

SUPERVISIÓN DE CONTROLADORES PREDICTIVOS MULTIVARIABLES

Rachid Aref, Ghraizi. Cesar, de Prada Moraga. Ernesto, Martínez.
 Dpt. Ingeniería de Sistemas y Automática
 Facultad de Ciencias, Universidad de Valladolid
 c/ Real de Burgos s/n, 47011, Valladolid, España.
 Teléfono: 983423162. FAX: 983423161
 e_mail: rachid@autom.uva.es
 prada@autom.uva.es

Resumen

El siguiente trabajo presenta el análisis de la supervisión del comportamiento del Control Predictivo Basado en Modelos (CPBM). Se analizan diferentes metodologías para evaluar el comportamiento del controlador, determinando un sistema de referencia (Benchmark) y supervisando éste en línea. Se propone una modificación del sistema de referencia para realizar el análisis de comportamiento del CPBM en línea. El caso se estudia simulando un reactor químico, ilustrando los resultados y demostrando las limitaciones y ventajas de estos métodos.

Palabras claves: Control Predictivo, Supervisión, EcosimPro, Benchmark, Función de Coste, Simulación, Optimización.

1. Introducción

El Control Predictivo multivariable Basado en Modelo (CPBM) está siendo gradualmente aplicado en la industria para manipular procesos complejos. Su gran éxito es su capacidad de ser aplicados en la mayoría de los procesos industriales y también las ventajas de sus características tales como: su naturaleza multivariable, manejo de restricciones, manejo de perturbaciones, predicciones futuras de las variables controladas y cálculo de las variables manipuladas... etc.

Estas razones han establecido al Control Predictivo Basado en Modelo durante la última década como la estrategia del control más destacada en las aplicaciones avanzadas de control en las industrias químicas [Sckaert & Rawlings, (1996)].

Existen dos elementos claves a la hora del diseño de una estrategia de Control Predictivo Basado en Modelo, de ahora en adelante será tratado como CPBM. Estos elementos son la identificación del modelo del proceso a controlar y la selección de los parámetros de la sintonía del controlador, donde un

uso inadecuado de estos elementos, junto con la influencia de otros factores internos o externos pueden desviar el controlador del punto óptimo de funcionamiento para el cual fue diseñado. Por lo tanto para garantizar su buen funcionamiento se necesita un método de supervisión que revele en cada instante de tiempo la eficacia de los resultados obtenidos indicando como se está comportando el controlador durante el funcionamiento en línea.

Esta evaluación es de gran importancia, si en la supervisión del comportamiento se revela un comportamiento inadecuado, entonces es importante establecer las causas y sugerir acciones para mejorar este comportamiento y mantener el proceso en condiciones óptimas de funcionamiento asegurando así un control adecuado.

Las etapas que se deben seguir para analizar el comportamiento de un controlador se dividen en tres partes y en la figura 1 se puede observar una representación gráfica de las mismas:

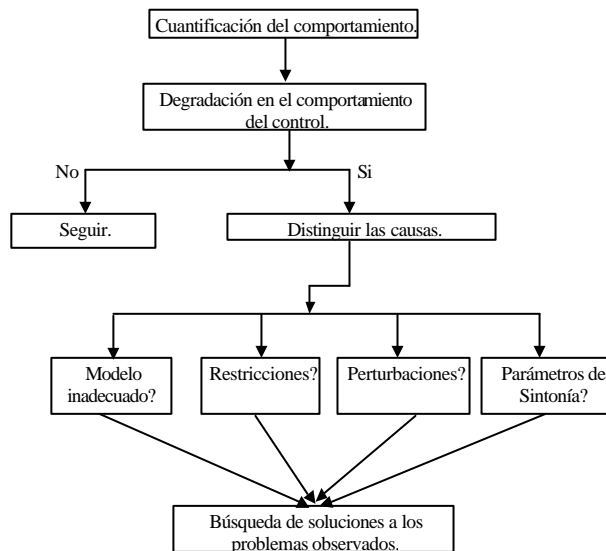


Figura 1: Diagrama representativo de las etapas del análisis del comportamiento del controlador.

- Primera etapa:** Evaluación del comportamiento, esta etapa cuantifica el comportamiento y trata de

definir y evaluar varios métodos para decidir si el sistema de control devuelve comportamientos satisfactorios o no, es decir, que discriminen si el problema reside en cambios o perturbaciones que sufre el proceso o bien se trata verdaderamente de problemas en el controlador y su sintonía.

2. **Segunda etapa:** Etapa de diagnóstico de los comportamientos inadecuados, si en la etapa anterior se indica malos comportamientos en el controlador, lo lógico es establecer sus causas y decidir si estos son debidos a:

- ☒ Modelo inadecuado.
- ☒ Restricciones.
- ☒ Parámetros de Sintonía.
- ☒ Perturbaciones.

3. **Tercera etapa:** Establecimiento de métodos y pautas para corregir los comportamientos pobres y restaurar el control según las causas:

- ☒ Si la causa de la degradación es el modelo, entonces deberá procederse a una identificación en línea o fuera de línea y determinar un nuevo modelo que cumpla con los requisitos necesarios.
- ☒ Si las causas están ligadas a la sintonía deberán proponerse pautas de cambio de horizontes, métodos o pesos para poder conseguir un control mejor. Estos parámetros se deben seleccionar como un compromiso eficiente entre el comportamiento de la trayectoria y la agresividad del control.
- ☒ Si las causas son las restricciones deberán proponerse métodos de flexibilización de las mismas.
- ☒ Si las causas son las perturbaciones se debe tratar de compensarlas de una manera más eficiente y encontrar un modelo que describa las perturbaciones de una manera más adecuada.

La supervisión y evaluación del comportamiento de controlador son necesarias debido a que muchos factores pueden causar deterioro abrupto o gradual del comportamiento de controladores. A menudo es difícil analizar los problemas de diagnósticos y supervisar el comportamiento con datos tomados de la planta sin tratamiento [Kozub, 1997]. Entonces, las herramientas del análisis estocásticas y estadísticas tienen que ser formuladas adecuadamente para detectar cambios estadísticamente significantes.

A pesar de que este tema es reciente y la información en la literatura no es muy amplia, se han propuesto algunos métodos para la supervisión del CPBM que incluyen:

la comparación del comportamiento real con el comportamiento óptimo estimado resolviendo el problema de LQG [Huang y Shah,1999]. La comparación del comportamiento del control real con el comportamiento histórico que usa el valor esperado

de la función de coste del CPBM usando una ventana de tiempo donde se sabe a priori que el controlador funciona bien [Patwardhan et al.,1998]. Estos autores también plantean la evaluación del comportamiento del controlador utilizando la función de coste objetivo del controlador y la del proceso. Otros trabajos fueron desarrollados por Zhang y Henson los cuales desarrollan su trabajo comparando valores de la función objetiva de la salida del modelo lineal de proceso y la salida de la planta real [Zhang y Henson,1999]. Recientemente se ha dado un nuevo enfoque al tema donde el trabajo no se limita solo a supervisar el comportamiento sino también a monitorear y determinar causas del mal comportamiento [Jochen Schaferand, Ali Cinar 2002]

Nuestro estudio está enfocado a evaluar el comportamiento del CPBM usando algunos de los métodos anteriores. El diagnóstico y supervisión se limita a evaluar el comportamiento del controlador en diferentes situaciones sin entrar por el momento en los pasos de distinguir las causas y proponer soluciones a los problemas que causaron la degradación del controlador. Se usan estudios del caso basados en un modelo del reactor químico para ilustrar la metodología propuesta. Se propone una modificación al índice de referencia y se realiza la simulación con el programa EcosimPro.

2. Métodos para evaluar el comportamiento del CPBM.

La metodología del CPBM se basa en la optimización en tiempo real de la siguiente función de coste presentada en la ecuación 2.1

$$J_{(t)}^* = \sum_{sal=1}^{NSAL} \sum_{j=1}^{N2} \hat{y}_{(sal)(k)} - \hat{w}_{(sal)(k)} + \sum_{ent=1}^{NENT} \sum_{j=1}^{Nu} u_{(ent)(k)} \quad (2.1)$$

donde: $k = t+j$, y \hat{W} , \hat{Y} , u^* son los vectores de la referencia interna del controlador, la predicción de la salida del proceso y el vector óptimo de la señal de control en el tiempo t, *NENT*, *NSAL* son numero de entradas y salidas, *Nu*, *N2* horizontes de control y de predicción respectivamente, \hat{Y} y \hat{W} son las matrices de peso de las variables controlada y manipulada *sal* número de salidas.

Esta ecuación incluye información sobre los horizontes de predicción y control. El controlador calcula los movimientos de control óptimos minimizando la función objetivo sobre los posibles movimientos del control, [Patwardhan R. et al, 1998].

$J^*(t)$ es el valor de diseño el cual se quiere lograr

durante el control, mientras la función de coste real es una variable aleatoria que se ve afectada por las perturbaciones y ruidos que afectan al proceso [Jochen Schaferand, Ali Cinar, 2002].

Si a $J(t)$ la dominamos como la función de coste deseado $J_{des}^*(t)$, la función de coste del proceso podemos llamarla, función de coste real $J_{real}(t)$, y al índice de referencia para evaluar el comportamiento del controlador (benchmark) ξ , el cual se calcula mediante la ecuación 2.2 de la siguiente forma:

$$\xi(t) = \frac{J_{des}^*(t+N_2)}{J_{real}(t)} \quad (2.2)$$

El valor esperado de este índice es: $E(\xi(t)) \approx 1$ donde E es la esperanza.

2.1 Enfoque del caso de diseño y del proceso.

Patwardhan R. et al., proponen la comparación de la función de coste deseada con la función de coste real para obtener un sistema de referencia (Benchmark) ξ que revele el comportamiento del controlador [Patwardhan R. et al.,1998], en la figura 2 se puede ver una representación grafica de este método.

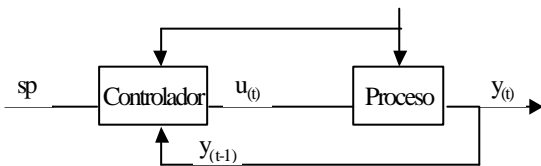


Figura 2: Representación gráfica del enfoque del caso de diseño y del proceso.

La función de coste deseada J_{des} tiene la misma forma que la ecuación 2.1. Su cálculo se realiza con datos del controlador que se obtiene en el instante actual como consecuencia de las predicciones futuras de las variables del proceso y las variables de control. El resultado de J_{des} se debe guardar para ser utilizado una vez pasados N_2 periodos de muestreo junto con la función de coste real que se calcula en el tiempo $t+N_2$ usando datos del proceso obtenidos en los últimos $t-N_2$ periodos de muestreos. El cálculo de la $J_{real}(t)$ se describe en la siguiente ecuación 2.3:

$$J_{real}(t) = \sum_{sal=1}^{NSAL} \sum_{j=1}^{N_2} y_{(sal)(k)} \hat{w}_{(sal)(k)} \sum_{ent=1}^{NENT} \sum_{j=1}^{Nu} u_{(ent)(k)} \quad (2.3)$$

donde $k = t + j - N_2$ e y es la salida del sistema, y el resto de los parámetros son los mismos explicados anteriormente en la ecuación 2.1. El cálculo de la función de coste real J_{real} se realiza utilizando los datos de las variables del proceso que se han obtenido

durante los últimos $t-N_2$ periodos de muestreo y las variables y parámetros de la sintonía del controlador en el mismo tiempo. En la Figura 3 se puede ver gráficamente la explicación anterior y como se realiza el análisis una vez pasado N_2 pasos de muestreo, ya que es el horizonte mínimo necesario para poder obtener informaciones del proceso real.

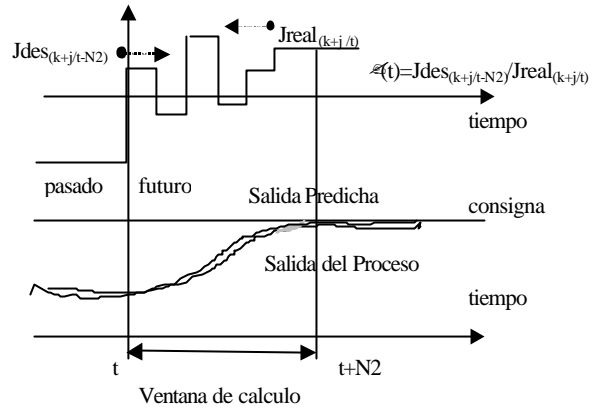


Figura 3: Representación gráfica del calculo de las funciones objetivos en el tiempo.

Una vez obtenidos estas dos funciones objetivos que utilizan informaciones del proceso real e informaciones de los resultados de la predicción del controlador, se puede conseguir el índice del comportamiento a través de una relación entre el diseño óptimo y la función objetivo real que se denotó como ξ y se calcula como en la ecuación 2.2.

2.2 Enfoque del caso del modelo lineal y del proceso.

Zhang y Henson en su trabajo proponen el cálculo del índice de referencia ξ realizando la comparación entre el proceso real y el modelo lineal para determinar el comportamiento del controlador [Zhang y Henson,1999]. La figura 4 resume gráficamente este método y representa en un esquema sus estructuras básicas.

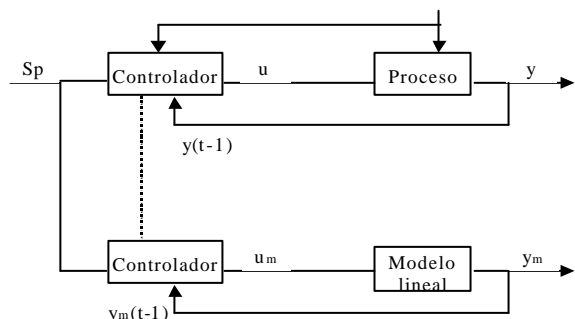


Figura 4: representación gráfica del enfoque del caso del modelo lineal y del proceso.

A diferencia del método anterior, el sistema de referencia ξ se obtiene con la función de coste

deseada (J_{des}) cuando la acción de control se aplica al modelo lineal del proceso en vez de la planta y no se consideran las perturbaciones en el modelo lineal. Teniendo en cuenta en este caso, que a la hora de la simulación, los parámetros de la sintonía del controlador y las consignas deben ser las mismas para ambos en cada instante de tiempo.

Las funciones de costes serán calculadas de la misma manera que la ecuación 2.3 en el epígrafe anterior, es decir, se van guardando los datos necesarios tanto para el proceso como para el modelo lineal y una vez pasado N_2 periodos de muestreos se pueda calcular las funciones de costes.

J_{real} se obtiene como se ve en la ecuación 2.4. La forma de cálculo es similar al de la ecuación 2.3, pero aquí no se consideran en este análisis la influencia de las entradas.

$$J_{real(t)} = \sum_{sal=1}^{NSAL} \sum_{j=1}^{N_2} y_{(sal)(k)} - \hat{y}_{(sal)(k)} \quad (2.4)$$

donde $k = t+j-N_2$, \hat{y} es vector de predicciones de las salidas en la planta real, el resto de los parámetros son los mismos explicados anteriormente.

sin embargo, como se explicó antes la función de coste deseada o J_{des} se obtiene cuando la acción de control se aplica al modelo lineal del proceso en vez de la planta. Es decir los datos para el calculo de J_{des} son del modelo lineal y su cálculo es similar a J_{real} de la ecuación 2.4, a continuación se presenta el cálculo de la J_{des} en la ecuación 2.5.

$$J_{des(t)} = \sum_{sal=1}^{NSAL} \sum_{j=1}^{N_2} y_{(sal)(k)}^m - \hat{y}_{(sal)(k)}^m \quad (2.5)$$

donde $k = t+j-N_2$. y^m y \hat{y}^m son los vectores de las salidas del modelo lineal y su predicción respectivamente. Una vez obtenidos estas dos funciones objetivos del proceso y del modelo lineal se consigue el índice del comportamiento a través de una relación entre ambos y se denotó como ρ , su cálculo se ve en la ecuación 2.2. la figura 5 presenta un demostración gráfica del cálculo de este método. Según los autores, el método no proporciona información sobre el efecto de los parámetros de la sintonía del CPBM. También según ellos el valor esperado de esta función debe ser uno cuando el modelo es perfecto y en ausencia de perturbaciones y ruidos. Pero no aporta informaciones significativas en caso contrario. Por lo que acuden a un análisis estadístico de la misma.

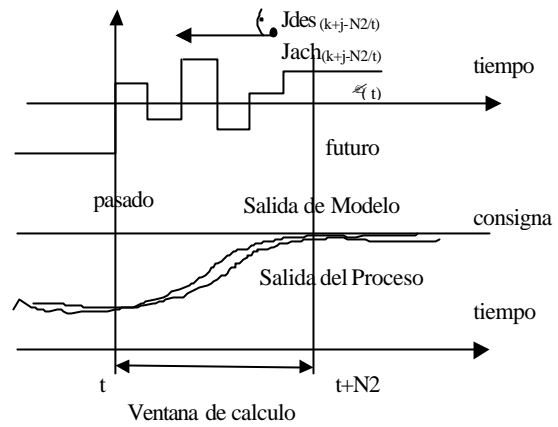


Figura 5: Representación gráfica del calculo de las funciones objetivos en el tiempo.

Zhang y Henson identifican ρ como una variable estocástica proponen una evaluación estadística para detectar cambios estadísticamente significativos en el comportamiento del controlador. Según los autores $\rho_{(t)}$ es una secuencia altamente correlacionada, por lo que proponen usar un modelo ARMA para su identificación como lo que se presenta en la ecuación 2.6.

$$A(q^{-1})\rho_{(t)} = C(q^{-1})r_{(t)} \quad (2.6)$$

Donde:

ρ es una variable gaussiana aleatoria de media cero y no correlacionada, $A(q^{-1})$ y $C(q^{-1})$ son los polinomios estimados utilizando técnicas estándar de identificación y q^{-1} es el operador de retardo. Estos coeficientes con la varianza se pueden identificar con métodos estándares de identificación. Pero debido a que $\rho_{(t)}$ es una secuencia altamente correlacionada, proponen usar un modelo AR en vez de ARMA $(1 - a(q^{-1}))r(t) = x(t)$ y por ultimo definen $Dr(t)$ como en la ecuación 2.7 y analizan su varianza.

$$Dr_{(t)} = \frac{\hat{A}(q^{-1})}{\hat{C}(q^{-1})} r_{(t)} \quad (0.1)$$

la estimación de la varianza de ruido se usa para determinar el 95% del limite de la confianza de ρ .

3. Método modificado del análisis del comportamiento del controlador.

El enfoque del caso de diseño probablemente no es aplicable a muchos de los casos reales, ya que los autores no consideran en sus análisis las perturbaciones no medibles que pueden afectar al proceso, que en casos reales existen y su consideración es de suma importancia. En la simulación de este método se ha observado que en ausencia de perturbaciones no medibles el

comportamiento del controlador es adecuado, mientras en la presencia de estos no aporta informaciones adecuada, el comportamiento se degrada notablemente a pesar de que el controlador funciona de manera apropiada, lo que limita este método. Mas adelante veremos una simulación de estos dos casos.

El método del enfoque del caso de diseño está limitado a un caso especial de estudio, para evitar esta inconveniente proponemos cambio en el cálculo del índice original por el índice que se indica en la ecuación 3.1:

$$\frac{J_{des}(t) - J_{real}(t)}{J_{des}(t)} \quad (3.1)$$

donde n es igual a N2 en el caso del enfoque del caso de diseño y del proceso y cero en el método del enfoque del caso del modelo lineal y del proceso.

El calculo de las funciones de coste J_{des} , J_{real} se realiza de la misma manera que en el método del enfoque del caso de diseño, el valor esperado de este nuevo índice debe ser cero en los casos óptimos, ya que se trata de la diferencia entre la función de coste esperada y la real, pero como esta función es una función aleatoria debido a las perturbaciones, el valor real de ella será próximo a cero, y se considera un valor adecuado un valor que no esté muy alejado de cero. Los resultados obtenidos que serán mostrados en el epígrafe 5, han sido satisfactorios.

Para el método del enfoque del caso del modelo lineal y del proceso, se ha realizado el mismo cambio en el cálculo de J , ya que pasaba lo mismo que en el caso anterior, donde consideramos que las informaciones que aportaba este método no eran suficientes para el análisis, ya que como se va a poder ver en las gráficas mas adelante J da casi cero después de pocas simulaciones y no cambia aun aumentando las perturbaciones o cambiando los parámetros de sintonía y es lógico ya que no se consideran perturbaciones en el modelo lineal y J_{des} se hace muy pequeña.

4. Descripción y simulación del proceso.

Los métodos para la evaluación del comportamiento del CPBM fueron aplicados a un reactor químico donde se le introduce un flujo a la entrada, y debido a la reacción química dentro del reactor se produce un nuevo componente que será devuelto a la salida junto con parte del componente original que no fue reaccionada. Debido al carácter exotérmico de la reacción se introducirle un flujo refrigerante a la camisa para mantener la temperatura del proceso en el punto de operación, generando un cambio en la temperatura final del mismo. Los flujos de masa se mantienen iguales, ya que, no hay pérdida durante la

reacción. En la Figura 6 se puede ver una representación gráfica del reactor, que tiene las siguientes características:

- Tres variables controladas: c_b Concentración del producto B a la salida, t_l Temperatura en el reactor a la salida, c_a Concentración del producto A a la salida,
- Dos variables manipuladas f_l Flujo de entrada de producto A, f_r Flujo refrigerante,
- Tres mediciones de Perturbaciones medibles c_{a0} Concentración de entrada del producto A, t_{l0} Temperatura del flujo de entrada, t_{r0} Temperatura de entrada del refrigerante.

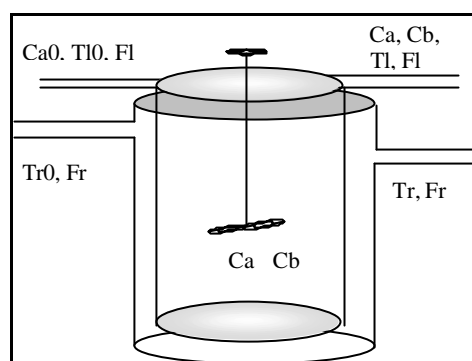


Figura 6: Representación gráfica del reactor químico.

4.1. Implementación del modelo en EcosimPro.

Para poder realizar la simulación y evaluar del comportamiento del CPBM, se ha realizado la implementación en EcosimPro del modelo matemático del proceso en cada uno de los casos, lineal y no lineal necesarios para poder llevar a cabo las simulaciones y estudiar el comportamiento según los métodos anteriores, por lo que se ha llamado desde EcosimPro a cada una de las funciones necesarias programadas en C++ que intervienen en el control del proceso o en la supervisión del controlador creándose una librería independiente para cada caso.

Para poder realizar pruebas en simulación del proceso con el objetivo de ver el comportamiento del controlador, por ejemplo ante cambios en la referencia, en sus parámetros o en las perturbaciones del proceso, fueron incluidas en el bloque continuo del componente *react_proceso* de la librería *react_proceso.el* las ecuaciones diferenciales que servirán como planta real, a continuación se muestra un resumen de estas:

```
COMPONENT react_proceso(INTEGER nsa=3,
INTEGER nent=2, INTEGER nper=3, INTEGER
futuro=40)
```

DATA

...

DISCRETE

```

WHEN ( sample == TRUE ) THEN
....
CONTINUOUS
    Ca' = f(FI, Ca0, VI, k)
    Cb' = f(FI, Ca, VI, k, Cb)
    Tl' = f(FI, Ca, VI, Tl0, Tl, k, H)
    Tr' = f(Fr, Tr0, Tr, T, Vr)
....
END COMPONENT

```

El modelo lineal del proceso ha sido programado en el bloque discreto del componente *react_modelo* de la librería *react_modelo.el*, el código se muestra a continuación:

```

COMPONENT react_modelo (INTEGER nsal=3,
INTEGER nent=2, INTEGER nper=3, INTEGER
futuro=40)
DATA
....
DISCRETE
    WHEN ( sample == TRUE ) THEN
....
    Cb_mod      =      3.0534*Cbmod_old[2] -
3.369994*Cbmod_old[3] \.... etc.
    Tl_mod      =      3.5111*Tlmod_old[2] -
4.6254905*Tlmod_old[3] \... etc.
    Ca_mod      =      3.6213*Camod_old[2] -
5.1413589*Camod_old[3] \... etc
....
END COMPONENT

```

Para poder llevar a cabo estas simulaciones se ha recurrido al programa C++ para programar las funciones necesarias del controlador, utilizando el código fuente del programa HITO el cual facilitó la aplicación del controlador predictivo y el cálculo de las funciones de costes.

Al principio y en cada una de las librerías se declararon las funciones de C++ que posteriormente serán llamadas en la parte discreta. Algunas de estas funciones se muestran a continuación:

```

"C" FUNCTION REAL regulador (IN REAL y[], IN
REAL u[], IN REAL v[], IN REAL ref[], IN REAL
y_predic[], IN REAL res_free[], IN INTEGER
predic_ok[], IN REAL err_y_predic[], IN REAL
ref_int[], IN REAL u_futura[], OUT REAL ro)

```

```

"C" FUNCTION NO_TYPE ini_hito (IN INTEGER
NSAL, IN INTEGER NENT, IN INTEGER NPER,
IN REAL y00[], IN REAL u0[], IN REAL v0[], IN
INTEGER tipo_modelo, ...etc)

```

```

"C" FUNCTION NO_TYPE reini_hito (IN
INTEGER cambio, IN INTEGER cambio_N1, IN
INTEGER cambio_N2, IN INTEGER cambio_N3,
IN INTEGER cambio_N4, IN INTEGER

```

```

cambio_NU, IN INTEGER cambio_res, IN
INTEGER cambio_modelo, ...etc)

```

En los dos casos la parte inicial deberá incluir la llamada a una función de inicialización del controlador (*ini_hito*) que realiza el paso de ciertos parámetros necesarios en la ejecución del regulador (como los horizontes de predicción – *N1*, *N2* y de control – *Nu*, los pesos de la función de coste – *gama*, *beta*, los límites de las variables – *Linf*, *Lsup*, *Uinf*, *Usup*, *Dinf*, *Dsup*, etc.) y también asegura la inicialización del experimento asociado a la partición de cada componente

Dentro de la zona discreta se hace la llamada a la función *regulador* en cada periodo de muestreo (*tsamp*) para conseguir el nuevo valor de la variable manipulada que se debe aplicar al proceso y el cálculo del índice de referencia ? .

```

DISCRETE
....
WHEN ( sample == TRUE ) THEN
....
regulador (y, u, v, refy, y_predic, y_libre, predic_ok,
err_y_predic, ref_int, u_futura, ro)
....
sample = FALSE
sample = TRUE AFTER tsamp
END WHEN

```

De una manera similar dentro de la zona discreta se realizan las llamadas a las otras funciones necesarias. El código C++ del controlador, para poder ser utilizado desde el entorno EcosimPro, se ha incluido como objeto externo en el momento en que se crea las particiones asociadas a los componentes junto con la otra librería utilizada (la Nag) para indicar el sitio de donde se puedan acceder las funciones utilizadas.

4.2. Aplicación de la simulación en los casos específicos.

En el método del enfoque del caso de diseño y del proceso, la simulación y la llamada al controlador para el cálculo del sistema de referencia ? no originaba complicaciones ya lo que se necesita era solo datos del proceso y del controlador donde en cada periodo de muestreo se realizaban las llamadas a las funciones necesarias y se actualizaban de los datos.

Sin embargo, el cálculo del índice de referencia presentado en el método del enfoque del caso del modelo lineal y del proceso del epígrafe 2.2, necesitaba datos del modelo lineal y del proceso, estos últimos y como se ha dicho antes se presentan en las librerías de EcosimPro como *reactor_modelo* y *reactor_proceso*. El análisis del comportamiento del

controlador exige que los parámetros de sintonía de los controladores sean los mismos así como las referencias del proceso y del modelo lineal y cualquier cambio se refleja en los dos.

Para asegurar estas condiciones era necesario permitir que se realizase intercambios y transferencias de datos entre los dos componentes haciendo que cada uno tome o cede los datos necesarios al otro. La política a seguir era que en cada periodo de muestro, cada componente y de forma individual chequease primero si es su turno, si fuese así, tomaba el permiso y deshabilitaba el permiso al otro componente hasta tanto haya terminado y actualizado los datos y así cediese el permiso para que se pueda enterar el otro componente que está en espera. En el momento del chequeo si el componente encuentra que no tiene el permiso tenía que espera a que el otro se lo entregase. De nuevo con la ayuda de C++ y dentro del código fuente del programa HITO se programaron las funciones necesarias que serían llamadas en las zonas discretas para llevar a cabo la operación.

Esta operación necesitó tres funciones básicas guardar los datos y parámetros, actualización de los datos y parámetros y chequeo del turno. En esta última operación chequea la existencia de un fichero que se crea cuando un componente toma el turno y se borra cuando el componente termina su trabajo para que así el otro pueda seguir. Cuando un componente encuentra este fichero espera a que el otro lo elimine, y una vez eliminado toma el turno.

Como se hizo anteriormente, las funciones de C++ que serían llamadas durante la simulación fueron declaradas global en cada librería. A continuación presentaremos un resumen de cómo fueron declaradas y utilizadas estas funciones.

Las declaraciones de las funciones que daban el turno a un componente o se lo quitaba fueron de la siguiente manera.

```
"C" FUNCTION NO_TYPE borrar_arch ()
"C" FUNCTION NO_TYPE crea_arch ()
"C" FUNCTION NO_TYPE chequea_arch (OUT
INTEGER stado_arch)
```

las funciones que daban o quitaban el turno a un componente fueron escritas como se muestra a continuación:

```
COMPONENT react_X (INTEGER nsal=3,
INTEGER nent=2, INTEGER nper=3, INTEGER
futuro=40)
DATA
....
DISCRETE
    WHEN ( sample == TRUE ) THEN
        chequea_arch (stado_arch)
```

```
WHILE (stado_arch == 0)
    chequea_arch (stado_arch)
END WHILE
```

```
-- Llamada al controlador predictivo
....
borrar_arch()
END COMPONENT
```

La actualización de parámetros lo realizaba cada componente una vez tomado el turno, es decir, después de que el otro componente guardara los parámetros y terminaba su turno,

```
guarda_param (ref_Cb, ref_Tl, ref_Ca, cambio,
cambio_N1, cambio_N2,...etc)
```

```
actua_param (ref_Cb, ref_Tl, ref_Ca, cambio,
cambio_N1, cambio_N2,...etc).
```

finalizando este epígrafe añadimos que como es obvio las variables y los datos necesarios fueron declarados como DECLS o DATA según se necesitaba que el tipo de dato.

5. Resultado de la simulación.

En este epígrafe se presentan los resultados del índice de referencia simulando de los métodos explicados anteriormente y su modificación.

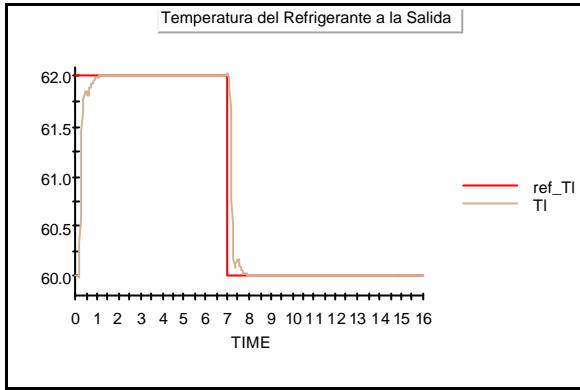
Primero vamos a presentar los resultados del índice de referencia obtenidos utilizando el método del enfoque del caso de diseño y del proceso, luego con los mismos datos y sin realizar cambios en los parámetros de sintonía del controlador presentaremos los resultados del método modificado, seguidamente comparamos los dos métodos y comentamos los resultados. Después de este análisis se procede a analizar método el método del enfoque del caso del modelo lineal y del proceso.

Para realizar esta simulación se consideraron los siguientes datos:

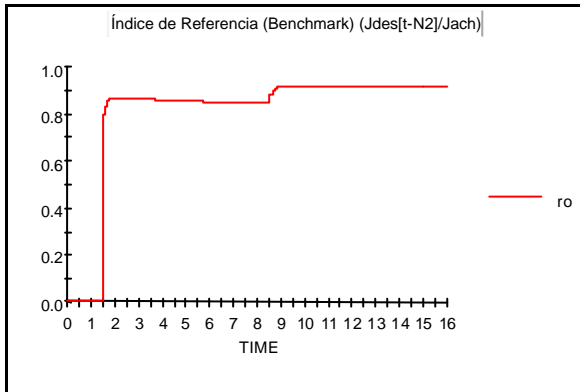
```
Controlador GPC
Horizontes de predicción N2 = { 25, 25, 25 }
Horizonte de control NU = { 2, 2 }
Limites físicos inferior Linf = { 6.7, 57.0, 0.0 }
Limites físicos superior Lsup = { 8.3, 72.0, 1.0 }
Los factores de peso relativo al cumplimiento de
cada consigna - gamma = {3.0, 3.0, 0.1}
Los factores de peso que penalizan los cambios
en las variables manipuladas - beta = {0.5, 0.5}
Puntos de consigna
Cb = 7.2, Tl = 62/60, Ca = 0.8
```

En este caso durante la simulación solo se realizará cambios en la consigna. Desde el inicio de la simulación y hasta los 7 horas se fija la consigna de la

temperatura del refrigerante a la salida en 62 y a partir de allí y hasta el final de la simulación se cambia a 60 como se ve en la gráfica 1.

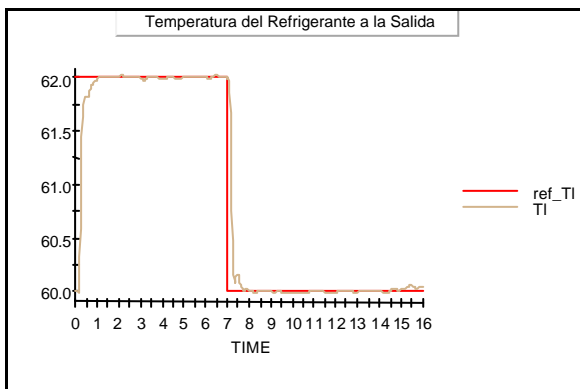


Gráfica 1: Temperatura del refrigerante a la salida del proceso.



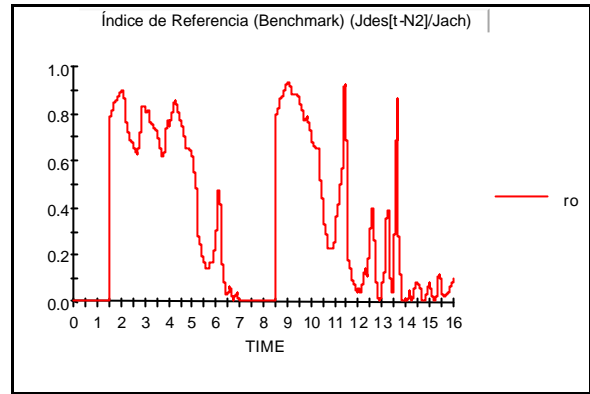
Gráfica 2: Índice de referencia sin perturbaciones en el proceso.

En las gráficas 1 podemos ver la temperatura del refrigerante a la salida del proceso sin considerar las perturbaciones y en la gráfica 2 observamos el comportamiento del controlador con el método del enfoque del caso de diseño y del proceso. En este método tenía que ser uno para que el comportamiento sea el óptimo, en ausencia de perturbaciones se ve que el comportamiento está muy cerca de uno indicando un resultado adecuado.



Gráfica 3

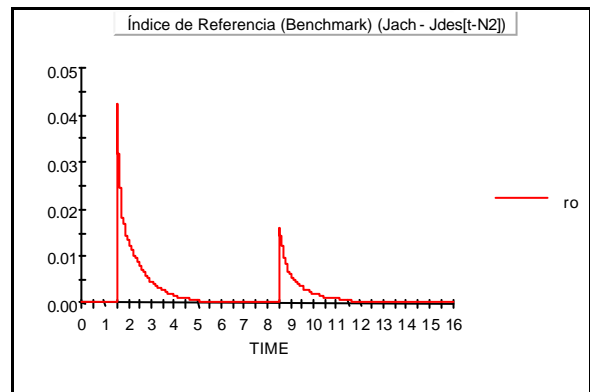
En las gráficas 3 podemos observar la temperatura del



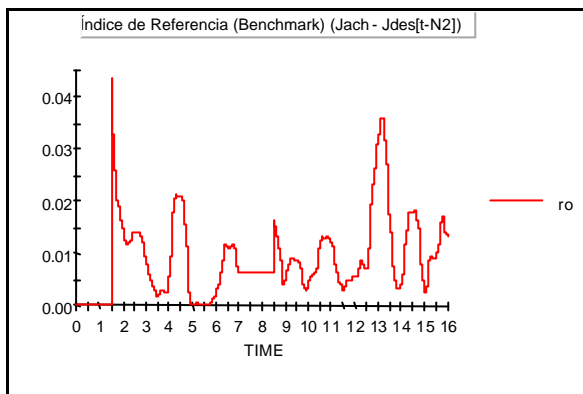
Gráfica 4 Índice de referencia con perturbaciones en el proceso.

refrigerante a la salida del proceso pero con perturbaciones que fueron aplicadas a la entrada del proceso, en la gráfica 4 observamos el comportamiento del controlador con mismo método anterior pero con perturbaciones, se observa que a diferencia de los resultados de la gráfica 2 la presencia de perturbaciones ha degradado el comportamiento mientras que en la realidad el controlador está funcionando adecuadamente y las variables controladas están alcanzando la referencia correctamente, por lo que esta manera de calcular el índice de referencia no aporta informaciones aptas para tener idea del comportamiento del controlador bajo ruido en el proceso.

Las gráficas 5 y 6 representan el resultado del comportamiento del controlador sin y con la presencia de perturbaciones respectivamente, pero aplicando el método del enfoque del caso de diseño y del proceso modificado. En la gráfica 5 observamos el comportamiento del controlador sin la presencia de ruido y como es debido el valor del índice va a cero, en la gráfica 6 se nota la presencia de ruidos ya que el valor del índice no es exactamente cero, pero si se mantiene en un valor pequeño cercano a cero, lo que justifica que esta forma de implementar este índice es más adecuada que la anterior.

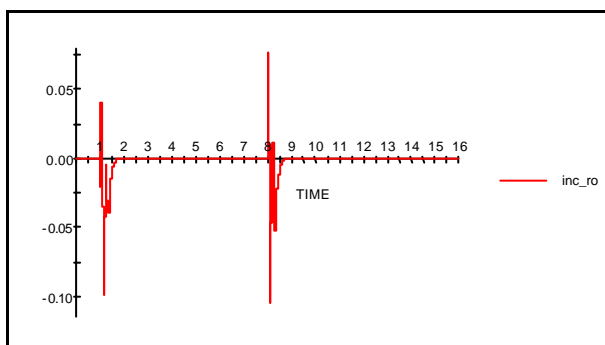


Gráfica 5: Índice de referencia modificado sin perturbaciones.

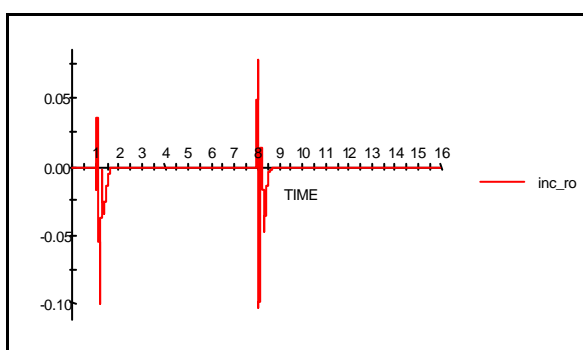


Gráfica 6: Índice de referencia ? modificado con perturbaciones.

Ahora vamos a presentar los resultados del índice de referencia obtenidos utilizando el método del enfoque del caso del modelo lineal y del proceso, luego como se hizo al principio, con los mismos datos y sin realizar cambios en los parámetros de sintonía del controlador presentaremos los resultados del método modificado, seguidamente comparamos los dos métodos y comentamos los resultados.

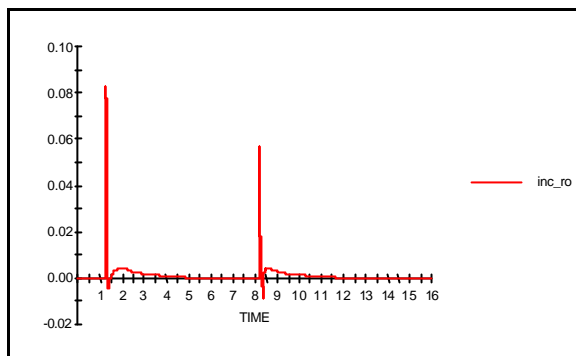


Gráfica 7: Incremento de índice de referencia ? sin perturbaciones.

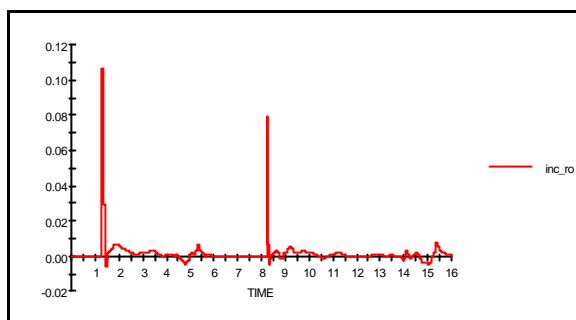


Gráfica 8: Incremento de índice de referencia ? con perturbaciones.

las gráficas 7 y 8 presentan el resultado de la simulación de ?? sin y con perturbaciones respectivamente, como se observa no hay cambios significativos entre las dos gráficas aun habiendo perturbaciones que afectan al proceso en el caso de la segunda.



Gráfica 9: Resultado de ?? modificado (caso sin perturbaciones).



Gráfica 10: Resultado de ?? modificado (caso con perturbaciones).

Las últimas dos gráficas 9 y 10 presenta los resultado de ?? modificado, aquí se puede notar que los resultados de las dos gráficas son diferentes, y es obvio, ya que en los resultados de la gráfica 10 hubo perturbaciones que afectaron al.

Por último, lo presentado en nuestro trabajo sobre los métodos modificados, se limita por el momento a estudiar el comportamiento del controlador predictivo reflejado en los resultados de las gráficas presentadas a lo largo de este trabajo. No se presenta la influencia de cambios en los parámetros de sintonía del controlador en su comportamiento aunque han sido probadas y han mostrado resultados satisfactorios.

6. Conclusiones.

Se han presentado dos métodos para la supervisión del comportamiento del Control Predictivo Basado en Modelos y se ha hecho una modificación a los mismos. Se ha realizado la simulación de los métodos explicados y a su versiones modificadas para evaluar el comportamiento del CPBM.

Se ha elegido como simulador la herramienta EcosimPro por ser aplicable a cualquier problema que pueda ser formulado con ecuaciones diferenciales y/o algebraicas y eventos discretos y también por la facilidad que ofrece de integrar el modelo con otros módulos de software, en concreto otras aplicaciones escritas en C++.

Se ha aplicado un controlador GPC para el control, utilizando para ellos el código fuente del programa

HITO.

Se han presentado los resultados de la simulación de la supervisión del Controlador Predictivo y se han comentado.

Referencias

- [1] Huang Biao, Kadali Ramesh, Zhao Xia, Tamayo, Edgar C., Hana Ahmed., (2000) An investigation into the poor performance of a model predictive control system on an industrial CGO cooker, *Control Engineering Practice*, 8, pp 619-631.
- [2] Huang, B. and S.L. Shah. (1999) *Performance Assessment of control loop*. Springer-verlag, London.
- [3] Huang L., Shah S. L., Kwok E., (1997). Good, bad or optimal? Performance assessment of multivariable process. *Automatica*, 33, pp 1175-1183.
- [4] Huang L., Shah S. L., Fujii H., (1997) The unitary interactor matrix and its estimation using closed-loop data. *Journal of Process Control*, 7, pp 195-207.
- [5] Jochen Schaferand, Ali Cinar. 15th triennial congress, Balcelona, Spain.
- [6] Kozub, D. J. (1999) controller performance monitoring and challenges. *CPC V Proceeding*.
- [7] Patwardhan R. S., Shah S. L., GenichiEmoto, (1998) Performance analysis of model_based predictive controller: An industrial case study. *AIChE Annual meeting*.
- [8] Scokaert P. O. M., Rawlings J. B., (1996) On infeasibilities in model predictive control. *Proceedings of CPC-V*. Tahoe City, CA , pp 331-334,
- [9] Zhang Y., Henson M. A., (1999) A performance measure for constrained model predictive controllers. *European control conference*.