

PRIMERAS EXPERIENCIAS DEL COPROCESADOR ORBEX EN EL CONTROL DE VEHÍCULOS SIN CONDUCTOR

Ricardo García Rosa

Instituto de Automática Industrial, CSIC
La Poveda, 28500 Arganda del Rey, Madrid
e-mail: ricardo@iai.csic.es

Teresa de Pedro

Instituto de Automática Industrial, CSIC
La Poveda, 28500 Arganda del Rey, Madrid
e-mail: tere@iai.csic.es

Resumen

El coprocesador ORBEX se ha utilizado para implementar el controlador borroso de un coche eléctrico. Los valores de las variables de entrada al controlador se obtienen procesando los datos proporcionados por un GPS colocado en el vehículo, esencialmente la posición y orientación relativas del coche respecto a la línea de referencia de la trayectoria. Los valores de las variables de salida obtenidos por el controlador permiten girar el volante el ángulo adecuado y ejercer la fuerza necesaria sobre el freno y el acelerador. Las propiedades del ORBEX hacen que basten pocas reglas de control porque pueden ser muy potentes.

Palabras clave: coprocesador borroso, control borroso, vehículos sin conductor.

1 EL PROCESADOR ORBEX

El coprocesador borroso ORBEX se está construyendo con el propósito de conseguir una unidad de proceso de sistemas borrosos en general. Dos son los tópicos que califican esta unidad *co*-procesador *borroso* y *propósito general*. El primero significa que el procesador es sólo para las operaciones borrosas típicas, el segundo destaca que, en su versión última, el coprocesador será adecuado para tratar cualquier problema modelado en términos de conjuntos y reglas borrosas y variables lingüísticas, no se quiere limitar su uso a aplicaciones de control.

La unidad de proceso tiene instrucciones elementales necesarias para realizar las operaciones aritméticas y lógicas requeridas en razonamiento y control borrosos y también procesa modificadores borrosos convencionales

(*no, muy, más o menos*) y otros exclusivos (*mayor que, menor que y entre*) que se han revelado muy potentes. Se pretende que versiones posteriores del ORBEX manejen también cuantificadores borrosos (como *la mayoría, algunos, muchos*, etc.).

El presente artículo describe el diseño funcional de la máquina, las instrucciones de utilización para el usuario y la comprobación de sus prestaciones. Esto último se hace ejecutando en el ORBEX una aplicación real y comprometida: el control de un vehículo sin conductor.

1.1 Evolución del modelo funcional

Desde el primer diseño funcional [1] [2] hasta el actual modelo [3] el procesador ORBEX¹ se ha ido adaptando a diversos requerimientos tecnológicos. Por ejemplo, la variedad de formas de las funciones de pertenencia del primer modelo se ha reducido a trapecios y los cálculos se han limitado a operaciones entre números enteros de longitud fija. Actualmente, y ante la caída de precios de los procesadores convencionales, se acepta que el ORBEX puede ser construido sobre un procesador convencional, con toda la riqueza de su primer modelo, de forma que la implementación de ORBEX coincide con la simulación realizada para comprobar sus características.

Este artículo presenta una realización del ORBEX sobre un procesador CIRYX incluido en una tarjeta de desarrollo de aplicaciones industriales que, además, permite la conexión de periféricos. La aplicación elegida se integra en el proyecto COVAN² idóneo para la comprobación de las prestaciones del ORBEX, ya que requiere un controlador borroso en tiempo real que se ejecute en un computador embarcado en un vehículo.

¹ ORBEX: CICYT, TIC 96 1393-C06-03

² COVAN: CAM 06T/042/9

1.2 Contexto del problema

El contexto o base de conocimiento de un problema definido en términos borrosos, contiene la definición de las variables y reglas. Las variables, entradas y salidas, están definidas por su universo de discurso y por sus valores lingüísticos, definidos, a su vez, por las funciones de pertenencia. Las condiciones de las reglas son proposiciones borrosas sobre las entradas y las conclusiones proposiciones borrosas sobre salidas.

Por ejemplo, para un controlador del proceso de fusión de chatarra en un horno eléctrico, una variable de entrada puede ser *temperatura* y una condición puede ser *temperatura alta*, siendo *alta* un valor lingüístico de la variable *temperatura*, cuya función de pertenencia puede ser un trapecio definido por los valores 800°C, 1000°C, 1500°C y 2000°C. Análogamente una variable de salida puede ser *válvula* y una conclusión puede ser *válvula abierta*, siendo *abierta* uno de sus valores lingüísticos, cuya función de pertenencia puede ser una rampa ascendente entre 0 y 1. Finalmente la expresión: "SI *temperatura* MUY *alta* Y ... O ... ENTONCES *válvula abierta* Y ..." puede ser una regla de control

1.3 Instrucciones borrosas típicas

El ORBEX posee dos modos de funcionamiento carga y ejecución. En el modo de carga el procesador construye la representación en memoria del problema que se desea resolver y en el modo de ejecución realiza las operaciones del ciclo de control.

En el modo de carga se almacenan las variables lingüísticas, conjuntos borrosos y reglas introducidas por el usuario en una base de variables y una base de instrucciones [4]. Existen dos instrucciones de carga: *carga variable* y *carga instrucción*. La primera interpreta las definiciones de las variables hechas por el usuario y las almacena en la base de variables. La segunda descompone las reglas en instrucciones borrosas elementales y las almacena en la base de instrucciones. Ambas bases de datos constituyen la representación en memoria del contexto que define el problema.

En el modo de ejecución hay tres instrucciones, que también definen tres fases del cálculo y acceden a las bases de datos construidas en la fase de carga; estas instrucciones son: *difuminar*, *inferir* y *concretar*.

La instrucción *difuminar* se ejecuta para cada variable de entrada y calcula el grado de pertenencia del valor físico de la variable, medido o calculado, a cada uno de sus valores lingüísticos. Por ejemplo, si en un instante determinado la temperatura de horno de fusión de chatarra es de 1450°C, calculará cuáles son los grados

de pertenencia a los valores lingüísticos *alta*, *media* y *baja* de la variable temperatura.

La instrucción *inferir* calcula el grado de verdad que se atribuye a una regla. Como es habitual el procedimiento de inferencia calcula primero el grado de certeza del antecedente y lo propaga al consecuente. Conviene destacar que la potencia del procesador ORBEX se apoya en la capacidad de esta instrucción para tratar reglas de gran poder expresivo.

Las proposiciones de los antecedentes pueden ser afirmativas o negativas, estar concatenadas por conjunciones copulativas "y" o disyuntivas "o" y admiten modificadores borrosos. Los modificadores admitidos son los convencionales *muy*, *poco* y *extra* y otros exclusivos del procesador ORBEX *mayor que*, *menor que* y *entre*. En consecuencia el código de la instrucción *inferir* debe contener instrucciones elementales para calcular la intersección de proposiciones, la agregación de proposiciones y el complemento de proposiciones, así como para calcular las funciones de pertenencia de los valores lingüísticos modificados.

Desde el punto de vista del diseño e implementación de su modelo funcional, el ORBEX podría admitir que los consecuentes estuviesen formados por proposiciones análogas a las de los antecedentes, los procedimientos para tratarlos serían los existentes. Sin embargo se ha considerado suficiente que el consecuente sea un conjunto de proposiciones borrosas afirmativas y sin modificadores unidas por la conjunción y. Esta aparente limitación no es tal en la práctica. En efecto, en un sistema de control, las salidas del controlador son las entradas del proceso a controlar, que son siempre bien conocidas, por eso no tiene sentido que la conclusión del procedimiento de control sea no hacer una acción (por ejemplo, *válvula no abierta*) o que dé igual hacer una acción u otra (por ejemplo, *válvula1 abierta* o *válvula2 abierta*). Creemos que estas consideraciones son válidas para otros sistemas basados en reglas borrosas distintos de los de control.

El grado de certeza de la regla o, en el caso de los sistemas de control, el grado en el que dicha regla influye en las acciones de control, es el resultado de la inferencia. Este valor se guarda en la base de variables, en los elementos correspondientes a las variables de salida que aparecen en el consecuente de la regla.

La instrucción *concretar* se encarga de asignar un valor numérico determinado a una variable de salida, por ejemplo los grados que debe girar un volante. El resultado de esta instrucción se obtiene tras un proceso de ponderación en el que participan todas las reglas del contexto.

1.4 Instrucciones para el usuario

El usuario utiliza el procesador ORBEX incluyendo una secuencia de instrucciones específicas en su programa de aplicación. Este conjunto de instrucciones constituye una interfase de utilización que contiene cuatro tipos de instrucciones:

| | |
|---------------------------|---|
| <i>fzzinicio</i> (...) | lectura del contexto y formación de las bases de datos. |
| <i>fzzentrada</i> (...) | lectura y difuminación de los valores actuales de las entradas. |
| <i>fzzreglas</i> | realización de las inferencias. |
| <i>fzzsalida</i> (...) | obtención de los valores de las salidas en tiempo de ejecución. |

La instrucción *fzzinicio* debe ser única y la primera de la secuencia, el ORBEX la ejecuta en modo de carga. Esta instrucción tiene como parámetro el nombre de un fichero externo que contiene el contexto del problema. Las otras instrucciones las realiza el ORBEX en modo de ejecución. Las instrucciones *fzzentrada* tienen como parámetro el nombre de una variable de entrada, hay tantas como entradas tenga el problema y pueden ir en cualquier orden en la secuencia pero siempre detrás de la instrucción *fzzinicio*. La instrucción *fzzreglas* no tiene parámetros, es única y tiene que ir después de la última instrucción *fzzentrada*. Las últimas instrucciones de la secuencia son las *fzzsalida*, tienen un parámetro que es el nombre de una variable de salida y habrá tantas como salidas haya en el sistema. Un ejemplo de interfase de utilización puede ser:

```
fzzinicio ( control de vehículos)
fzzentrada (separación)
fzzentrada (orientación)
...
fzzreglas
fzzsalida (volante)
fzzsalida (pedal)
...
```

1.5 Primera implementación

En la actualidad, dadas las líneas de investigación del IAI, la primera implementación real del ORBEX está dirigida a control, como ya se ha dicho el control de un vehículo eléctrico. El problema concreto que se aborda es el control de la dirección, freno y acelerador de un CITROËN BERLINGO provisto de un DGPS (sistema de posicionamiento global diferencial) que se mueve por unas pistas privadas del IAI, no abiertas al tráfico convencional. El conjunto de estas pistas denominado ZOCO³ (zona de experimentación de coches) simula las calles de un barrio urbano.

³ ZOCO: CICYT IN 960118

La tarea fundamental del ORBEX es inferir en cada ciclo de control, la posición del volante, la del freno y la del acelerador de forma que el coche siga la trayectoria deseada a la velocidad conveniente, todo ello en función de la posición actual del coche proporcionada por el DGPS.

2 APLICACIÓN: EL CONTROL DE UN COCHE SIN CONDUCTOR

En el proyecto COVAN se pretende desarrollar técnicas básicas para que un coche sea capaz de realizar algunas maniobras sin conductor. El objetivo inmediato es ayudar al conductor, en tareas tediosas o peligrosas como aparcar o circular en caravana, mover vehículos en casos de incendio, en zonas contaminadas, así como obtener algunos sistemas de ayuda a conductores minusválidos. De momento no se quiere sustituir al conductor, aunque sí se pueden considerar aplicaciones en las que esto sería posible y deseable.

La zona de experimentación, ZOCO, consiste en un entramado de calles asfaltadas de una longitud total cercana al kilómetro y una anchura de seis metros, para permitir circulación de dos vehículos en paralelo, también existen dos plazas. Calles y plazas tienen nombres asignados, de forma que el recorrido de un coche se puede fijar como una secuencia de nombres. Se ha construido un mapa de ZOCO con los datos proporcionados por el DGPS colocado en un coche que recorrió la zona conducido por una persona. El mapa es imprescindible para que el sistema de control pueda establecer las referencias de la trayectoria cuando se fija un recorrido simbólico para el coche.

Los experimentos reales se hacen con dos vehículos eléctricos, CITROËN BERLINGO, aunque hasta ahora sólo se ha instrumentado uno. En el coche se ha colocado un sensor de posición, basado en un DGPS y actuadores para mover el volante y el acelerador (que también actúa de freno). Además se ha instalado un computador industrial con un procesador CIRYX, en el que se ha implementado el modelo funcional del ORBEX ya descrito, para ejecutar los procedimientos de control. Finalmente un sistema de comunicaciones vía radio permite pasar programas, trayectorias y posiciones, etc. entre los computadores de los vehículos que circulan por la zona.

La figura representa el hardware de control. El computador de a bordo recibe por una línea serie los datos del DGPS, otra línea serie conecta el controlador del acelerador, que actúa en tensión sobre el acelerador y recibe del tacómetro la velocidad medida; finalmente la dirección se controla mediante una tarjeta KELVIN que acciona un motor de 100w engranado con el volante de la dirección.

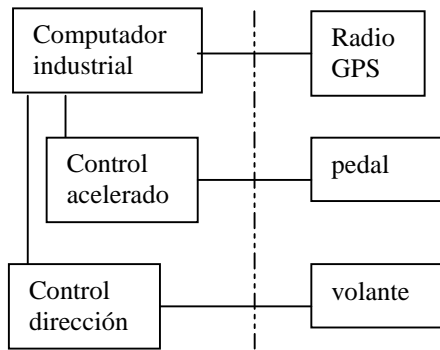


Figura 1. Hardware de control

2.1 Control básico de un vehículo

La posición y orientación del coche deben adaptarse a la trayectoria de referencia. Los desvíos de esta referencia, deriva lateral y cabeceo angular se utilizan en el control del volante, mientras que el control del acelerador depende de la velocidad. La estrategia de control es:

```

    velocidad{referencia -2 0 0 2 }
    pedal {levanta -100 -100 -1 0; pisa 0 1 100 100 }
    cabeceo { cero -20 0 0 20 }
    deriva { negativa -100 -100 -2 0; positiva 0 2 100 100 }
    volante { negativo -100 -100 -1 0; positivo 0 1 100 100 }
  
```

```

    si velocidad mayor que referencia entonces pedal levanta
    si velocidad menor que referencia entonces pedal pisa
    si cabeceo menor que cero entonces volante positivo
    si cabeceo mayor que cero entonces volante negativo
    si cabeceo cero y deriva positiva entonces volante negativo
    si cabeceo cero y deriva negativa entonces volante positivo
    fin
  
```

2.2 Primeros experimentos con automóviles

Algunos recorridos han sido simulados en un Silicon Graphics representando el entorno real en el que tienen lugar los experimentos. Para ello se han introducido el modelo cinemático de los coches y el mapa de la zona. La simulación muestra varios coches circulando por las pistas según las leyes de tráfico. El comportamiento de la velocidad, aceleración y potencia consumida depende del juego de reglas de control. Interesa destacar que la simulación aconsejó cambiar las reglas de control iniciales y las funciones de pertenencia.

También se han realizado maniobras simples pero reales con un coche: conducción en línea recta y giros. En

concreto el recorrido se inicia al principio de la calle Zadeh, llega a la plaza Sugeno gira y vuelve por la misma calle Zadeh hasta la Puerta del Sol.

Por ahora el controlador borroso se maneja desde el teclado del computador, mediante un programa que inicia y acaba su funcionamiento. Las primeras pruebas nos han llevado a ampliar el número de reglas y a incluir como nueva entrada la variable aceleración, que se calcula mediante diferenciación numérica.



Referencias

- [1] R. García Rosa, T. De Pedro. Un modelo de coprocesador borroso. *Actas del ESTYLF'96, VI Congreso Español sobre Tecnologías y Lógica Fuzzy*, 1996.
- [2] R. García Rosa, T. De Pedro. Modeling a fuzzy coprocessor and its programming language. *Mathware and Soft Computing*, vol V., n°. 2-3, pp 167-174, 1998.
- [3] R. García Rosa, T. De Pedro. A powerfull model for a fuzzy coprocessor. *Proc. of the CLCA'98 and IFAC'98* 1998.
- [4] R. García Rosa, T. De Pedro, M.T. de Andrade. Parallel inference engine for fuzzy controllers. *Cybernetics and Systems. An International Journal*, vol 25, n° 2, pg 359-371, 1994.
- [5] R. García Rosa, T. De Pedro, A. Rosetti. Fuzzy driving strategies for cars in several traffic situations. *IFAC Symposium on Intelligent Autonomous Vehicles*, 1998.
- [6] R. García Rosa, T. De Pedro. Automatic car drivers. *31st ISATA International Symposium on Automotive Technology and Automation*, 1998.
- [7] M. Sugeno, Murofushi. Fuzzy algorithmic control of model car by oral Instructions. *2nd IFSA Congress*, Tokyo.

