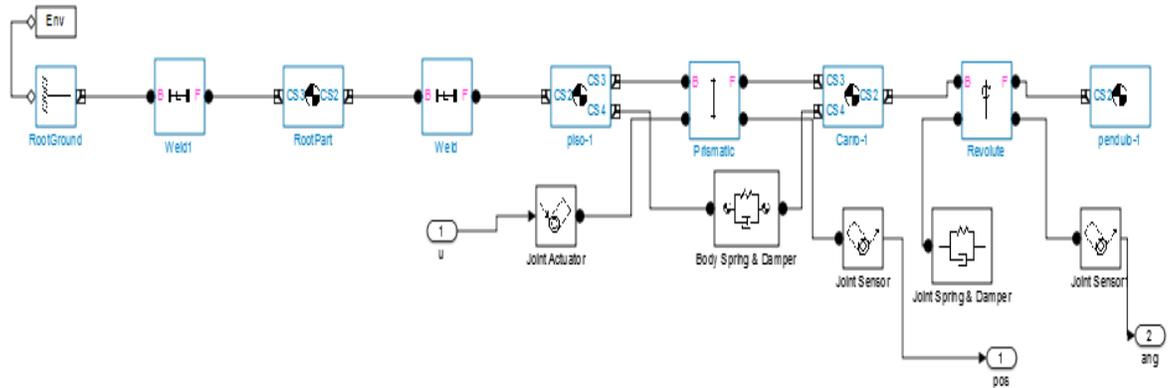


Figura 1.4. Diagrama de bloques del sistema completo

Finalmente, se crea un subsistema el cual aloja la parte mecánica, y se corre el programa para ver el diseño importado en la ventana de visualización de Matlab.

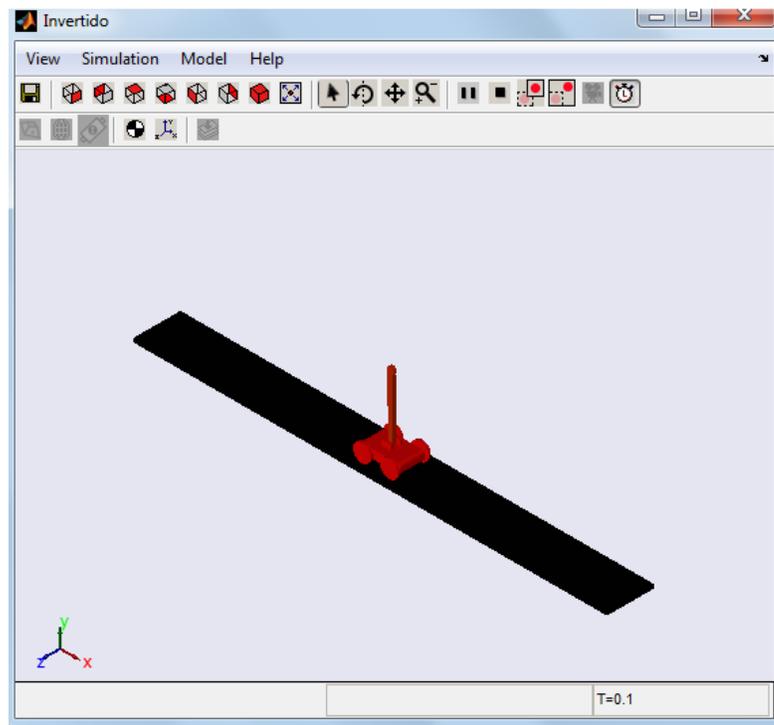


Figura 1.5 Sistema importado en Matlab

2. IDENTIFICACIÓN DE LOS PARÁMETROS DEL SISTEMA

Para el control de sistemas, se conocen diversos procedimientos de diseño para casos en los cuales los parámetros de la planta son constantes y el sistema resultante es invariante en el tiempo. Cuando los parámetros de la planta son escasamente conocidos o varían durante la operación normal, el uso de técnicas adaptativas se requiere para obtener un alto desempeño en los sistemas de control. El control adaptativo se considera como cualquier estrategia de control que usa estimación de los parámetros en tiempo real (identificación recursiva). El controlador adaptativo al ser diseñado está basado en el principio de equivalencia de certidumbre: se diseña el controlador suponiendo que los parámetros de la planta son conocidos y debido a que estos en realidad no se conocen, en el tiempo k se reemplazan por su estimado. [1]

A continuación en la figura 2.1 se muestra el sistema a controlar, el cual es un sistema inestable.

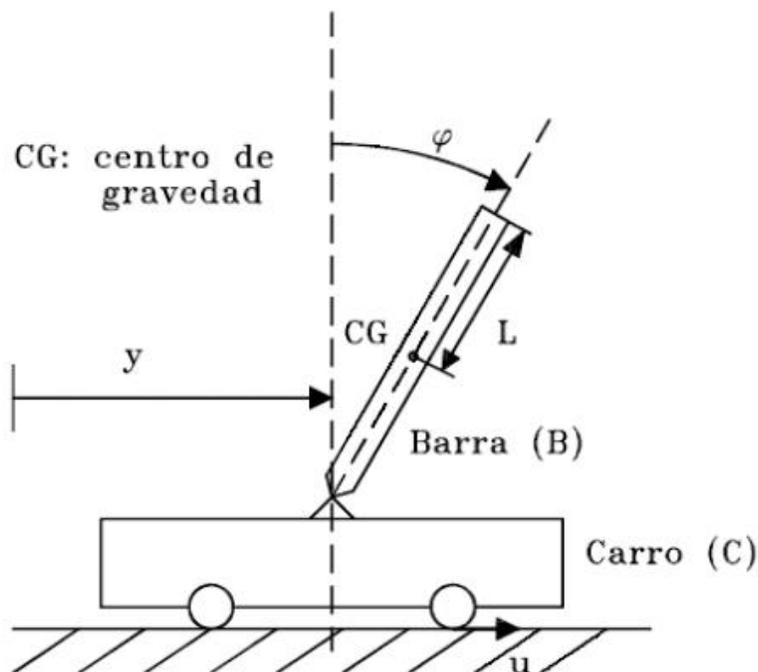


Figura 2.1 Sistema de péndulo invertido [2]

Teniendo en cuenta la figura 2.1, el sistema es del tipo SIMO (Single input Multiple output) donde la variable de entrada es la fuerza ejercida al carro y las variables de salida son la posición angular y la posición traslacional.

La mayoría de técnicas para el diseño de sistemas de control se basan en un buen entendimiento de la planta bajo estudio y su ambiente. Sin embargo, en un número de tareas, la planta a ser controlada es muy compleja y los procesos básicos internos de ella no se entienden del todo. Debido al poco conocimiento que se puede tener de un sistema, se hace el uso de técnicas adaptativas que permitan controlar el sistema [3].

Básicamente la técnica de control se fundamenta en una identificación paramétrica en línea del sistema, modelo al cual se le pueden implementar diferentes tipos de control, generando así varias técnicas de control adaptativo o controladores auto-sintonizados.

Las características de entrada-salida de una amplia clase de sistemas dinámicos deterministas tanto lineales como no lineales pueden ser descritas por un modelo simple de la siguiente forma:

$$y(t) = \phi(t-1)^T \theta_0 \quad (1)$$

Donde:

$y(t)$ Denota la salida estimada (escalar) del sistema en el tiempo t .

$\phi(t-1)$ Denota un vector lineal o no lineal en función de las entradas y salidas anteriores del sistema:

$$Y(t-1) = [y(t-1), y(t-2), \dots]$$

$$U(t-1) = [u(t-1), u(t-2), \dots]$$

θ_0 Denota el vector de parámetros (desconocido).

Con base en la ecuación (1) es posible mostrar el siguiente algoritmo para la identificación de parámetros en línea.

2.1 ALGORITMO DE PROYECCIÓN

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\phi(t-1)}{\phi(t-1)^T \phi(t-1)} [y(t) - \phi(t-1)^T \hat{\theta}(t-1)] \quad (2)$$

Siendo $\hat{e}(t) = y(t) - \phi(t-1)^T \hat{\theta}(t-1)$

El anterior algoritmo puede ser explicado geoméricamente como se observa en la figura 2.2. Dados todos los posibles valores que satisfagan la ecuación (1) se selecciona entre todos los valores candidatos a ser θ_0 , aquel denotado por $\hat{\theta}(t)$, el cual es el más cercano a $\hat{\theta}(t-1)$. Entonces el criterio J dado a continuación es minimizado:

$$J = \frac{1}{2} \|\hat{\theta}(t) - \hat{\theta}(t-1)\|^2 \quad (3)$$

Algunas propiedades de este algoritmo aseguran que $\hat{\theta}(t)$ nunca estará más lejos de θ_0 que $\hat{\theta}(0)$. Otra propiedad implica que cuando el modelo de error $e(t)$, esté apropiadamente normalizado, la suma de los errores al cuadrado es finita. Esto se vuelve una condición suficiente para establecer una convergencia global [4].

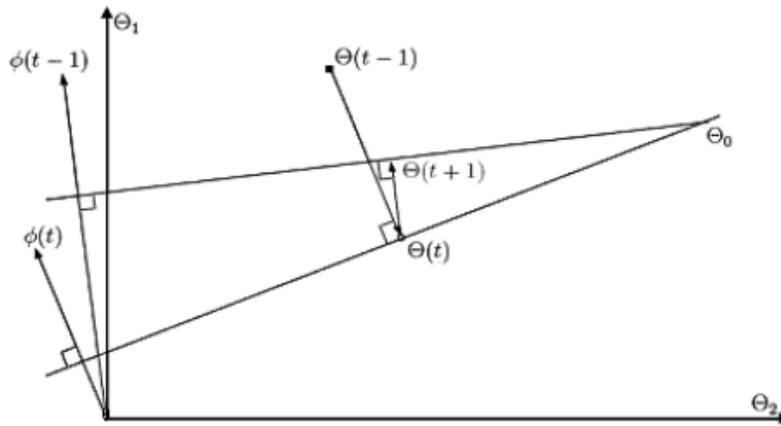


Figura 2.2 Explicación geométrica del algoritmo de proyección.

Una alternativa para evitar la división por cero es agregar una pequeña constante c en el algoritmo.

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\phi(t-1)}{c + \phi(t-1)^T \phi(t-1)} [y(t) - \phi(t-1)^T \hat{\theta}(t-1)] \quad (4)$$

2.2 ALGORITMO DE MÍNIMOS CUADRADOS

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-1)\phi(t-1)}{1+\phi(t-1)^T P(t-1)\phi(t-1)} [y(t) - \phi(t-1)^T \hat{\theta}(t-1)] \quad (5)$$

$$P(t) = P(t-1) - \frac{P(t-1)\phi(t-1)\phi(t-1)^T P(t-1)}{1+\phi(t-1)^T P(t-1)\phi(t-1)} \quad (6)$$

Este algoritmo resulta de la minimización de la siguiente función cuadrática [3]:

$$J_n(\theta) = \frac{1}{2} \sum_{t=1}^N (y(t) - \phi(t-1)^T \theta)^2 + \frac{1}{2} (\theta - \hat{\theta}(0))^T P_0^{-1} (\theta - \hat{\theta}(0)) \quad (7)$$

Básicamente, la función de costo anterior representa la suma de los errores al cuadrado. El segundo término de la derecha se incluyó para tener en cuenta las condiciones iniciales.

3. DISEÑO DEL CONTROLADOR ADAPTATIVO

Teniendo muy claro el método de identificación que se selecciona, en este caso mínimos cuadrados, se procede a diseñar un controlador el cual no es fijo, sino que depende de los parámetros de la planta. En la figura 3.1 se puede observar el esquema general de un sistema de control adaptativo.

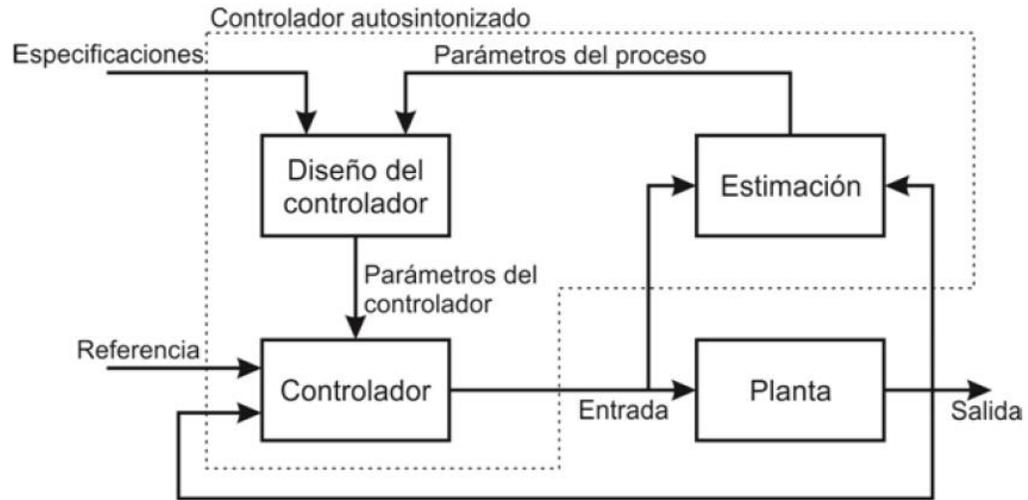


Figura 3.1 Esquema general de un sistema de control adaptativo [5].

Se asume el sistema lineal como una aproximación de segundo orden el cual tiene función de transferencia de la siguiente forma:

$$\frac{y(k)}{u(k)} = \frac{b_1q^{-1}+b_2q^{-2}}{1+a_1q^{-1}+a_2q^{-2}} \quad (8)$$

Donde a_1 , a_2 , b_1 y b_2 son los parámetros estimados en la etapa de identificación.

Se procede a implementar el controlador, el cual será un control PID tipo 2 con reubicación de polos, en realidad serán dos controladores ya que el modelo se asumirá como dos lazos de control, uno para la posición angular y otro para la posición traslacional.

En la figura número 3.2 se puede observar el diagrama de bloques para el lazo de control del sistema.

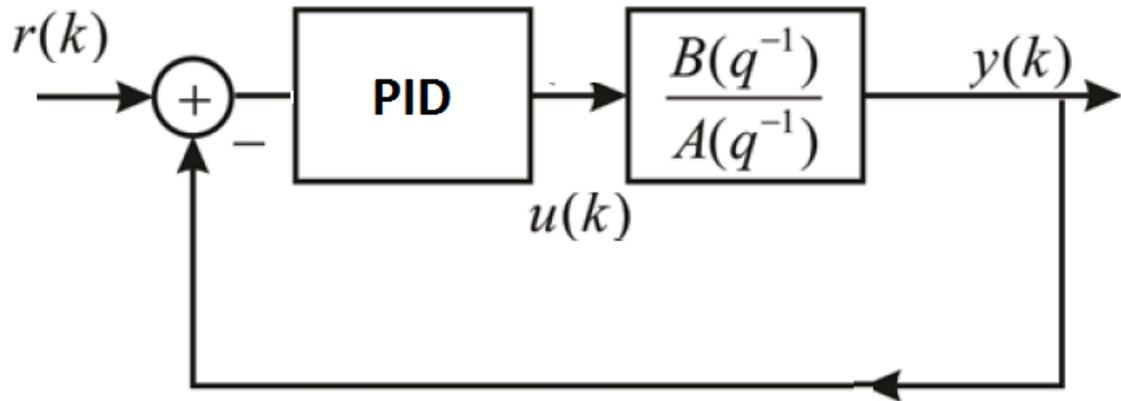


Figura 3.2 Lazo de control para PID tipo 2

Sobre el sistema aproximado de segundo orden se implementa el control PID el cual es de la forma:

$$\frac{C_0 + C_1 q^{-1} + C_2 q^{-2}}{(1 - q^{-1})(1 + C_3 q^{-1})} \quad (9)$$

Teniendo en cuenta (8) y (9) se procede a determinar el polinomio característico en lazo cerrado:

$$PLC = \frac{\frac{C_0 + C_1 q^{-1} + C_2 q^{-2}}{(1 - q^{-1})(1 + C_3 q^{-1})} * \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}}{1 + \frac{C_0 + C_1 q^{-1} + C_2 q^{-2}}{(1 - q^{-1})(1 + C_3 q^{-1})} * \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}} \quad (10)$$

De 10 se obtiene el polinomio característico en lazo cerrado que se muestra a continuación:

$$PLC = Z^4 + (a_1 + c_3 - 1 + c_0 b_0)Z^3 + (a_2 + c_3 a_1 - a_1 - c_3 + c_0 b_1 + c_1 b_0)Z^2 + (c_3 a_2 - a_2 - c_3 a_1 + c_1 b_1 + c_2 b_0)Z + (c_2 b_1 - c_3 a_2) \quad (11)$$

Una vez hallado el polinomio característico se compara con el polinomio deseado:

$$PD = Z^4 + \beta_1 Z^3 + \beta_2 Z^2 + \beta_3 Z + \beta_4 \quad (12)$$

Se iguala término a término donde se tiene un sistema de ecuaciones con 4 incógnitas y 4 ecuaciones.

$$a_1 + c_3 - 1 + c_0 b_0 = \beta_1 \quad (13)$$

$$a_2 + c_3 a_1 - a_1 - c_3 + c_0 b_1 + c_1 b_0 = \beta_2 \quad (14)$$

$$c_3 a_2 - a_2 - c_3 a_1 + c_1 b_1 + c_2 b_0 = \beta_3 \quad (15)$$

$$c_2 b_1 - c_3 a_2 = \beta_4 \quad (16)$$

Teniendo en cuenta lo anterior se plantea en términos de matrices el sistema para encontrar los valores de C_i .

$$\begin{bmatrix} b_0 & 0 & 0 & 1 \\ b_1 & b_0 & 0 & a_1 - 1 \\ 0 & b_1 & b_0 & a_2 - a_1 \\ 0 & 0 & b_1 & -a_2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \beta_1 + 1 - a_1 \\ \beta_2 - a_2 + a_1 \\ \beta_3 + a_2 \\ \beta_4 \end{bmatrix}$$

Planteado el sistema en términos de matrices se pueden calcular las constantes del controlador mediante

$$C = \text{inv}(A) * B \quad (17)$$

Siendo A la matriz de los coeficientes y B el vector de las constantes.

Finalmente se procede a generar la acción de control, la cual es de la forma:

$$u[k] = -(C_3 - 1)u[k - 1] + C_3 u[k - 2] + C_0 e[k] + C_1 e[k - 1] + C_2 e[k - 2] \quad (18)$$

4. DISEÑO DEL CONTROLADOR ÓPTIMO

Cuando se habla de optimización, se hace referencia a que el vector de control $u(k)$ permite que el índice de desempeño se minimice o se maximice, según sea el caso, para que el sistema trabaje restringiendo alguna de sus variables. Tómese el índice de desempeño como una función cuyo valor se considera una indicación de qué tanto se parece el desempeño del sistema real al desempeño del sistema deseado.

Teniendo en cuenta lo anterior, lo ideal sería optimizar la señal de control $u(k)$ del controlador PID diseñado en el capítulo anterior, sin embargo se decide diseñar el controlador en espacio de estados ya que de esta forma es más viable optimizar el proceso.

Se plantea el sistema de la forma:

$$x(k + 1) = G(k)x(k) + H(k)u(k) \quad (19)$$

$$y(k) = C(k)x(k) + D(k)u(k) \quad (20)$$

Donde:

$$x(k) = \text{vector } n \quad (\text{Vector de estado})$$

$$y(k) = \text{vector } m \quad (\text{Vector de salida})$$

$$u(k) = \text{vector } r \quad (\text{Vector de entrada})$$

$$G(k) = \text{matriz } n \times n \quad (\text{Matriz de estado})$$

$$H(k) = \text{matriz } n \times r \quad (\text{Matriz de entrada})$$

$$C(k) = \text{matriz } m \times n \quad (\text{Matriz de salida})$$

$$D(k) = \text{matriz } m \times r \quad (\text{Matriz de transmisión Directa})$$

La presencia de la variable k en los argumentos de las matrices $G(k)$, $H(k)$, $C(k)$, y $D(k)$ implica que estas matrices varían con el tiempo.

Teniendo en cuenta (19) y (20) se plantea la ecuación de estados y de salida según la forma canónica controlador para el sistema de segundo orden.

$$\begin{bmatrix} \hat{x}1(k+1) \\ \hat{x}2(k+1) \end{bmatrix} = \begin{bmatrix} -a1 & -a2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}1(k) \\ \hat{x}2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) \quad (21)$$

$$y(k) = [b1 \quad b2] \begin{bmatrix} \hat{x}1(k) \\ \hat{x}2(k) \end{bmatrix} \quad (22)$$

Una vez teniendo el sistema en espacio de estados, es relevante abordar los criterios de controlabilidad y observabilidad los cuales fueron introducidos por R.E. Kalman. Tienen un papel importante en el control óptimo de sistemas multivariantes.

4.1 CONTROLABILIDAD

Se dice que un sistema de control es de estado completamente controlable, si es posible transferir el sistema de un estado inicial arbitrario a cualquier estado deseado (también un estado arbitrario), en un período finito. Es decir, un sistema de control es controlable si todas las variables de estado pueden ser controladas en un período finito, mediante alguna señal de control no restringida. Si cualquiera de las variables de estado es independiente de la señal de control, entonces resulta imposible controlar esa variable de estado, por lo tanto el sistema sería no controlable [6].

Para definir si un sistema es controlable o no, se utiliza la matriz de controlabilidad

$$[H : GH : \dots : G^{n-1}H] \quad (\text{Matriz de controlabilidad})$$

Con base en lo anterior, se puede observar que todos los estados que pueden ser alcanzados desde el origen, están abarcados por las columnas de la matriz. Por lo tanto, si el rango de la matriz es n , entonces, para un estado arbitrario $x(nT) = x_f$, existirá una secuencia de señales de control no acotadas $u(0), u(t) \dots, u(n-1)T$.

En conclusión, el sistema es totalmente controlable siempre y cuando:

$$\text{Rango } [H : GH : \dots : G^{n-1}H] \Rightarrow n$$

Entendiéndose el rango de una matriz como el número de líneas de esa matriz (filas o columnas) que son linealmente independientes. Una línea es linealmente independiente de una u otras cuando no se puede establecer una combinación lineal entre ellas.

4.2 OBSERVABILIDAD

La observabilidad se ocupa del problema de determinar el estado de un sistema dinámico a partir de observaciones de los vectores de salida y de control en un número finito de períodos de muestreo. Un sistema es observable si, con el sistema en el estado $x(0)$, se puede determinar el estado a partir de la observación de los vectores de salida y de control a lo largo de un número finito de períodos de muestreo.

Para determinar si un sistema es observable se utiliza el criterio de la matriz de observabilidad, el cual será mostrado a continuación:

$$\begin{bmatrix} C \\ CG \\ \vdots \\ CG^{n-1} \end{bmatrix} \quad (\text{Matriz de observabilidad})$$

Se determina que el sistema es observable si el rango de la matriz es mayor o igual a n .

4.3 CONTROL EN ESPACIO DE ESTADOS

Suponiendo que todas las variables de estado son medibles y disponibles para la realimentación. Se podrá llevar a cabo el control por reubicación de polos en lazo cerrado mediante una matriz de ganancia de realimentación del estado apropiada, siempre y cuando el sistema esté considerado completamente controlable.

Luego de determinar que el sistema es controlable, se procede a realizar un controlador de la forma:

$$u(k) = -Kx(k) \quad (23)$$

Donde:

$u(k)$: Señal de control.

K = Matriz de ganancia de realimentación del estado.

$x(k)$ = Vector de estados.

El sistema en lazo cerrado se muestra en la figura 4.1, el cual tiene como acción de control la ecuación 23.

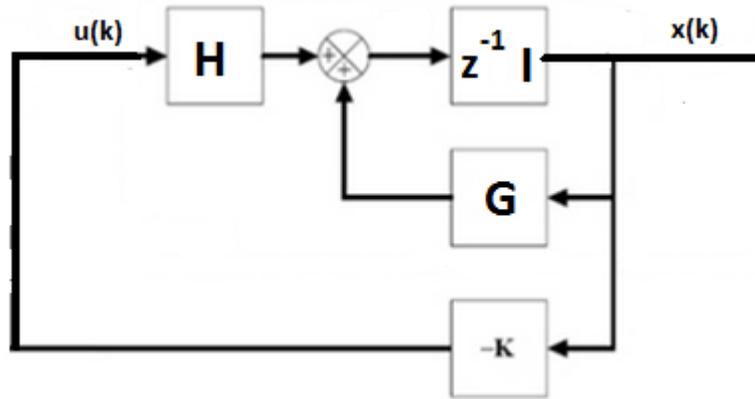


Figura 4.1 Sistema de control en lazo cerrado con $u(k) = -Kx(k)$

Tomando como base la señal de control de la ecuación 23 se decide aplicar al sistema en espacio de estados de la ecuación 21. Sin embargo no todas las variables de estado se encuentran disponibles para medición directa, por lo tanto es necesario estimarlas mediante el observador de estados.

4.3.1 OBSERVADOR DE ESTADOS

Un observador de estados, también conocido como estimador de estados, es un subsistema del sistema de control que lleva a cabo una estimación de las variables de estado, a partir de las mediciones de las variables de salida y control. Aquí el concepto de observabilidad analizado en la sección 4.2 juega un papel muy importante.

Para designar el vector de estados observado se utiliza la notación $\hat{x}(k)$, siendo este vector utilizado posteriormente para generar el vector de control óptimo.

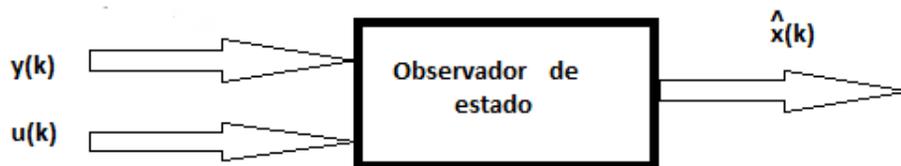


Figura 4.2 Diagrama esquemático del observador de estados

Con el fin de resolver este tipo de casos, un método muy poderoso para la estimación del vector de estado es utilizar un modelo dinámico del sistema original, como sigue:

$$\hat{x}(k + 1) = G\hat{x}(k) + Hu(k) \quad (24)$$

$$\hat{y}(k) = C\hat{x}(k) \quad (25)$$

Donde las matrices G, H y C son las mismas que las del sistema original (19) y (20) respectivamente. Así mismo, se supone que el sistema dinámico está sujeto a la misma señal de control $u(k)$ que el modelo original. Si las condiciones iniciales son iguales tanto para el sistema original como para el estimado, entonces el estado $\hat{x}(k)$ y el estado $x(k)$ serán iguales

Los modelos dinámicos de las ecuaciones 24 y 25 no utilizan la salida medida $y(k)$. El desempeño de este último se puede mejorar si se utiliza la diferencia entre la salida medida $y(k)$ y la salida estimada $C\hat{x}(k)$. Teniendo como base lo anteriormente dicho, el vector de estados estimado $\hat{x}(k + 1)$ queda de la siguiente forma:

$$\hat{x}(k + 1) = G\hat{x}(k) + Hu(k) + L[y(k) - C\hat{x}(k)] \quad (26)$$

Donde L sirve como matriz de ponderación para la diferencia entre la salida real y la estimada, de esta forma se reduce la diferencia entre el modelo dinámico y el real.

4.3.2 FÓRMULA DE ACKERMAN

Para determinar la matriz de ganancia de realimentación de estados L , existe un método llamado fórmula de Ackerman, la cual se puede utilizar siempre y cuando se haya verificado con anterioridad que el sistema sea completamente observable, es decir, que el rango de la matriz de observabilidad sea mayor o igual al número de estados del sistema.

A continuación se muestra la fórmula de Ackerman [6].

$$L = \varphi(G) \begin{bmatrix} C \\ CG \\ \vdots \\ CG^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (27)$$

Donde $\varphi(G)$ es el polinomio característico deseado de la dinámica de error y tiene la siguiente forma:

$$\varphi(G) = G^n + \beta_1 G^{n-1} + \dots + \beta_{n-1} G + \beta_n I \quad (28)$$

Siendo β_n los coeficientes que acompañan al polinomio deseado.

Con base en lo anterior, es posible calcular L y de esta forma determinar el vector de estados estimados de la ecuación 26.

En la figura 4.3 se puede observar la estructura del sistema de control por regulación utilizando el observador de estados

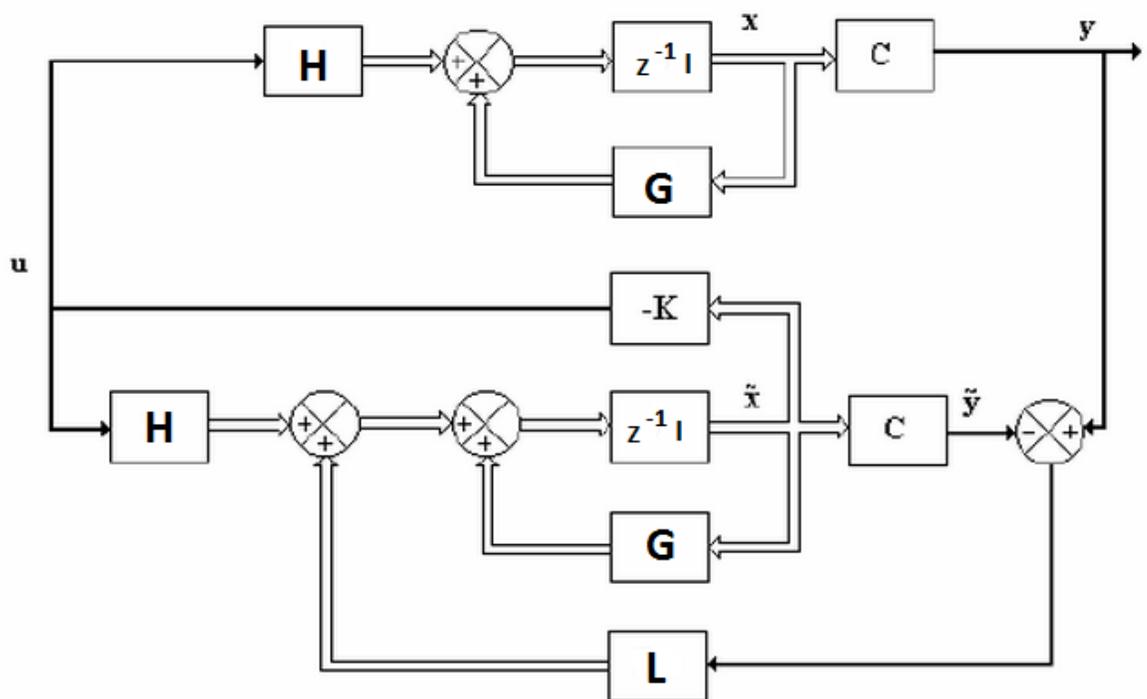


Figura 4.3 Sistema de control regulado mediante la realimentación del estado observado

4.4 CONTROL ÓPTIMO CUADRÁTICO

El problema que logra solucionar el control óptimo es el de saber dónde han de reubicarse los auto-valores del sistema para conseguir unos fines determinados, como puede ser la minimización de una matriz de costo J . Esta función de costo, o índice de desempeño, suele estar relacionada con la energía y por lo tanto con el costo económico del proceso.

Una forma general del índice de desempeño cuadrático es la siguiente:

$$J = \frac{1}{2} \sum_{k=0}^{N-1} x^T(k) Q x(k) + \frac{1}{2} \sum_{k=0}^{N-1} u^T(k) R u(k) \quad (29)$$

Donde:

Q = Matriz de pesos para los estados. $n \times n$

R = Matriz de pesos para las entradas. $q \times 1$

Como se puede observar en la ecuación 29, el vector de señales de entrada $u(k)$ hace parte del índice de comportamiento, debido a que son las señales de entrada las que hacen que el sistema exija un determinado consumo energético, por lo tanto, las que generan un costo.

De forma general, son los coeficientes de las matrices Q y R los que inciden sobre los valores de las señales $x(k)$ y $u(k)$, entre más grandes son los coeficientes de estas matrices, mejor optimizada se ve la variable a la que corresponden.

Finalmente, se considera el sistema de la ecuación 21 para hacer el controlador óptimo adaptativo del lazo de control de la posición angular el cual será por regulación de estados, seleccionando éste ya que la referencia siempre es cero. El índice de desempeño J está dado por la ecuación 29 y la ley de control que minimiza a J está dada por:

$$u(k) = -K \hat{x}(k) \quad (30)$$

Por lo tanto, se observa que hasta el momento ya se tiene el vector de estados estimados calculado mediante la ecuación 26, lo que falta es la ganancia de realimentación óptima de estados que está dada por:

$$K = (R + H'PH)^{-1} H'PG \quad (31)$$

Donde P es la matriz de covarianza óptima del error del estimador de estado, obtenida a partir de la ecuación de Riccati.

$$P = Q + G'PG - G'PH(R + H'PH)^{-1}H'PG \quad (32)$$

4.4.1 DETERMINACIÓN DE LAS MATRICES Q Y R

El cálculo presentado para el controlador LQR no considera ninguna restricción en los valores de las variables de control ni las variables de estado. De manera que cuando por criterio del diseñador se desea introducir restricciones para una o varias variables de estado o de control, sin necesidad de aumentar la complejidad del problema, se pueden tener en cuenta estas restricciones y proponer las matrices Q y R apropiadas para que en ningún caso, se alcancen los límites permitidos de diseño o saturación física de las variables de estado y de control. La determinación de estas matrices puede ser intuitiva en mayor o menor medida dependiendo del control que se quiera sobre cada variable; pero la magnitud de los valores asignados influye decisivamente en la respuesta de las acciones de control y de los estados [7].

Será necesario, por tanto, proponer diferentes valores para las matrices Q y R, simular el comportamiento del sistema y escoger aquel control para el cual las variables de control y de estado se mantengan dentro de los límites permitidos de diseño.

Lo primero que se debe tener en cuenta para determinar las matrices Q y R consiste en determinar la configuración que tendrán dichas matrices, esta configuración depende principalmente del orden del sistema, el cual determinará el tamaño de la matriz Q y de la cantidad de señales de control se determinará el tamaño de la matriz R.

Para este sistema se utilizan unas matrices Q y R con las siguientes configuraciones:

Q diagonal

$$Q = \begin{bmatrix} q_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & q_{ii} \end{bmatrix} \quad (33)$$

R diagonal

$$R = \begin{bmatrix} r_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_{ii} \end{bmatrix} \quad (34)$$

En donde q_{ii} representa el peso que se le da al estado x_i y r_{ii} representa el peso que se le da a la señal de control.

Los valores Q y R deben ser siempre positivos o cero. El conocimiento que se posea del sistema resulta siempre fundamental en la selección de estas matrices. Aun así, es conveniente calcular diferentes señales de control K, con base en distintos valores para las matrices Q y R, y verificar su efectividad mediante simulación.

Una ventaja importante del control LQR es que, independiente al valor de las matrices Q y R, se preserva la estabilidad asintótica y la robustez del controlador.

Para la determinación de las matrices Q y R después de tener en cuenta lo anterior, no existen métodos analíticos simples que ayuden al diseñador a definir los valores que harán parte de dichas matrices, las cuales están en función del sistema, del control que se desee realizar y de los esfuerzos de las variables de control.

De modo que no existe más opción que proponer unas matrices Q y R, calcular el controlador y comprobar el comportamiento del sistema en lazo cerrado mediante simulación. Luego verificando los resultados, se modifican los valores de las matrices, se recalcula el controlador y se simula de nuevo. Este proceso se repite el número de veces necesario hasta llegar al comportamiento de diseño especificado del sistema. En conclusión, se trata de un proceso iterativo, en donde el conocimiento de la planta, por parte del diseñador puede ser decisivo a la hora de reducir el número de iteraciones. El proceso de simulación, efectuado con una herramienta informática (Matlab), es simple pero puede ser extenso y tedioso en términos de tiempo cuando hay que realizar un número elevado de simulaciones.

4.4.2 CONTROL EN ESPACIO DE ESTADOS CON GANANCIA EN LAZO DIRECTO

Para el caso del lazo de control de la posición traslacional, se emplea el control óptimo cuadrático mencionado anteriormente pero se le adiciona al sistema una ganancia en lazo directo, por lo tanto la acción de control en este caso sería de la forma:

$$u(k) = -K\hat{x}(k) + Kg * e(k) \quad (35)$$

Donde Kg se calcula aplicando el teorema del valor final a la ecuación en lazo cerrado, quedando de la siguiente manera:

$$Kg = \frac{1}{C(I-G+HK)^{-1}H} \quad (36)$$

En la figura 4.4, se muestra el sistema de control óptimo adaptativo sobre el sistema en espacio de estados con observador de estados y ganancia en lazo directo.

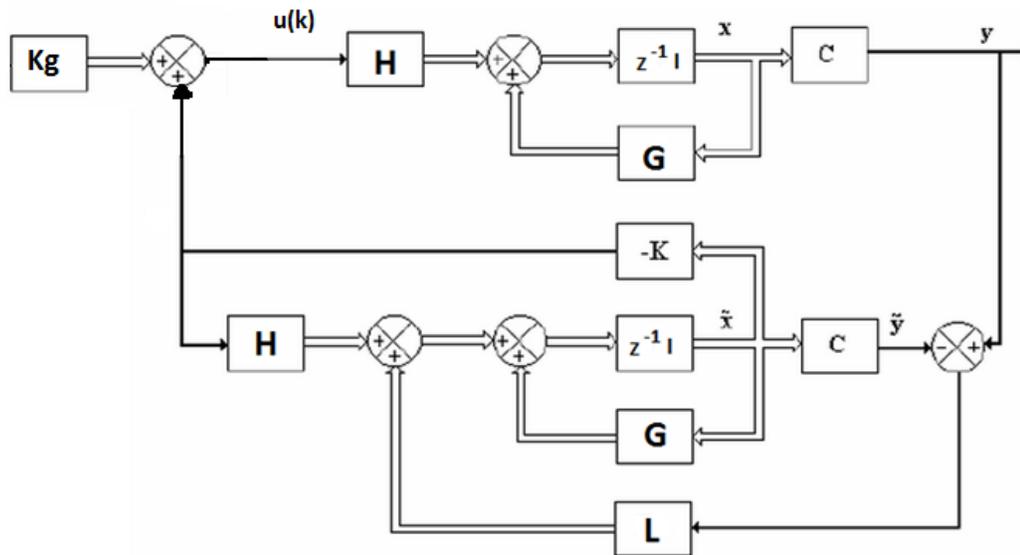


Figura 4.4 Sistema de control en lazo directo mediante la realimentación del estado observado

5. IMPLEMENTACIÓN Y RESULTADOS

5.1 ALGORITMO DE PROYECCIÓN

Se diseña el algoritmo de identificación de la ecuación 4 para un sistema de segundo orden donde:

$$\phi(k-1) = \begin{bmatrix} -y(k-1) \\ -y(k-2) \\ u(k-1) \\ u(k-2) \end{bmatrix} \quad \hat{\theta}(k-1) = \begin{bmatrix} \hat{a}1 \\ \hat{a}2 \\ \hat{b}0 \\ \hat{b}1 \end{bmatrix} \quad c = 0.1$$

Para el algoritmo es necesario tener una entrada u , y como en este caso sólo se pretende identificar se procede a generar una señal de control arbitraria.

$$e(k) = r(k) - y(k) \quad u(k) = 0.5 * e(k)$$

Teniendo claras las variables del algoritmo se procede a simular la identificación del sistema, donde es necesario hacer uso de herramientas del toolbox SimMechanics como son los **joint sensor**, los cuales se conectan al diagrama de bloques para obtener la salida $y(t)$ y también el **joint actuator** para la entrada $u(t)$.

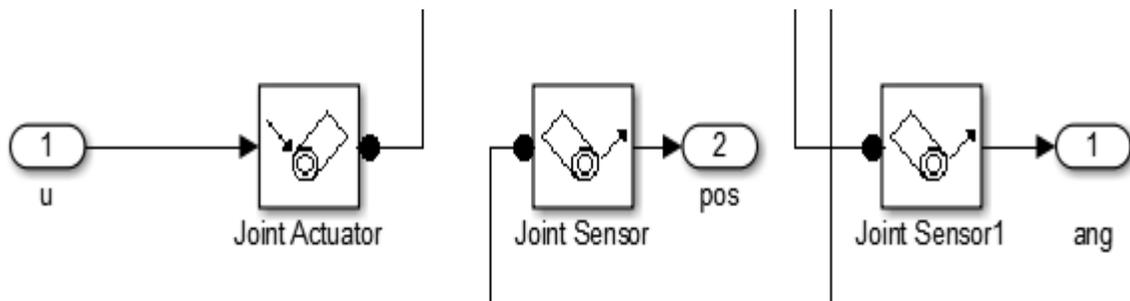


Figura 5.1 Obtención de entradas y salidas del sistema

Una vez diseñado el algoritmo y organizada la interfaz, como es el caso de los visualizadores, se procede a identificar el sistema.

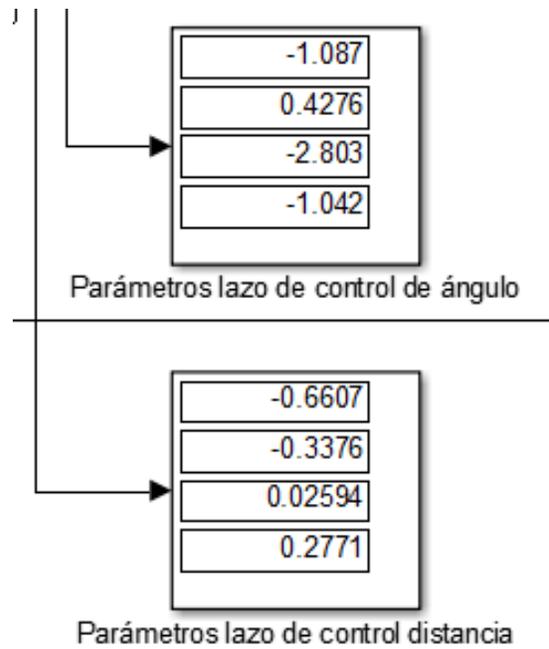


Figura 5.2 Parámetros del sistema mediante proyección

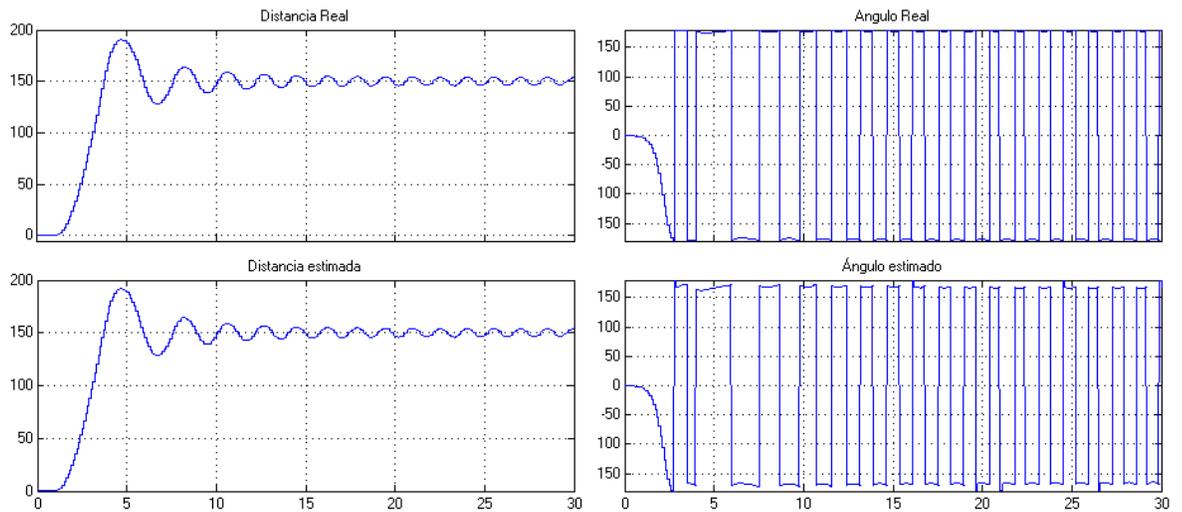


Figura 5.3 Convergencia entre las salidas reales y las estimadas

Se aplicó un controlador proporcional cualquiera ya que la intención es verificar que el algoritmo de proyección converja, lo cual se puede comprobar en la figura 5.3.

5.2 ALGORITMO DE MÍNIMOS CUADRADOS

Básicamente el procedimiento para implementar este algoritmo es igual al anterior, lo único es que para hallar el vector $\hat{\theta}(t)$ es necesario calcular previamente $P(t)$, como lo indican las ecuaciones 5 y 6.

En la figura 5.4 se puede observar en el bloque de visualización de SimuLink, la convergencia entre los valores estimados y los valores reales y en la figura 5.5 los parámetros identificados.

Finalmente se decide identificar el sistema con el algoritmo de mínimos cuadrados ya que converge más rápido y además la matriz $P(t)$, ayuda para que el algoritmo no sea tan sensible al ruido.

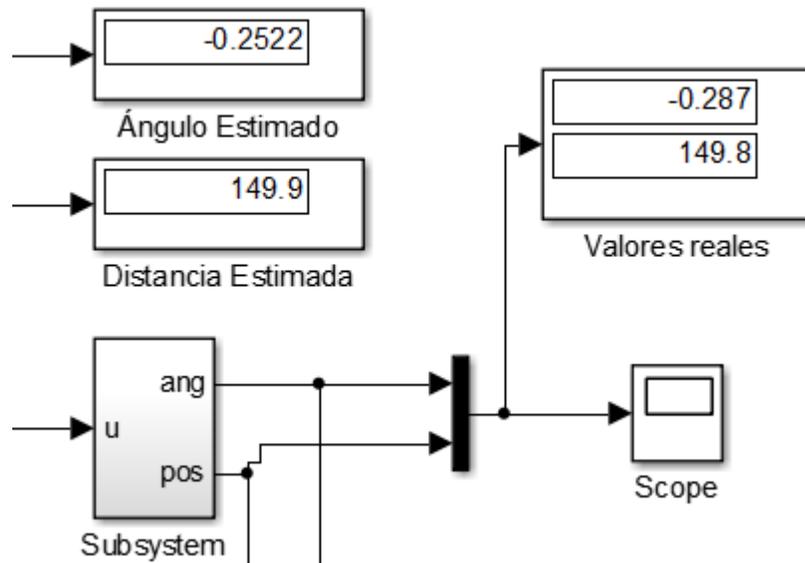


Figura 5.4 Convergencia entre salidas estimadas y reales

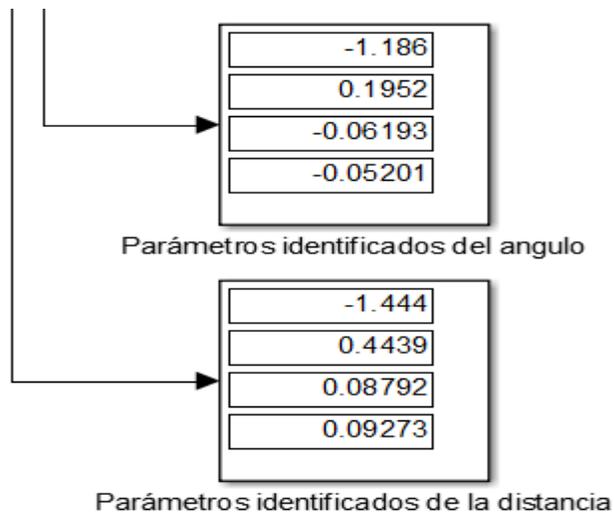


Figura 5.5 Identificación de parámetros mediante mínimos cuadrados

5.3 CONTROL PID ADAPTATIVO

Una vez calculadas las constantes del controlador PID adaptativo mediante el sistema de matrices (17) el cual para el polinomio característico se reubicaron los polos en un valor de 0.5. Se procede a generar la acción de control de la ecuación (18), de esta forma se puede implementar en el software SimuLink, allí se hace uso del bloque llamado *matlab function*, dicho bloque es de gran ayuda, ya que es posible programar la etapa de identificación y de control mediante código, y esto simplifica el programa debido a que las operaciones entre vectores y matrices no necesariamente se hacen con bloques. En la figura 5.6 se muestran los bloques utilizados para la implementación del controlador.

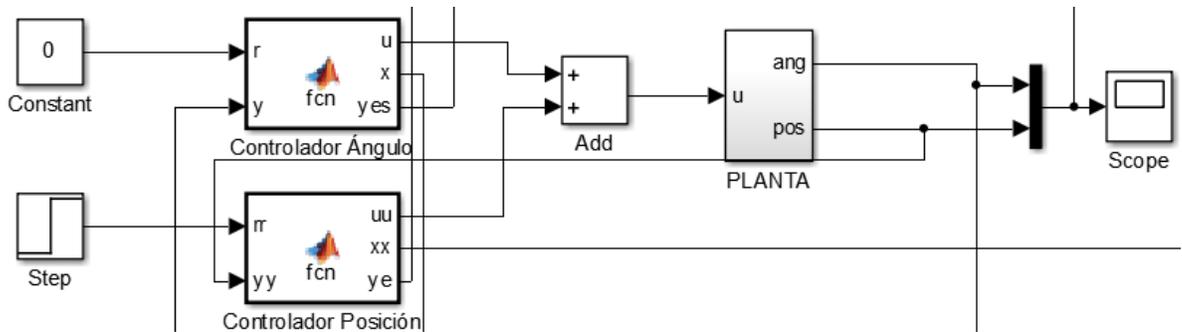


Figura 5.6 Implementación de control PID adaptativo.

Teniendo en cuenta el sistema, la referencia para el ángulo siempre va a ser cero, por lo tanto se colocó una constante, mientras que para la distancia se colocó una función escalón, la cual se varió para probar el sistema.

En la figura 5.7 se muestra la respuesta del sistema con referencias en 0 y -120 para el ángulo y la distancia respectivamente, además en la figura 5.8 se muestra el sistema en la referencia seleccionada.

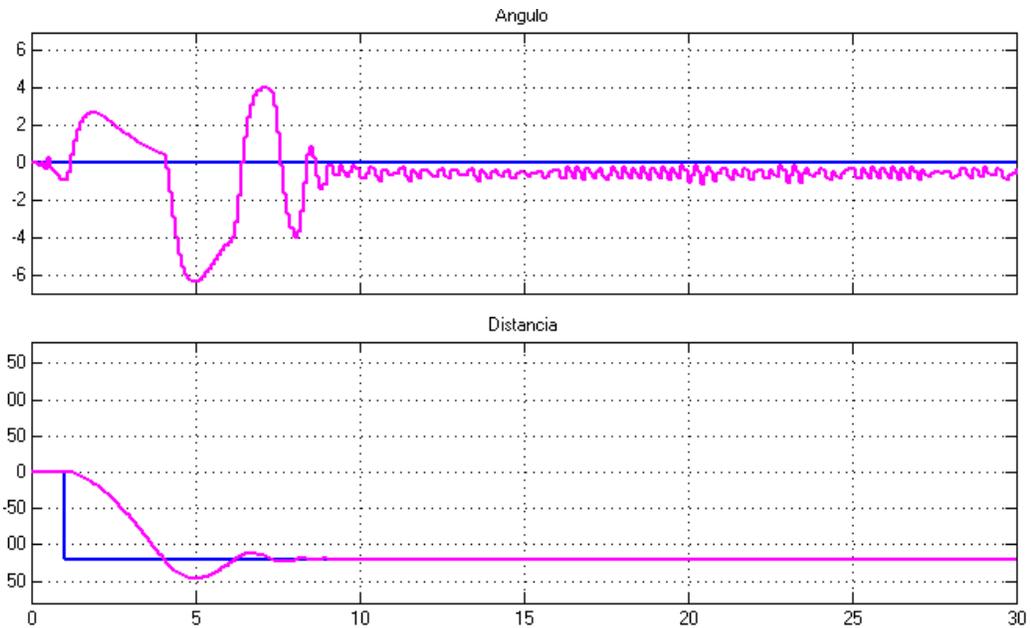


Figura 5.7 Respuesta del sistema

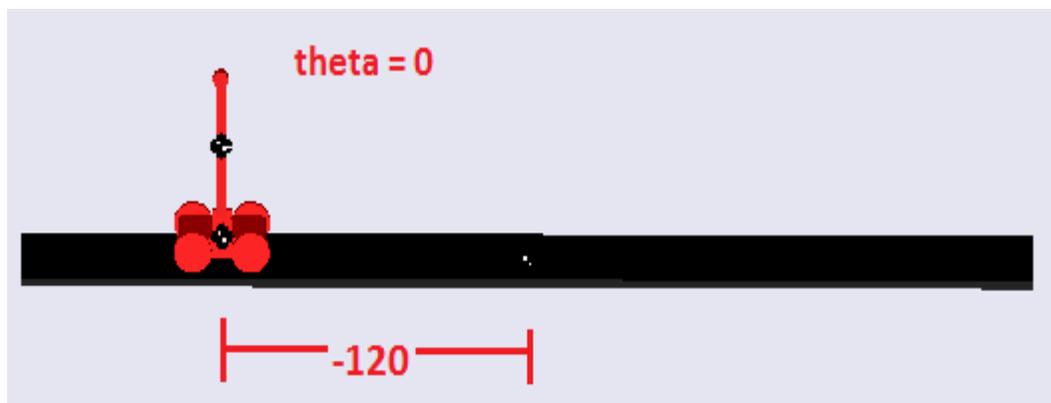


Figura 5.8 Respuesta del sistema simulado

5.4 CONTROL ÓPTIMO CUADRÁTICO

Para definir la señal de control óptima, se muestra el diagrama de bloques de la figura 5.9

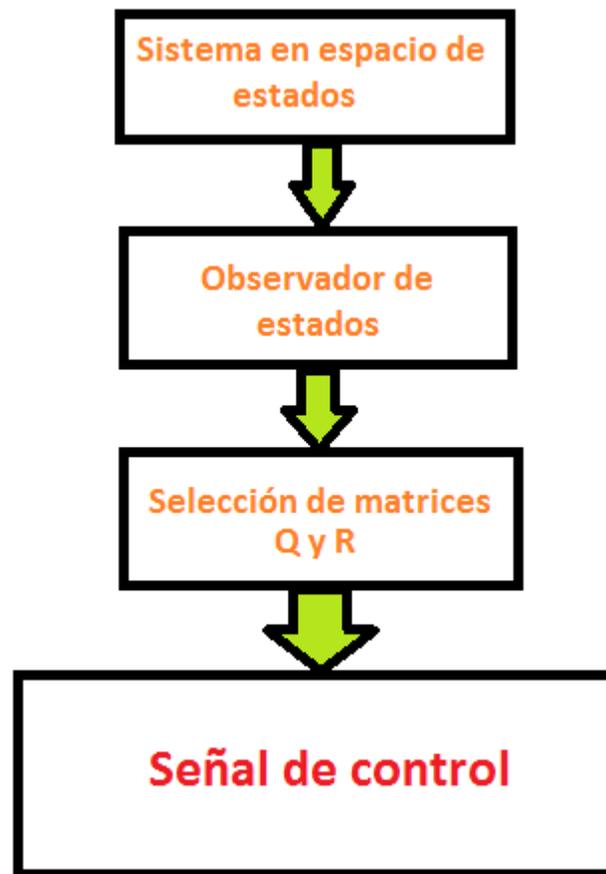


Figura 5.9 Diagrama de bloques control óptimo

Con base en el anterior diagrama de bloques se implementa el control óptimo descrito en el capítulo 4, en el cual se muestran las respuestas a la selección de diferentes matrices de peso Q y R , las cuales actúan sobre la señal de control de las ecuaciones 30 y 35.

Seleccionando las matrices Q y R como identidad, se obtuvo la siguiente respuesta.

Para el ángulo.

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad R = 1$$

Para la distancia.

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad R = 1$$

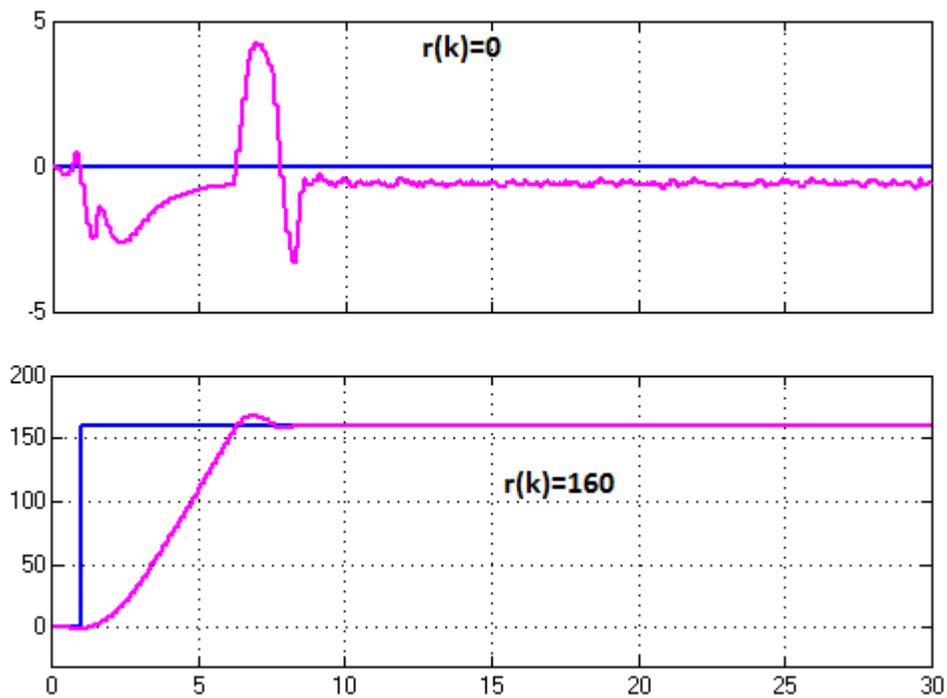


Figura 5.10 Respuesta del sistema con matrices de pesos identidad

Se puede observar en la figura 5.10 que aunque la respuesta es mejor que en el caso del controlador PID de la figura 5.7, se siguen presentando sobre picos, por lo cual se decide modificar los valores de las matrices Q y R llegando a una mejor respuesta como se muestra a continuación.

Para el ángulo.

$$Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix} \quad R = 0.01$$

Para la distancia.

$$Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix} \quad R = 1$$

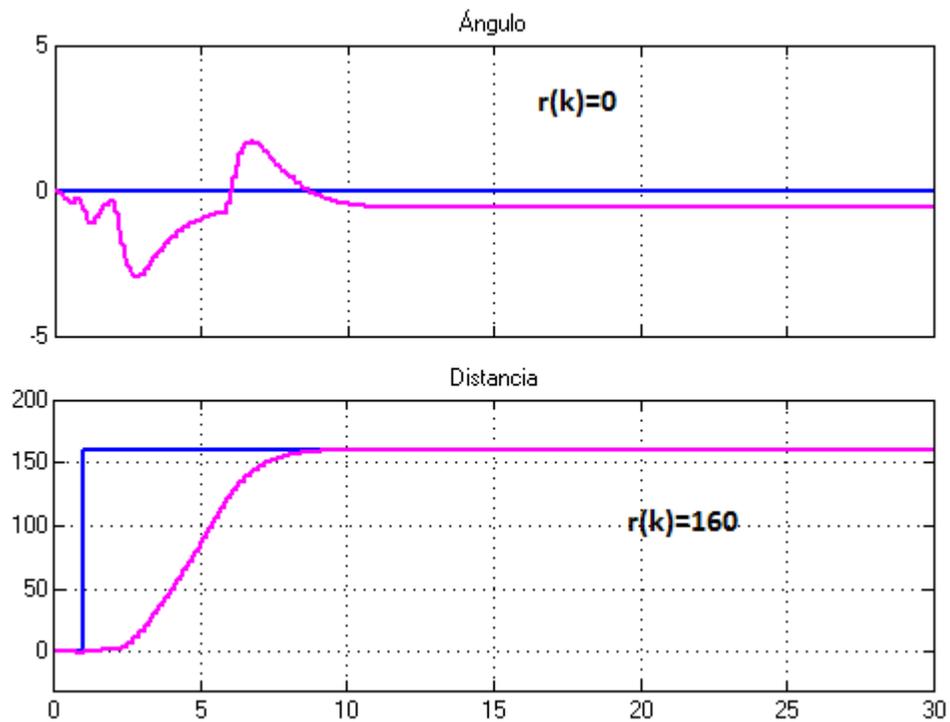


Figura 5.11 Respuesta del sistema con matrices Q y R seleccionadas

Finalmente en la figura 5.11 se observa que con dichas matrices el sistema responde mejor ante las referencias y ya se acentúan mejor las señales con respecto a la figura 5.10.

5.5 IMPLEMENTACIÓN DEL CONTROL ÓPTIMO SOBRE UNA PLANTA FÍSICA

Con la finalidad de probar uno de los algoritmos de control óptimo cuadrático, se decidió construir una planta para el control de ángulo utilizando la herramienta de LEGO MINDSTORMS NXT, la cual, mediante el software LabVIEW de National Instruments permite llevar a cabo el procedimiento.

Primero que todo es necesario tener el toolkit de Lego instalado en el LabVIEW, ya que de lo contrario no sería posible crear la interfaz entre el computador y el sistema físico. En la figura 5.12 se muestra la planta, la cual consta de un servomotor como elemento de acción, y un giroscopio para medir el ángulo, además del ladrillo NXT. Seguidamente en la figura 5.13 se muestra el mismo sistema pero en el software.



Figura 5.12 Planta para control de ángulo construida con NXT

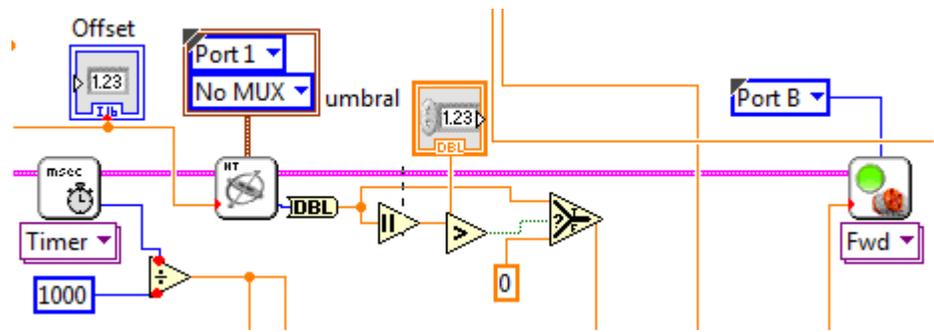


Figura 5.13 Visualización de la planta en el software LabView

Teniendo en cuenta la figura 5.13, para medir el ángulo es necesario integrar la salida del giroscopio ya que éste mide es la velocidad angular, por otro lado es necesario crear un umbral para que la medida no tenga tanto ruido.

Seguidamente se genera un ciclo de tiempo, el cual corre el programa según sea el tiempo de muestreo establecido. Una vez teniendo la interfaz definida se procede a generar el algoritmo de identificación y de control en una herramienta llamada Mathscrip, siendo esta última una ventana donde es posible programar en lenguaje C.

Finalmente se diseña el control en espacio de estados con ganancia en lazo directo, optimizándolo con la función de costo de la ecuación 29 sobre la acción de control de la ecuación 35.

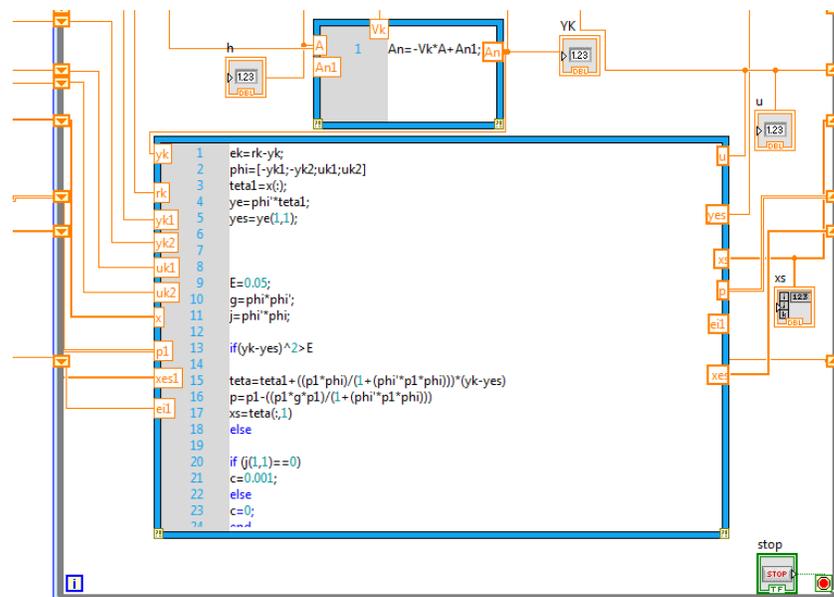


Figura 5.14 Diseño del controlador en LabVIEW

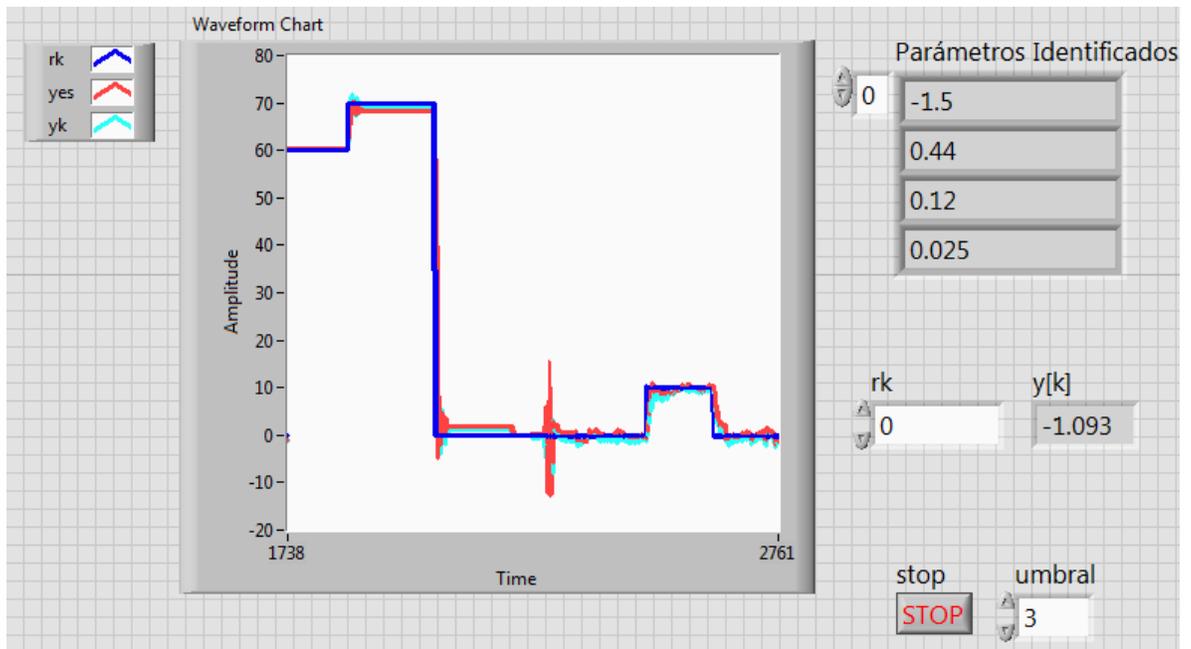


Figura 5.15 Panel frontal en LabVIEW

En la figura 5.15 se puede observar la respuesta de la planta al implementar el control óptimo cuadrático en espacio de estados con ganancia en lazo directo.

CONCLUSIONES

- Se llevó a cabo con éxito el objetivo principal del trabajo el cual consistía en diseñar un controlador óptimo adaptativo para un sistema de péndulo invertido multivariable.
- La identificación de los parámetros del sistema fue fundamental para diseñar el proceso, ya que estos parámetros son los que hacen parte del sistema en espacio de estados.
- A pesar de que el control PID adaptativo que se diseñó en el capítulo 3 llevaba las variables a la referencia, fue necesario representar el sistema en espacio de estados para aplicar la técnica de control óptimo cuadrático.
- Es importante concluir que el control óptimo se puede implementar para cualquier sistema, es decir, no tiene restricción alguna pudiéndose implementar a sistemas lineales, no lineales, variantes e invariantes en el tiempo y multivariables.
- Teniendo en cuenta que el control óptimo cuadrático necesita realimentar las variables de estado, es necesario medir cada una de dichas variables, sin embargo en muchos casos no se poseen los instrumentos de medida o simplemente se hace imposible medirlas, por lo tanto es allí donde se hace necesario recurrir a un observador de estados el cual mediante las salidas y entradas del sistema permite estimar las variables de estado.
- Debido a que no se cuenta con una técnica precisa para determinar las matrices de restricción $Q Y R$ es necesario optar por el método de sintonización prueba y error, que a pesar de ser un poco tedioso permite encontrar valores acertados.
- Se puede concluir que el control óptimo mejoró la respuesta del sistema controlado previamente mediante un PID. Siendo las técnicas de control moderno unas herramientas muy potentes para implementar en cualquier sistema de la industria.
- El tiempo de muestreo es un factor muy importante a la hora de simular e implementar, ya que siendo éste muy grande la acción de control tendrá retardos y el resultado no será el esperado.

RECOMENDACIONES

- Como el diseño del controlador óptimo es la parte más importante para este trabajo, se hace fundamental el seleccionar unas matrices Q y R adecuadas, y es allí donde es determinante conocer el comportamiento del sistema, ya que como no hay restricciones definidas a la hora de escoger los valores de estas matrices, se pueden cometer errores que generen acciones de control y de estado que no mantengan estas variables dentro de los límites de diseño permitidos. De igual forma, es recomendable seleccionar suficientes valores y hacer las respectivas simulaciones, encontrando así aquéllos que entregan las mejores respuestas con respecto a las necesidades y exigencias que son requeridas; teniendo en cuenta la relación entre gasto energético y respuesta del controlador.
- En la implementación de los controladores en el programa de simulación, se presentó un inconveniente con la respuesta del sistema, ya que el péndulo no respondía bien ante las referencias que se le ponían. Este problema era producto de un error en las unidades de la señal de entrada, que teniendo en cuenta las dimensiones con las que fue diseñada la planta, eran desproporcionadas y por lo tanto no había forma de que el controlador mantuviera la planta dentro de los parámetros requeridos. Por lo tanto, se sugiere realizar un diagrama que permita tener presentes estos aspectos, ya que si bien en algunos casos no son fundamentales, en otros pueden determinar que un sistema funcione o no adecuadamente.
- Implementar los controladores de posición angular y posición traslacional sobre una planta física eran una de las ideas a realizar, pero debido a un problema de retardo que se presentaba con la comunicación entre la computadora y el Lego, sólo fue posible implementar el controlador para la posición angular. Se recomienda tener en cuenta el método que se quiere usar para una comunicación entre sistemas, ya que si bien en la simulación ambos controladores funcionaban correctamente, en la planta física no se obtuvieron los mismos resultados.

6. ANEXOS

6.1 CÓDIGO DE MÍNIMOS CUADRADOS EN MATLAB FUNCTION

```
function [u,x,yes] = fcn(r,y)
%#codegen

persistent u1 u2 e1 e2 teta y1 y2 p
if isempty(u1)
    u1=0;
    u2=0;
    e1=0;
    e2=0;
    teta=[1;1];
    y1=0;
    y2=0;
    p=100*eye(4);
end
c=1;
fi=[-y1; -y2;u1; u2];
teta=teta+(fi*p/(c+fi'*p*fi))*(y-fi'*teta);
p=p-p*fi*fi'*p/(1+fi'*p*fi)
x=teta;
yes=(fi'*teta);

e=r-y;
a1=x(1,1);
a2=x(2,1);
b0=x(3,1);
b1=x(4,1);
u = 0.5*e;
if u>=3;
    u=3;
end
if u<=-3;
    u=-3;
end
u2=u1;
u1=u;
e2=e1;
e1=e;
y2=y1;
y1=y;
```

6.2 CÓDIGO DE CONTROL PID ADAPTATIVO

```
function [u,x,yes] = fcn(r,y)
%#codegen

persistent u1 u2 e1 e2 teta y1 y2 p
if isempty(u1)
    u1=0;
    u2=0;
    e1=0;
    e2=0;
    teta=[-1.09;-0.9;1;-0.05];
    y1=0;
    y2=0;
    p=100*eye(4);
end
teta=teta+p*fi/(1+fi'*p*fi)*(y-fi'*teta);
p=p-p*fi*fi'*p/(1+fi'*p*fi);
x=teta;
yes=(fi'*teta);
e=r-y;
    a1=x(1,1);
    a2=x(2,1);
    b0=x(3,1);
    b1=x(4,1);
d1=0.5;
d2=0.5;
d3=0.5;
d4=0.5;

    T=[b0 0 0 1; b1 b0 0 (a1-1);0 b1 b0 (a2-a1); 0 0 b1 -a2 ];
    D=[(d1+1-a1);(d2-a2+a1);(d3+a2);d4];
    C=inv(T)*D;
    C0=C(1,1);
    C1=C(2,1);
    C2=C(3,1);
    C3=C(4,1);

u=( -(C3-1)*u1)+C3*u2+C0*e+C1*e1+C2*e2);

if u>=1;
    u=1;
end
if u<=-1;
    u=-1;
end

u2=u1;
u1=u;
e2=e1;
e1=e;
y2=y1;
y1=y;
```

6.3 CÓDIGO DE CONTROL ÓPTIMO ADAPTATIVO

```
function [u,x,yes] = fcn(r,y)
%#codegen

persistent u1 u2 e1 e2 teta y1 y2 p
if isempty(u1)
    u1=0;
    u2=0;
    e1=0;
    e2=0;
    teta=[-1.09;-0.9;1;-0.05];
    y1=0;
    y2=0;
    p=100*eye(4);
end
teta=teta+p*fi/(1+fi'*p*fi)*(y-fi'*teta);
p=p-p*fi*fi'*p/(1+fi'*p*fi);
x=teta;
yes=(fi'*teta);

e=r-y;
a1=x(1,1);
a2=x(2,1);
b0=x(3,1);
b1=x(4,1);

G=[-a1 -a2; 1 0];
H= [1;0];
C=[b0 b1];
Q=[0.001 0;0 0.001];
R=0.01;

coder.extrinsic('acker','dlqr');
k=dlqr(G,H,Q,R);
%k=acker(G,H,[0.5,0.5]);
L=acker(G',C',[0,0])';
xes1=G*xes1(:)+H*u1+L*(y-(C*xes1(:)));
if (C*F*H)==0;
Kg=1;
else
Kg=1/(C*F*H);
end

u=-k*xes1(:)+Kg*e;
u=u(1,1);

if u>=0.2;
u=0.2;
end
```

```
if u<=-0.2;
    u=-0.2;
end
    u2=u1;
u1=u;
e2=e1;
e1=e;
y2=y1;
y1=y;
```

BIBLIOGRAFÍA

- [1] GIRALDO, Didier, TABARES, Iván. Model reference adaptive control. 11TH ISPE/IEEE/IFAC INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS, FACTORIES OF THE FUTURE: CARS & FOF, 1995.
- [2] GIRALDO, Didier, GIRALDO Eduardo. Teoría de Control Análogo, Primera edición 2010.
- [3] SASTRY, Shankar, BODSON, Mark. Adaptive Control: Stability, Convergence and Robustness. Prentice Hall. USA. 1989.
- [4] GOODWIN G. , SIN, K. S. Adaptive Filtering, Prediction and Control. Prentice Hall, Inc. Englewood, New Jersey., 1984.
- [5] LOZANO, Leonardo, RODRÍGUEZ, Luis, GIRALDO, Didier. Diseño, implementación y validación de un controlador PID auto-sintonizado. Revista Tecno Lógicas. Enero-Junio 2012.
- [6] OGATA, K. Sistemas de control en tiempo discreto. Prentice-Hall, 2 edición, 1996. Capítulo 5, 6 y 8.
- [7] BETANCUR, Carlos, TAFUR, Leonardo. Sintonización de un control óptimo cuadrático con computación evolutiva para un puente grúa. 2008. Capítulo 4.