

Implementación de un Sistema de Visión para Localización Bidimensional con Estimación de Velocidad

Luis García, Tadeo Espinoza, Ricardo Parada y Alejandro Dzul
Instituto Tecnológico de la Laguna
Torréon, Coahuila, México

Resumen—Este artículo presenta una metodología para la implementación de un sistema de localización de vehículos, en dos dimensiones, por medio de visión artificial. Como aportaciones del artículo, se plantea la utilización de una red neuronal para mapear la posición de coordenadas de la cámara a un marco ubicado en el suelo; así como la estimación de la velocidad en la que se aplica, de manera práctica, un diferenciador por medio de modos deslizantes de orden superior. El sistema de localización planteado se puede utilizar con cualquier tipo de vehículo autónomo que pueda desplazarse dentro del área de alcance de la cámara. Derecho reservado © UNAM-AMCA.

Palabras clave: Sistema de visión, red neuronal, modos deslizantes, diferenciador.

I. INTRODUCCIÓN

Entre los diversos problemas relacionados con la navegación autónoma de robots móviles, la localización de los vehículos es una problemática importante a resolver. Para ello, tecnológicamente se utilizan diferentes equipos como: módulos GPS, ondas de radio frecuencia, visión artificial, etc.

Con los recientes avances en el campo de la visión artificial, ha sido posible estimar las coordenadas de localización con una alta frecuencia de muestreo. Los sistemas de visión artificial han sido aplicados a robots tipo carro, submarinos y robots aéreos no tripulados (W. Bath and J. Paxman, 2005).

Distintos métodos avanzados de control han sido desarrollados para responder a las demandas cada vez mayores en el desempeño de los sistemas, no obstante, requieren información completa de los estados que puede limitar su utilización práctica. Aun si es posible medir todos los estados, generalmente están alterados por ruido. Más aún, el incremento en el número de sensores hace al sistema más complejo en su implementación, incrementando también el costo de su realización (A. Benallegue et al, 2007).

En este artículo se reporta un método de medición de posición y velocidad lineal en el plano $x - y$ por medio de una cámara montada en el techo y apuntando hacia el suelo. Dado que es una cámara especializada para sistemas de visión, tanto la resolución como la velocidad de muestreo

son mayores que las tradicionales cámaras web, por lo que permite un mejor sentido.

En el método tradicionalmente utilizado para transformar coordenadas de la cámara a un marco en el mundo real, es necesario realizar un buen modelado de la cámara, esto es, conocer los parámetros intrínsecos de la cámara. Una vez que se conoce el modelo de la cámara, se aplican operaciones matriciales, en donde estas operaciones transforman las coordenadas de píxeles a coordenadas de distancia, eliminando además los errores debidos a la distorsión radial (R. Hartley and A. Zisserman, 2004). Obtener parámetros adecuados para la cámara es bastante difícil y se debe hacer a “prueba y error”, con la posibilidad de que no se pueda modelar adecuadamente la cámara. Para evitar esta forma de modelado, en este trabajo se recurrió a la utilización de una red neuronal. La red es entrenada mapeando ciertos puntos (coordenadas) en píxeles y como objetivo (*target*) de cada punto, la posición en centímetros en un marco definido en el suelo. Una vez entrenada la red, se miden distintos puntos para comprobar la precisión. Con la red neuronal, el error de medición no se va incrementando a las orillas de la imagen como sucede en el método tradicional. La desventaja de este tipo de calibración por medio de redes neuronales es que al cambiar la altura del objeto a medir, o al variar la altura de la cámara, se debe entrenar otra vez la red neuronal con los datos de las nuevas condiciones.

Al implementar un sistema de localización, resulta casi indispensable proporcionar también una estimación de la velocidad del objeto sentido. Una forma de hacerlo puede ser midiendo el flujo óptico, sin embargo los algoritmos para determinarlo involucrarían un incremento en el tiempo de muestreo. Otra forma es recurrir a métodos de diferenciación de la posición. Uno de los algoritmos más sencillos de diferenciación discreta es el de “diferencia hacia atrás”, definido como $\frac{dx(t)}{dt} = \frac{x(t) - x(t-T)}{T}$, sin embargo este método es muy impreciso para calcular la derivada en este tipo de sistema. Una solución al aplicar la diferencia hacia atrás es utilizar un filtro pasa-bajas, después de lo cual resulta una aproximación bastante buena de la derivada. En (A. Levant, 2001) se afirma que el diferenciador exacto por modos deslizantes de orden superior es posiblemente la mejor solución para el problema simple de diferenciación

numérica en tiempo real. En el presente artículo se implementa un estimador de velocidad por modos deslizantes de orden superior, basado en el trabajo de (A. Levant, 2001). Este método brinda también gran precisión con muestreos discretos (A. Levant and L. Alelishvili, 2004).

El artículo está compuesto por las siguientes secciones: en la sección II se presenta el uso de las redes neuronales para la transformación de coordenadas; la sección III contiene la aplicación del diferenciador exacto por modos deslizantes de orden superior; en la sección IV se tiene la descripción del software utilizado, del robot móvil experimental, así como los resultados prácticos obtenidos; por último, la sección V contiene las conclusiones.

II. REDES NEURONALES PARA TRANSFORMACIÓN DE COORDENADAS

Para el correcto mapeo de las coordenadas $X - Y$ de la imagen hacia coordenadas $x - y$ en el plano del suelo, se desarrolló una red neuronal con una capa de entrada correspondiente a 2 señales (posición $X - Y$ de la imagen), una capa oculta de 10 neuronas, y una capa de salida de 2 neuronas, como la que se muestra en la Fig. 1.

El sensor de posición utilizado corresponde a una cámara montada en el techo, la cual apunta hacia el suelo, como se muestra en la figura 2. El área que se mide es aproximadamente de 3.7 x 1.6 mts.

Para adquirir los datos necesarios para entrenar la red neuronal, se midió en distintos puntos la posición del vehículo en el plano del mapa de bits, se guardaron estos datos en una matriz de entrada a la red, y a su vez se creó una matriz con las posiciones correspondientes a cada punto medido, pero en el plano del suelo (en centímetros).

La red neuronal fue creada y entrenada en Matlab R2010b con la herramienta *nftool*.

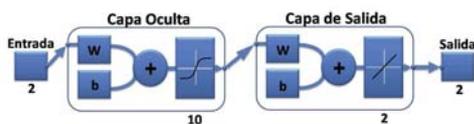


Figura 1: Esquema a bloques de la red neuronal utilizada.

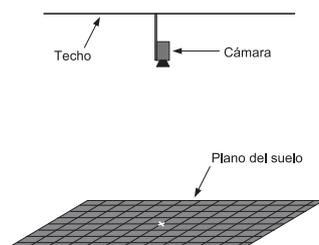


Figura 2: Esquema de la implementación del sensor de posición.

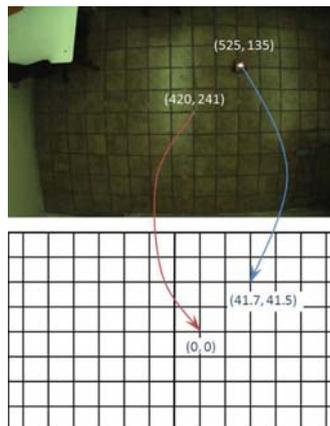


Figura 3: Determinación de coordenadas.

III. DIFERENCIADORES

III-A. Aproximación Discreta con el Método de la Diferencia Hacia Atrás

La derivada de una función en el tiempo representa la variación de dicha función en cada intervalo infinitesimal de tiempo. La aproximación más simple de una derivada, en tiempo discreto, se obtiene calculando la diferencia entre la muestra actual y la anterior, y dividiendo entre el período de muestreo en ese instante de tiempo.

$$\frac{df(t)}{dt} = \frac{f(t) - f(t - \tau)}{\tau} \quad (1)$$

La desventaja de utilizar este diferenciador en el sistema de localización consiste en que la cámara llega a muestrear en ocasiones consecutivas un mismo valor de posición, aunque el objeto se esté moviendo (debido a que la cámara muestrea continuamente y almacena datos en el búfer) y el diferenciador estará interpretando en esos momentos una diferencia de cero. Otra desventaja en este tipo de diferenciador es que los valores de la derivada instantánea van a depender mucho del período de muestreo, por lo que pueden resultar valores muy irreales. Una solución al usar la diferencia hacia atrás en este tipo de sistemas es la utilización de un filtro pasa-bajas.

III-B. Filtro Discreto Pasa-Bajas

El filtro discreto que se utilizará se define como

$$G(z) = \frac{(1 - e^{-\alpha\tau})z^{-1}}{1 - e^{-\alpha\tau}z^{-1}} \quad (2)$$

Al filtrar la derivada discreta con la función de transferencia (2) se obtiene una muy aceptable aproximación de la derivada. El único problema consiste en seleccionar adecuadamente la constante de tiempo α , pues con un valor muy pequeño la derivada filtrada estará por debajo del valor instantáneo que debería tener, además de que la señal se va defasando (atrasando con respecto a la derivada real). Si se selecciona un valor muy alto de α , la derivada filtrada conservará mucho del ruido de la señal que se está tratando de filtrar.

III-C. Diferenciador Exacto por Modos Deslizantes

La técnica de modos deslizantes está relacionada a la teoría de sistemas de estructura variable. Los modos deslizantes estándar son robustos y de gran precisión, sin embargo presentan una característica indeseada que es el llamado efecto *chattering* causado por la alternancia del control (A. Levant and L. Alelishvili, 2004).

Una solución para tratar de reducir el *chattering* es reemplazar la función *signo* con una función *saturación* con pendiente alta. Desafortunadamente con esta solución se ve comprometido el desempeño del controlador por modos deslizantes. Para reducir o eliminar el *chattering* se pueden utilizar Modos Deslizantes de Orden Superior (HOSM - High Order Sliding Modes), los cuales generalizan la idea básica de los modos deslizantes, pero actúan en una derivada temporal de orden más alto, en lugar de influenciar la primera derivada, como ocurre en los modos deslizantes estándar. Los HOSM remueven el *chattering* a la vez que conservan las ventajas del esquema original (robustez) (M. Rolink, et al, 2006).

Los observadores por HOSM poseen atractivas características como lo son: insensibilidad con respecto a entradas desconocidas; posibilidad de usar los valores de la inyección de salida equivalente para la identificación de entradas desconocidas; y convergencia en tiempo finito a los valores exactos de los vectores de estados (A. Benallegue et al., 2007) (Jean-Pierre Barbot and T. Floquet, 2009). La convergencia en tiempo finito puede ser hecha arbitrariamente pequeña, y para ello se debe ajustar sólo un parámetro escalar (A. Levant, 2001).

Otro tipo de diferenciadores, aun los más modernos, no dan una diferenciación exacta con tiempo de convergencia finito o requieren algún conocimiento del ruido y eliminarían las ventajas de los HOSM. Las derivadas, entonces, deben ser calculadas por la implementación sucesiva de un diferenciador exacto robusto de primer orden (A. Levant, 2001).

III-C.1. Diferenciador Exacto de Orden Arbitrario: Sea la señal de entrada $f(t)$ una función definida en $[0, \infty)$ que consiste en un ruido *Lebesgue-medible* acotado con características desconocidas, y una señal base desconocida $f_0(t)$, donde la primera derivada tiene una constante Lipschitz $C > 0$. El problema consiste en encontrar una estimación robusta de $\dot{f}(t)$ que sea exacta en ausencia de ruido de medición.

El diferenciador de primer orden está dado como

$$\dot{z}_0 = -\lambda_0 |z_0 - f(t)|^{2/3} \text{sign}(z_0 - f(t)) + z_1 \quad (3)$$

$$\dot{z}_1 = -\lambda_1 \text{sign}(z_0 - f(t)) \quad (4)$$

donde

$$\text{sign}(x) = \begin{cases} -1, & \text{si } x < 0 \\ 1, & \text{si } x \geq 0 \end{cases} \quad (5)$$

Debido a que el anterior derivador fue obtenido de un esquema recursivo, se puede obtener de forma similar, un diferenciador de segundo orden.

$$\dot{z}_0 = -\lambda_0 |z_0 - f(t)|^{2/3} \text{sign}(z_0 - f(t)) + z_1 \quad (6)$$

$$\dot{z}_1 = -\lambda_1 |z_1 - \dot{z}_0|^{1/2} \text{sign}(z_1 - \dot{z}_0) + z_2 \quad (7)$$

$$\dot{z}_2 = -\lambda_2 \text{sign}(z_2 - \dot{z}_1) \quad (8)$$

Teorema 1: (A. Levant, 2001) Seleccionando adecuadamente los parámetros, se cumplen las siguientes igualdades, en ausencia de ruido en la entrada, después de un tiempo finito de un proceso transitorio:

$$z_0 = f_0(t) \quad (9)$$

$$z_1 = \dot{f}_0(t) \quad (10)$$

Demostración:

La demostración se encuentra en (A. Levant, 2001). ■

IV. PLATAFORMA Y RESULTADOS EXPERIMENTALES



Figura 4: AVT Guppy Cam.

IV-1. Cámara: Especializada en sistemas de visión. De la marca AVT modelo Guppy Cam, con las siguientes características: video a color, con resolución de 752×480 píxeles; velocidad nominal de 60 cuadros por segundo y tipo de conexión Firewire

Esta cámara permite modificar por software características como brillo, contraste, tiempo de exposición, etc. Con el objeto de poder medir un marcador de manera sencilla y con poco "ruido", la característica que nos interesa modificar es el tiempo de exposición de la imagen a un valor muy pequeño, con el fin de que la imagen captada por la cámara sea muy oscura, de modo que sólo se vea el brillo de un led, el cual va a ser el marcador a captar.

IV-2. Software: Se realizó una interfaz en Visual C++ 2005, la cual se encarga de realizar las siguientes acciones:

- Inicializa la cámara.
- Toma imágenes en bruto desde la cámara y las convierte a un formato en RGB, al mismo tiempo que identifica la posición del marcador.
- Una función con la red neuronal cambia las coordenadas del marcador ($X - Y$ en píxeles) a coordenadas en un marco en el suelo.
- Se realiza el algoritmo diferenciador para obtener las velocidades en los ejes x y y .
- Visualiza la imagen de la cámara encerrando en cada muestreo al marcador dentro de la imagen.
- Visualiza gráficas de posición y velocidad en el tiempo, además de poder graficar la trayectoria x contra y seguida por el vehículo.

- Graba en tiempo de ejecución los datos de interés en un archivo de texto.
- Transmite los datos de interés vía Zigbee.

IV-A. Precisión

Se tomaron 9 mediciones en los cuatro cuadrantes y el centro, obteniendo los siguientes valores

Cuadrante	Real x	Real y	Calculada X	Calculada Y
O	0	0	0.10	0.13
I	30	31	30.29	31.50
I	104.5	82	104.32	81.87
II	-21.5	41.5	-21.75	41.09
II	-105.5	21	-105.5	21.22
III	-126.5	-20.9	-126.81	-20.26
III	-42	-62.5	-41.82	-62.49
IV	21	-20.9	21.09	-20.50
IV	83.5	-62.5	83.74	-62.13

IV-B. Robot móvil

Se construyó e instrumentó un Robot Móvil con Ruedas (RMR) tipo carro unicycle, el cual en sus ruedas contiene encoders de 48 cuentas en modo cuádruple, que son leídos por microcontroladores PIC28F331 y a su vez envían los datos a un microprocesador Rabbit. En este microprocesador es donde se programan las tareas deseadas para el robot móvil. El vehículo cuenta también con un dispositivo Xbee de la serie 2, para transmitir y recibir datos provenientes de una computadora (u otro dispositivo). Este vehículo tiene montado un led ultra-brillante, en su parte más alta, el cual es utilizado para que la cámara capte su posición. En la figura 5 se muestra el RMR tipo unicycle utilizado.

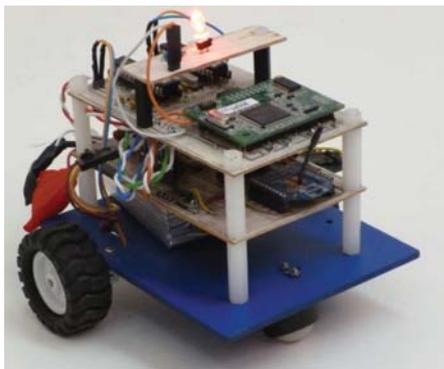


Figura 5: RMR tipo unicycle utilizado en los experimentos.

IV-C. Resultados Experimentales

Se realizaron 2 tipos de pruebas, en la primera el RMR gira en círculos (en lazo abierto, es decir, sin control de posición); y en la segunda se movió manualmente el vehículo para captar distintas trayectorias y velocidades.

En el primer caso se programó el carro para enviar un nivel de voltaje diferente a cada rueda, de modo que el carro gire en círculos. La velocidad de avance del carro fue

relativamente baja. En la figura 6 se muestran las gráficas de posición en cada uno de los ejes con respecto al tiempo, mientras que en la figura 7 se muestra la trayectoria que fue recorriendo el carro (gráfica x, y).

La velocidad fue estimada por dos métodos, primero por medio del diferenciador discreto filtrado (DDF) y el segundo por medio de un diferenciador exacto por modos deslizantes (DEMD). El diferenciador por modos deslizantes muestra una señal menos ruidosa debida en parte por la baja velocidad de desplazamiento del vehículo. Si se desea hacer menos ruidosa la señal del DDF, se debe reducir la constante de tiempo α del filtro (2), pero llega un punto en que la señal se empieza a desfazar en atraso, lo cual no es conveniente. La figura 8 muestra la derivada estimada que resulta ruidosa y un tanto deformada por respecto la gráfica de posición, pero haciendo un acercamiento a la posición medida por la cámara, puede notarse que la señal se ve en algunos lados como “escalonada”, es decir que en ocasiones repite, indeseadamente, una misma posición aunque el objeto se esté moviendo.

En la figura 9 se muestra un acercamiento de la gráfica de posición en x donde se observa el “escalonamiento” de la señal de Posición Medida por la Cámara (PMC), mientras que la Posición Estimada por Modos Deslizantes (PEMD) resulta en una señal “suave”. Si se desea que la PEMD se aproxime más a la PMC, entonces la derivada se verá muy ruidosa y se irá alejando de los valores reales de velocidad.

En el segundo experimento se movió manualmente al vehículo para un desplazamiento en forma diagonal; la prueba empezó con el carro detenido, comenzando a moverlo y a mitad del camino se le detuvo por un pequeño lapso, enseguida se volvió a mover siguiendo la línea diagonal, pero con una velocidad un poco mayor para finalmente detenerlo. Las figuras 10 y 11 muestran lo descrito anteriormente.

Nuevamente, se procedió a calcular la velocidad por DDF y DEMD, pero aumentando las constantes en ambos casos para que puedan seguir o estimar mejor la velocidad mayor que se aplicó al vehículo. El resultado se muestra en la figura 12.

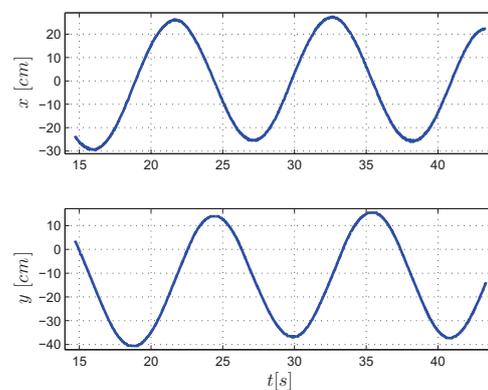


Figura 6: Posición del vehículo medida por la cámara.

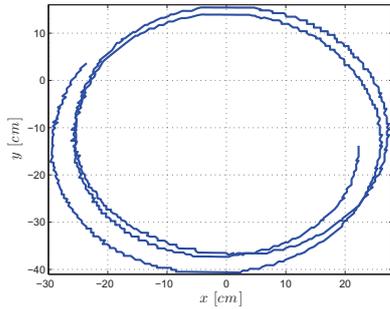


Figura 7: Posición del vehículo medida por la cámara.

En la gráfica de velocidad, en la figura 12, se puede apreciar que el DDF se aproxima más en magnitud al valor que debería tener la velocidad, y el DEMD está casi siempre por debajo del valor del DDF; esto se debe en parte a 2 factores: el primero, como ya se había mencionado, es que la señal medida presenta brincos o “escalones”, en vez de cambiar suavemente; y lo segundo es que para poder seguir adecuadamente las velocidades mayores (o tal vez deberíamos decir aceleraciones mayores) que se presentaron en esta prueba experimental, se debieron aumentar las ganancias del DEMD, es decir, si conserváramos las ganancias pequeñas el derivador sería menos sensible a los escalonamientos, pero no alcanzaría a seguir la velocidad más alta a la que llegó el experimento, y por el contrario, si aumentamos bastante las ganancias del diferenciador, el escalonamiento de la señal sensada provocará una mala estimación de la velocidad real. La desventaja en este caso del DDF, es que cada vez que el vehículo se detuvo, tardó un poco más el DEMD en llevar la velocidad estimada a cero.

A pesar de que los HOSM afirman eliminar el *chattering*, el derivador de primer orden lo presentó en esta prueba cada vez que el vehículo se detuvo. La figura 13 muestra un acercamiento en la señal estimada de velocidad, donde se aprecia el efecto *chattering*, cuya magnitud se debe

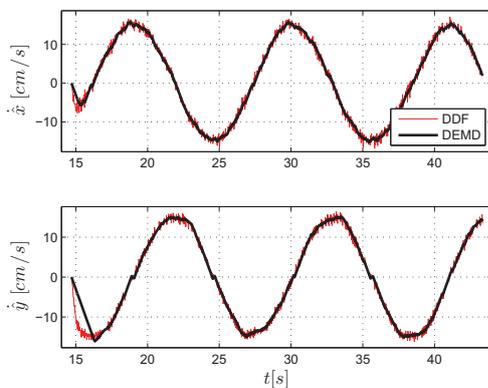


Figura 8: Velocidad estimada DDF y DEMD.

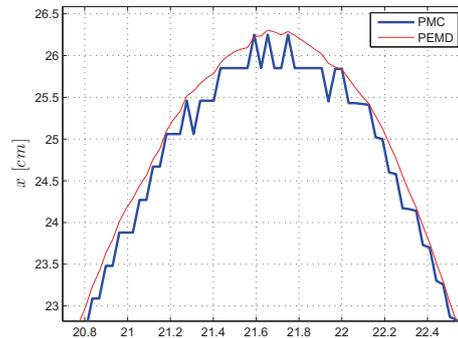


Figura 9: Acercamiento en gráfica de posición PMC y PEMD.

principalmente a que es alta la diferencia entre las ganancias λ_0 y λ_1 .

Una solución para reducir el *chattering*, pero conservar una buena aproximación de la derivada es utilizar un DEMD de segundo orden, del que sólo interesará la primera derivada que corresponde a la velocidad. En la figura 14 se muestra la estimación de velocidad del segundo experimento, con DEMD de segundo orden, comparado también con el DDF. En esta gráfica el DEMD se aproxima más a los valores del DDF que en el caso del diferenciador de primer orden, además que el *chattering* se reduce considerablemente.

La utilización del DEMD de segundo orden parece ser la opción adecuada para estimar la posición del vehículo, la desventaja (si se le puede llamar así) es que consume mayor tiempo de cálculo, aunque para las velocidades de procesamiento de las computadoras actuales no es gran cosa. El DEMD de primer orden es más sencillo en su implementación y es una buena opción para estimar velocidades pequeñas. El DDF muestra pequeños retardos en la estimación de velocidad al detener el vehículo y en caso de estar bien sintonizado el DEMD, la señal del DDF aparenta ser más ruidosa.

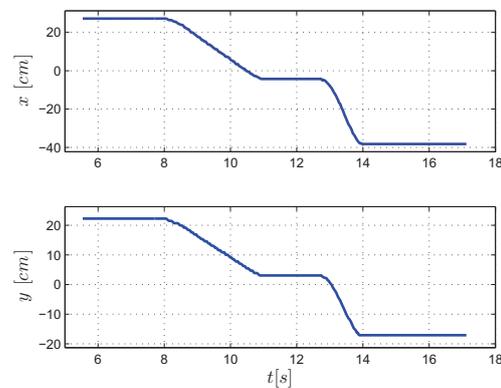


Figura 10: Posición del vehículo con respecto al tiempo.

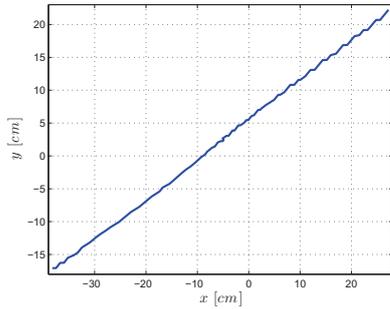


Figura 11: Posición del vehículo en el plano visto por la cámara.

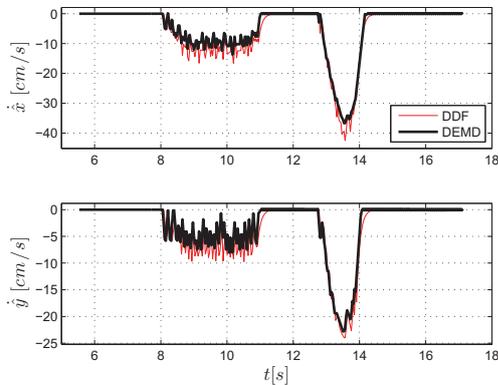


Figura 12: Velocidad estimada DDF y DEMD.

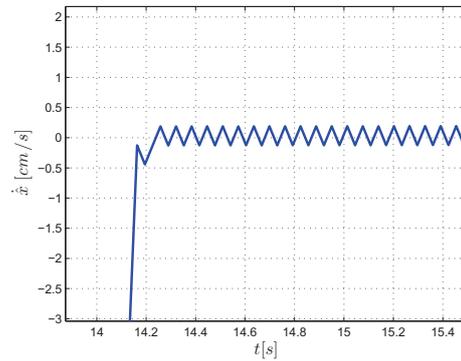


Figura 13: Acercamiento en gráfica de velocidad del DEMD de primer orden.

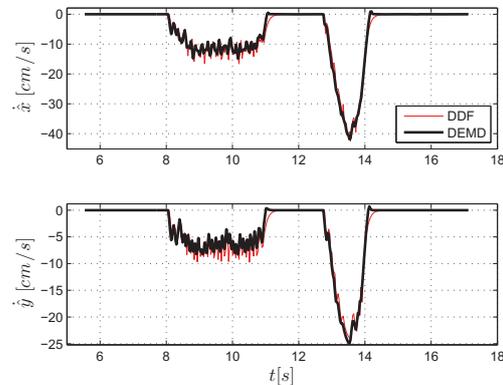


Figura 14: Velocidad estimada DDF y DEMD.

V. CONCLUSIONES

En este artículo se ha mostrado la manera de implementar un sistema de localización bidimensional por medio de visión por computadora. La conversión de píxeles a unidades decimales del área medida se realiza por medio de una red neuronal, para evitar la calibración de la cámara. Parte fundamental de este trabajo corresponde a la implementación de un estimador de velocidad. Una sencilla y buena técnica para estimar la velocidad en este sistema es la utilización de un DDF, sin embargo, por medio del DEMD se pueden obtener mejores resultados, no obstante, por buenos que sean estos estimadores, se ven afectados por el hecho de que el sistema de visión que se ha implementado llega a repetir (indeseadamente) lecturas consecutivas de posición, con lo cual entrega una señal “escalonada” en lugar de hacer cambios suaves, deformando en cierta manera la señal estimada de velocidad.

REFERENCIAS

W. Bath and J. Paxman (2005). UAV localisation & control through computer vision. In Proceedings of Australasian Conference on Robotics and Automation, Sidney, Australia, December 2005.

E. Bompiani, E. Majorana, F. Ricci, G. Testi (2002). Single device telemetric algorithm for absolute position measurement using a CCD camera. Elsevier Science B.V. 2002

A. Levant (2001). Higher order sliding modes and arbitrary-order exact robust differentiation. Proceedings of the European Control Conference, 2001

M. Rolink, T. Boukhobza and D. Sauter (2006). High order sliding mode observer for fault actuator estimation and its application to three tanks Bechman. hal-00121029, version 1-19 Dec 2006

R. Hartley and A. Zisserman (2004). Multiple View Geometry in Computer Vision. Cambridge University Press, 2nd edition, 2004.

A. Levant, L. Alelishvili (2004). Transient adjustment of high-order sliding modes. in Proc. of the 7th Scientific Workshop “Variable Structure Systems VSS’2004”, Vilanova, Spain, September 6-8, 2004.

Jean-Pierre Barbot and T. Floquet (2009). Iterative higher order sliding mode observer for nonlinear systems with unknown inputs. inria-00442129, version 1 - 18 Dec 2009.

A. Benallegue, A. Mokhtari and L. Fridman (2007). High-order sliding-mode observer for a quadrotor UAV. International Journal of Robust and Nonlinear Control. 2007