

Citation:

C. A. Cruz-Villar, J. Álvarez-Gallegos, M. G. Villarreal-Cervantes, "Rediseño paramétrico del Pendubot para posicionamiento vertical en tiempo mínimo", Revista Iberoamericana de Automática e Informática Industrial, Vol. 5, No. 1, pp. 104-114, ISSN: 1697-7912, DOI:10.1016/S1697-7912(08)70128-6, January 2008.

**REDISEÑO PARAMÉTRICO DEL PENDUBOT
PARA POSICIONAMIENTO VERTICAL EN
TIEMPO MÍNIMO**

**Carlos A. Cruz-Villar^{*}, Jaime Alvarez-Gallegos^{*},
Miguel G. Villarreal-Cervantes^{*}**

^{} Departamento de Ingeniería Eléctrica, Sección
Mecatrónica, Cinvestav-IPN, Av. Instituto Politécnico
Nacional, No. 2508, Col. San Pedro Zacatenco, México
D.F. 07360, Apdo. Postal 14-740*

Resumen: Se propone una metodología de rediseño paramétrico de sistemas no lineales para control en tiempo mínimo sin restricciones de trayectoria en los estados y con control acotado. Tal metodología consiste en plantear el problema de rediseño óptimo como uno de optimización dinámica, permitiendo encontrar simultáneamente los parámetros estructurales del sistema y los instantes de conmutación de la señal de control que minimizan el tiempo requerido para llevar al sistema de un estado inicial a uno final. Se muestran resultados experimentales del Pendubot original con control en tiempo mínimo y del Pendubot rediseñado siguiendo la metodología propuesta. *Copyright © 2008 CEA-IFAC*

Palabras clave: Diseño integrado, Rediseño paramétrico, Optimización dinámica, Pendubot, Tiempo mínimo.

1. INTRODUCCIÓN

Una de las principales funciones que un sistema electromecánico desempeña, es la de moverse entre puntos específicos en el espacio de trabajo en forma repetitiva. Esto lo podemos observar en los manipuladores robóticos de las industrias cuando necesitan trasladar una pieza de un lugar a otro. Generalmente estos robots son diseñados de manera convencional para realizar la tarea en el menor tiempo posible. Este método consiste en diseñar primeramente una estructura mecánica que será óptima tanto desde el punto de vista estático como del cinemático (Castano y Will, 2000). Una vez dada la estructura mecánica se diseña un sistema de control que será óptimo desde el punto de vista dinámico (Bryson, 1999; Sadegh y Driessen, 1999) y si en caso de que se desee mejorar aún más el desempeño del robot, los diseñadores primeramente optimizan

nuevamente el controlador por ser lo más fácil de modificar y/u optan por un rediseño de los parámetros de la estructura mecánica (Li, et al., 2001; Zhang, et al., 2001) que podría resultar en un cambio físico completo del sistema, no muy recomendable ya que implica principalmente un costo mayor. Por otra parte, el desempeño óptimo de un sistema electromecánico no sólo depende del diseño del controlador, sino también del diseño de su estructura mecánica. Es así, que se debe considerar una estrategia de diseño que integre tanto el diseño de la estructura mecánica como del sistema de control, de tal manera que las interacciones y los compromisos entre el sistema mecánico y el controlador sean simultáneamente considerados. La filosofía de esta estrategia es la llamada "*metodología de diseño mecatrónico*" (Fu y Mills, 2005; Park y Asada, 1994) cuya idea fundamental es crear un diseño integrado que haga posible el diseño

simultáneo de la estructura mecánica y del controlador, con el propósito de lograr un desempeño óptimo del sistema electromecánico. Sin embargo se presenta el inconveniente de que no encuentra solución cuando el sistema es no lineal.

En este artículo se propone una estrategia de rediseño paramétrico para un sistema no lineal, que mejora el desempeño en el tiempo de posicionamiento (tiempo que se tarda un sistema electromecánico en ir de un lugar a otro). Esta nueva estrategia de rediseño paramétrico permite encontrar los parámetros estructurales del sistema, así como los parámetros de la señal de control en lazo abierto, que minimizan en forma simultánea un criterio de desempeño dinámico (tiempo de posicionamiento). Se puede utilizar cuando el sistema ya está construido y se desea mejorar el tiempo de posicionamiento con pequeñas modificaciones en la estructura física del sistema y con cambios en los parámetros de la señal de control. Sin embargo también se puede aplicar dicha metodología para realizar un diseño completo de la estructura y del controlador.

Para aplicar la metodología de este artículo, se debe considerar que la posible trayectoria para llevar los estados del sistema de la condición inicial a la final, esté libre de obstáculos, es decir que la trayectoria en los estados no presente restricciones. Además se debe de obtener el modelo matemático del comportamiento dinámico del sistema, el cual debe ser suave, y continuamente diferenciable. Si el sistema no presenta las características anteriormente mencionadas, no es posible aplicar dicha metodología ya que no se podrían calcular las sensibilidades del sistema con respecto a las variables de diseño. Otro detalle a considerar es que el modelo del sistema sea lineal o no, debe de presentar un comportamiento aproximado al sistema real.

Para realizar lo metodología de rediseño paramétrico, se propone una modificación de la metodología de control óptimo de Sadegh y Driessen (1999), con el propósito de encontrar, además de la señal de control en lazo abierto, los parámetros estructurales óptimos que mejoren el desempeño en el tiempo de posicionamiento del sistema.

La metodología de rediseño se ejemplifica con un prototipo de laboratorio, el "Pendubot" (Block, 1991), con el propósito de evaluar el desempeño en el tiempo de posicionamiento. El Pendubot es un sistema electromecánico subactuado de dos eslabones rígidos, el primer eslabón está acoplado a un motor y el segundo gira libremente alrededor del extremo del primer eslabón. Desde su introducción en 1991 (Block, 1991), el objetivo

de control ha sido llevar al Pendubot del equilibrio estable o posición de descanso $[q^T, \dot{q}^T] = [-\frac{\pi}{2}, 0, 0, 0]$ al equilibrio inestable o posición de arriba $[q^T, \dot{q}^T] = [\frac{\pi}{2}, 0, 0, 0]$ (ver figura 1) y para esto se han desarrollado a través del tiempo varios algoritmos de control, entre los que se incluyen, control basado en la energía (Fantoni, et al., 2000), control bang-bang sintonizado con un algoritmo genético (Zhao y YI, 2003) y control PD (Block, 1991).

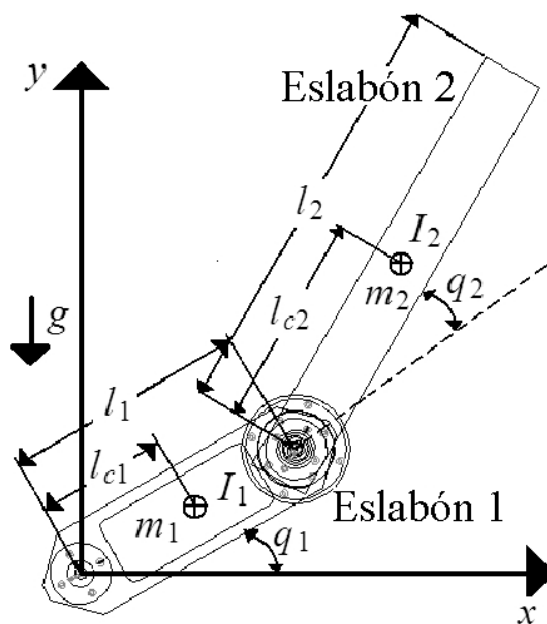


Figura 1. Dibujo esquemático del Pendubot

Para este sistema subactuado, el rediseño consistirá en colocar un peso de forma cilíndrica a cada eslabón con el propósito de no modificar en gran medida su estructura física. Este peso se podrá desplazar a través del eslabón con el propósito de modificar sus parámetros estructurales, es decir, la masa, la distancia del eje de giro con el centro de gravedad y el momento de inercia.

2. REDISEÑO PARAMÉTRICO PARA POSICIONAMIENTO EN TIEMPO MÍNIMO

En esta sección se plantea la metodología de rediseño paramétrico para posicionamiento en tiempo mínimo. Primero se plantea de manera general el problema de rediseño paramétrico, dando pie a la formulación del problema de optimización dinámica. Finalmente se presenta el algoritmo de optimización que encuentra los parámetros estructurales óptimos del sistema, así como a los parámetros óptimos de la señal de control.

2.1 Planteamiento del problema

Considere el problema de encontrar simultáneamente el vector de entrada de la señal de control $u(t) \in R^m$ para todo el tiempo $t_0 \leq t \leq t_f$ y el vector con las variables de diseño estructurales $\Upsilon \in R^i$ del sistema dinámico no lineal invariante en el tiempo (2) que minimizan el funcional $\bar{J} \in R$ (1) y que satisfacen las restricciones (3) - (6),

$$\bar{J} = t_f \quad (1)$$

Sujeto a:

1. El sistema dinámico no lineal invariante en el tiempo en representación de estados,

$$\dot{x} = f(x, u, \Upsilon) \quad (2)$$

2. En los estados iniciales que presenta el sistema,

$$x(t_0) = x_0 \quad (3)$$

3. En los estados finales a los que se quiere llegar,

$$x(t_f) = x_f \quad (4)$$

4. En los límites de la señal de control,

$$u_{\min} \leq u(t) \leq u_{\max} \quad (5)$$

5. En los límites de los parámetros estructurales:

$$\Upsilon_{\min} \leq \Upsilon \leq \Upsilon_{\max} \quad (6)$$

donde $x \in R^n$ es el vector de estado, y $f(x, u, \Upsilon) : R^{n+m+i} \rightarrow R^n$ es una función de clase C^1 . $x_0 \in R^n$ y $x_f \in R^n$ son los estados iniciales y finales de referencia, respectivamente; $x(t_0) \in R^n$ y $x(t_f) \in R^n$ son los estados iniciales y finales en el tiempo inicial t_0 y el tiempo final t_f , respectivamente.

2.2 Formulación del problema de optimización

En (Weinreb y Bryson, 1985) se observó primeramente en resultados de simulación que el control en tiempo mínimo (óptimo) para llevar un sistema de una posición inicial a una posición final sin restricciones en la trayectoria produce una señal de control con un comportamiento tipo bang-bang o conmutado, después esto fue demostrado teóricamente en (Vakhrameev, 1997). Con estos resultados, en varias investigaciones de control en tiempo óptimo como en (Meier y Bryson, 1990) y (Wie y Sunkel, 1990), las condiciones necesarias de optimalidad del problema de control en tiempo mínimo se derivan con el uso del principio del mínimo de Pontryagin (Bryson, 1999) y asumiendo una señal de control del tipo conmutada. Sin embargo éste problema es difícil de resolver numéricamente. Es así que en el planteamiento mostrado en la Sección

2.1, la señal de control se parametriza en intervalos de tiempo de conmutación y ahora estos intervalos son las variables de diseño para la señal de control. Por otra parte, con el propósito de satisfacer las restricciones (6) en los parámetros estructurales Υ , éstas se incorporan al índice de desempeño como funciones de barrera (Bazaraa, 1993). De esta forma, el problema de rediseño paramétrico para el posicionamiento de un sistema electromecánico en tiempo mínimo, se plantea como uno de optimización dinámica que consiste en encontrar los instantes de tiempo de conmutación de la señal de control $\Delta t = [\Delta t_1 \cdots \Delta t_o] \in R^o$ y los parámetros estructurales $\Upsilon = [\Upsilon_1 \cdots \Upsilon_i] \in R^i$ que minimizan el funcional $J : R^{o+i} \rightarrow R$ (7), sujeto a restricciones en el sistema dinámico no lineal invariante en el tiempo (8), en el estado inicial (9) y en el estado final (10) que se desea alcanzar, asumiendo que existe una señal de control conmutado o bang-bang,

$$J(\Delta t, \Upsilon) \equiv t_f + \bar{\Xi} = \sum_{j=1}^o |\Delta t_j| + \sum_{l=1}^i \bar{\Xi}_l \quad (7)$$

Sujeto a las siguientes restricciones:

$$\dot{x} = f(x(t), \Delta t, \Upsilon) \quad (8)$$

$$x(t_0) = x_0 \quad (9)$$

$$x(t_f) = x_f \quad (10)$$

$\bar{\Xi}_l = \omega_{\Upsilon_l} \left[\frac{1}{-(\Upsilon_l - \Upsilon_{l\max})} + \frac{1}{-(\Upsilon_l + \Upsilon_{l\min})} \right] : R \rightarrow R$ es una función de clase C^1 que limita al parámetro Υ_l , a permanecer dentro de su región factible ($\Upsilon_{l\min} \leq \Upsilon_l \leq \Upsilon_{l\max}$), Υ_l es el parámetro l del vector $\Upsilon \in R^i$ donde i es el número de parámetros, $\omega_{\Upsilon_l} \in R$, $x \in R^n$ y $u \in R^m$, $\Delta t \in R^o$ donde "o" es el número de instantes de conmutación.

Sea Q el conjunto de soluciones factibles para el problema (7) sujeto a las restricciones (8)-(10), es decir,

$$Q = \left\{ \Delta t, \Upsilon \mid \dot{x} = f(x, \Delta t, \Upsilon), x(t_0) = x_0, x(t_f) = x_f, \right\} \quad (11)$$

Si $x \in Q$, se dice que $x^* \subseteq x$ es un mínimo global que minimiza el índice de desempeño (7), si la relación $J(x^*) \leq J(x)$ se cumple para todo $x \in Q$. Si existe una vecindad $N_\epsilon(x^*)$ alrededor de x^* tal que $J(x^*) \leq J(x)$ para cada $x \in N_\epsilon(x^*)$, entonces x^* se le llama mínimo local (Bazaraa, 1993).

Así para poder encontrar un mínimo global o local que minimiza J con el algoritmo de optimización, se asume lo siguiente:

1. El conjunto Q es no vacío.
2. Las funciones $f(x, \Delta t, \Upsilon)$ y J son funciones de clase C^1 .

2.3 Algoritmo numérico de optimización

Se define $u(t)$ para $0 = t_0 < t_1 < t_2 < \dots < t_f$ como:

$$u(t) = \begin{cases} u_{\text{máx}} & \text{si } \Delta t_j > 0 \\ u_{\text{mín}} & \text{si } \Delta t_j < 0 \end{cases} \quad (12)$$

El intervalo de conmutación Δt_j se calcula por el método de proyección del gradiente (13). El signo positivo/negativo de Δt_j en (12), indica que la señal de control tendrá un valor fijo de $u_{\text{máx}}/u_{\text{mín}}$ para el intervalo de tiempo $|\Delta t_j|$.

Es importante aclarar que las ecuaciones (7)-(10) están en función implícita o explícita del intervalo de conmutación Δt y haciendo la diferencia con lo ya reportado en Sadegh y Driessen (Sadegh y Driessen, 1999), están también en función de los parámetros estructurales Υ , los cuales modifican en forma simultánea a dichas ecuaciones, y por lo tanto al desempeño del sistema.

Para resolver el problema establecido anteriormente se usa el método de proyección del gradiente (Bazaraa, 1993), el cual encuentra el vector de intervalos de tiempo de conmutación de la señal de control y los parámetros estructurales Υ óptimos que minimizan (7) sujeto a (8)-(10):

$$\begin{bmatrix} \Delta t_{\text{new}} \\ \Upsilon_{\text{new}} \end{bmatrix} = \begin{bmatrix} \Delta t \\ \Upsilon \end{bmatrix} - k_r \check{A}^+ e_f - k_n P \nabla J \quad (13)$$

donde $e_f \equiv x(t_f) - x_f = 0$ es el error en los estados finales, $k_r, k_n \in R^{o+i}$ son matrices diagonales en las cuales $1 \geq k_{r11}, k_{r22}, \dots, k_{r_{o+i, o+i}} \geq k_{n11}, k_{n22}, \dots, k_{n_{o+i, o+i}} > 0$ son las constantes de tamaño de paso, I es una matriz identidad, \check{A} es la matriz Jacobiana de derivadas parciales de $x(t_f)$ con respecto a Δt y Υ , ésto es, $\check{A} = \left[\frac{\partial x(t_f)}{\partial \Delta t}, \frac{\partial x(t_f)}{\partial \Upsilon} \right]$, ∇J es el gradiente de J con

respecto a Δt y Υ , es decir, $\nabla J = \left[\frac{\partial J}{\partial \Delta t}, \frac{\partial J}{\partial \Upsilon} \right]^T$,

$P = (I - A^+ A)$ es la matriz de proyección, \check{A}^+ es la pseudo inversa Moore-Penrose de \check{A} ($\check{A}^+ = \check{A}^T (\check{A} \check{A}^T)^{-1} \check{A}$) cuando \check{A} tiene rango completo $n \leq o+i$, la cual es el caso normal del algoritmo.

Hay que hacer notar que en este artículo, además de encontrar los instantes de conmutación Δt óptimos que minimizan el funcional (7), se encuentran también los parámetros estructurales óptimos que minimizan el mismo criterio.

En la ecuación (13) se observa que $P = (I - \check{A}^T (\check{A} \check{A}^T)^{-1} \check{A})$ es el gradiente proyectado del funcional J sobre la restricción de la matriz A , y $\check{A}^+ e_f$ es la mínima solución para e_f . La teoría de convergencia del método del gradiente proyectado puede ser encontrada en (Bazaraa, 1993). De esta teoría, la convergencia es asegurada con tamaños

de paso suficientemente pequeños tal que la matriz Jacobiana \check{A} presente rango completo. Bajo estas condiciones, los errores proyectados $\check{A}^+ e_f$ y $P \nabla J$ convergen a cero asintóticamente.

La sensibilidad de $x(t_f)$ con respecto a cada componente Δt_j del vector de intervalos de conmutación Δt puede ser obtenida de la siguiente forma (Sadegh y Driessen, 1999):

$$\begin{aligned} \frac{\partial x(t_f)}{\partial \Delta t_j} &= \frac{\partial x(t_f)}{\partial t_j} \frac{\partial t_j}{\partial \Delta t_j} \frac{\partial x_j}{\partial x_j} \\ &= \frac{\partial x(t_f)}{\partial x_j} \dot{x}_j \text{signo}(\Delta t_j) \end{aligned} \quad (14)$$

donde $\text{signo}(\cdot)$ es la función signo, $x_j \equiv x(t_j)$ y \dot{x}_j es el término que está a la derecha de la ecuación (8) evaluada en x_j y Δt_j , así si $A_j(t_j) = \frac{\partial x_j}{\partial x_{j-1}}$ por la regla de la cadena $\frac{\partial x(t_f)}{\partial x_j} = A_o A_{o-1} \dots A_{j+1}$.

Las matrices A_j se obtienen al integrar la primera variación de la ecuación (8), si $A_j(t) = \frac{\partial x(t)}{\partial x_{j-1}}$ para $t_{j-1} \leq t \leq t_j$,

$$\begin{aligned} \frac{\partial}{\partial x_{j-1}} \frac{dx(t)}{\partial t} &= \frac{\partial f(x(t), \Delta t_j, \Upsilon)}{\partial x_{j-1}} \\ \frac{d}{dt} \frac{\partial x(t)}{\partial x_{j-1}} &= \frac{\partial f(x(t), \Delta t_j, \Upsilon)}{\partial x} \frac{\partial x(t)}{\partial x_{j-1}} \end{aligned}$$

$$\frac{dA_j(t)}{dt} = \left[\frac{\partial f(x(t), \Delta t_j, \Upsilon)}{\partial x} \right] A_j(t) \quad (15)$$

con condición inicial $A_j(t_{j-1}) = I$ para cada $t_{j-1} \leq t \leq t_j$, donde I es la matriz identidad, $A_j = A_j(t_j)$ y $x(t)$ es la trayectoria resultante para Δt .

La sensibilidad de $x(t_f)$ con respecto a los parámetros estructurales Υ_l se obtiene al integrar la derivada de la ecuación (8) con respecto a Υ , tomando solamente el último valor, esto es, $\frac{\partial x(t_f)}{\partial \Upsilon_l}$:

$$\begin{aligned} \frac{d \left(\frac{\partial x(t)}{\partial \Upsilon_l} \right)}{dt} &= \frac{\partial f(x(t), \Delta t, \Upsilon)}{\partial x(t)} \left(\frac{\partial x(t)}{\partial \Upsilon_l} \right) \\ &\quad + \frac{\partial f(x(t), \Delta t, \Upsilon)}{\partial \Upsilon_l} \end{aligned} \quad (16)$$

Esta manera de encontrar la sensibilidad del sistema con respecto a una variable se explica detalladamente en (Frank, 1978).

Para el buen funcionamiento del algoritmo del gradiente proyectado, se deben eliminar del vector de intervalos de tiempo de conmutación Δt , aquellos Δt_j que cambien de signo en diferentes iteraciones. Para cada Δt_j que cambie de signo, se debe agregar un intervalo de conmutación Δt_{o+1} de longitud $\epsilon \approx 0$ y opuesto de signo a su intervalo anterior, esto es, $\Delta t_{o+1} = -\epsilon \text{signo}(\Delta t_o)$. Este

nuevo intervalo Δt_{o+1} podrá aumentar su valor si el algoritmo del gradiente proyectado lo requiere.

Por otra parte, un error en el modelo dinámico propuesto producirá que el algoritmo del gradiente proyectado sólo encuentre los parámetros óptimos para dicho modelo pero no encontrará los parámetros óptimos para el sistema real, esto es, para los resultados experimentales. Un parámetro que no es óptimo para el sistema real producirá un error en el estado final al que se desea llegar. Una forma de incrementar la robustez de dicho algoritmo o de mejorar los resultados experimentales, es implementar los parámetros Δt y Υ óptimos de simulación al sistema real. Se guarda la trayectoria de los estados tomados del sistema real $x_m(t)$ y se utiliza para calcular Δt y Υ con (17). Donde c representa el número de pruebas, x_m^c es el estado terminal medido durante la prueba c y A^c es la matriz Jacobiana de A evaluada en x_m^c .

$$\begin{bmatrix} \Delta t_{new}^{c+1} \\ \Upsilon_{new}^{c+1} \end{bmatrix} = \begin{bmatrix} \Delta t^c \\ \Upsilon^c \end{bmatrix} - k_r \check{A}^{c+}(x_m^c(t_f) - x_f) \quad (17)$$

$$\begin{aligned} & -k_r \check{A}^{c+}(x_m^c(t_f) - x_f) \\ & -k_n(I - A^{c+}A^c)\nabla J^c \end{aligned}$$

Una vez que se encontraron las mejoras en Δt_{new}^{c+1} y Υ_{new}^{c+1} , se vuelve a guardar la trayectoria de los estados tomados del sistema real y se repite el proceso antes mencionado.

La solución óptima real es obtenida al realizar varias pruebas hasta que se satisfagan las restricciones terminales.

2.4 Resumen del algoritmo

A continuación se muestra paso a paso el desarrollo del algoritmo propuesto arriba.

Paso 1: Proponer las matrices de tamaño de paso ($0 < k_n \leq k_r < 1$) del método del gradiente proyectado, así como $0 < \omega_{\Upsilon_l} < 1$ y los parámetros estructurales Υ_l para $1 \leq l \leq i$ siempre que se encuentren en la región factible ($\Upsilon_{l\text{mín}} \leq \Upsilon_l \leq \Upsilon_{l\text{máx}}$). Escoger una tolerancia de convergencia (tc) para \check{A}^+e_f que es la mínima solución para $e_f \equiv x(t_f) - x_f$ y otra tolerancia para $P\nabla J$ que es la proyección del gradiente del funcional, nombrándose $tc_{A^+e_f}$ y $tc_{P\nabla J}$, respectivamente. Proponer intervalos de tiempo de conmutación $\Delta t \in R^o$ donde "o" es la dimensión del vector o el número de intervalos de conmutación.

Paso 2: Evaluar $u(t) = u_{\text{máx}}$ o $u(t) = u_{\text{mín}}$ para $0 \leq t \leq t_f$ dependiendo del signo de Δt (12).

Paso 3: Resolver la ecuación del sistema $\dot{x} = f(x(t), \Delta t, \Upsilon)$ mediante algún método de integración numérica (Runge-Kutta) hacia adelante con condiciones iniciales $x(t_0) = x_0$ y $u(t)$ para $0 \leq t \leq t_f$ obtenido del paso 2.

Paso 4: Calcular $\check{A} = \left[\frac{\partial x(t_f)}{\partial \Delta t}, \frac{\partial x(t_f)}{\partial \Upsilon} \right]$ donde

$$\frac{\partial x(t_f)}{\partial \Delta t} = \left[\frac{\partial x(t_f)}{\partial \Delta t_1} \cdots \frac{\partial x(t_f)}{\partial \Delta t_o} \right]$$

$$\frac{\partial x(t_f)}{\partial \Upsilon} = \left[\frac{\partial x(t_f)}{\partial \Upsilon_1} \cdots \frac{\partial x(t_f)}{\partial \Upsilon_i} \right]$$

Para obtener la derivada de $x(t_f)$ con respecto a cada componente Δt_j del vector del intervalo de conmutación Δt se hace lo siguiente:

Paso 4.1: Resolver la ecuación (15) por integración hacia adelante con condiciones iniciales $x(t_{j-1}) = x_{j-1}$, $A_j(t_{j-1}) = I$ y $u(t_j)$ para cada intervalo $\Delta t = [\Delta t_1 \dots \Delta t_o] \in R^o$.

Paso 4.2: Una vez obtenidas todas las $A_j(t)$ para $\Delta t_1, \Delta t_2, \dots, \Delta t_o$, se obtiene $\frac{\partial x(t_f)}{\partial x_j}$ para $1 \leq j \leq o$.

$$\frac{\partial x(t_f)}{\partial x_j} = A_o A_{o-1} \dots A_{j+1}$$

Paso 4.3: Calcular $\frac{\partial x(t_f)}{\partial \Delta t_j}$ de (14).

Paso 4.4: Calcular $\frac{\partial x(t_f)}{\partial \Upsilon}$ de (16)

$$\frac{\partial x(t_f)}{\partial \Upsilon} = \left[\frac{\partial x(t_f)}{\partial \Upsilon_1} \cdots \frac{\partial x(t_f)}{\partial \Upsilon_i} \right]$$

Paso 4.5: Calcular \check{A}

$$\check{A} = \left[\frac{\partial x(t_f)}{\partial \Delta t_1} \cdots \frac{\partial x(t_f)}{\partial \Delta t_o}, \frac{\partial x(t_f)}{\partial \Upsilon_1} \cdots \frac{\partial x(t_f)}{\partial \Upsilon_i} \right]$$

Paso 5: Calcular $(\check{A}^+e_f)_{avg}$ y $(P\nabla J)_{avg}$

$$(\check{A}^+e_f)_{avg} = \sqrt{\frac{1}{(o+i)} \left(-\check{A}^+e_f \right)^T \left(-\check{A}^+e_f \right)}$$

$$(P\nabla J)_{avg} = \sqrt{\frac{1}{(o+i)} \left(-P\nabla J \right)^T \left(-P\nabla J \right)}$$

donde $\check{A}^+ = \check{A}^T (\check{A}\check{A}^T)^{-1}$, $P = (I - \check{A}^+\check{A})$,

$$\nabla J = \left[\frac{\partial t_f}{\partial \Delta t}, \frac{\partial \Xi}{\partial \Upsilon} \right]^T$$

$$\nabla J = \left[\text{signo}(\Delta t_1) \cdots \text{signo}(\Delta t_n), \frac{\partial \Xi}{\partial \Upsilon_1} \cdots \frac{\partial \Xi}{\partial \Upsilon_i} \right].$$

Paso 6: Si $(\check{A}^+e_f)_{avg}$ y $(P\nabla J)_{avg}$ convergen a la tolerancia deseada ($tc_{A^+e_f}$ y $tc_{P\nabla J}$) se concluye el programa.

Paso 7: Si no se cumple el paso 6, se calcula Δt_{new} y Υ_{new} de (13) para mejorar la aproximación a un mínimo local.

Paso 7.1: Se suman los Δt_{new} del mismo signo que estén juntos uno de otro y se eliminan los intervalos de tiempo Δt_{new} que son sumados, una forma de ejemplificarlo es:

$$\begin{aligned}\Delta t_{new} &= \begin{bmatrix} + & - & + & - & - \end{bmatrix}^T \in R^5 \\ \implies \\ \Delta t_{new} &= \begin{bmatrix} + & - & + & (-\Delta t_{new_4} - \Delta t_{new_5}) \end{bmatrix}^T \in R^4\end{aligned}$$

Paso 7.2: Se elimina un intervalo de conmutación Δt_j cuando se reduce, pasa por cero y cambia de signo, si esto sucede, se agrega un intervalo de conmutación $\Delta t_{new_{o+1}} = -\epsilon \text{signo}(\Delta t_{new_o})$ de longitud ϵ cercano a cero y opuesto de signo a su intervalo precedido Δt_o . Una manera de ejemplificarlo es:

$$\begin{aligned}\Delta t &= \begin{bmatrix} +0,1 & -0,07 & +0,05 \end{bmatrix}^T \in R^3 \\ \Delta t_{new} &= \begin{bmatrix} +0,12 & +0,01 & +0,058 \end{bmatrix}^T \in R^3 \\ \implies \\ \Delta t_{new} &= \begin{bmatrix} +0,12 & +0,058 & \Delta t_{new_{o+1}} \end{bmatrix}^T \in R^3\end{aligned}$$

En este caso el Δt_2 que se encuentra en la segunda columna tiene un valor muy cercano a cero y cuando se obtiene Δt_{new_2} cambia de signo, por lo que se elimina y a su vez se agrega un intervalo de conmutación $\Delta t_{new_{o+1}}$.

Paso 7.3: Se regresa al paso 2.

3. DINÁMICA DEL PENDUBOT

El desarrollo de las ecuaciones de movimiento del Pendubot pueden ser encontradas en (Spong y Vidyasagar, 1989). En forma matricial las ecuaciones son:

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \quad (18)$$

en donde:

$$\begin{aligned}D(q) &= \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, \\ G(q) &= [\phi_1 \ \phi_2]^T, \tau = [\tau_1 \ 0]^T\end{aligned}$$

$$D_{11} = \Theta_1 + \Theta_2 + 2\Theta_3 \cos(q_2)$$

$$D_{12} = \Theta_2 + \Theta_3 \cos(q_2)$$

$$D_{21} = D_{12}$$

$$D_{22} = \Theta_2$$

$$C_{11} = -\Theta_3 \dot{q}_2 \sin(q_2)$$

$$C_{12} = -\Theta_3 \sin(q_2)(\dot{q}_1 + \dot{q}_2)$$

$$C_{21} = \Theta_3 \dot{q}_1 \sin(q_2)$$

$$C_{22} = 0$$

$$\phi_1 = \Theta_4 g \cos(q_1) + \Theta_5 g \cos(q_1 + q_2)$$

$$\phi_2 = \Theta_5 g \cos(q_1 + q_2)$$

$$\Theta_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1$$

$$\Theta_2 = m_2 l_{c2}^2 + I_2$$

$$\Theta_3 = m_2 l_1 l_{c2}$$

$$\Theta_4 = m_1 l_{c1} + m_2 l_1$$

$$\Theta_5 = m_2 l_{c2}$$

$\tau = [\tau_1, 0]^T$ es el par externo, $D(q)$ es la matriz simétrica de inercia definida positiva, $C(q, \dot{q})$ es la matriz centrífuga y de Coriolis, $G(q)$ es el vector de fuerzas o pares gravitacionales, $q = [q_1, q_2]^T$, $\dot{q} = [\dot{q}_1, \dot{q}_2]^T$ y $\ddot{q} = [\ddot{q}_1, \ddot{q}_2]^T$ son los ángulos, velocidades angulares y aceleraciones angulares para los eslabones 1 y 2, respectivamente.

Se debe incluir el modelo dinámico del motor al modelo dinámico del Pendubot con el propósito de modelar de la manera más aproximada al sistema, y así, evitar soluciones no óptimas para los resultados experimentales como se explicó en la sección 2. El desarrollo del modelo dinámico del motor se encuentra en (Ogata, 1998) y se muestra en (19), donde L_a es la inductancia de armadura, k_e es la constante de la fuerza contra-electromotriz, V_{in} es el voltaje de entrada de armadura, b_o es el coeficiente de fricción viscosa referido a la flecha, R_a es la resistencia de armadura, J_o es la inercia del motor referida a la flecha del motor, $q_m, \dot{q}_m, \ddot{q}_m$ son las posiciones, velocidades y aceleraciones del rotor respectivamente.

$$\begin{aligned}L_a \dot{i}_a + R_a i_a + k_e \dot{q}_m &= V_{in} \\ J_o \ddot{q}_m + b_o \dot{q}_m &= \tau_1 = k_m i_a\end{aligned} \quad (19)$$

Es importante incluir el modelo del motor al sistema ya que existen constantes de tiempo eléctricas τ_e y mecánicas τ_m en el motor que retardan la conmutación del voltaje de entrada (señal de control).

El par producido por el motor (19) se debe igualar al par requerido por el Pendubot (18), con el propósito de acoplar la dinámica del Pendubot

con la dinámica del motor (19), además $q_m = q_1$, $\dot{q}_m = \dot{q}_1$, $\ddot{q}_m = \ddot{q}_1$. Así, la ecuación de movimiento del Pendubot acoplado con el modelo dinámico del motor (Villarreal, 2005) es la siguiente:

$$\frac{d}{dt} \begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \\ i_a \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \bar{D}(q) \left(\begin{bmatrix} k_m i_a \\ 0 \end{bmatrix} - (C(q, \dot{q}) + \bar{b}_0) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} - G(q) \right) \\ \frac{V_{in} - R_a i_a - k_e \dot{q}_1}{L_a} \end{bmatrix} \quad (20)$$

en donde:

$$\bar{D}(q) = (D(q) + \bar{J}_o)^{-1}, \quad \bar{J}_o = \begin{bmatrix} J_o & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{b}_0 = \begin{bmatrix} b_o & 0 \\ 0 & 0 \end{bmatrix}$$

El modelo dinámico (20) se puede formular como una representación en espacios de estado no lineal $\dot{x} = f(x(t), u(t))$ donde $x = (q_1, q_2, \dot{q}_1, \dot{q}_2, i_a)$ es el vector de estado, i_a es la corriente de armadura del motor y $u(t) = [V_{in}, 0]^T$ es el vector de la señal de control (voltaje de entrada).

En la figura 1 se muestra el diagrama esquemático del Pendubot donde $m_1, m_2, l_1, l_2, l_{c1}, l_{c2}, I_1, I_2$ son las masas, las longitudes, las longitudes del centro de masa medido a partir del eje de rotación y los momentos de inercia de los eslabones 1 y 2 respectivamente.

Para implementar la metodología de rediseño paramétrico se deberán modificar los parámetros estructurales del Pendubot (m, l_c, I), para esto, es necesario implementar un sistema de contrapesos al Pendubot que modifiquen dichos parámetros. Este sistema de contrapesos se muestra en la figura 2 y consiste en adicionar un peso de forma cilíndrica a cada eslabón, por lo que ahora las ecuaciones de movimiento en representación de estados $\dot{x} = f(x(t), u(t), \Upsilon)$ están en función del vector de estados $x(t)$, de la entrada de control $u(t)$ y de los parámetros estructurales del Pendubot con el sistema de contrapeso. A su vez, los parámetros estructurales del Pendubot con el sistema de contrapesos están en función de la suma de los parámetros constantes del Pendubot ($m_1, m_2, l_{c1}, l_{c2}, I_1, I_2$) más los parámetros variables del sistema de contrapesos $\Upsilon = \{m_{p1}, l_{cp1}, m_{p2}, l_{cp2}\} \in R^4$, donde $m_{p1}, l_{cp1}, m_{p2}, l_{cp2}$ son la masa y la longitud del centro de masa del peso en forma cilíndrica medido a partir del eje de rotación como se muestra en la figura 2.

En (Villarreal, 2005) se desarrollan los parámetros estructurales del Pendubot con el sistema de contrapesos ($\tilde{m}_1, \tilde{l}_{c1}, \tilde{I}_1, \tilde{m}_2, \tilde{l}_{c2}, \tilde{I}_2$), mostrándose a continuación los resultados obtenidos:

Eslabón 1 con el sistema de contrapeso:

$$\begin{aligned} \tilde{m}_1 &= 0,8435Kg + m_{p1} \\ \tilde{l}_{c1} &= 0,1573m + \frac{(l_{cp1})(m_{p1})}{\tilde{m}_1} \\ \tilde{I}_1 &= 0,0057Kg m^2 + I_{z_{p1}} \end{aligned} \quad (21)$$

Eslabón 2 con el sistema de contrapeso:

$$\begin{aligned} \tilde{m}_2 &= 0,3859Kg + m_{p2} \\ \tilde{l}_{c2} &= 0,1416m + \frac{(l_{cp2})(m_{p2})}{\tilde{m}_2} \\ \tilde{I}_2 &= 0,00539Kg m^2 + I_{z_{p2}} \end{aligned} \quad (22)$$

En las ecuaciones (21) y (22), $r_{p1} = r_{p2} = 0,0254m$ son los radios de los pesos cilíndricos 1 y 2 respectivamente, $I_{z_{p1}} = \frac{1}{2}m_{p1}(r_{p1})^2 + (m_{p1})(l_{cp1} - \tilde{l}_{c1})^2$ y $I_{z_{p2}} = \frac{1}{2}m_{p2}(r_{p2})^2 + m_{p2}(l_{cp2} - \tilde{l}_{c2})^2$ son los momentos de inercia de los pesos cilíndricos 1 y 2 con respecto al centro de masa de los eslabones 1 y 2 respectivamente.

El rango en el que se pueden variar los parámetros del sistema de contrapeso se define con base en las dimensiones de la estructura del Pendubot y a que sean físicamente realizables. Dichos rangos se muestran en (23)-(26).

$$m_{p1 \text{ mín}} = 0Kg \leq m_{p1} \leq 0,2Kg = m_{p1 \text{ máx}} \quad (23)$$

$$l_{cp1 \text{ mín}} = -0,2m \leq l_{cp1} \leq 0,127m = l_{cp1 \text{ máx}} \quad (24)$$

$$m_{p2 \text{ mín}} = 0Kg \leq m_{p2} \leq 0,135Kg = m_{p2 \text{ máx}} \quad (25)$$

$$l_{cp2 \text{ mín}} = 0,0381m \leq l_{cp2} \leq 0,3048m = l_{cp2 \text{ máx}} \quad (26)$$

4. RESULTADOS DE SIMULACIÓN Y EXPERIMENTALES

Aplicando la metodología de la sección 2 al problema de balanceo del Pendubot en tiempo

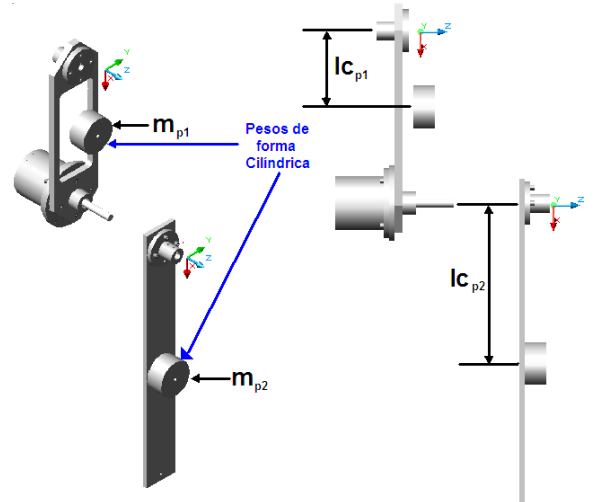


Figura 2. Sistema de contrapesos del Pendubot

mínimo, en donde se desea encontrar los instantes de conmutación $\Delta t = [\Delta t_1 \dots \Delta t_o] \in R^o$, y los parámetros estructurales del Pendubot con el sistema de contrapesos $\Upsilon = [m_{p1}, m_{p2}, lc_{p1}, lc_{p2}]^T \in R^{i=4}$ que minimizan en forma simultánea el índice de desempeño (7), sujeto a las restricciones impuestas por las ecuaciones de movimiento del Pendubot $\dot{x} = f(x(t), \Delta t, \Upsilon)$, a las restricciones de estado inicial (9) donde $x_0 = (-\frac{\pi}{2}, 0, 0, 0)$ es la posición de descanso del Pendubot, y a las restricciones de estado final (10) donde $x_f = (\frac{\pi}{2}, 0, 0, 0)$ es la posición vertical del Pendubot al cual se desea llegar (ver figura 1), asumiendo que existe una señal de voltaje $-45volts \leq u(t) \leq 45volts$ de control conmutado (bang-bang).

En (7), $\bar{\Xi} = \sum_{l=1}^4 \frac{\Xi}{l}$ es el conjunto de funciones de barrera de los parámetros estructurales, y se representan de la siguiente manera:

$$\begin{aligned} \bar{\Xi} = & \omega_{\Upsilon_{m_{p1}}} \left[\frac{1}{-(m_{p1} - m_{p1 \text{ máx}})} + \frac{1}{-(-m_{p1} + m_{p1 \text{ mín}})} \right] \\ & + \omega_{\Upsilon_{m_{p2}}} \left[\frac{1}{-(m_{p2} - m_{p2 \text{ máx}})} + \frac{1}{-(-m_{p2} + m_{p2 \text{ mín}})} \right] \\ & + \omega_{\Upsilon_{lc_{p1}}} \left[\frac{1}{-(lc_{p1} - lc_{p1 \text{ máx}})} + \frac{1}{-(-lc_{p1} + lc_{p1 \text{ mín}})} \right] \\ & + \omega_{\Upsilon_{lc_{p2}}} \left[\frac{1}{-(lc_{p2} - lc_{p2 \text{ máx}})} + \frac{1}{-(-lc_{p2} + lc_{p2 \text{ mín}})} \right] \end{aligned}$$

donde los límites de los parámetros se encuentran en ec's (23-26).

Uno de los principales factores para que el algoritmo no converge a su mínimo local es el de la selección del intervalo de conmutación Δt . Con base en la experiencia adquirida se puede decir que para un Δt grande se necesita de $k_{n_{\Delta t}}, k_{r_{\Delta t}}$ muy pequeñas aunque no en todos los casos se converge al mínimo, por tal motivo es preferible encontrar un Δt cercano al mínimo local cuyas ganancias $k_{n_{\Delta t}}, k_{r_{\Delta t}}$ pueden ser moderadamente pequeñas. Además se tiene que seleccionar un buen ω_{Υ_l} para que la función barrera limite el paso hacia la región no factible del parámetro Υ_l . La selección de ω_{Υ_l} está en relación con los tamaños de paso de los parámetros k_{n_l}, k_{r_l} , estos se deben proponer a prueba y error, observando la tendencia hacia donde se esta moviendo el parámetro y verificando que no se salga de la región factible ($\Upsilon_{\text{mín}} \leq \Upsilon \leq \Upsilon_{\text{máx}}$). De esta forma las condiciones iniciales para este problema son:

$$\begin{aligned} m_{p1ini} &= 0,10Kg, \quad lc_{p1ini} = 0,05m \\ m_{p2ini} &= 0,034Kg, \quad lc_{p2ini} = 0,1m \\ \Delta t_{ini} &= [-0,08 \quad 0,65 \quad -0,4 \quad 0,1 \quad -0,05]^T \end{aligned} \quad (27)$$

con $\omega_{\Upsilon_{m_{p1}}} = \omega_{\Upsilon_{lc_{p1}}} = \omega_{\Upsilon_{m_{p2}}} = \omega_{\Upsilon_{lc_{p2}}} = 5E - 8$, $k_{r_{11}}, k_{r_{22}}, \dots, k_{r_{5,5}} = 1E - 3$, $k_{n_{11}}, k_{n_{22}}, \dots, k_{n_{o,o}} =$

$$\begin{aligned} 1E - 4, \quad k_{r_{o+1,o+1}} &= 3E - 4, \quad k_{r_{o+2,o+2}} = 3E - 4, \\ k_{r_{o+3,o+3}} &= 5E - 4, \quad k_{r_{o+4,o+4}} = 5E - 4, \quad k_{n_{o+1,o+1}} = \\ 1E - 4, \quad k_{n_{o+2,o+2}} &= 1E - 4, \quad k_{n_{o+3,o+3}} = 1E - 4, \\ k_{n_{o+4,o+4}} &= 1E - 4. \end{aligned}$$

La condición inicial Δt_{ini} se calcula con base en la metodología de control en tiempo mínimo de (Sadegh y Driessen, 1999) con los parámetros Υ_{ini} constantes.

Los parámetros óptimos en simulación para el problema de rediseño paramétrico del Pendubot para posicionamiento vertical en tiempo mínimo se muestran a continuación:

$$t_f^* = \sum_1^5 |\Delta t^*| = 1,09s \quad (28)$$

$$\Delta t^* = [-0,029 \quad 0,614 \quad -0,356 \quad 0,088 \quad -0,007]$$

$$\begin{aligned} m_{p1}^* &= 0,1990Kg, \quad lc_{p1}^* = -0,1425m \\ m_{p2}^* &= 3E - 4Kg, \quad lc_{p2}^* = 0,1532m \end{aligned} \quad (29)$$

Se observa en (29) que los parámetros estructurales que afectan el tiempo de posicionamiento son la masa y la distancia del eje de giro del eslabón 1 al centro de masa del peso en forma cilíndrica del primer eslabón.

Una vez obtenidos los parámetros óptimos en simulación, se implementan dichos parámetros al sistema real (Pendubot con el sistema de contrapesos). Usando la trayectoria de los estados tomados del sistema real se procede a calcular nuevamente el algoritmo del gradiente proyectado (17) como se explica en la sección 2. Esto se repite hasta que se encuentren los parámetros óptimos para los resultados experimentales, esto es, cuando el error en los estados sea mínimo.

Para este caso, se encuentra una mejora en el vector del tiempo de conmutación Δt_{new}^{c+1} y no para los parámetros estructurales Υ_{new}^{c+1} ; esto se debe a que la mejora o los cambios en dichos parámetros son de alrededor de 0,1%, haciendo difícil su implementación práctica.

Para este caso la solución óptima experimental es:

$$t_f^{c*} = \sum_1^5 |\Delta t^{c*}| = 0,955s \quad (30)$$

$$\Delta t^{c*} = \sum_1^5 |[-0,07 \quad 0,58 \quad -0,245 \quad 0,06 \quad 0]|$$

con los parámetros estructurales óptimos de (29).

Una vez que el control en lazo abierto se aplica al motor del Pendubot y éste alcanza una vecindad de la posición final deseada, es necesario cambiar la ley de control bang-bang

de lazo abierto por una ley de control en lazo cerrado que estabilice o mantenga los eslabones del Pendubot en la posición de arriba ($x_0 = [\frac{\pi}{2}, 0, 0, 0, 0]$), ya que la metodología integrada que se propone en este artículo lleva al sistema de un lugar a otro en tiempo mínimo y no se garantiza que permanezca en la posición final en la presencia de perturbaciones o fuerzas externas como la fuerza de gravedad. Se implementó un regulador cuadrático lineal (LQR por sus siglas en inglés) con ganancias $K_{p-m} = [-403,533 \quad -421,792 \quad -91,846 \quad -64,012 \quad 2,93]$ para dicha estabilización. A la vecindad de la posición final deseada la llamaremos en éste artículo como región de atracción del LQR y ésta a su vez indica el momento en el cual la señal de control conmuta de una ley de control en lazo abierto a una ley de control en lazo cerrado, y está definida cuando: $|e_{x_1}| \leq 0,13963rad$, $|e_{x_2}| \leq 0,10472rad$, $|e_{x_3}| \leq 4\frac{rad}{s}$ y $|e_{x_4}| \leq 8\frac{rad}{s}$, donde $e_x = x(t) - x_f$.

En la figura 3 se muestra el resultado experimental del balanceo del Pendubot con la metodología de rediseño paramétrico para posicionamiento vertical en tiempo mínimo. Una vez que se encuentra el Pendubot en la posición vertical arriba, se intercambia la ley de control de lazo abierto por un LQR. Le toma un tiempo de 0,895s para cambiar la ley de control, el cual es menor que en (30), ésto se debe a que antes de haberse cumplido todos los instantes de conmutación Δt^{c*} de la señal de control, los eslabones ya se encuentran en la región de atracción (Block, 1991) del LQR intercambiándose la ley de control.

Este resultado se compara con los resultados obtenidos con una metodología de control en tiempo mínimo en donde se consideran únicamente los parámetros de la señal de control a optimizar (Sadegh y Driessen, 1999).

En la figura 4 se observan los resultados experimentales del balanceo del Pendubot con la metodología de tiempo mínimo propuesta por Sadegh y Driessen. Considerando el mismo LQR con la misma región de atracción, al sistema le toma un tiempo final de 0,915s para llegar a la posición de arriba ($x_f = [\frac{\pi}{2}, 0, 0, 0, 0]$) e intercambiándose en ese instante la ley de control. Esta diferencia en tiempo con respecto a (30) se debe en gran parte a que en la metodología de control en tiempo mínimo se minimiza el tiempo final de posicionamiento (índice de desempeño) modificando solamente los instantes de tiempo de conmutación de la señal de control. Así, al aplicar la metodología de rediseño paramétrico propuesta en este artículo, es posible minimizar el tiempo de posicionamiento modificando en forma simultánea los parámetros estructurales del sistema y los instantes de

tiempo de conmutación de la señal de control, dando como resultado una señal de control y unos parámetros estructurales que mejoran el desempeño en el tiempo de posicionamiento respecto a los resultados obtenidos con una metodología que contempla solamente variaciones en los parámetros del controlador.

5. CONCLUSIONES

Se planteó una metodología de rediseño paramétrico para posicionamiento en tiempo mínimo y se aplicó al problema de balanceo del Pendubot. Esta metodología de rediseño hace que los parámetros estructurales, en conjunto con los instantes de tiempo de conmutación de la señal de control, minimicen en forma simultánea un índice de desempeño, que en este caso es el tiempo de posicionamiento. La señal de control y los parámetros estructurales óptimos son encontrados numéricamente con base en el algoritmo del gradiente proyectado.

Se compara la metodología integrada propuesta aquí con una metodología de control en tiempo mínimo. Se mejora un 2,18% el tiempo de posicionamiento con dicha metodología integrada, esto se debe a que además de contemplar los instantes de tiempo de conmutación de la señal de control se involucran los parámetros estructurales (diseño mecatrónico) que modifican simultáneamente el tiempo de posicionamiento. Si se considera que el balanceo del pendubot es una tarea repetitiva, una ganancia en el tiempo de posicionamiento del 2,18% por ciclo, que involucra una disminución de 0,02s, significaría una ganancia en tiempo de 72s por hora, lo que resultaría en un incremento en la productividad. De esta forma, para sistemas electromecánicos que trabajan en forma repetitiva y que responden lentamente, una ganancia del 2,18% es muy favorable. Así los beneficios que ofrece la metodología que se propone en este artículo es el de mejorar el tiempo de posicionamiento de un sistema electromecánico ya definido en su estructura física (longitud, peso, forma, etc.), al realizar pequeños cambios en su estructura y un cambio en los parámetros de la señal de control, sin tener la necesidad de un cambio completo del sistema. Además, esta misma metodología de diseño se puede aplicar a sistemas electromecánicos que están en la fase inicial de su diseño, con tan solo proporcionar la forma del sistema.

Para el ejemplo del Pendubot que ilustra la metodología integrada de este artículo, se mejoró su desempeño en el tiempo de posicionamiento al agregar un sistema de contrapesos que modifica los valores de los parámetros estructurales del

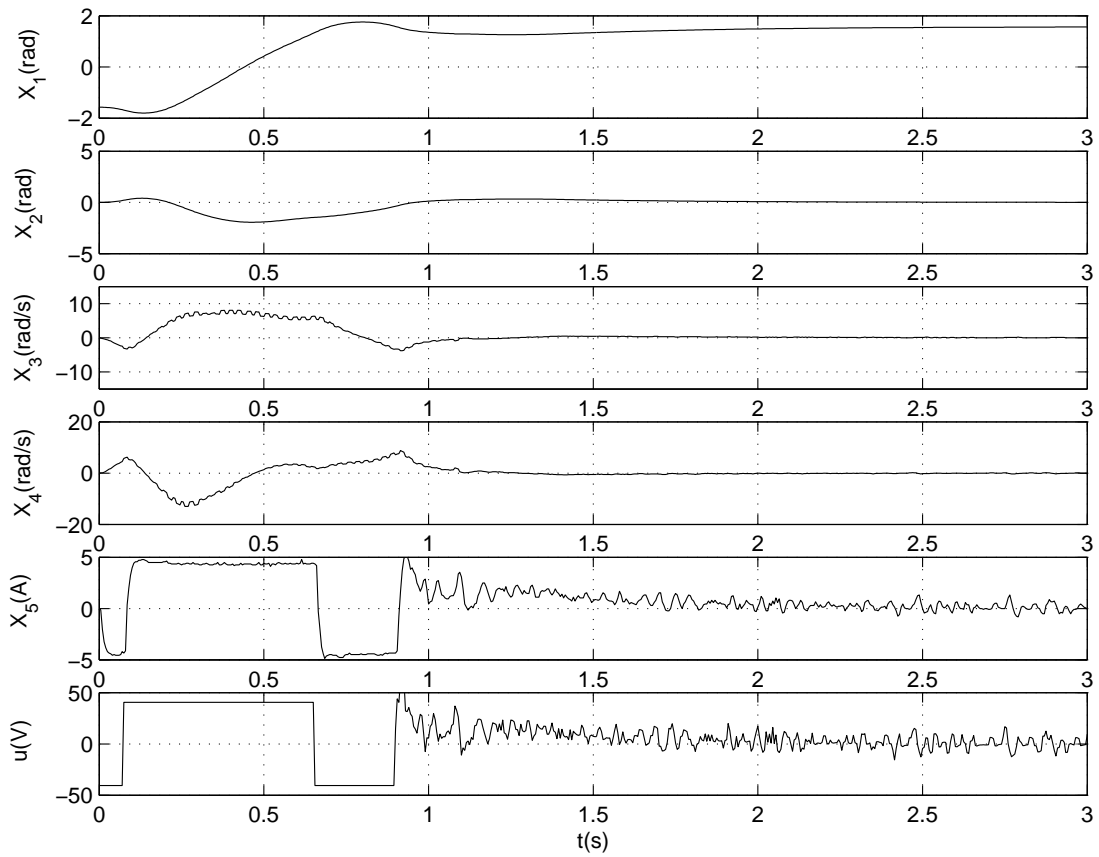


Figura 3. Balanceo del Pendubot con la metodología propuesta

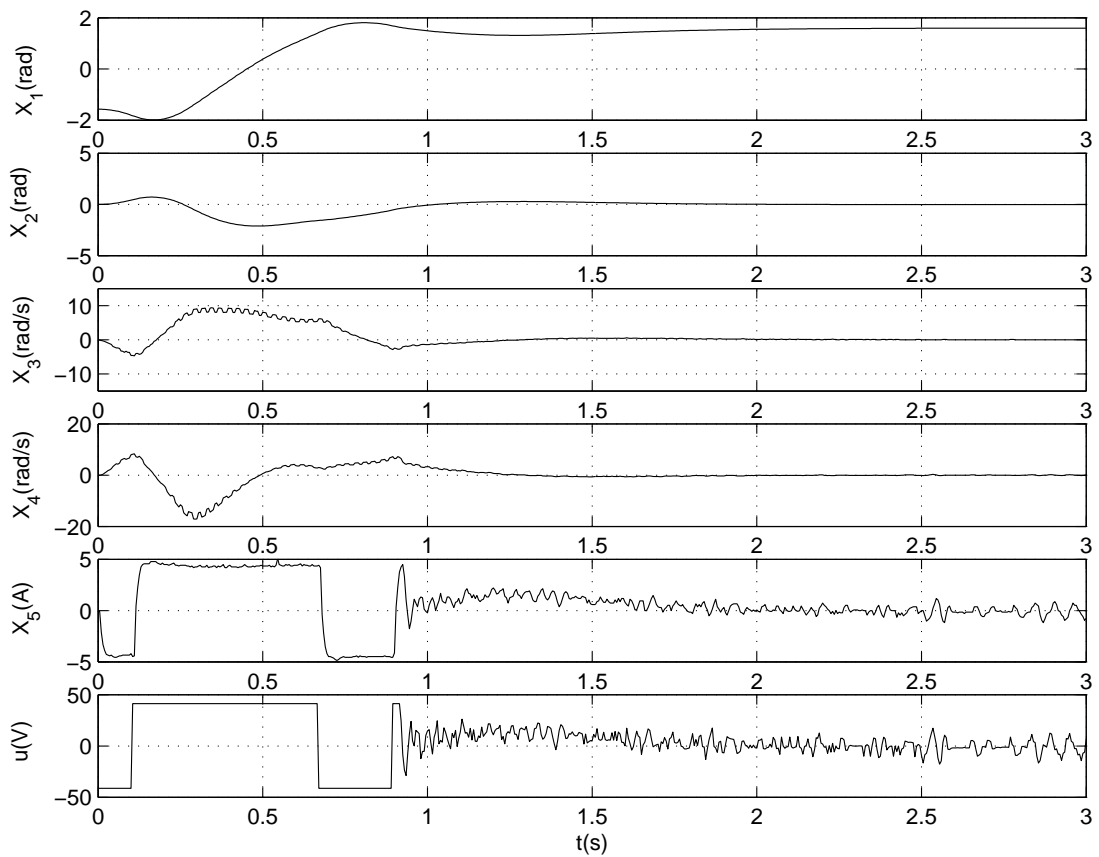


Figura 4. Balanceo del Pendubot con la metodología de control en tiempo mínimo

sistema original y en forma simultánea se realiza un diseño de los parámetros de la señal de control conmutada. Los parámetros estructurales por su parte, alteran las matrices de inercia, de Coriolis y de gravedad del sistema $\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)$ pudiendo, en un momento dado, cancelarse o restarse los términos $D(q)\ddot{q}$, $C(q, \dot{q})\dot{q}$, $G(q)$, produciendo que el par aplicado pueda mover con mas facilidad al sistema electromecánico y de esta manera junto con la señal de control, reducir el tiempo de establecimiento.

Un buen diseño de un sistema electromecánico es aquel que involucra tanto a los parámetros del controlador como a su estructura mecánica como variables a optimizar en forma integrada para minimizar/maximizar uno o varios índices de desempeños (diseño mecatrónico).

REFERENCIAS

- Bazaraa, M. S., Sherali, H. D. and Shetty C. M. (1993). *Nonlinear programming*. John Wiley & Sons, New York Inc.
- Block Jerome, D. (1991). *Mechanical design and control of the Pendubot*. Tesis de Maestría, Departamento de Ingeniería General, Universidad de Illinois.
- Bryson Jr, A. E. (1999). *Dynamic optimization*. Addison Wesley, New York.
- Castano, A. and P. Will (2000). Mechanical design of a module for reconfigurable robots. *IEEE/RSJ International Conference 2000* **3**, 2203–2209.
- Fantoni, I., R. Lozano and M. Spong (2000). Energy based control of the Pendubot. *IEEE Transactions on Automatic Control* **45(4)**, 725–729.
- Frank, P. (1978). *Introduction to system sensitivity theory*. Academic Pres, New York.
- Fu, K. and J. K. Mills (2005). Convex integrated design (CID) Method and its applications to the design of a linear positioning system. *IEEE Trans. on Control Systems Technology* **13(5)**, 701–707.
- Li, Q, W. L. Zhang and L. Chen (2001). Design for control: a concurrent engineering approach for mechatronic system design. *IEEE/ASME Trans. Mechatronics* **6(2)**, 161–169.
- Meier, E-B. and Jr, A. E. Bryson (1990). Efficient algorithm for time-optimal control of a two-link manipulator. *Journal of Guidance* **13(5)**, 859–866.
- Park, J. H. and H. Asada (1994). Convex integrated design (CID) Concurrent design optimization of mechanical structure and control for high speed robots. *ASME Trans. J. of Dyn. Sys., Meas. Ctrl.* **116**, 344–356.
- Ogata, K. (1998). *Ingeniería de Control Moderna*. Prentice-Hall, Hispanoamérica, México.
- Sadegh, N. and B. Driessen (1999). Minimum time trajectory optimization and learning. *ASME, Journal of Dynamics Systems, Measurement, and Control* **121(2)**, 213–217.
- Spong, M. W. and M. Vidyasagar (1989). *Robot Dynamics and Control*. John Wiley & Sons, Inc., New York.
- Vakhrameev S.A. (1997). A bang-bang theorem with a finite number of switchings for nonlinear smooth control systems. *Journal of Mathematical Sciences* **85(3)**, 2002–2016.
- Villarreal, M. G. (2005). *Rediseño paramétrico del Pendubot para posicionamiento vertical en tiempo mínimo*. Tesis de Maestría, Departamento de Ingeniería Eléctrica, Sección de Mecatrónica, CINVESTAV-IPN.
- Weinreb, A. and Jr A.E. Bryson (1985). Optimal control of systems with hard control bounds. *IEEE Transactions on Automatic Control* **30(11)**, 1135–1138.
- Wie, B. and J. Sunkel (1990). Minimum-time pointing control of a two-link manipulator. *Journal of Guidance* **13(5)**, 867–872.
- Zhao, D. and J. YI (2003). Swing up Pendubot with a GA-tuned Bang-bang Controller. *IEEE Int. Conf. on Robotics, Intelligent Systems and Signal Processing* **2**, 752–757.
- Zhang, W. J., Q. Li and L. S. Guo (2001). Integrated design of mechanical structure and control algorithm for a programmable four-bar linkage. *IEEE/ASME Trans. Mechatronics* **4(4)**, 354–362.