Revista Iberoamericana de Automática e Informática Industrial

ISSN: 1697-7912. Vol. 7, Núm. 2, Abril 2010, pp. 81-90 http://www.revista-riai.org

# Fusión Borrosa de Estimadores para Aplicaciones de Control Basado en Imagen

**Carlos Perez-Vidal** * **Luis Gracia** ** **Oscar Reinoso** *

* *Departamento de Ingeniería de Sistemas Industriales, Universidad Miguel Hernández, Avda. de la Universidad s/n, 03202 Elche (Alicante), España, (e-mail: carlos.perez@umh.es, o.reinoso@umh.es)*
** *Instituto IDF, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, España, (e-mail: luigraca@isa.upv.es)*

Resumen: El control visual es una disciplina de gran actualidad dentro del control de robots, y dentro de ésta, los algoritmos de predicción se usan para estimar la localización de objetos o características visuales proporcionadas por un sensor con retardo (cámara). Algunos de los algoritmos más utilizados son: el filtro de Kalman; los filtros alpha-beta/gamma ($\alpha\beta/\gamma$); el AKF; el SKF; etc. El mayor problema de algunos de ellos es conseguir que su implementación permita trabajar en aplicaciones con fuertes restricciones temporales o de tiempo real. En este artículo se presenta un nuevo método de predicción, denominado FMF, basado en la fusión o combinación borrosa de varios filtros, y por tanto con un alto coste computacional. En el artículo se estudia a través de simulación la mejora obtenida con la predicción del FMF respecto a los filtros individuales, lo que justifica su interés. Así mismo, se desarrolla su implementación de tiempo real en una FPGA empleando técnicas de paralelización y segmentado. La viabilidad, robustez y fiabilidad del algoritmo propuesto se ha comprobado mediante una aplicación experimental de control visual. 

Palabras Clave: Métodos predictivos, algoritmos paralelos, sistemas fuzzy, visión por computador, control automático.

## 1. INTRODUCCIÓN

El control visual (*vision-based control*) es una de las soluciones existentes para controlar el movimiento de un robot en entornos no estructurados (Hutchinson et al., 1996). Esto es posible gracias a la realimentación visual que permite minimizar una determinada función de error predefinida. Existen básicamente dos aproximaciones para realizar el control visual: control visual basado en imagen, en la que la señal de error se obtiene directamente de la imagen y se traduce en una acción de control; y, control visual basado en posición, en la que a partir de la imagen adquirida se reconstruye el espacio 3D de trabajo y la acción de control se obtiene respecto a éste (Corke and Hutchinson, 2000). El sistema de visión es el componente más lento en el esquema de control en dos aspectos diferentes: la adquisición de la imagen; y el procesamiento de los datos. El primero se puede resolver usando cámaras de alta velocidad o técnicas *sub-window* (Vincze and Hager, 2000), pero el segundo puede ser un gran problema cuando el ordenador no es capaz de procesar la información visual y generar la acción de control en tiempo real. El término "tiempo real" o RT se refiere a los requisitos temporales impuestos por el sistema bajo control y en general dependen de su dinámica. El problema indicado anteriormente se presenta a menudo cuando a un grupo de características en una secuencia de imágenes se les aplica filtros predictivos con alto coste computacional, como el *Switching Kalman Filter* SKF (Chroust and Vincze, 2003) u otros (Kawase et al., 1997) (Kolodziej and Singh, 2000).

En este artículo se presenta un nuevo método de predicción, denominado FMF (*Fuzzy Mix of Filters*), basado en mezclar o *fusionar* con técnicas borrosas la predicción realizada por otros filtros/estimadores (interpolación lineal, filtro de Kalman, filtro alpha-beta/gamma, circulares, etc.) obteniendo una predicción mejorada. El coste computacional del FMF en un procesador secuencial es la suma de los tiempos de cómputo de cada uno de los filtros que lo componen más el tiempo necesario para realizar la mezcla borrosa. Sin embargo, utilizando un procesador paralelo, todos los filtros se pueden calcular al mismo tiempo, reduciéndose considerablemente el tiempo de computo. Para implementar el filtro FMF se ha seleccionado un dispositivo FPGA, ya que posee la mencionada capacidad de paralelización y dispone de los suficientes recursos. El artículo está estructurado de la siguiente forma. La sección 2, presenta el filtro FMF a lo largo de diferentes apartados que describen su planteamiento conceptual y teórico, así como su funcionamiento en simulación. La sección 3 desarrolla la implementación del nuevo filtro en un procesador paralelo o FPGA e incluye una comparativa con un procesador secuencial Intel Pentium. La sección 4 muestra la viabilidad, robustez y fiabilidad del algoritmo propuesto mediante una aplicación experimental de control visual. Finalmente, las conclusiones del artículo y los trabajos futuros se indican en la sección 5.

## 2. FUSIÓN BORROSA DE FILTROS

### 2.1 Introducción

Existen gran cantidad de estimadores o filtros para estimar el estado de un sistema cara a poder realizar el seguimiento (*tracking*) de un objeto. El funcionamiento óptimo o ideal de

**Publicado electrónicamente: 09/04/2010**
**DOI: 10.4995/RIAI.2010.02.08**

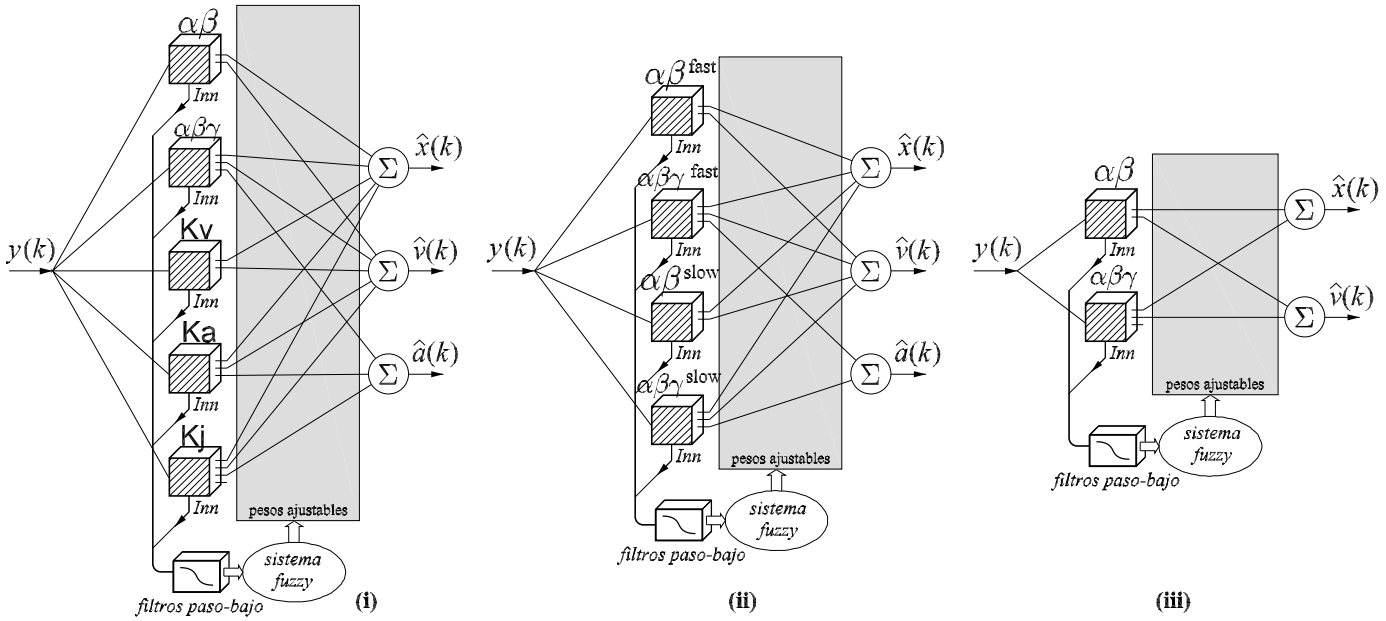


Figura 1. **(i)** FMF de tipo soft-Real-Time (sRT) donde se utilizan 5 filtros de predicción distintos. **(ii)** FMF formado por 4 filtros rápidos y lentos. **(iii)** FMF de tipo hard-Real-Time (hRT) formado por un filtro $\alpha\beta$ y otro $\alpha\beta\gamma$.

estos filtros depende de las condiciones del sistema: dinámica no modelada, ruido de medida, etc. que generalmente son cambiantes. Por tanto, si se utiliza sólo uno de estos filtros como estimador, se hace necesario cambiar en línea (*on-line*) sus parámetros (e.g. índice de maniobra) para obtener un funcionamiento óptimo o quasi-óptimo la mayor parte del tiempo (Efe and Atherton, 1998). La idea anterior es la que subyace en los filtros adaptativos. Sin embargo, en la mayoría de casos el método de actualización de estos parámetros es complejo de establecer. Otra alternativa es utilizar simultáneamente varios filtros estáticos (no adaptativos), cada uno ajustado con diferentes condiciones, y combinar sus predicciones de forma adecuada para obtener una predicción global mejor que la realizada por cada uno de ellos. De modo que, la salida de aquellos que funcionen más cerca de su comportamiento óptimo tendrán mayor influencia en la salida del estimador final. Los siguientes trabajos utilizan este procedimiento:

- En (Chroust and Vincze, 2003) se desarrolla el SKF *Switching Kalman Filter*, el cual utiliza un "monitor de predicción" que supervisa la calidad de la predicción de un filtro adaptativo de Kalman (AKF). Si el monitor detecta una discontinuidad, se conmuta a un filtro de Kalman estacionario ($\alpha\beta/\gamma$) adecuado;
- En (Kawase et al., 1997) se combina un predictor alpha-beta con un predictor circular o elipsoidal;
- En (Kolodziej and Singh, 2000) y (Tenne and Singh, 2002) se emplea un filtro alpha-beta para los tramos rectilíneos del movimiento del objeto y un predictor circular dinámico para las zonas de maniobra;
- En (Yoo and Kimb, 2003) se usa un algoritmo que conmuta entre un filtro alpha-beta convencional cuando el movimiento del objeto alcanza el régimen permanente y un algoritmo de aproximación de ganancia cuando el objeto empieza a maniobrar;
- En (Moore and Wang, 2001) se presenta un algoritmo que combina cinco filtros de Kalman distintos.

### *2.2 Trasfondo teórico de la fusión de filtros propuesta*

El problema de los trabajos mencionados en el apartado anterior es su complejidad matemática y sustentación empírica. En este sentido, el algoritmo aquí propuesto presenta como novedad el emplear para combinar los filtros una fusión borrosa (*fuzzy mix*). La lógica borrosa está basada en reglas empíricas que no dejan de ser una "formalización" de un procedimiento basado en los conocimientos y experiencia del diseñador. No obstante, tiene la ventaja de ser sencilla e intuitiva, lo que para este caso supone una mejora respecto a los complejos y poco explicados filtros basados en mezcla existentes hasta el momento. Esta idea se presento por primera vez en el artículo de congreso (Pérez et al., 2007), pero aquí se han introducido varias modificaciones para mejorar la mezcla borrosa y los resultados obtenidos. La principal diferencia es que aquí se usa la innovación del filtro como entrada del sistema borroso, cambiando las reglas, funciones de pertenencia, motor de inferencia, etc.

El filtro propuesto o FMF pretende ser configurable por el usuario según sus necesidades. A modo de ejemplo, la Figura 1 muestra tres variantes posibles con los distintos bloques que intervienen. En los tres casos la entrada a todos los filtros que componen el FMF es la medida $y$ en el instante actual $k$ de la posición del objeto o característica visual. El FMF de la Figura 1(i), que es el más complejo de los tres, está compuesto por los siguientes 5 filtros: filtro estacionario de velocidad constante ($\alpha\beta$); filtro estacionario de aceleración constante ($\alpha\beta\gamma$); filtro de Kalman de velocidad constante (Kv); filtro de Kalman de aceleración constante (Ka); y filtro de Kalman de jerk (derivada de aceleración) constante (Kj). La salida de cada uno de ellos es la innovación $Inn$ y la estimación de las siguientes variables: posición $x$; velocidad $v$; aceleración $a$ (sólo para los filtros $\alpha\beta\gamma$, Ka y Kj); y jerk (sólo para el filtro Kj). Estas estimaciones se mezclan dependiendo de los valores de innovación (por los motivos indicados más adelante en este apartado) para obtener la estimación global de las siguientes variables: posición $(\mathrm{k}_1 \cdot \hat{x}_{\alpha\beta} + \mathrm{k}_2 \cdot \hat{x}_{\alpha\beta\gamma} + \mathrm{k}_3 \cdot \hat{x}_{\mathrm{Kv}} + \mathrm{k}_4 \cdot \hat{x}_{\mathrm{Ka}} +$

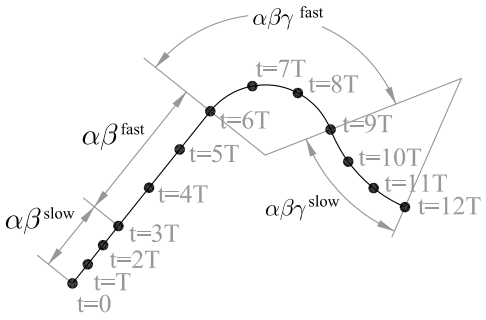


Figura 2. Ejemplo cualitativo de trayectoria para un objeto con movimiento en 2 dimensiones.

$k_5 \cdot \hat{x}_{Kj}$); velocidad ($k_6 \cdot \hat{v}_{\alpha\beta} + k_7 \cdot \hat{v}_{\alpha\beta\gamma} + k_8 \cdot \hat{v}_{Kv} + k_9 \cdot \hat{v}_{Ka} + k_{10} \cdot \hat{v}_{Kj}$); y aceleración ($k_{11} \cdot \hat{a}_{\alpha\beta\gamma} + k_{12} \cdot \hat{a}_{Ka} + k_{13} \cdot \hat{a}_{Kj}$). En las expresiones anteriores: $k_i$ es la ponderación de cada filtro, que se establece a partir de su valor de innovación; el símbolo $\hat{}$ encima de la variable indica que es una estimación; y el subíndice de la variable indica con que filtro se ha realizado la estimación. Notar que no se puede realizar una combinación de estimaciones para el jerk, ya que sólo se obtiene en uno de los filtros, y por tanto no forma parte de la salida del FMF. La estimación de las variables se puede realizar para el instante temporal discreto $k$, $k+1$, etc. según los requerimientos del sistema de control visual.

La selección del FMF más adecuado en cada caso depende de la aplicación a realizar y del cómputo que se pueda asumir. En este artículo se clasifican los filtros de predicción como soft-Real-Time (sRT) y hard-Real-Time (hRT) dependiendo del nivel de prestaciones demandadas. La diferencia entre ambos es que el cómputo del hRT es significativamente más bajo que el del sRT. A modo de ejemplo el FMF de la Figura 1(i) sería de tipo sRT y el FMF de la Figura 1(iii) (formado tan sólo por un filtro $\alpha\beta$ y otro $\alpha\beta\gamma$) se podría clasificar como hRT. Para seleccionar uno u otro otro tipo de filtro habría que llegar a un compromiso entre precisión y coste. Gran cantidad de aplicaciones no requieren elevadas prestaciones (Corke, 1996): el control de juguetes, cámaras de vigilancia (CCTV), etc. Sin embargo, otras aplicaciones pueden producir un fallo fatal por la ausencia o retardo de la acción de control pertinente, como son: la automoción, el transporte ferroviario, la aviónica, etc. El FMF mostrado en la Figura 1(ii) muestra dos tipos diferentes de filtro alpha-beta/gamma, llamados $\alpha\beta/\gamma^{\text{fast}}$ y $\alpha\beta/\gamma^{\text{slow}}$. El filtro rápido (*fast*) tiene un tiempo de respuesta más corto que el lento (*slow*). El primero predice con mayor calidad los movimientos rápidos y el segundo los lentos.

A continuación se justifica conceptual e ilustrativamente la filosofía que subyace en el FMF en particular y en los estimadores que combinan varios tipos de filtro en general. El comportamiento genérico de un objeto móvil no responde a una estructura única (e.g. de velocidad o aceleración constante) sino que sufre modificaciones en su dirección y avance. Este comportamiento se suele descomponer (Efe and Atherton, 1998) (Yoo and Kimb, 2003) en tramos con maniobra y sin maniobra. De modo que, el FMF aprovecha en cada tramo la estimación realizada por aquellos filtros que tienen mejor predicción, i.e. de aquellos que están más cerca de su funcionamiento ideal. A modo de ejemplo, se considerará la trayectoria cualitativa de la Figura 2 (objeto con movimiento en 2 dimensiones) y el FMF formado por cuatro filtros de la Figura 1(ii). Esta trayectoria se puede descomponer en cuatro partes diferenciadas. La primera es un tramo rectilíneo de velocidad constante baja, para el cual el filtro que mejor predicción dará es el $\alpha\beta^{\mathbf{slow}}$. La segunda es un tramo también rectilíneo de velocidad constante pero mayor, y por tanto el filtro que mejor predicción dará es el $\alpha\beta^{\mathbf{fast}}$. La tercera parte es un arco de circunferencia (el objeto está maniobrando y por tanto tiene una aceleración normal) con curvatura grande, donde el filtro que mejor predicción dará es el $\alpha\beta\gamma^{\mathbf{fast}}$. Finalmente, el último tramo es un arco de circunferencia con menor curvatura (mayor radio) y menor velocidad de desplazamiento que el anterior, con lo que el filtro que mejor predicción dará es el $\alpha\beta\gamma^{\mathbf{slow}}$. De forma que, si se consigue identificar adecuadamente que filtro está dando un mejor resultado en cada tramo de trayectoria, la estimación combinada de filtros mejorará mucho la obtenida utilizando uno sólo de los filtros individuales. Esta es la filosofía del FMF, donde en cada momento se pondera más (i.e. tiene más influencia en la estimación global) el filtro que funciona mejor en cada tramo. Notar que, en el caso de que toda la trayectoria pertenezca únicamente a un tipo comportamiento, el FMF puede no dar mejor predicción que el filtro individual óptimo para ese comportamiento, aunque ambas predicciones serían muy próximas.

Para saber si un filtro está funcionando cerca de sus condiciones óptimas o ideales (hipótesis de diseño que dan lugar a ese filtro concreto) se utilizará el valor medio de la innovación. El motivo es porque, cuando se dan dichas condiciones, la innovación es un ruido blanco (sin correlación entre sus propios valores en diferentes instantes) con media nula y matriz de covarianza determinada (Simon, 2006). Para obtener el valor medio de la innovación de cada estimador se puede utilizar por ejemplo un filtro paso bajo, ver la Figura 1. De modo que, cuanto más próximo a cero esté el valor medio de la innovación de un filtro, más próximo está éste a su funcionamiento ideal y por tanto más ponderación se le dará en la combinación de filtros. En realidad no se puede afirmar que la fusión de filtros sea óptima. No obstante, si la función de pertenencia utilizada para ponderar cada filtro (a la que se le pasa el valor medio de su innovación) se selecciona y/o ajustan sus parámetros asociados de forma adecuada, con algún procedimiento de búsqueda, sí que se podría hablar de que dicha fusión es subóptima. Esta búsqueda puede consistir en hacer un barrido de los parámetros de una o varias funciones de pertenencia entre extremos alejados. De modo que, para una de las funciones de pertenencia analizadas y un conjunto de valores de sus parámetros, se tendrá un error de predicción global mínimo. Dicha función y valores de parámetros serán los que se utilizarán en la práctica. El barrido anterior se realizaría para una trayectoria característica que recoja todos los movimientos típicos del objeto móvil a seguir. Como se puede comprobar, el proceso de selección y ajuste anterior es de naturaleza empírica (propio de la lógica borrosa) y está basado fundamentalmente en la experiencia del diseñador y en el resultado de las pruebas realizadas.

### *2.3 El algoritmo del filtro* hRT FMF

El FMF de la Figura 1(iii), que es el que tiene menor tiempo de cómputo (hRT) de los descritos en el apartado anterior, se desarrolla a continuación en detalle. El filtro de Kalman estacionario o de régimen permanente se puede describir mediante las ecuaciones:

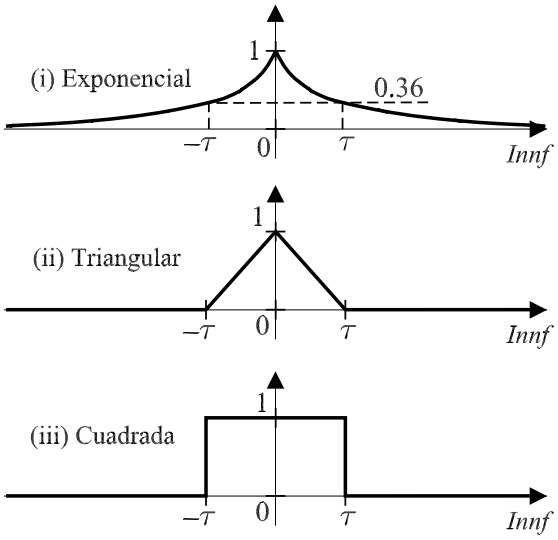


Figura 3. Funciones de pertenencia probadas en la implementación del filtro hRT FMF.

$$Inn(k) = y(k) - \mathbf{C} \cdot \hat{\mathbf{z}}(k|k-1) \quad (1)$$

$$\hat{\mathbf{z}}(k|k) = \hat{\mathbf{z}}(k|k-1) + \mathbf{K} \cdot Inn(k) \quad (2)$$

$$\hat{\mathbf{z}}(k+1|k) = \mathbf{A} \cdot \hat{\mathbf{z}}(k|k) \quad (3)$$

donde $Inn(k)$ es la innovación en el instante $k$; $y(k)$ es la medida de la posición del objeto en el instante $k$; $\mathbf{z}$ es el vector de estado del sistema; $\hat{\mathbf{z}}(n|m)$ es la estimación del estado en el instante $n$ dada por las observaciones hasta e incluyendo el instante $m$; $\mathbf{K}$ es la ganancia de corrección del filtro en régimen permanente; $\mathbf{A}$ es la matriz de estado del sistema; y $\mathbf{C}$ es la matriz de salida del sistema (el modelo considerado es LTI). Dependiendo de la dimensión de los vectores y matrices, el filtro de Kalman estacionario puede corresponderse a un filtro $\alpha\beta$ o a uno $\alpha\beta\gamma$. En el caso del filtro $\alpha\beta$, filtro de Kalman estacionario con modelo DWNA (*discrete white noise acceleration model*), las matrices anteriores se particularizan como (Bar-Shalom et al., 2001):

$$\mathbf{K}_{ab} = \begin{bmatrix} \alpha_{ab} \\ \beta_{ab}/\mathrm{T} \end{bmatrix}; \mathbf{A}_{ab} = \begin{bmatrix} 1 & \mathrm{T} \\ 0 & 1 \end{bmatrix}; \mathbf{C}_{ab} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

donde T es el periodo de muestreo del sistema discreto y $\{\alpha_{ab}, \beta_{ab}\}$ son los parámetros del filtro. Mientras que para el caso del filtro $\alpha\beta\gamma$, filtro de Kalman estacionario con modelo DWPA (*discrete wiener process acceleration model*), las matrices son (Bar-Shalom et al., 2001):

$$\mathbf{K}_{abg} = \begin{bmatrix} \alpha_{abg} \\ \beta_{abg}/\mathrm{T} \\ \gamma_{abg}/(2\mathrm{T}^2) \end{bmatrix}; \mathbf{A}_{abg} = \begin{bmatrix} 1 & \mathrm{T} & \mathrm{T}^2 \\ 0 & 1 & \mathrm{T} \\ 0 & 0 & 1 \end{bmatrix}; \mathbf{C}_{abg} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

donde $\{\alpha_{abg}, \beta_{abg}, \gamma_{abg}\}$ son los parámetros del filtro. Los valores de los parámetros de los filtros anteriores se obtienen con las expresiones indicadas en (Bar-Shalom et al., 2001) a partir del ruido de proceso $\sigma_v$ y ruido de medida $\sigma_w$ considerados.

Las reglas $R_i$ que se han utilizado para el filtro hRT FMF son:
$R_1$: SI la innovación del filtro ES alta, ENTONCES ponderar ligeramente el filtro

$R_2$: SI la innovación del filtro ES baja, ENTONCES ponderar fuertemente el filtro

Como función de pertenencia para filtro FMF se han contemplado tres funciones típicas (ver la Figura 3): exponencial, triangular y rectangular, cuyas expresiones son:

$$f_{\exp}(Innf) = e^{-|Innf|/\tau} \quad (4)$$

$$f_{\mathrm{tri}}(Innf) = \begin{cases} 1 - |Innf|/\tau & \text{para} \quad Innf \le \tau \\ \varepsilon & \text{para} \quad Innf > \tau \end{cases} \quad (5)$$

$$f_{\mathrm{rect}}(Innf) = \begin{cases} 1 & \text{para} \quad Innf \le \tau \\ \varepsilon & \text{para} \quad Innf > \tau \end{cases} \quad (6)$$

donde *Innf* es el valor medio o filtrado de la innovación en un determinado instante; $\tau$ es el parámetro de diseño de la función de pertenencia; y $\varepsilon$ es un valor muy pequeño (e.g. $10^{-7}$) no nulo para evitar la singularidad en el cálculo del filtro FMF. La predicción realizada por éste se define mediante las expresiones siguientes:

$$\mathrm{W}_{\alpha\beta}(k) = f(Innf_{\alpha\beta}(k)) \quad (7)$$

$$\mathrm{W}_{\alpha\beta\gamma}(k) = f(Innf_{\alpha\beta\gamma}(k)) \quad (8)$$

$$\hat{\mathbf{z}}_{\mathrm{FMF}}(k) = \frac{\mathrm{W}_{\alpha\beta}(k) \cdot \hat{\mathbf{z}}_{\alpha\beta}(k) + \mathrm{W}_{\alpha\beta\gamma}(k) \cdot \hat{\mathbf{z}}_{\alpha\beta\gamma}(k)}{\mathrm{W}_{\alpha\beta}(k) + \mathrm{W}_{\alpha\beta\gamma}(k)} \quad (9)$$

donde $f(\cdot)$ es alguna de las tres funciones de pertenencia indicadas; $\hat{\mathbf{z}}_{\mathrm{X}}$ es la estimación realizada por el filtro X del vector de estado (posición y velocidad); y W es la ponderación de cada filtro. La expresión (9) se extiende a $\mathrm{N_f}$ filtros con:

$$\hat{\mathbf{z}}_{\mathrm{FMF}}(k) = \sum_{i=1}^{\mathrm{N_f}} \left( \mathrm{W}_i(k) \cdot \hat{\mathbf{z}}_i(k) \right) \Big/ \sum_{i=1}^{\mathrm{N_f}} \mathrm{W}_i(k) \quad (10)$$

### *2.4 Simulaciones del hRT FMF*

En este apartado se realizan diferentes simulaciones, empleando el método de Monte Carlo (Robert and Casella, 1999), para mostrar el buen funcionamiento del filtro desarrollado. Para analizar estas simulaciones se definen los siguientes términos: RMS (media cuadrática o *root mean square*); RMSE (valor RMS del error de predicción); NRMSE (RMSE normalizado); TARMSE (media cuadrática temporal del RMSE); TANRMSE (media cuadrática temporal del NRMSE); y RMSI (valor RMS de la innovación del filtro). Las expresiones siguientes definen dichos términos:

$$\mathrm{RMSE}(k) = \sqrt{\frac{1}{\mathrm{Nr}} \sum_{i=1}^{\mathrm{Nr}} (x^i(k) - \hat{x}^i(k))^2} \quad (11)$$

$$\mathrm{NRMSE}(k) = \frac{\mathrm{RMSE}(k)}{\sqrt{\frac{1}{\mathrm{Nr}} \sum_{i=1}^{\mathrm{Nr}} (x^i(k) - y^i(k))^2}} \quad (12)$$

$$\mathrm{TARMSE} = \sqrt{\frac{1}{\mathrm{N} - \mathrm{D}} \sum_{k=D+1}^{\mathrm{N}} \mathrm{RMSE}^2(k)} \quad (13)$$

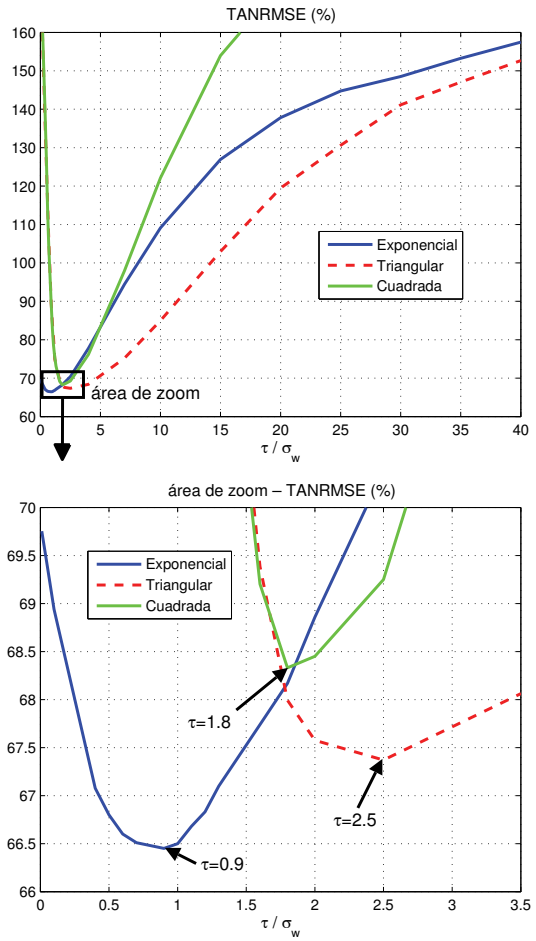


Figura 4. Búsqueda del parámetro óptimo en las tres funciones de pertenencia haciendo un barrido con *tray*$_3$.

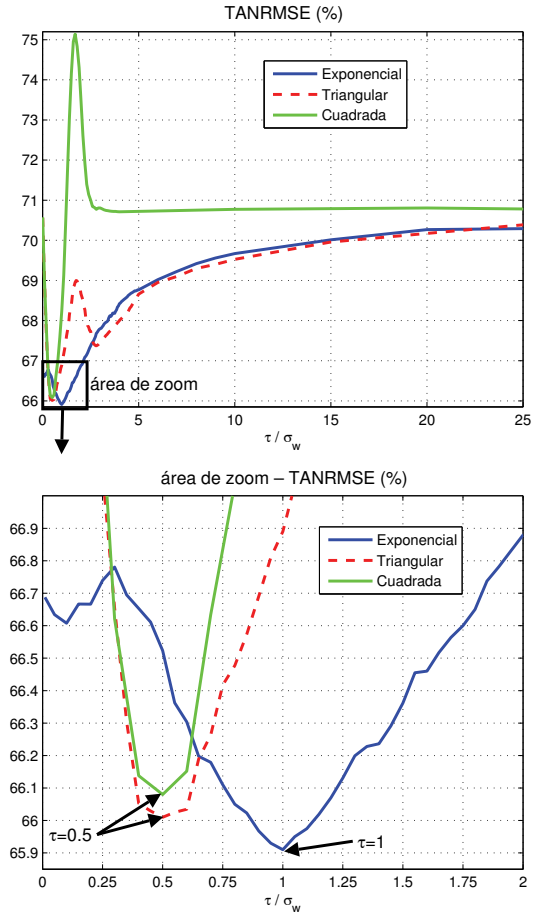


Figura 5. Búsqueda del parámetro óptimo en las tres funciones de pertenencia haciendo un barrido con *tray*$_4$.

$$\mathrm{TANRMSE} = \sqrt{\frac{1}{\mathrm{N}-\mathrm{D}} \sum_{k=D+1}^{\mathrm{N}} \mathrm{NRMSE}^2(k)} \qquad (14)$$

$$\mathrm{RMSI}(k) = \sqrt{\frac{1}{\mathrm{Nr}} \sum_{i=1}^{\mathrm{Nr}} \left(Inn^i(k)\right)^2} \qquad (15)$$

donde $x^i(k)$ es la posición real del objeto para la réplica $i$ en el instante $k$; $\hat{x}^i(k)$ es la predicción de la posición del objeto para la réplica $i$ en el instante $k$; $y^i(k)$ es la medida de la posición del objeto para la réplica $i$ en el instante $k$; $Inn^i(k)$ es la innovación del filtro para la réplica $i$ en el instante $k$; Nr es el número de réplicas de Monte Carlo simuladas; N es el número de muestras simuladas; y D es el número de muestras iniciales rechazadas en el cálculo de los valores de media cuadrática temporal. Los valores definidos en (11)-(15) se usan especificando el tipo de trayectoria con un superíndice y el tipo de filtro con un subíndice, por ejemplo: $\mathrm{RMSE}^1_{\alpha\beta}$, $\mathrm{RMSE}^2_{\alpha\beta\gamma}$, etc.

El número de réplicas de una simulación de Monte Carlo afecta a la calidad de los resultados. En general, cuantas más réplicas de Monte Carlo se simulen, más suaves y exactos son los resultados obtenidos, i.e. tienen menos irregularidades y están más próximos a sus valores asintóticos. En este sentido, el número de réplicas seleccionado en cada simulación de este apartado es tal que los resultados sean suficientemente suaves y por tanto se haya alcanzado prácticamente el valor asintótico. De la misma forma, cuantos más instantes N se consideren, los valores obtenidos con la media cuadrática temporal son más exactos.

Para las simulaciones se consideran las cuatro trayectorias siguientes:

(1) *tray*$_1$: Trayectoria obtenida con el modelo DWNA (coincide con la hipótesis de diseño del filtro $\alpha\beta$).
(2) *tray*$_2$: Trayectoria obtenida con el modelo DWPA (coincide con la hipótesis de diseño del filtro $\alpha\beta\gamma$).
(3) *tray*$_3$: Trayectoria con 3 tramos, obtenida por la unión de: modelo DWNA, modelo DWPA y modelo DWNA.
(4) *tray*$_4$: Trayectoria sinusoidal.

Estas trayectorias se han generado con un periodo T=0.04s (tiempo de adquisición de una cámara PAL trabajando a 25 frames por segundo) y 250 muestras (N=250). Para generar las tres primeras se ha utilizado un ruido de proceso de aceleración discreto con desviación típica $\sigma_v$=0.56m/s$^2$. La cuarta se ha generado con una sinusoide de amplitud 2m y periodo 10 segundos. Para todas las trayectorias se ha considerado una desviación típica del ruido de medida $\sigma_w$=0.02. La inicialización utilizada en los modelos DWNA/DWPA para generar las tres primeras trayectorias es: *tray*$_1$ $\rightarrow$ $\mathbf{x}(0)$ =(0,0.4)$^{\mathrm{T}}$; *tray*$_2$ $\rightarrow$ $\mathbf{x}(0)$ =(0,0,0.08)$^{\mathrm{T}}$; y *tray*$_3$ $\rightarrow$ $\mathbf{x}(0)$ =(0,0.4,0)$^{\mathrm{T}}$.

Con objeto de identificar los valores óptimos del parámetro $\tau$ para las funciones de pertenencia de la Figura 3, así como para saber cuál funciona mejor, se ha realizado el proceso de barrido descrito al final del apartado 2.2. En concreto, las Figuras 4 y 5 muestran el valor del TANRMSE (indicador global del error de predicción) para el barrido en $\tau$ utilizando las trayectorias *tray*$_3$ y *tray*$_4$, que son más generales que las otras dos. Notar que el eje horizontal de ambos barridos, en el que se representa el valor de $\tau$, está normalizado respecto a la desviación típica del ruido de medida $\sigma_w$, ya que el valor medio de la innovación será de ese orden de magnitud. En los dos barridos la función de pertenencia que mejor funciona (con un valor mínimo de TANRMSE más pequeño) es la exponencial, luego la triangular y finalmente la rectangular. Lo anterior está en consonancia con el grado de suavidad la transición de 1 a 0 en cada función: la exponencial es más suave (elaborada) que la triangular, y ésta a su vez que la rectangular. Se puede apreciar que los valores óptimos de $\tau$ para la función exponencial en ambos barridos está muy próximo: $0.9\sigma_w$ y $\sigma_w$.

Los filtros $\alpha\beta$ y $\alpha\beta\gamma$ se han simulado para las cuatro trayectorias anteriores (los valores de $\sigma_v$ y $\sigma_w$ considerados para los filtros son los mismos con los que se han generado las trayectorias), mientras que el filtro FMF se simula sólo para las dos últimas trayectorias. En todas las simulaciones, se han despreciado las primeras 50 muestras para calcular los valores de media cuadrática temporal (D=50). El valor medio de la innovación de los filtros $\alpha\beta$ y $\alpha\beta\gamma$ (necesaria para calcular el FMF) se obtenido con un filtro paso bajo de primer orden de Butterworth con frecuencia normalizada de corte igual a 0.1. La función de pertenencia utilizada para el FMF es la exponencial (que es la que mejor resultado ha dado de las tres probadas) y la constante $\tau$ se ha ajustado al valor de $\sigma_w$, que está muy próximo al óptimo obtenido en los barridos de las Figuras 4 y 5.

Simulaciones usando *tray*$_1$:

En la Figura 6, se muestra el error de predicción normalizado para esta trayectoria usando $5 \cdot 10^3$ réplicas de Monte Carlo. El filtro $\alpha\beta\gamma$ estima un 5.3 % mejor en la realidad ($\mathrm{TARMSE}^1_{\alpha\beta\gamma}$=0.01351m) que lo ideal considerado en el algoritmo, 0.0143m (valor dado por la desviación típica obtenida con la matriz de incertidumbre), debido a que las condiciones de funcionamiento son más sencillas que las que contempla el filtro $\alpha\beta\gamma$, al no haber propagación de errores a través del estado de aceleración. De igual modo el filtro $\alpha\beta$ estima mejor ($\mathrm{TARMSE}^1_{\alpha\beta}$=0.01016m) que el filtro $\alpha\beta\gamma$ porque se le ha simplificado al algoritmo la tarea de estimación (sólo hay propagación de 2 estados, posición y velocidad) y esta simplificación coincide con la situación real de la trayectoria. Debido a que el cálculo del error de predicción normalizado implica simplemente dividir por la desviación estándar del ruido de medida, desde este momento, se graficará solamente el error de predicción absoluto.

Simulaciones usando *tray*$_2$ (Nr=$5 \cdot 10^3$):

En la Figura 7 se puede comprobar que la corrección es insuficiente para el filtro $\alpha\beta$ y que su error de predicción o incertidumbre es monótonamente creciente con el tiempo (habría que aumentar el ruido de proceso al diseñar el filtro para acotar su incertidumbre). Este filtro por tanto no es adecuado para este tipo de trayectoria.

Simulaciones usando *tray*$_3$ (Nr=$5 \cdot 10^3$):

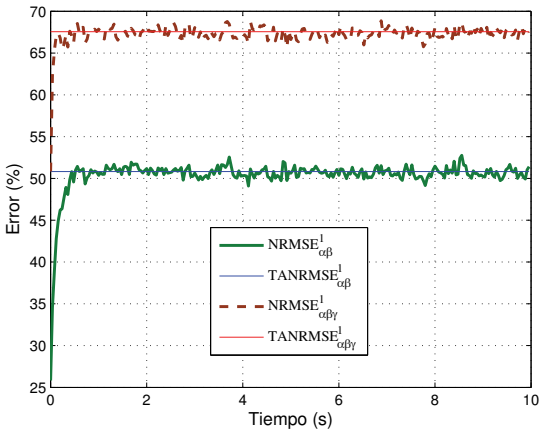


Figura 6. Errores de predicción normalizados para *tray*$_1$.

La media cuadrática temporal de los errores absolutos (mostrados graficamente en la Figura 8) y normalizados son:

$$\mathrm{TARMSE}^3_{\alpha\beta} = 0.06909\mathrm{m},\ \mathrm{TANRMSE}^3_{\alpha\beta} = 345.45\,\%,$$
$$\mathrm{TARMSE}^3_{\alpha\beta\gamma} = 0.01413\mathrm{m},\ \mathrm{TANRMSE}^3_{\alpha\beta\gamma} = 70.65\,\%,$$
$$\mathrm{TARMSE}^3_{\mathrm{FMF}} = 0.01329\mathrm{m},\ \mathrm{TANRMSE}^3_{\mathrm{FMF}} = \mathbf{66.45}\,\%$$

Estos datos muestran que el filtro FMF mejora en un 6 % al filtro $\alpha\beta\gamma$. La mejora del FMF frente al filtro $\alpha\beta$ es muy grande porque en el tramo de modelo DWPA éste acumula mucho error. Notar que el filtro $\alpha\beta\gamma$ sufre un transitorio en t=6.68 segundos (ver Figura 8), debido a que la trayectoria sufre una discontinuidad al pasar del modelo DWPA al modelo DWNA y desaparecer súbitamente el estado de aceleración.

Simulaciones usando *tray*$_4$ (Nr=$15 \cdot 10^3$):

La media cuadrática temporal de los errores absolutos y normalizados son:

$$\mathrm{TARMSE}^4_{\alpha\beta} = 0.02035\mathrm{m},\ \mathrm{TANRMSE}^4_{\alpha\beta} = 101.84\,\%,$$
$$\mathrm{TARMSE}^4_{\alpha\beta\gamma} = 0.01374\mathrm{m},\ \mathrm{TANRMSE}^4_{\alpha\beta\gamma} = 68.74\,\%,$$
$$\mathrm{TARMSE}^4_{\mathrm{FMF}} = 0.013176\mathrm{m},\ \mathrm{TANRMSE}^4_{\mathrm{FMF}} = \mathbf{65.92}\,\%$$

Estos datos muestran que el filtro FMF mejora en un 35 % al filtro $\alpha\beta$ y en un 4 % al $\alpha\beta\gamma$. Notar que el filtro FMF mejora al $\alpha\beta\gamma$ a pesar de que el $\alpha\beta$ está lejos de su funcionamiento ideal gran parte de la trayectoria. En la Figura 9 se muestra el error de predicción del filtro $\alpha\beta$ que tiene forma de senoide desplazada, siendo unas veces mayor y otras menor que el error del filtro $\alpha\beta\gamma$, el cual es bastante uniforme. El error del FMF también tiene forma de senoide desplazada pero de menor amplitud y en algún tramo es simultáneamente menor que los de los filtros $\alpha\beta$ y $\alpha\beta\gamma$. Esto se debe a que en la suma ponderada de los dos filtros uno tiene error positivo y el otro negativo, resultando un error absoluto en el FMF menor que el de cualquiera de los dos.

## 3. IMPLEMENTACIÓN DEL HRT FMF

El mayor problema de la implementación del filtro FMF es el tiempo necesario para calcular todos los filtros involucrados en la predicción final. La solución que se presenta en este artículo

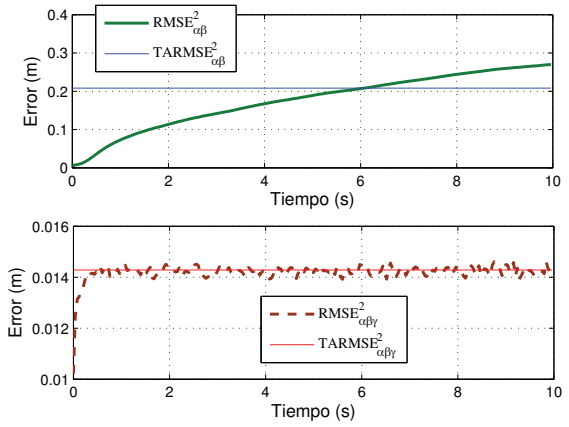


Figura 7. Errores de predicción para $tray_2$.

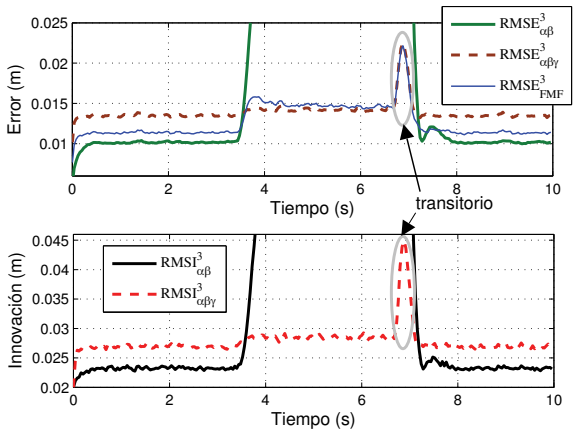


Figura 8. Errores e innovaciones para $tray_3$.

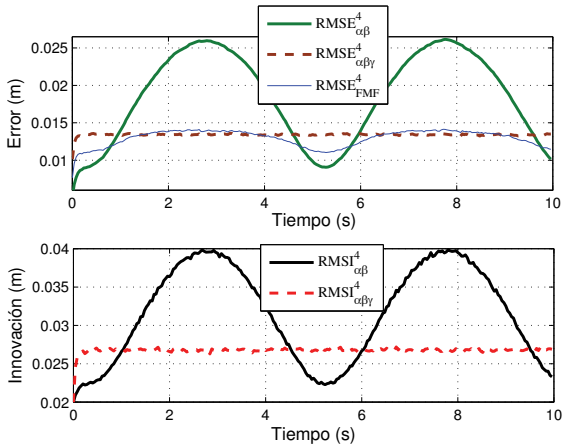


Figura 9. Errores e Innovaciones para $tray_4$

es la implementación del FMF en una FPGA (Perez-Vidal and Gracia, 2009) donde se calculen todos los filtros que componen el FMF de forma simultánea e independiente. Para la implementación en tiempo real se han usado las tarjetas de desarrollo Diligent Spartan3E (Xilinx xc3s500e-4) y Nallatech Ballynuey 3 (Virtex xc2v3000-4), así como el módulo DCM (*Digital Clock Manager*), que permite obtener un periodo de muestreo muy bajo: T=10ns (100MHz) para un filtro de Kalman estacionario. En cuanto a la programación, se ha utilizado la librería VHDL-200x y las variables se han definido en coma fija con una precisión de 0.00003 (32 bits).

### *3.1 Análisis del tiempo de ejecución*

Denominando $\mathrm{T_{comp}}$ al tiempo necesario para calcular la salida el filtro completo, incluyendo el procesamiento de la imagen $\mathrm{T_p}$, y $\mathrm{T_a}$ al tiempo de adquisición de la cámara, se tienen tres casos prácticos distintos:

(a) $\mathrm{T_{comp}} \ll \mathrm{T_a}$: El tiempo necesario para calcular el filtro completo ($\mathrm{T_{comp}}$) es mucho menor que el tiempo de adquisición ($\mathrm{T_a}$), lo que da lugar al denominado procesamiento en línea (*on-the-fly processing*) (Chroust and Vincze, 2003). En este caso (ver Figura 10(i)): $\mathrm{T} = \mathrm{T_a}$; Latencia $= \mathrm{T}$; y $\mathrm{A_1}$ propaga $\mathrm{T_a}$, i.e. $\mathrm{A_1} = e^{\mathrm{A_c}\cdot\mathrm{T}}$, donde $\mathrm{A_c}$ es la matriz continua del sistema. La innovación se obtiene utilizarse en las reglas del filtro FMF.

(b) $\mathrm{T_{comp}} \approx \mathrm{T_a}$: El tiempo necesario para calcular el filtro completo ($\mathrm{T_{comp}}$) es aproximadamente el mismo que el tiempo necesario para adquirir la imagen ($\mathrm{T_a}$). En este caso (Figura 10(ii)): $\mathrm{T} = \max(\mathrm{T_a},\mathrm{T_{comp}})$; Latencia $= 2\mathrm{T}$; la matriz $\mathrm{A_1}$ propaga $\mathrm{T} - \mathrm{T_p}$, i.e. $\mathrm{A_1} = e^{\mathrm{A_c}\cdot(\mathrm{T}-\mathrm{T_p})}$; y la matriz $\mathrm{A_2}$ propaga $2\mathrm{T}$, i.e. $\mathrm{A_2} = e^{\mathrm{A_c}\cdot 2\mathrm{T}}$.

(c) $\mathrm{T_{comp}} > \mathrm{T_a}$: El tiempo necesario para calcular el filtro FMF es mayor que el tiempo de adquisición. Por tanto, el algoritmo no se puede calcular a tiempo. Una posible solución para evitarlo es segmentar. La Figura 10(iii) muestra el filtro FMF segmentado: las líneas discontinuas son registros de la segmentación; los subíndices "v" y "a" son para el filtro $\alpha\beta$ y $\alpha\beta\gamma$, respectivamente; las matrices $\mathrm{A_{1v}}$ y $\mathrm{A_{1a}}$ propagan $\mathrm{T_1}$+$\mathrm{T_2}$+$\mathrm{T_3}$; las matrices $\mathrm{A_{2v}}$ y $\mathrm{A_{2a}}$ propagan $\mathrm{T_1}$+...+$\mathrm{T_6}$+$\mathrm{T_d}$; y Latencia $= \mathrm{T_1} + ... + \mathrm{T_6} + \mathrm{T_d}$. Los segmentos del algoritmo se deben diseñar de forma que su tiempo de ejecución sea lo más parecido posible.

Las causas que producen los retardos en los sistemas de control visual se describen en (Sharkey and Murray, 1996) y se consideran uno de los mayores problemas en los sistemas de control visual. En el mejor de los casos el retardo introducido por el sistema de visión será de un ciclo del lazo de control, i.e. el caso (a) indicado. Prácticamente para todas las implementaciones y arquitecturas (Hutchinson et al., 1996) el sistema de visión introduce un retardo mínimo de dos ciclos (Vincze and Hager, 2000).

### *3.2 Comparativa: PC-MATLAB vs FPGA-VHDL*

El código fuente de los filtros para la implementación en MATLAB y VHDL se puede consultar en: http://personales.upv.es/luigraca/RIAI/fuentes.htm. La comparativa entre ambas implementaciones se realiza en este apartado para las dos últimas trayectorias, $tray_3$ y $tray_4$, por ser las más próximas a un caso práctico. Para la implementación en la FPGA, se han utilizado trayectorias de 250 puntos e inicialización por el método TPD (Bar-Shalom et al., 2001). Para analizar las resultados de este apartado, se definen los siguientes términos: RMSET (valor RMS del error de predicción con respecto al tiempo); RMSME (valor RMS de los errores de medida con respecto al tiempo); y NRMSET (RMSET normalizado). Las expresiones siguientes definen dichos términos:

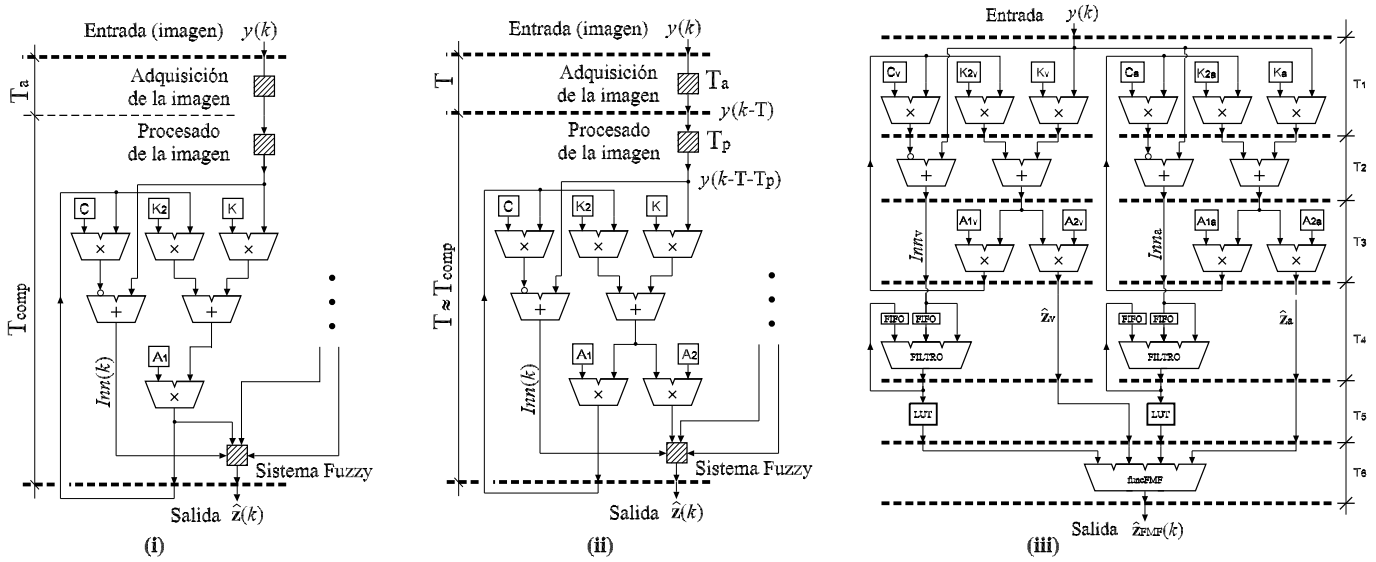


Figura 10. **(i)** Implementación de un filtro de Kalman de régimen permanente: caso (a). **(ii)** Implementación de un filtro de Kalman de régimen permanente: caso (b). **(iii)** Estructura de tipo segmentado (*pipe-line*) del filtro FMF propuesto en este artículo.

$$\mathrm{RMSET} = \sqrt{\sum_{k=\mathrm{D}+1}^{\mathrm{N}} \frac{\big(x(k) - \hat{x}(k)\big)^2}{(\mathrm{N}-\mathrm{D})}} \tag{16}$$

$$\mathrm{RMSMET} = \sqrt{\sum_{k=\mathrm{D}+1}^{\mathrm{N}} \frac{\big(x(k) - y(k)\big)^2}{(\mathrm{N}-\mathrm{D})}} \tag{17}$$

$$\mathrm{NRMSET} = \frac{\mathrm{RMSET}}{\mathrm{RMSMET}} \tag{18}$$

donde $x(k)$ es la posición real del objeto en el instante $k$; $\hat{x}(k)$ es la predicción de la posición del objeto en el instante $k$; $y(k)$ es la medida de la posición del objeto en el instante $k$; N es el número de muestras de la trayectoria; y D es el número de muestras iniciales despreciadas en el cálculo de los valores RMS. Los valores definidos por las expresiones anteriores se usan nuevamente especificando el tipo de trayectoria con un superíndice y el tipo de filtro con un subíndice. A continuación se muestran los valores de los indicadores de error (16)-(18) correspondientes a la implementación en PC, programada en MATLAB, del caso (a) y caso (b) con las dos trayectorias consideradas.

Implementación del caso (a) ($\mathrm{T_{comp}} \ll \mathrm{T_a}$):

- Usando *tray*$_3$:
  $\mathrm{RMSMET}^3 = 0.065537\mathrm{m}$
  $\mathrm{RMSET}^3_{\alpha\beta} = 0.035294\mathrm{m}$, $\mathrm{NRMSET}^3_{\alpha\beta} = 53.85\,\%$,
  $\mathrm{RMSET}^3_{\alpha\beta\gamma} = 0.017082\mathrm{m}$, $\mathrm{NRMSET}^3_{\alpha\beta\gamma} = 26.06\,\%$,
  $\mathrm{RMSET}^3_{\mathrm{FMF}} = 0.016212\mathrm{m}$, $\mathrm{NRMSET}^3_{\mathrm{FMF}} = \mathbf{24.74}\,\%$,

- Usando *tray*$_4$:
  $\mathrm{RMSMET}^4 = 0.025874\mathrm{m}$
  $\mathrm{RMSET}^4_{\alpha\beta} = 0.016169\mathrm{m}$, $\mathrm{NRMSET}^4_{\alpha\beta} = 62.49\,\%$,
  $\mathrm{RMSET}^4_{\alpha\beta\gamma} = 0.017650\mathrm{m}$, $\mathrm{NRMSET}^4_{\alpha\beta\gamma} = 68.22\,\%$,
  $\mathrm{RMSET}^4_{\mathrm{FMF}} = 0.014917\mathrm{m}$, $\mathrm{NRMSET}^4_{\mathrm{FMF}} = \mathbf{57.65}\,\%$,

Implementación del caso (b) ($\mathrm{T_{comp}} \approx \mathrm{T_a}$):

- Usando *tray*$_3$:
  $\mathrm{RMSMET}^3 = 0.127752\mathrm{m}$
  $\mathrm{RMSET}^3_{\alpha\beta} = 0.045095\mathrm{m}$, $\mathrm{NRMSET}^3_{\alpha\beta} = 68.81\,\%$,
  $\mathrm{RMSET}^3_{\alpha\beta\gamma} = 0.023025\mathrm{m}$, $\mathrm{NRMSET}^3_{\alpha\beta\gamma} = 35.13\,\%$,
  $\mathrm{RMSET}^3_{\mathrm{FMF}} = 0.020842\mathrm{m}$, $\mathrm{NRMSET}^3_{\mathrm{FMF}} = \mathbf{31.80}\,\%$,

- Usando *tray*$_4$:
  $\mathrm{RMSMET}^4 = 0.039662\mathrm{m}$
  $\mathrm{RMSET}^4_{\alpha\beta} = 0.019787\mathrm{m}$, $\mathrm{NRMSET}^4_{\alpha\beta} = 76.47\,\%$,
  $\mathrm{RMSET}^4_{\alpha\beta\gamma} = 0.023044\mathrm{m}$, $\mathrm{NRMSET}^4_{\alpha\beta\gamma} = 89.07\,\%$,
  $\mathrm{RMSET}^4_{\mathrm{FMF}} = 0.018491\mathrm{m}$, $\mathrm{NRMSET}^4_{\mathrm{FMF}} = \mathbf{71.46}\,\%$,

La implementación en la FPGA, programada con VHDL, ha dado prácticamente los mismos datos que antes para todos los experimentos. Las pequeñas diferencias entre ambas implementaciones son estables, tienen un valor medio de $2.5 \cdot 10^{-5}$m y se deben principalmente a que en la FPGA los datos se representan en coma fija con 32 bits. Como se observa en los datos mostrados, el comportamiento del filtro FMF es mejor en todos los casos que el comportamiento de los filtros $\alpha\beta$ y $\alpha\beta\gamma$. La Tabla 1 contiene los valores de coste computacional de cada filtro usando un Intel Core 2 Duo 2,13 GHz. El tiempo de ejecución de los filtros $\alpha\beta$ y $\alpha\beta\gamma$ es muy similar a pesar de que en el segundo los vectores y matrices son de mayor dimensión. También se puede apreciar que el caso (a) requiere, como era previsible, un poco menos tiempo de cómputo que el caso (b). Por otro lado, el tiempo necesario para calcular el filtro FMF es mayor que la suma de los filtros $\alpha\beta$ y $\alpha\beta\gamma$ debido a la posterior mezcla borrosa (sistema *fuzzy*) para tener la predicción final. La Tabla 2 muestra los recursos utilizados para implementar en la FPGA el filtro $\alpha\beta$ y el filtro hRT FMF desarrollado.

**Tabla 1. Coste computacional de los filtros implementados en PC-MATLAB ($\mu$s / iteración)**

| Filtro | $\alpha\beta$ | $\alpha\beta\gamma$ | FMF |
|---|---|---|---|
| Tiempo en caso (a) | 3.5 | 3.55 | 19.64 |
| Tiempo en caso (b) | 4.59 | 4.59 | 21.85 |

**Tabla 2. Utilización de dispositivos para la implementación de un filtro $\alpha\beta$ y de un filtro FMF**

| Filtro | $\alpha\beta$ | FMF |
|---|---|---|
| Tarjeta | Spartan3E | Ballynuey 3 |
| Dispositivo | Xilinx xc3s500e-4 | Virtex xc2v3000-4 |
| f max (sin segm.) | 50.74MHz | 12.49MHz |
| f max (segm.) | 101.72MHz | 32.78MHz |
| Latencia | 3 ciclos de reloj | 6 ciclos de reloj |
| MULT18X18SIO | 6 de 20 | 36 de 96 |
| ($T_1$,$T_2$,$T_3$) | (9.83,3.54,6.34)ns | (9.83,3.54,6.34) |
| ($T_4$,$T_5$,$T_6$) | (–,–,–) | (26.7,3.1,30.5)ns |
| Nº de puertas | 27878 de 5e5 | 277079 de 3e6 |

## 4. RESULTADOS EXPERIMENTALES

Las pruebas experimentales se han realizado sobre la plataforma de pruebas mostrada esquemáticamente en la Figura 11 (Corke, 1996). La Figura 12 muestra el montaje real, que consta de un disco de color claro en el que se ha dibujado una característica visual negra cerca del bordo. Esta característica visual gira gracias a un motor accionado por un variador de frecuencia. Frente a este disco se ubica el conjunto cámara ST-VL6524 y unidad pan/tilt (PTU), ver la Figura 14, la cual está dotada de dos motores para realizar el seguimiento. El objetivo es tener la característica visual siempre centrada en el plano de imagen (Perez-Vidal et al., 2009). Se ha implementado en la FPGA (Nallatech Ballynuey 3) la adquisición y el procesamiento de la imagen junto al algoritmo de control y el filtro de predicción propuesto (uno en cada coordenada de la PTU). El periodo de muestreo utilizado para cerrar el lazo de control visual es de $T = T_a = 40$ms. El tiempo total de cómputo $T_{comp}$, incluyendo la ley de control, está entorno a 92.5ns para la implementación FPGA-VHDL (algoritmo no segmentado) y a 39.61$\mu$s para la implementación PC-MATLAB. Teniendo en cuenta estos valores, el experimento corresponde al caso (a) ($T_{comp} \ll T_a$) para ambas implementaciones y la trayectoria descrita por la característica es similar a *tray*$_4$. Las Figuras 15 y 16 muestran, para un experimento concreto, la posición y velocidad en 2D de dos variables en el plano de movimiento del disco. Estas variables son la característica y la intersección entre el eje óptico de la cámara y el plano de movimiento.

Es interesante señalar que, aunque el experimento anterior se puede ejecutar en ambas plataformas, FPGA-VHDL y PC-MATLAB, en las aplicaciones reales de control visual (e.g. guiado de vehículos, SLAM de robots móviles o aplicaciones médicas) existen diferentes factores que incrementan el coste computacional de los algoritmos:

- Puede haber más de una característica visual: usualmente se realiza el seguimiento de varios cientos de características.
- El filtro de predicción se ha calculado para dos dimensiones, pero para control visual basado en posición, éste cálculo se puede realizar para 6 grados de libertad.
- En la actualidad, cada vez son más comunes las cámaras de alta velocidad, lo que puede reducir la adquisición de 40ms (25 fps) a por ejemplo 4ms (250 fps).
- Se puede utilizar más de una cámara (e.g. visión estéreo).

Un ejemplo de aplicación real, podría ser el guiado de un vehículo, donde se puede tener una configuración de este tipo: dos cámaras (visión estéreo) rápidas (de

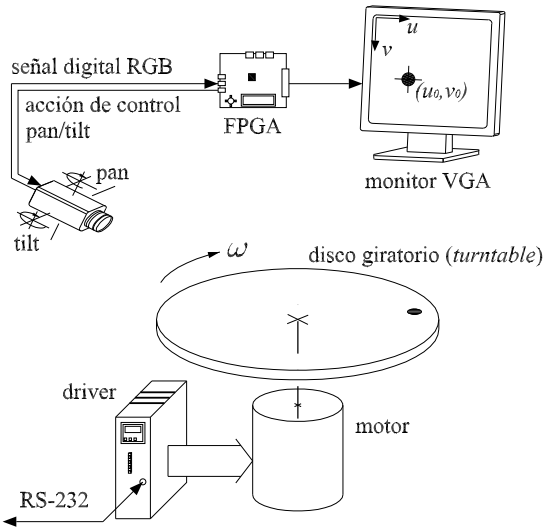


Figura 11. Disposición de la plataforma experimental.

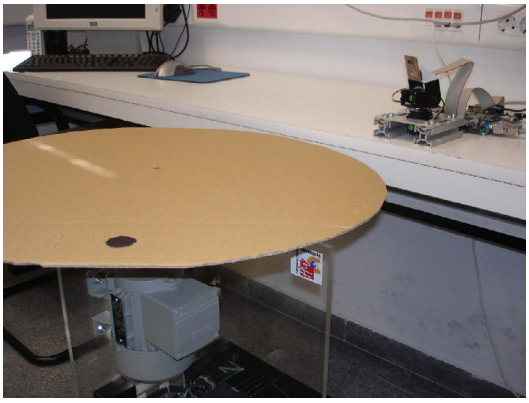

Figura 12. Fotografía de la plataforma experimental.

Figura 13. Disco giratorio

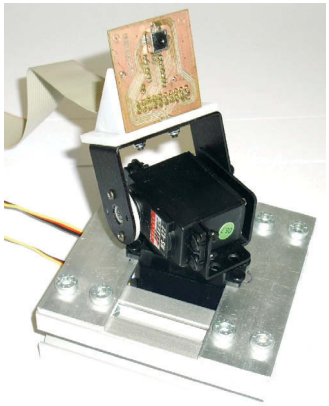

Figura 14. Unidad pan-tilt con cámara ST-VL6524.

250 fps), 300 características visuales y 6 grados de libertad (posición 3D y ángulos yaw, pitch y roll). Toda esta información necesita ser calculada más rápido que 4ms(250fps)/[2(cámaras)·300(características visuales)·6(gdl)]=1.11$\mu$s. Este cómputo no se podría realizar con la implementación PC-MATLAB ($T_{comp}$ =39.61$\mu$s) y sólo la implementación paralelizada en la FPGA cumpliría los requisitos temporales ($T_{comp}$ =92.5ns). Además, si se

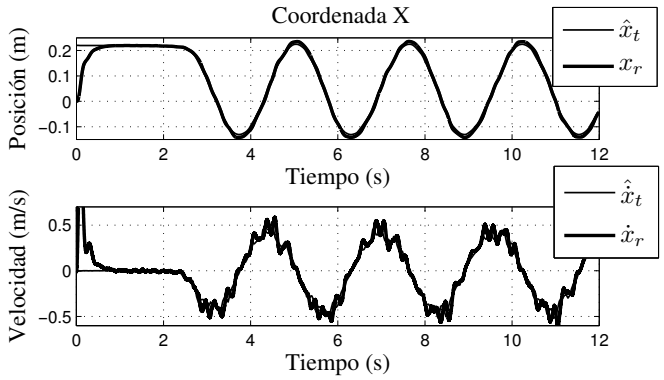


Figura 15. Posición y velocidad en el eje del "pan" de la PTU.

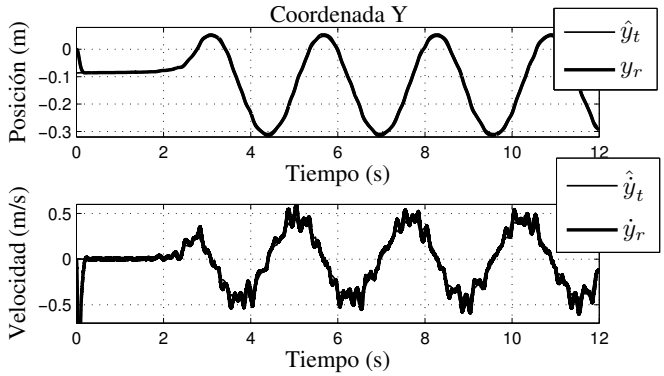


Figura 16. Posición y velocidad en el eje del "tilt" de la PTU.

implementa un filtro FMF más complejo, como el de la Figura 1(i), su coste computacional puede verse incrementado fácilmente de 5 a 10 veces.

## 5. CONCLUSIÓN

En este artículo se propone el desarrollo de un filtro de predicción, denominado FMF (*Fuzzy Mix of Filters*), que combina o fusiona varios filtros utilizando técnicas borrosas y la variable de funcionamiento óptimo o innovación de cada filtro. El uso de un sistema borroso es una buena solución porque hace intuitiva la difícil tarea de combinar varios filtros de diferentes tipos. En particular, para un FMF específico se muestra que la estimación es mejor que la realizada por cualquiera de los filtros que componen la mezcla o fusión. El filtro propuesto tiene un mayor coste computacional que los filtros utilizados tradicionalmente en control visual y puede ser utilizado en aplicaciones sin fuertes restricciones temporales. Para el resto de aplicaciones, se ha mostrado cómo el FMF puede implementarse en un procesador paralelo o FPGA para reducir su coste computacional significativamente. Además, se han mostrado resultados experimentales para una aplicación concreta de control visual que permiten validar los algoritmos y métodos propuestos así como apreciar sus ventajas. Como trabajo futuro queda el desarrollar e implementar un filtro FMF más complejo que combine mayor número de filtros, y que por tanto proporcione una mejor predicción.

## AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por Bancaja. Los autores agradecen a los revisores sus comentarios para la mejora de este artículo.

## REFERENCIAS

Bar-Shalom, Y., X. Li, and T. Kirubarajan (2001). *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, New York, USA.

Chroust, S. and M. Vincze (2003). Improvement of the prediction quality for visual servoing with a switching kalman filter. *International Journal of Robotics Research* **22**(10-11), 905–922.

Corke, P. (1996). *Visual Control of Robots: High performance Visual Servoing.* Research Studies Press. Taunton, Somerset, England.

Corke, P. and S. Hutchinson (2000). Real-time vision, tracking and control. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*. p. 622–629. San Francisco, CA, USA.

Efe, M. and D. Atherton (1998). Maneuvering target tracking with an adaptative kalman filter. In: *Proc. of the 37th IEEE Conference on Decision and Control*. p. 737–742. Tampa, Florida, USA.

Hutchinson, S., G. Hager and P. Corke (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation* **12**(5), 651–670.

Kawase, T., H. Tsurunosono, N. Ehara, and I. Sasase (1997). Alpha-beta tracking filter combined with ellipsoidal prediction using generalized hough transform. *Radar* **97**, 609–613.

Kolodziej, J. and T. Singh (2000). Target tracking via a dynamic circular filter/linear alpha-beta filters in 2-d. In: *Proc. of the American Control Conference*. p. 4343–4347. Chicago, IL, USA.

Moore, M. and J. Wang (2001). Adaptive dynamic modelling for kinematic positioning. In: *IAG Scientific Meeting*. Budapest, Hungary.

Pérez, C., N. García, J.M. Sabater, J.M. Azorín, O. Reinoso and L. Gracia (2007). Improvement of the visual servoing task with a new trajectory predictor. the fuzzy kalman filter. In: *Proc. of the Int. Conf. on Informatics in Control, Automation and Robotics*. p. 133–140. Angers, France.

Perez-Vidal, C. and L. Gracia (2009). High speed filtering using reconfigurable hardware. *Journal of Parallel and Distributed Computing* **69**(11), 896–904.

Perez-Vidal, C., L. Gracia, N. Garcia and E. Cervera (2009). Visual control of robots with delayed images. *Advanced Robotics* **23**(6), 725–745.

Robert, C. and G. Casella (1999). *Monte Carlo Statistical Methods.* Springer-Verlag. New York, USA.

Sharkey, P. and D. Murray (1996). Delays versus performance of visually guided systems. *IEE Proceedings on Control Theory and Applications* **143**(5), 436–447.

Simon, D. (2006). *Optimal State Estimation: Kalman, Hinf and Nonlinear Approaches.* Wiley-Interscience. Hoboken, NJ, USA.

Tenne, D. and T. Singh (2002). Circular prediction algorithms - hybrid filters. In: *Proc. of the American Control Conference*. p. 172–177. Anchorage, AK, USA.

Vincze, M. and G. Hager (2000). *Robust Vision for Vision-Based Control of Motion.* IEEE Press. Piscataway, NJ, USA.

Yoo, J.-C. and Y.-S. Kimb (2003). Alpha-beta-tracking index (Alpha-beta-Lambda) tracking filter. *Signal Process* **83**(11), 169–180.