

Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor. Simulación y evaluación de propuestas.

Xavier Blasco*, Sergio García-Nieto, Gilberto Reynoso-Meza

*Instituto Universitario de Automática e Informática Industrial
Universitat Politècnica de València
Camino de Vera s/n Apdo. 46022, Valencia*

Resumen

Este documento muestra en detalle el proceso de simulación y evaluación multiobjetivo de las propuestas al Concurso en Ingeniería de Control 2012, *Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor*, organizado por el Comité Español de Automática (CEA). El concurso busca ofrecer una herramienta de trabajo adicional que pueda servir para reforzar la educación en control y orientar a los alumnos de ingeniería sobre el tipo de problemas que un ingeniero de control debe enfrentar. *Copyright © 2012 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.*

Palabras Clave:

Educación en control, Control de helicópteros, Identificación de procesos, Diseño de sistemas de control, Decisión multiobjetivo.

1. Introducción

El Concurso en Ingeniería de Control 2012, *Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor* organizado por el Comité Español de Automática, se desarrolla en varias fases según se indica en las bases que se pueden encontrar en la web: <http://www.ceautomatica.es/og/ingenieria-de-control/benchmark-2011-2012>. Como objetivo general del concurso, se busca ofrecer a los estudiantes de ingeniería de control una plataforma que reproduzca un sistema real, complejo, mediante el cual puedan reforzar los conocimientos adquiridos en las clases de ingeniería de control. El mismo demandará, por medio de un aprendizaje basado en problemas, la integración de conocimientos en dinámica, modelado, identificación y control así como el desarrollo de habilidades para el trabajo en equipo.

Los objetivos específicos de la primera fase del concurso son:

1. Obtener un modelo que reproduzca lo mejor posible el comportamiento dinámico de las variables de posición x e y del vehículo ante cambios en las variables manipuladas (las referencias *RefPitch* y *RefRoll*). Los datos

suministrados a los participantes se obtienen de ensayos sobre un modelo no-lineal del proceso con el que posteriormente se evaluará la calidad del modelo presentado.

2. Obtener un controlador que, con las medidas disponibles x , y , v_x y v_y (posiciones y velocidades), actúe sobre las variables manipuladas *RefPitch* y *RefRoll*, para conseguir el mejor seguimiento posible de trayectorias sobre el plano xy . La evaluación del controlador se realizará también con el modelo no-lineal del proceso.

Para facilitar los trabajos de los participantes se suministran unas funciones Matlab©/Simulink© que reproducen el proceso de evaluación aunque no incluyen el modelo no-lineal que se utilizará. El documento describe, por una parte, estas funciones y cómo se deben utilizar, y por otra parte, cómo se va a realizar la evaluación.

2. Simulador de modelos

Para simular los modelos que se desarrollen, compararlos con los datos experimentales suministrados para la fase 1 del concurso y evaluar la calidad de los resultados obtenidos, se suministra el diagrama simulink© *ComparadorModelo_continuo.mdl*, la función *pintaResultExp()* y el *script simulaModelo.m* (disponibles en: <http://www.ceautomatica.es/og/ingenieria-de-control/benchmark-2011-2012>).

La función *pintaResultExp()* tiene como parámetro de entrada la variable con los datos que se obtienen de la simu-

* Autor en correspondencia.

Correos electrónicos: xblasco@isa.upv.es (Xavier Blasco),
sergarro@isa.upv.es (Sergio García-Nieto), gilreyme@upv.es (Gilberto Reynoso-Meza)

lación (variable `ResultExp`) y como parámetro de salida una estructura con los indicadores necesarios para la evaluación de prestaciones. Además esta función representa gráficamente, en dos figuras, los resultados de la simulación del modelo comparados con los datos experimentales (figura 1) y los indicadores que se van a utilizar (figura 2).

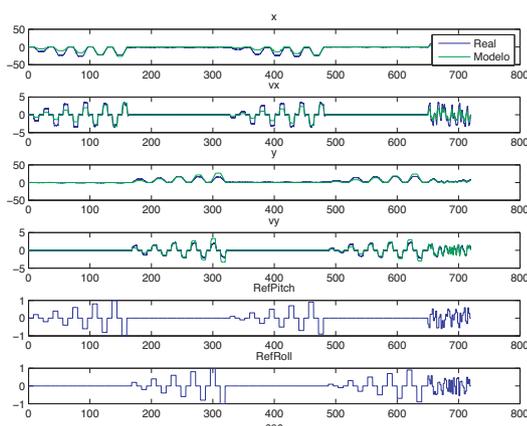


Figura 1: Figura 1 que aparece al ejecutar `pintaResultExp()`. Muestra los resultados de la simulación comparándolos con los datos experimentales.

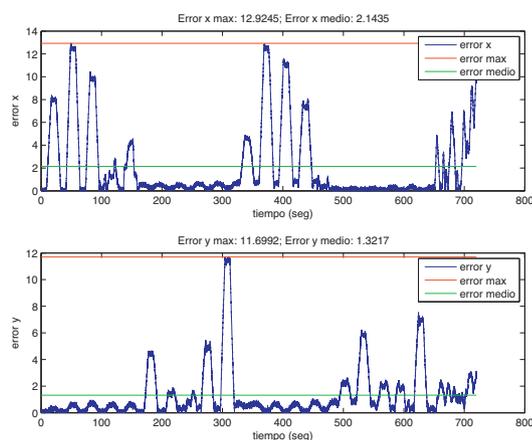


Figura 2: Figura 2 que aparece al ejecutar `pintaResultExp()`. Muestra gráficamente la información de los indicadores que se utilizarán para la evaluación.

Todo el proceso de evaluación queda automatizado en `simulaModelo.m`; se cargan los datos experimentales, se simula, se calculan los indicadores y presentan los datos de evaluación. El procesamiento de la información de evaluación se describe con más detalle en el apartado 4.

Para la evaluación formal de cada propuesta los ensayos que se realizarán vendrán dados por la organización y no necesariamente coincidirán con los datos que se suministraron para la fase 1.

3. Simulador de alternativas de control

La simulación de una alternativa de control se realiza mediante la función en Simulink© `simuladorControlCuatrirrotor.mdl` (ver figura 3).

Para poder ejecutar la simulación es imprescindible que las variables `ruta` y `radio` estén definidas en el espacio de trabajo de Matlab©. La variable `ruta` contiene las coordenadas (en metros) de la trayectoria a seguir, cada fila de esta matriz corresponde con coordenadas xy de un punto. La variable `radio` tiene asignado el valor del radio (en metros) del disco alrededor de cada uno de los puntos de la trayectoria para el que se considera que el vehículo ha alcanzado dicha coordenada de la trayectoria. **Es importante destacar que estas variables están disponibles para el controlador**, es decir, el control puede conocer, si así lo requiere, los valores de la trayectoria y el radio permitido al arrancar el ensayo.

El modelo suministrado en el bloque Simulink© no corresponde al modelo que se utilizará para la evaluación, el control suministrado para mostrar el funcionamiento trata de controlar ese modelo y no el modelo no-lineal. Se aportan exclusivamente como ejemplo para entender el sistema de ensayo. Cada participante que quiera utilizar esta utilidad debe aportar su modelo y su controlador sustituyendo los que vienen por defecto.

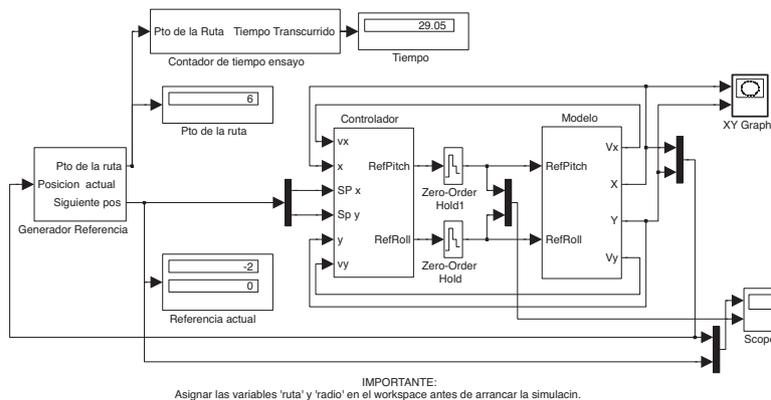
Durante la ejecución de este diagrama Simulink© es posible ver una representación xy de la trayectoria (figura *Graph XY*) y un conjunto de gráficas con las variables manipuladas y controladas del sistema (figura *Scope*).

Al finalizar la simulación aparece la variable `sdata` en el espacio de trabajo que contiene la información de la simulación. Para representar los resultados de la simulación y calcular la información necesaria para la evaluación, se suministra la función `pintasdata()`. La función tiene como parámetros de entrada los datos del ensayo (`sdata`), la ruta (`ruta`) y el radio (`radio`) y, como parámetro de salida una variable con la información necesaria para la evaluación (`indicadores`):

```
>> indicadores=pintasdata(sdata,ruta,radio)
```

Al ejecutar esta función aparecen dos figuras (ver figuras 4 y 5). La primera figura muestra las referencias y evolución para las variables x e y y por separado, la evolución de las variables manipuladas `RefPitch` y `RefRoll` y las evoluciones de la distancia instantánea, media y máxima del vehículo a la trayectoria mínima (segmento que une dos puntos consecutivos de la ruta). La segunda figura muestra la trayectoria seguida por el vehículo y la trayectoria mínima (unión de los puntos de la ruta) en el plano xy . Se han trazado además las circunferencias que marcan la zona de alcance, la trayectoria empieza en la coordenada $(0,0)$ marcada con un círculo rojo y acaba en el círculo verde (último punto de la ruta).

El `script` `simulaControl.m` automatiza todas las funciones descritas anteriormente y añade la parte de evaluación que se describirá más adelante (apartado 4). Por tanto, para evaluar una propuesta, el usuario debe sustituir el modelo y controlador por sus propuestas, abrir el `script` `simulaControl.m` para establecer la ruta que desea ensayar y ejecutarlo. Este tipo de



Asignar las variables 'ruta' y 'radio' en el workspace antes de arrancar la simulacin.

Figura 3: simuladorControlCuatrirrotor.mdl. Simulador del control. Disponible en: <http://www.ceautomatica.es/sites/default/files/upload/13/files/EvaluacionCIC2012.zip>.

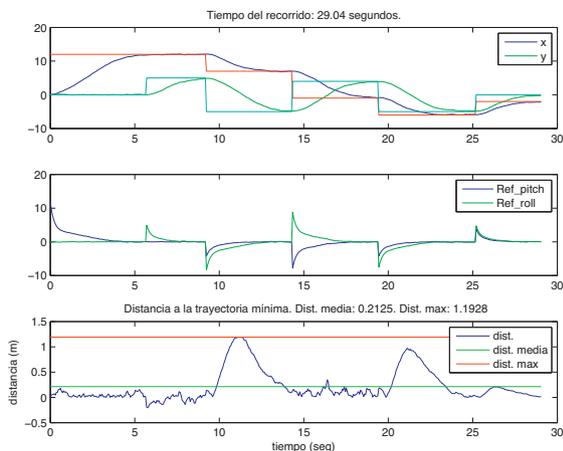


Figura 4: Figura 1 que aparece al ejecutar pintascdata(). Muestra tiempo empleado en el recorrido, las referencias, variables controladas, variables manipuladas y variables de evaluación.

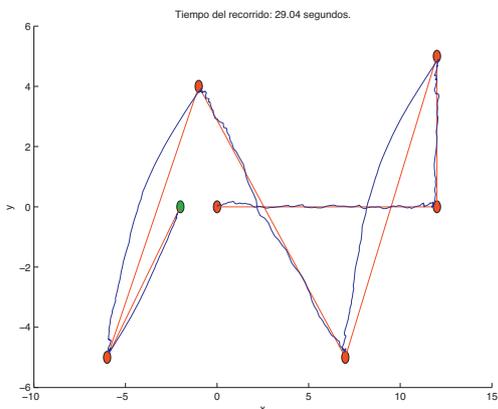


Figura 5: Figura 2 que aparece al ejecutar pintascdata(). Muestra la trayectoria XY del vehículo y de la trayectoria mínima. Muestra el tiempo empleado en el recorrido.

evaluación sólo sirve para que cada participante pueda comparar sus diferentes propuestas de control con su modelo y tener algún tipo de realimentación en los diseños. La evaluación final se realiza por parte de la organización con el modelo no-lineal.

4. Metodología de evaluación

Acorde con las bases del concurso, para valorar la calidad de las propuestas de los participantes se evaluará:

1. En el caso del modelo identificado, se medirá el error absoluto medio en las variables x e y así como el error máximo en dichas variables (todas las medidas en metros):

$$I_x^{em} = \frac{\sum_{k=1}^n |x(k)_{medida} - x(k)_{modelo}|}{n} \tag{1}$$

$$I_x^{emax} = \max \sum_{k=1}^n |x(k)_{medida} - x(k)_{modelo}| \tag{2}$$

$$I_y^{em} = \frac{\sum_{k=1}^n |y(k)_{medida} - y(k)_{modelo}|}{n} \tag{3}$$

$$I_y^{emax} = \max \sum_{k=1}^n |y(k)_{medida} - y(k)_{modelo}| \tag{4}$$

2. En el caso de la fase de control, el error medio, el error máximo (ambos en metros) y el tiempo del ensayo (en segundos):

$$C_d^{em} = \frac{\sum_{k=1}^n d(k)}{n} \tag{5}$$

$$C_d^{emax} = \max_k d(k) \tag{6}$$

$$T_{recorrido} \tag{7}$$

Dado que en ambos casos se tienen múltiples indicadores para valorar la calidad del modelo o del controlador, se puede hablar de una evaluación multiobjetivo. Ante esta situación, diversas soluciones pueden ser óptimas, y las diferencias entre ellas radican en el grado de compromiso entre sus objetivos (Marler and Arora (2004), Reynoso-Meza et al. (2009)).

Se utilizará la filosofía derivada de (*Physical Programming* Messac. (1996), Martínez et al. (2006)). Se trata de una técnica de agregación de varios requerimientos en un único valor numérico; se diferencia del empleo de un vector de pesos en que la codificación de los objetivos se realiza en una base lingüística, intuitiva y con significado para el diseñador.

Para cada uno de los objetivos, asumiendo que se pretende minimizar cada uno de ellos, se establecen unos rangos de preferencia (definidos por los valores de sus extremos) etiquetados acorde con la experiencia y preferencias del diseñador, tales como:

AD: *Altamente deseable.*

D: *Deseable.*

T: *Tolerable.*

I: *Indeseable.*

AI: *Altamente Indeseable.*

Con las definiciones anteriores, se construye una nueva función para cada objetivo (función de clase), de manera que cada uno se traslada a un dominio e imagen diferentes. En su nueva imagen, se pretende que los objetivos de diseño sean equitativamente comparables entre sí (ver Figura 6).

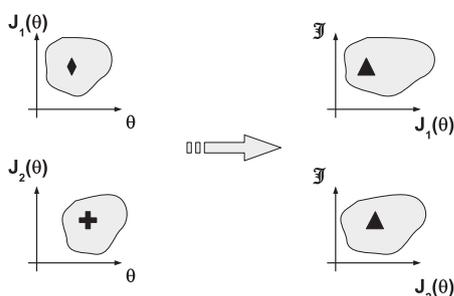


Figura 6: Concepto de la función de clase. La función de clase traslada los objetivos a un nuevo dominio, donde la imagen de los objetivos es comparable entre sí.

Mientras que el método original construye las funciones de clase con requerimientos específicos de curvatura (para poder emplear métodos de optimización no-lineal), aquí emplearemos la simplificación propuesta en Sanchis et al. (2010), denominada *Global Physical Programming*.

La función de clase se construye además para asegurar la regla heurística *One-vs-others* (Messac. (1996)). Este criterio heurístico pretende garantizar la preferencia por soluciones equilibradas y evitar que soluciones extremas (muy buenas en un indicador pero muy malas en otros) queden bien posicionadas.

La nueva función de clase es construida para cada objetivo con segmentos lineales acorde con la figura 7 y teniendo en

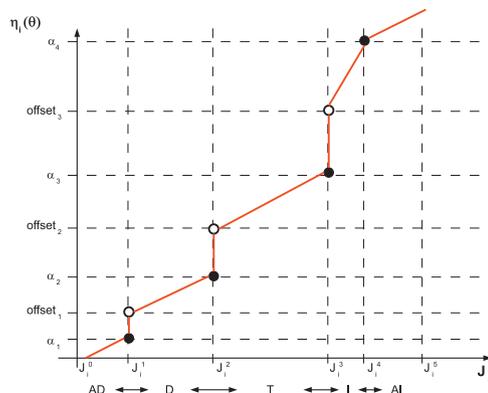


Figura 7: Función de clase propuesta.

cuenta las propiedades citadas anteriormente. Una vez construidas las funciones de clase, para obtener el indicador global basta agregar todas las funciones de clase. La función de agregación, dados m objetivos queda entonces definida como:

$$J_{gpp}(\theta) = \sum_{q=1}^m \eta_q(\theta) \tag{8}$$

donde $\eta_q(\theta) : \mathcal{R} \rightarrow \mathcal{R}$ es la función utilizada para traducir $J_q(\theta), q \in [1, \dots, m]$ en su correspondiente imagen en su función de clase. Para este caso,

$$\eta_q(\theta) = \alpha_{k-1} \cdot (m + 1) + \Delta\alpha_k \frac{J_q(\theta) - J_q^{k-1}(\theta)}{J_q^k(\theta) - J_q^{k-1}(\theta)} \tag{9}$$

$$\Delta\alpha_k = \alpha_k - \alpha_{k-1} \quad (1 \leq k \leq M) \tag{10}$$

$$J_q^{k-1}(\theta) \leq J_q(\theta) < J_q^k(\theta) \tag{11}$$

$$\alpha_0 = 0 \tag{12}$$

$$\alpha_1 = \alpha_{ini} = \frac{1}{m^2}, \quad \alpha_{ini} \geq 0 \tag{13}$$

$$\alpha_k = \alpha_{k-1} \cdot m^2 \quad (1 < k \leq M) \tag{14}$$

Donde M representa el número de rangos de clase. Para esta edición del concurso se proponen los rangos descritos en las tablas 2 y 1.

Para calcular lo anterior, se ha desarrollado la función:

```
>> [Rendimiento, Porcentajes, Zonas] =
    Rendimiento_GPP(J,Phy,Labels)
```

J: Es un vector fila con N_{ind} indicadores a evaluar. Puede aceptar varias filas de manera que se pueden evaluar varias soluciones simultáneamente de forma independiente.

Phy: Es una matriz de N_{ind} filas y 6 columnas que determinan los extremos de los rangos de preferencias para cada uno de los objetivos en J . Cada fila corresponde a un objetivo.

Labels: Es un parámetro auxiliar, donde se determina el nombre y las unidades de cada uno de los objetivos a evaluar.

Tabla 1: Preferencias para el modelo identificado (Fase 1)

| Objetivo | J_i^0 | ← AD → Altamente Deseable J_i^1 | ← D → Deseable J_i^2 | ← T → Tolerable J_i^3 | ← I → Indeseable J_i^4 | ← AI → Altamente Indeseable J_i^5 |
|---------------------------|--------------------------|--|------------------------------|-------------------------------|--------------------------------|--|
| | Distancia x(k) Media (m) | 0 | 0.20 | 0.60 | 1.00 | 1.50 |
| Distancia y(k) Media (m) | 0 | 0.20 | 0.60 | 1.00 | 1.50 | 2.50 |
| Distancia x(k) Máxima (m) | 0 | 0.50 | 1.80 | 3.50 | 5.50 | 8.00 |
| Distancia y(k) Máxima (m) | 0 | 0.50 | 1.80 | 3.50 | 5.50 | 8.00 |

Tabla 2: Preferencias para la fase de control (Fase 1)

| Objetivo | J_i^0 | ← AD → Altamente Deseable J_i^1 | ← D → Deseable J_i^2 | ← T → Tolerable J_i^3 | ← I → Indeseable J_i^4 | ← AI → Altamente Indeseable J_i^5 |
|---------------------------|---------------------|--|------------------------------|-------------------------------|--------------------------------|--|
| | Distancia Media (m) | 0 | 0.05 | 0.30 | 0.60 | 1.00 |
| Distancia Máxima (m) | 0 | 0.30 | 1.20 | 2.00 | 4.00 | 8.00 |
| Tiempo de Recorrido (seg) | 0 | $1,3t_{min}$ | $1,7t_{min}$ | $3,0t_{min}$ | $6,0t_{min}$ | $12t_{min}$ |

Rendimiento: El rendimiento global del objetivo, de acuerdo a las preferencias establecidas en Phy.

R: Rango (en porcentaje) que un objetivo tiene dentro de su preferencia.

Zonas: es la zona de preferencia donde se encuentra cada objetivo.

Esta función se ha incorporado dentro de los *script*: *simulaControl.m* y *simulaModelo.m* de manera que al ejecutar cada uno de ellos, finalizada la simulación, se obtiene el valor de rendimiento y una descripción de los indicadores que se han conseguido.

4.1. Ejemplo de evaluación de un controlador

En el fichero del simulador de control, se usa un controlador PD por defecto. Para la ruta especificada, se tiene la siguiente estructura de resultados:

```
>> indicadores
```

```
indicadores =
    ruta: [7x2 double]
    longitudRuta: 56.9207
    velMax: 3
    tiempoMin: 18.9736
    distancia: [485x1 double]
    dmedia: 0.2125
    dmax: 1.1928
    tiempoRecorrido: 29.0400
```

al pasar dichos resultados por el script de evaluación, con las preferencias especificadas anteriormente (tabla 1) se tiene:

```
[GPP, Porcentajes, Zonas] =
    Rendimiento_GPP(J,PhyMatrix,Etiquetas)
```

```
***** Rendimiento GPP *****
```

```
Vector Indicador 1 : valor GPP : 9.1813
```

```
---I1: Distancia Media = 0.2125 metros
```

```
---> D - Deseable (64.999%)
```

```
---I2: Distancia Máxima = 1.1928 metros
```

```
---> D - Deseable (99.2043%)
```

```
---I3: Tiempo de Recorrido = 29.04 segundos
```

```
---> D - Deseable (57.6377%)
```

```
GPP =
```

```
9.1813
```

```
Porcentajes =
```

```
64.9990 99.2043 57.6377
```

```
Zonas =
```

```
'D - Deseable'
```

```
'D - Deseable'
```

```
'D - Deseable'
```

El valor GPP, que es de 9.1813 en el ejemplo, puede ser empleado como criterio de evaluación para que cada participante pueda comprobar la calidad de sus distintas propuestas. Se han elegido los rangos de preferencias y las funciones de clase, de tal manera que 0 corresponde a la solución ideal (0 distancia media, 0 distancia máxima, en el tiempo mínimo de recorrido, con un modelo perfecto) y 100 corresponde a un valor al límite de la tolerancia en cada uno de los objetivos. Cualquier valor mayor a 100 tiene, al menos, un objetivo en el rango *Indeseable*.

Así mismo, como información adicional al participante, se devuelven los rangos en los que se ubican cada uno de los objetivos (*Zonas*) así como el porcentaje que alcanzan dentro del mismo. Por ejemplo, para el objetivo 1, la distancia media, tiene un rango *Deseable* al 64,99 %, lo que significa que se encuentra

en el rango *Deseable*, pero más cerca del valor límite *Tolerable* que del valor límite *Deseable*. Sin embargo, el objetivo 2, la distancia máxima, se encuentra en la zona *Deseable* al 99,2043 %, lo que significa que está próxima a entrar en el rango *Tolerable*.

5. Evaluación global

El proceso de evaluación final para clasificar a los participantes que pasarán a la fase final siguiendo los criterios que se describen a continuación.

Para la evaluación del modelo se realizará un ensayo sobre el modelo de cada participante modificando el valor de las variables manipuladas de manera que se ensayen cambios de amplitud de todo tipo (bajos, medios y elevados). Se comparará la respuesta con la del modelo no-lineal para obtener los indicadores de calidad descritos anteriormente I_x^{em} , I_x^{emax} , I_y^{em} e I_y^{emax} .

La evaluación del control propuesto por los participantes se realizará mediante un ensayo con el modelo no-lineal de la organización y con una trayectoria a seguir en el plano xy (no hay variaciones en z), tratando de evaluar el comportamiento ante distintos tipos de trayectorias. Los puntos de la trayectoria se deben alcanzar en el orden en que aparecen en la variable ruta. Para la evaluación, la variable radio tendrá un valor de 0.2 metros, es decir, se considera que el vehículo ha alcanzado un punto de la trayectoria si alcanza un disco de radio 0.2 metros y de centro las coordenadas del punto de la trayectoria en cuestión. Tanto la variable ruta como radio están disponible para el controlador cuando arranca el ensayo. A partir de este ensayo se obtendrán los tres indicadores necesarios C_d^{em} , C_d^{emax} y *Trecorrido*.

La puntuación de cada propuesta (modelo y controlador) se obtiene de la evaluación de los siete indicadores simultáneamente, utilizando la metodología descrita en la sección anterior y con las tablas 1 y 2 conjuntamente.

6. Conclusiones

El presente documento detalla el procedimiento para simular y evaluar las propuestas de modelo y control para el Concurso en Ingeniería de Control 2012: *Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor*. Asimismo se presenta una plataforma práctica para ofrecer a los estudiantes de ingeniería de control un problema real, complejo, que requiere de la integración de sus conocimientos y el trabajo en equipo.

Los participantes tendrán un acercamiento a un problema de control que demanda el cumplimiento de una serie de especificaciones y donde los resultados son evaluados mediante indicadores cuantitativos.

Para la simulación de las propuestas, se han desarrollado una serie de *scripts* y simulaciones en Matlab©/Simulink© que reproducen el comportamiento del vehículo cuatrirrotor. La evaluación es multiobjetivo (multi-criterio), ya que son varios los

indicadores a tomar en cuenta para apreciar el rendimiento de un modelo y una propuesta de control. Para construir una función que evalúe el rendimiento global de una propuesta y que sirva para determinar al ganador, se emplea la metodología basada en la técnica *Physical Programming*. Dicha técnica consiste en elaborar una función agregada de un conjunto de indicadores a través de una codificación lingüística, intuitiva y con significado para el diseñador.

English Summary

Autonomous trajectory control of a quadricopter vehicle. Simulation and evaluation.

Abstract

In this document, the simulation and multi-objective evaluation of the competition on Control Engineering 2012, *Autonomous trajectory control of a quadricopter vehicle* is shown and explained. This competition is organized by the Comité Español de Automática (CEA). This benchmark is presented as an educational platform to reinforce control education as well as to show the participants the kind of problems that a control engineer could face.

Keywords:

Control education, Helicopter control, Control system design, Process identification, Multiobjective decision.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo del Ministerio de Ciencia e Innovación (Gobierno de España) mediante la Acción Complementaria DPI2010-12026-E y la Universitat Politècnica de València a través de la beca UPV-FPI-2010/19.

Referencias

- Marler, R., Arora, J., 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* (26), 369 – 395.
- Martínez, M. A., Sanchis, J., Blasco, X., 2006. Diseño multiobjetivo de controladores pid para el benchmark de control 2008-2009. *Revista Iberoamericana de Automática e Informática Industrial* 3 (1), 39 – 51.
- Messac, A., 1996. Physical programming: effective optimization for computational design. *AIAA Journal* 34 (1), 149 – 158.
- Reynoso-Meza, G., Blasco, X., Sanchis, J., 2009. Diseño multiobjetivo de controladores pid para el benchmark de control 2008-2009. *Revista Iberoamericana de Automática e Informática Industrial* 6 (4), 93 – 103.
- Sanchis, J., Martínez, M. A., Blasco, X., Reynoso-Meza, G., 2010. Modelling preferences in multiobjective engineering design. *Engineering Applications of Artificial Intelligence*, Article in press. doi:10.1016/j.engappai.2010.07.005.