

TÉCNICAS DE CONTROL A TRAVÉS DE RED INALÁMBRICA DE VEHÍCULO BASADO EN PÉNDULO INVERTIDO

J. Arriaga, P. Millán, I. Jurado, C. Vivas y F.R. Rubio
Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, España
arriaga,pmillan,ijurado,vivas@cartuja.us.es y rubio@esi.us.es

Resumen

Este trabajo presenta la aplicación de una técnica de control robusto \mathcal{H}_∞ en el contexto de sistemas controlados a través de redes (NCS). La aplicación se lleva a cabo sobre un Vehículo personal basado en Péndulo Invertido (PPCar) que se controla remotamente a través de una red de comunicaciones inalámbrica tipo Bluetooth. El trabajo realiza un estudio comparativo del comportamiento de esta técnica de control frente a un control clásico LQR, presentando resultados experimentales que ponen de manifiesto las dificultades intrínsecas de control a través de redes.

Palabras clave: Péndulo Invertido, Control a través de redes, Control Robusto.

1. INTRODUCCIÓN

Esta última década ha sido testigo de un notable desarrollo de las tecnologías de comunicación inalámbricas, que han motivado la aparición de una amplia variedad de aplicaciones en multitud de disciplinas. La ingeniería de control no es ajena a estos desarrollos y podemos observar cómo en los últimos tiempos son cada vez más frecuentes los trabajos y sesiones técnicas especializadas dedicadas al control de sistemas dinámicos a través de redes de comunicaciones, o NCS (Networked Control Systems) en su denominación anglosajona.

Los sistemas NCS son aquellos en los cuales los diferentes elementos del lazo de control (sensores, controladores y actuadores) se encuentran físicamente distribuidos y unidos a una red de comunicaciones de propósito general, compartiendo ancho de banda con otras aplicaciones, no necesariamente relacionadas con el control.

El empleo de tecnologías de control a través de redes de comunicaciones presenta múltiples ventajas respecto a los sistemas tradicionales, como son arquitecturas más flexibles, reducción de costes de instalación y mantenimiento, posibilidades de control remoto de dispositivos móviles mediante enlaces inalámbricos, etc.

Consecuentemente, las tecnologías NCS encuentran aplicación en la actualidad en campos tan variados como el control de tráfico en autovías [9] o intervenciones quirúrgicas remotas [8].

Sin embargo el uso de redes de comunicación no dedicadas, en contraste con los tradicionales enlaces de comunicaciones independientes dedicados a control, introduce una serie de interesantes desafíos desde el punto de vista del diseño de controladores, protocolos de comunicaciones, arquitecturas de control, y en general todos los elementos asociados al lazo de control.

El control NCS surge así como una disciplina híbrida de las, hoy día bien establecidas, teoría de control automático y tecnología de telecomunicaciones. Tradicionalmente la teoría de control se ocupa de sistemas dinámicos interconectados a través de *enlaces de comunicaciones ideales* (instantáneos, sin pérdidas, etc), mientras que la tecnología de telecomunicaciones se ocupa de la transmisión de información sobre *canales imperfectos*. Una combinación de ambos marcos es necesario para modelar apropiadamente un sistema NCS.

El presente trabajo trata de poner manifiesto algunas de las peculiaridades asociadas a los sistemas de control NCS, presentando un análisis comparativo sobre resultados experimentales del comportamiento del comportamiento de una técnica de control clásico frente a técnicas específicas NCS.

Concretamente, se plantea el control de un *Vehículo personal basado en Péndulo Invertido* (PPCar) a través de una red de comunicaciones inalámbrica tipo Bluetooth. El sistema es especialmente apropiado para ejemplificar los problemas asociados al control NCS, debido a que se trata de un sistema de dinámica inestable relativamente simple, aunque fácil de estabilizar si se aplican las hipótesis del control clásico. La dificultad de control se pone de manifiesto en este trabajo al comparar el comportamiento de un control clásico LQR diseñado bajo la hipótesis de *enlace de comunicaciones perfecto* frente a una técnica de control \mathcal{H}_∞ adaptado a las condiciones de tiempo de muestreo variable y pérdida de información asociado a sistemas NCS.

El artículo se ha estructurado del siguiente modo: En primer lugar, en la sección 2, se describen los aspectos más relevantes de la arquitectura hardware y software del sistema PPCar. A continuación, la sección 3 describe el modelo dinámico no lineal del sistema que se empleará para el diseño de controladores. En la sección 4 se describen las estrategias de control local LQR y control NCS \mathcal{H}_∞ que se ensayan en este trabajo. Para finalizar, la sección 5 describe los resultados experimentales obtenidos y realiza una breve discusión de los mismos.

2. DESCRIPCIÓN DEL SISTEMA

Se ha trabajado con un vehículo personal basado en péndulo invertido, el PPCar -Personal Pendulum Car-, descrito en [5]. Dicho vehículo, destinado al transporte de personas, está montado sobre dos ruedas paralelas donde el péndulo está constituido por la persona que está sobre el vehículo, que se apoya sobre una barra de control. En la fig. 1 se representa un esquema del vehículo. El movimiento de avance está provocado por la inclinación de la persona con respecto a la posición de equilibrio, la verticalidad. De hecho, los controladores calculados para lograr la verticalidad se han diseñado teniendo en cuenta tan sólo el movimiento en línea recta del vehículo, mientras que el movimiento de rotación es controlado directamente por el piloto mediante un dispositivo eléctrico ubicado en el puño.

En los trabajos anteriores con este vehículo, se trabajó siempre utilizando como núcleo del sistema un microcontrolador ubicado en el propio vehículo, que recogía la información de los sensores y a partir de dicha información calculaba las acciones de control. Adicionalmente, el microcontrolador se comunicaba con una estación base (un PC) mediante un software diseñado a tal efecto que permitía ajustar los parámetros de los controladores en línea y capturar información relevante. Sin embargo, en este trabajo el microcontrolador ha quedado como simple intérprete de órdenes: al PC le sigue llegando la información, pero es éste (mediante un nuevo software) que calcula las acciones de control. El PC transmite las consignas al microcontrolador, que se limita a transferirlas a los motores. De hecho, si se dispusiera de sensores externos para obtener el estado del sistema, podríamos enviar directamente las consignas a la controladora de los motores sin necesidad de intervención del microcontrolador.

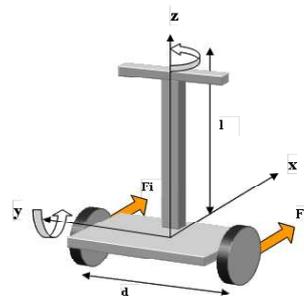


Figura 1: Esquema del vehículo

2.1. ARQUITECTURA HARDWARE

La plataforma donde debe montarse el piloto va sostenida por el eje de las ruedas, de radio 17.5 cm, cada una de ellas conectada a un motor alimentado a 24 V y capaces de llegar a 240 rpm.

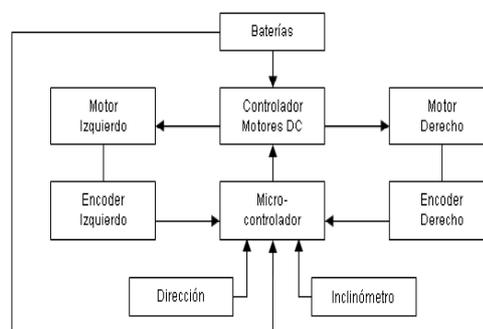


Figura 2: Diagrama de bloques del sistema

La fig. 2 representa un diagrama de bloques del conjunto. El núcleo del sistema es un microcontrolador de 8 bits de bajo coste, el modelo Atmega128 de Atmel, que se comunica con los distintos periféricos a través de diferentes puertos. El microcontrolador lleva incorporado un reloj interno de 16 Mhz, lo que ha permitido en trabajos anteriores utilizar tiempos de muestreo de decenas de milisegundos.

Al microcontrolador se conectan:

- Dos encoders, uno por cada rueda. Se emplean para medir las velocidades de ambas.
- Un sensor de inclinación MicroStrain, que incorpora preprocesamiento de señal basada en filtro de Kalman.
- Un sensor para indicar la consigna de giro, ordenada por el piloto a través de un puño con resistencia variable.
- Una controladora conectada a los dos motores, Roboteq AX1550, se comunica con el con-

trolador a través de una de las dos UARTs que posee.

- Finalmente, un radio-módem bluetooth HandyPort HPS-120 que trabaja en la banda de 2.4 Ghz, se conecta al microcontrolador a través de la segunda UART y al PC en el otro extremo mediante el puerto serie (también con interfaz RS-232).
- Una caja de control conectada tanto a la controladora de los motores como al microcontrolador y que tiene la potestad de detener la acción de ambos.

El equipo PC de la estación base es un Pentium IV a 2.80 Ghz de frecuencia de reloj con 512 MB de memoria RAM.

La controladora AX1550 permite obviar el problema de la alimentación de los motores: recibe las consignas del microcontrolador y se ocupa de mantener la consigna a ambos motores. En la fig. 3 se ilustra la arquitectura completa del sistema.

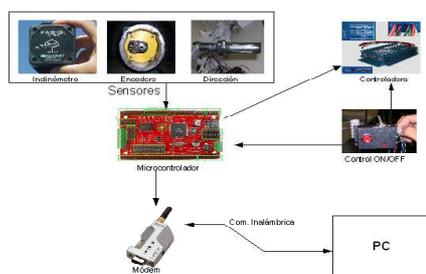


Figura 3: Arquitectura del sistema

2.2. SOFTWARE DEL MICROCONTROLADOR

El software para el microcontrolador se ha basado en el sistema operativo para sistemas embebidos TinyOS, programando en lenguaje nescC. Se partió de la base del trabajo original reutilizando gran parte del código.

TinyOS es un software de código abierto con una estructura de módulos jerárquica que permite al programador abstraerse del hardware, así como reutilizar código de manera sencilla.

NesC permite desarrollar programas compactos que consuman poca memoria, basados en un núcleo multihilo con dos tipos de estructuras: tareas y eventos.

Para poder tener un sistema de control en tiempo real se programó un reloj de tiempo real de 10ms de período que dota al sistema de una base de tiempo para poder realizar las tareas necesarias.

Para las comunicaciones por puerto serie, tanto para lectura de datos de los sensores como para la comunicación con el PC, se han utilizado interrupciones.

Para el desarrollo del software empleado se aprovechó el desarrollo original; tal y como estaba previsto, la estructura basada en módulos facilitó enormemente la reprogramación al abstraerlos del nivel de hardware. De este modo, empleamos los módulos de convertidor analógico/digital, puerto serie, encoder y temporización.

2.3. SOFTWARE DE PLATAFORMA PC

Como ya se ha mencionado, la comunicación entre microcontrolador y PC se lleva a cabo mediante comunicación inalámbrica siguiendo el protocolo Bluetooth en su versión 1.1 (norma IEEE 802.15.1-2002); sin embargo, al estar conectado ese módem al puerto serie, el programador trabaja directamente sobre el interfaz RFCOMM. La capa de control y adaptación del enlace lógico (L2CAP) y la capa física quedan abstraídas del programador. La tasa de bits (configurable) elegida fue de 19200 bits/s.

Se estudió la posibilidad de emplear TCP/IP sobre PPP, sin embargo eso hubiera implicado la instalación de dichos protocolos en el sistema operativo TinyOS y hubiera complicado innecesariamente el sistema: en su lugar, creamos un protocolo simple de comunicación que no posee control de flujo ni de errores (aunque sí detección de éstos).

El protocolo implementado consta de tres tipos de PDUs: las tramas de conexión, las de asentimiento y las de datos.

- PDU de conexión: son tramas simples, de un sólo octeto, cuyo objetivo es arrancar o detener el vehículo; debido a su naturaleza crítica son las únicas que requieren asentimiento.
- PDU de asentimiento: sólo se utilizan en respuesta a las anteriores. También de un sólo octeto.
- PDU de datos: compuestas por una cabecera, un pie, un número par de octetos de información (si hace falta se añade un octeto nulo al final para lograr un número par) y dos octetos de checksum.

El software instalado en la estación base (PC) se desarrolló en C empleando el entorno de desarrollo Dev-C++. Para la comunicación con el puerto serie se utilizó la biblioteca LnxComm [3], que permite de forma sencilla la programación multipataforma (windows o *NIX) del puerto serie

abstrayendo al programador de las particularidades del sistema operativo. El programa consta de dos hilos, el de recepción de datos (prioritario) y el de generación de consignas: el hilo de recepción controla el flujo del programa, escuchando el puerto serie a la espera de una trama de datos; al recibirla, cede brevemente el control del flujo al hilo de generación de consignas para que, a partir de la información que acaba de recoger (y procesar), pueda calcular y enviar la señal de control de vuelta al vehículo. Para la creación de los hilos y la sincronización entre éstos se empleó el sistema de eventos de Win32; se plantea la conveniencia de emplear hilos POSIX para asegurar la portabilidad del código a plataformas *NIX.

Ambos hilos comparten información a través de una estructura de tipo CARRO, que contiene información sobre el puerto lógico que soporta la conexión, el estado actual del vehículo y la consigna que se envía. Dicha estructura puede ser ampliada de acuerdo con las necesidades del momento con estados (y consignas) anteriores o futuras.

El programa recibe la información y envía la orden de control de forma continua hasta que se cumple un máximo de iteraciones preestablecido o bien hasta que el operador de la estación base decida terminar. Cuando eso ocurre, se almacena en un fichero de registro toda la información obtenida (instante de tiempo en que se recogen muestras, contenido de éstas, señal de control enviada) en un formato adecuado para su lectura y procesamiento con software específico (MATLAB, Octave).

Es importante resaltar el hecho de que no existe temporización en la plataforma PC: sólo el microcontrolador lleva su propio temporizador que cada 10 ms (un ciclo de reloj) recaba datos, y cada cierto número de ciclos envía información. En el desarrollo del software de control y monitorización tan sólo hemos de preocuparnos de que el programa reciba la información y procese y envíe la respuesta lo antes posible: el microcontrolador aplicará la señal de control de forma asíncrona. Si una de las tramas llega con errores, se desecha y se espera a la siguiente; puesto que el tiempo es crítico en nuestro sistema, no podemos permitirnos pedir y esperar un reenvío. El programa incorpora un mecanismo de seguridad de tal forma que si llega un número determinado (y configurable) de tramas erróneas, se detiene el vehículo y se corta la conexión.

El programa incluye una función “controlador” que devuelve una señal de control entera a partir de los datos que obtiene; dichos datos, que conforman el estado y están almacenados en la estructura de tipo CARRO, son empleados para calcular la señal de control. Si se desea programar un nuevo

controlador, es tan sencillo como implementarlo en dicha función.

3. OBTENCIÓN DEL MODELO DINÁMICO DEL SISTEMA

En esta sección se detalla el proceso de obtención del modelo matemático del sistema. En la fig. 1 se ha representado el subsistema constituido por el péndulo invertido sobre plataforma móvil.

Del balance de fuerzas sobre el eje horizontal, se obtienen la ecuación (1).

$$(M + m)\ddot{x} + Ml\ddot{\theta}\cos\theta - Ml\dot{\theta}^2\sin\theta = F \quad (1)$$

donde:

- $M = 70$ Kg, masa del péndulo (incluyendo al piloto).
- $l = 1$ m, longitud del brazo del péndulo.
- $m = 35$ Kg, masa del carro.
- $g = 9.81$ m/s^2 , aceleración de la gravedad.
- F = fuerza aplicada por los motores sobre el carro.
- x = desplazamiento lineal del carro.
- θ = ángulo de inclinación del péndulo.

Del balance de momentos se obtiene la ecuación (2).

$$\ddot{x}Ml\cos\theta + Ml^2\ddot{\theta} - Mgl\sin\theta = 0 \quad (2)$$

Combinando las ecuaciones anteriores pueden despejarse las segundas derivadas de la posición y la inclinación. Resulta sencillo comprobar que este sistema tiene dos equilibrios, el estable (ángulo de inclinación 180°), y el inestable (ángulo 0°), que corresponde al péndulo en posición vertical. Si se linealiza en torno al punto de equilibrio inestable, se obtiene el siguiente sistema descrito ya en espacio de estados:

$$\begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ (M + m)\frac{g}{ml} & 0 & 0 \\ \frac{-Mg}{m} & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \\ x \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{-1}{ml} \\ \frac{1}{m} \end{pmatrix} F \quad (3)$$

3.1. MODELADO DE LA RED

A la hora de diseñar cualquier controlador a través de la red, resulta imprescindible conocer el retraso de la red. A lo largo de varias pruebas, se han recabado datos de tiempos de propagación de paquetes simples (un octeto) a través de la red.

Programamos el PC y el vehículo para enviar y recibir paquetes de forma instantánea, con el menor procesamiento posible. Se hizo un mínimo de 200 envíos y recepciones para cada una de las baterías de pruebas, en las que el vehículo fue desplazado para simular distintas posiciones con o sin visión directa.

Finalmente se almacenó un total de 2500 muestras de tiempos; comoquiera que en ese tiempo se incluyen tanto el envío como el asentimiento, hubo que dividir cada una de esas medidas por la mitad. Asimismo, hubo que restarle el tiempo de envío asociado al régimen binario. Una vez procesados, los datos arrojaron los siguientes resultados:

- Valor medio: 23.34 ms.
- Desviación típica: 1.8662.
- Valor mínimo del retraso: 18 ms.
- Paquetes perdidos: 0.04

La figura fig:retrasos muestra cómo los datos siguen una distribución normal con la media y la distribución típica calculada.

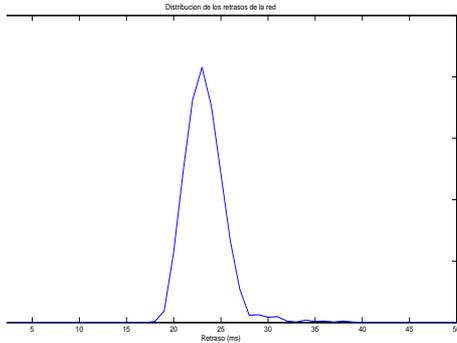


Figura 4: Distribución de los retardos en la red

4. DISEÑO DE CONTROLADORES

Para poner de manifiesto las peculiaridades de control a través de redes, se plantea en este trabajo una comparación del comportamiento un control clásico LQR, diseñado para el PPCar bajo la hipótesis de *enlace perfecto de comunicaciones*

frente a un control robusto \mathcal{H}_∞ que si tiene en cuenta aspectos propios de NCS, tal como tiempos de muestreo no uniformes y eventuales pérdidas de paquetes de información. Se describe a continuación el procedimiento de diseño de cada una de estas técnicas de control.

4.1. DISEÑO DE CONTROLADOR LOCAL

El controlador LQR es un control por realimentación de estados de la forma:

$$u = -Kx \quad (4)$$

donde la señal de control u , y la constante de realimentación K se obtiene a partir de un problema de minimización del funcional de costes J , definido en (5).

$$J = \int_0^\infty (x'Qx + u'Ru)dt. \quad (5)$$

Las matrices Q y R penalizan respectivamente el error del estado y el esfuerzo de control.

La solución del problema LQR para el sistema (3) obtiene el siguiente controlador:

$$u = F = -Kx = -K_1\theta - K_2\dot{\theta} - K_3\dot{x} \quad (6)$$

donde los parámetros K varían en función de los parámetros de las matrices Q y R .

En este caso, con el objetivo de comparar un controlador clásico diseñado sin tener en cuenta la red de comunicaciones, se adopta el diseño recogido en [5]:

$$Q = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad R = 0,1, \quad (7)$$

lo que nos lleva a la siguiente solución:

$$K_1 = 2070 \quad K_2 = 385 \quad K_3 = 3,2. \quad (8)$$

La fig. 5 muestra el buen comportamiento del controlador LQR obtenido en simulaciones de control local. Anteriores experiencias aplicando este control en el microcontrolador dieron lugar a resultados coherentes con la simulación.

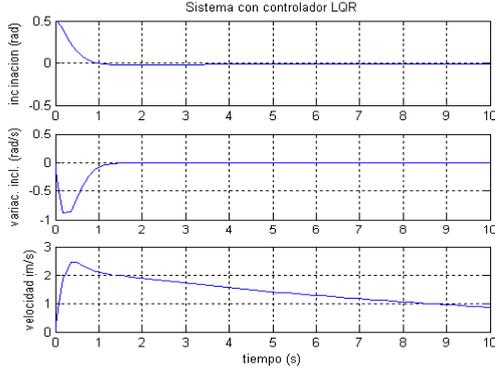


Figura 5: Simulación del controlador LQR

4.2. DISEÑO BASADO EN RED DE UN CONTROLADOR ROBUSTO \mathcal{H}_∞

En esta sección se describe el proceso de diseño y aplicación al de un controlador robusto \mathcal{H}_∞ basado en red, propuesto en [6]. En la sección 4.2.1, se detalla el modelado de sistemas para el diseño del controlador, y se enuncian las hipótesis fundamentales. En la sección 4.2.2 se detalla el procedimiento de diseño del controlador. Por último en la sección 4.2.3, se resuelve el problema de control para el caso del PPar.

4.2.1. Modelado para el control

En esta sección se va a considerar un sistema con incertidumbres paramétricas de la forma:

$$\begin{aligned}\dot{x}(t) &= [A + \Delta A(t)]x(t) + [B + \Delta B(t)]u(t) + B_w w(t) \\ x(t_0) &= x_0 \\ z(t) &= Cx(t) + Du(t)\end{aligned}\quad (9)$$

donde $x(t) \in \mathbb{R}^{n \times n}$, $u(t) \in \mathbb{R}^m$, $z(t) \in \mathbb{R}^q$ son el vector de estados, la señal de control y la salida controlada respectivamente. $x_0 \in \mathbb{R}^n$ denota la condición inicial; A , B , B_w , C y D son matrices constantes de dimensiones apropiadas y $\Delta A(t)$ y $\Delta B(t)$ son las incertidumbres paramétricas, que satisfacen:

$$[\Delta A(t) \quad \Delta B(t)] = GF(t)[E_a \quad E_b], \quad (10)$$

donde G , E_a y E_b son matrices constantes de dimensiones apropiadas y $F(t)$ es una matriz desconocida variable en el tiempo, que es medible en el sentido de Lebesgue en t y satisface $F^T(t)F(t) \leq I$. Por otra parte $w(t) \in L_2[t_0, \infty)$ denota a las perturbaciones externas.

La presencia de una red entre el controlador y la planta causa a menudo retrasos de transmisión y pérdida de datos que pueden degradar el comportamiento en bucle cerrado del sistema [7]. En presencia de una red de control, bajo ley de control lineal, el sistema (9) se puede escribir de la siguiente forma:

$$\begin{aligned}\dot{x}(t) &= [A + \Delta A(t)]x(t) + [B + \Delta B(t)]u(t) + B_w w(t) \\ z(t) &= Cx(t) + Du(t), t \in [i_k h + \tau_k, i_{k+1} h + \tau_{k+1}) \\ u(t^+) &= Kx(t - \tau_k), t \in i_k h + \tau_k, k = 1, 2, \dots\end{aligned}\quad (11)$$

donde h es el periodo de muestreo, $i_k (k = 1, 2, 3, \dots)$ son enteros tal que $\{i_1, i_2, i_3\} \subset \{1, 2, 3, \dots\}$. Por otra parte τ_k denota el retraso temporal desde el instante $i_k h$ cuando el sensor muestrea la planta hasta el instante en el que el actuador aplica la señal de control correspondiente. Se asume que $u(t) = 0$ hasta que la primera señal de control alcanza la planta. Obviamente se tiene que $\bigcup_{k=1}^{\infty} [i_k h + \tau_k, i_{k+1} h + \tau_{k+1}) = [t_0, \infty)$, $t_0 \geq 0$.

Es importante remarcar que $\{i_1, i_2, i_3, \dots\}$ es un subconjunto de $\{1, 2, 3, \dots\}$. No se requiere por tanto la condición $i_{k+1} > i_k$. Cuando $i_1, i_2, i_3 = 1, 2, 3, \dots$ significa que no se han producido pérdidas de paquetes durante la transmisión. Si $i_{k+1} = i_k + 1$ esto implica necesariamente que $h + \tau_{k+1} > \tau_k$, caso que incluye como condiciones particulares $\tau_k = \hat{\tau}$ y $\tau_k < h$, con $\hat{\tau}$ constante.

A continuación se enuncian las dos hipótesis en las que se basa el desarrollo posterior:

Hipótesis 1. El sensor es periódico, mientras que el controlador y el actuador se ejecutan tan pronto como disponen de una nueva señal (basados en eventos).

Hipótesis 2. Existen dos constantes $\eta > 0$ y $\tau_m > 0$ tal que:

$$(i_{k+1} - i_k)h + \tau_{k+1} \leq \eta, k = 1, 2, \dots \quad (12)$$

$$\tau_m \leq \tau_k, k = 1, 2, \dots \quad (13)$$

donde $\eta \leq \tau_m$. Bajo la Hipótesis 2, el sistema (11) puede ser escrito de forma equivalente como sigue:

$$\begin{aligned}\dot{x}(t) &= [A + \Delta A(t)]x(t) + [B + \Delta B(t)] \\ &\quad \times Kx(i_k h) + B_w w(t),\end{aligned}\quad (14)$$

$$\begin{aligned}x(t) &= \Phi(t, t_0 - \eta)x(t_0 - \eta) \\ &\doteq \phi(t), t \in [t_0 - \eta, t_0],\end{aligned}\quad (15)$$

$$\begin{aligned}z(t) &= Cx(t) + DKx(i_k h), \\ t &\in [i_k h + \tau_k, i_{k+1} h + \tau_{k+1})\end{aligned}\quad (16)$$

donde $\Phi(t, t_0 - \eta)$ es una solución de $\dot{\Phi}(t, t_0 - \eta) = [A + \Delta A(t)]\Phi(t, t_0 - \eta)$, $t \in [t_0 - \eta, t_0]$.

Definición. El sistema (14)-(16) es exponencialmente estable y robusto con norma \mathcal{H}_∞ acotada por γ si se cumplen las siguientes condiciones:

1. El sistema (14)-(16) con $w(t) \equiv 0$ es exponencialmente estable de manera robusta, es decir, existen unas constantes $\alpha > 0$ y $\beta > 0$ tal que $\|x(t)\| \leq \alpha \sup_{t_0 - \eta \leq s \leq t_0} \|\Phi(s)\| e^{-\beta t}$, $t \geq t_0$, para todas las incertidumbres admisibles $\Delta A(t)$ y $\Delta B(t)$.
2. Bajo condiciones iniciales nulas, la salida controlada $z(t)$ satisface $\|z(t)\|_2 \leq \gamma \|w(t)\|_2$ para toda perturbación $w(t) \in L_2[0, \infty)$.

4.2.2. Diseño del controlador robusto

A continuación se enuncia un teorema que permite diseñar, teniendo en cuenta la red de comunicaciones, el controlador K del sistema resolviendo un sistema de LMIs.

Teorema 1 [6]: Dados los escalares ρ_i ($i = 1, 2, 3, 4$), τ_m , η y γ , si existen matrices \tilde{P}_k ($k = 1, 2, 3$), $\tilde{T}_j > 0$, $\tilde{R}_j > 0$ ($j = 1, 2$), una matriz no singular X y cualesquiera matrices \tilde{N}_i , \tilde{S}_i ($i = 1, 2, 3, 4$) de dimensiones apropiadas, así como un escalar $\mu > 0$ tal que:

$$\begin{pmatrix} \Phi_{11} & * \\ \Phi_{21} & \Phi_{22} \end{pmatrix} < 0 \quad (17)$$

$$\begin{pmatrix} \tilde{P}_1 & \tilde{P}_2 \\ \tilde{P}_2^T & \tilde{P}_3 \end{pmatrix} > 0 \quad (18)$$

donde

$$\Phi_{11} = \begin{pmatrix} \Sigma_{11} & * & * & * \\ \Sigma_{21} & \Sigma_{22} & * & * \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} & * \\ \Sigma_{41} & \Sigma_{42} & \Sigma_{43} & \Sigma_{44} \end{pmatrix}$$

$$\Phi_{21} = \begin{pmatrix} \tau_0 \tilde{P}_3 & 0 & -\tau_0 \tilde{P}_3 & \tau_0 \tilde{P}_2^T \\ \eta \tilde{N}_1^T & \eta \tilde{N}_2^T & \eta \tilde{N}_3^T & \eta \tilde{N}_4^T \\ \delta \tilde{S}_1^T & \delta \tilde{S}_2^T & \delta \tilde{S}_3^T & \delta \tilde{S}_4^T \\ CX^T & DY & 0 & 0 \\ -B_w^T & -\rho_2 B_w^T & -\rho_3 B_w^T & -\rho_4 B_w^T \\ E_a X^T & E_b Y & 0 & 0 \end{pmatrix}$$

$$\Phi_{22} = \text{diag}(-\tau_0 \tilde{T}_2 \quad -\eta \tilde{R}_1 \quad -\delta \tilde{R}_2 \quad -I \\ -\gamma^2 I \quad -\mu I)$$

$$\Sigma_{11} = \tilde{P}_2 + \tilde{P}_2^T + \tilde{T}_1 + \tau_0 \tilde{T}_2 + \tilde{N}_1 \\ + \tilde{N}_1^T - AX^T - XA^T + \mu GG^T,$$

$$\Sigma_{21} = \tilde{N}_2 - \tilde{N}_1^T + \tilde{S}_1^T - \rho_2 AX^T \\ - Y^T B^T + \mu \rho_2 GG^T,$$

$$\Sigma_{31} = \tilde{N}_3 - \tilde{P}_2^T - \tilde{S}_1^T - \rho_3 AX^T + \mu \rho_3 GG^T,$$

$$\Sigma_{41} = X - \tilde{N}_4 - \tilde{P}_1^T - \rho_4 AX^T + \mu \rho_4 GG^T,$$

$$\Sigma_{22} = -\tilde{N}_2 - \tilde{N}_2^T + \tilde{S}_2 + \tilde{S}_2^T - \rho_2 BY \\ - \rho_2 B^T Y^T + \mu \rho_2^2 GG^T,$$

$$\Sigma_{32} = -\tilde{N}_3 + \tilde{S}_3 - \tilde{S}_2^T - \rho_3 BY + \mu \rho_3 \rho_2 GG^T,$$

$$\Sigma_{42} = -\tilde{N}_4 + \tilde{S}_4 + \rho_2 X - \rho_4 BY + \mu \rho_4 \rho_2 GG^T,$$

$$\Sigma_{43} = -\tilde{S}_4 + \rho_3 X + \mu \rho_4 \rho_3 GG^T,$$

$$\Sigma_{44} = \rho_4 X + \rho_4 X^T + \eta \tilde{R}_1 + 2, \delta \tilde{R}_2 + \mu \rho_4^2 GG^T,$$

entonces, bajo el controlador $u(t) = Kx(t)$, con $K = YX^{-T}$, el sistema (11) con una red de control satisfaciendo la Hipótesis 2 es exponencialmente estable y robusto con norma \mathcal{H}_∞ acotada por γ .

Demostración: La demostración del Teorema 1 se encuentra desarrollada en detalle en [6].

4.2.3. Aplicación del controlador al PPcar

El sistema bajo estudio, descrito por (3), puede escribirse directamente en la forma (9) con sólo definir de forma adecuada las matrices ΔA , ΔB y B_w , según las perturbaciones e incertidumbres que deseen incluirse. En este estudio, se va a considerar incertidumbre en la masa del piloto M , que se denotará por ΔM . Como perturbación principal se va a considerar el viento, que influye sobre el carro y el piloto. Esto se puede modelar como una fuerza resultante W del perfil de viento, que se repartirá entre la base del carro con un porcentaje c_m y el centro de gravedad del péndulo en un porcentaje c_M . La fig. 6 ilustra este fenómeno. De

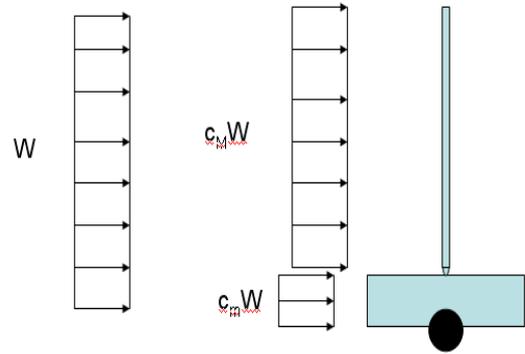


Figura 6: Viento incidiendo sobre el vehículo

esta forma se obtiene:

$$\Delta A = \begin{pmatrix} 0 & 0 & 0 \\ \frac{\Delta M g}{m} & 0 & 0 \\ -\frac{\Delta M g}{m} & 0 & 0 \end{pmatrix}$$

$$\Delta B = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$B_w = \begin{pmatrix} 0 \\ \frac{c_M}{m} - \frac{c_m}{m} \\ \frac{c_m}{l} \end{pmatrix} W$$

Para una incertidumbre de $\Delta M = 1Kg$, se obtiene, resolviendo el problema de LMI planteado en el Teorema 2, el siguiente controlador:

$$K_1 = 1550 \quad K_2 = 412 \quad K_3 = 53 \quad (19)$$

5. PRUEBAS Y RESULTADOS

Tras las simulaciones, se llevaron a cabo dos pruebas reales con el vehículo: en ambas, el objetivo fue, partiendo de unas condiciones de contorno, estudiar el comportamiento de ambos controladores tratando de estabilizar el sistema.

Es importante advertir que nuestro modelo matemático adolece de una gravísima debilidad: no modela adecuadamente el efecto del piloto. Éste ejerce un par de fuerzas sobre la base y el péndulo, haciendo que su comportamiento sea impredecible; es por eso que tendemos a considerarlo como una perturbación, y será la resistencia frente a la perturbación la que nos informe de la bondad de nuestro control.

La primera prueba se llevó a cabo programando en la aplicación de PC un control por realimentación por espacio de estados: concretamente, el LQR ya expuesto.

Puede apreciarse que la perturbación hace imposible estabilizar el sistema: al mantener la consigna durante 30 ms en lugar de los 10 ms del control local, ésta no puede reaccionar lo suficientemente rápido a los cambios en el sistema. El sistema no se inestabiliza, pero tampoco logra alcanzar el reposo que marcaría la estabilidad asintótica.

La segunda prueba consistió en la programación, también por realimentación del espacio de estados, del controlador H_∞ mediante LMI ya calculado.

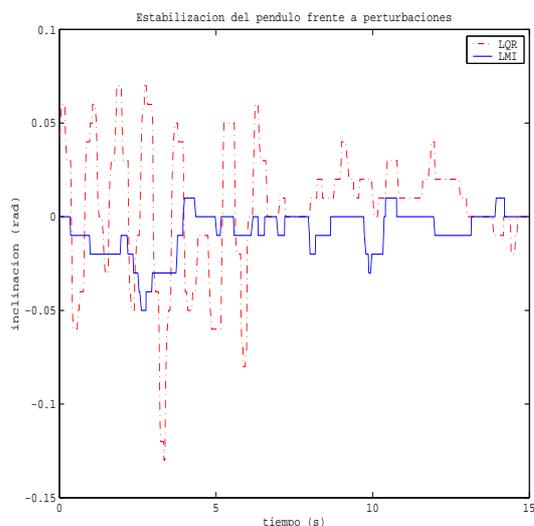


Figura 7: Estabilización del péndulo

La comparación entre ambas curvas (LQR en línea punteada, LMI con línea continua) muestra cómo el controlador H_∞ logra estabilizar el sistema de forma más rápida, siendo mucho menos sensible a las perturbaciones. Comoquiera que las perturbaciones son consustanciales al sistema, debido a la inevitable acción del piloto, el controlador resulta mucho más conveniente. Naturalmente, los retardos asociados a la red inalámbrica son los que convierten en críticas las perturbaciones, ya que el sistema no es capaz de reaccionar a éstas con tanta rapidez como en el caso del control local.

Agradecimientos

Los autores agradecen a CICYT por financiar este trabajo bajo el proyecto DPI2007-64697.

Referencias

- [1] Aström, K.J., Furuta, K., (1996) "Swinging-up a pendulum by energy control", *Proc. IFAC Congress*, pp. 37-95.
- [2] Clark, M.A., Field, J.B., McMahon, S.G., Philips, P.S., Cazzolato, B.S., (2005) "Edgar, a self-balancing scooter", Project Final Report. University of Adelaide, Australia
- [3] Pujaico F., "LnxComm, Linux Communication: Tutorial y Guía de Referencia"
- [4] Rubio, F.R., López, M.J., "Control adaptativo y robusto" (2006), Universidad de Sevilla
- [5] Viguria, A., Prieto, M., Fiacchini, M., Cano, R., Rubio, F.R., Aracil, J., Canudas-de-Wit, C., (2006) "Desarrollo y experimentación de un vehículo basado en péndulo invertido (PP-Car)", *Revista Iberoamericana de Automática e Informática Industrial. Vol. 3, Num. 4*, pp. 54-66,
- [6] Yue, Dong and Han, Qing-Long and Lam, James, (2005) "Network-based robust \mathcal{H}_∞ control of systems with uncertainty", *Automatica. Vol. 41, Num. 6*, pp. 999-1007,
- [7] Zhang, W. and Branicky, MS and Phillips, SM, "Stability of networked control systems" (2001) *Control Systems Magazine, IEEE. Vol. 21, Num. 1*, pp. 84-99,
- [8] C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian, "Remote surgery case: Robot-assisted teleneurosurgery", in *IEEE Int. Conf. Robot. and Auto. (ICRA '04)*, Apr. 2004, vol. 1, pp. 819-823.
- [9] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems", in *Proc. 2001 Amer. Contr. Conf.*, Jun. 2001, vol. 2, pp. 1491-1496.