

# 8º Workshop

## ROBOTS DE EXTERIORES

### ROBOCITY2030-II



**Editores:**

Roemi E. Fernández

Héctor Montes

Centro de Automática y Robótica

CSIC-UPM



UNIÓN EUROPEA  
Fondo Social Europeo  
El FSE invierte en tu futuro



**miod**

Diciembre 2010

## CAPÍTULO 12

### **SIMULACIÓN DE UN SISTEMA DE NAVEGACION PARA UN GRUPO DE ROBOTS DE EXTERIORES UTILIZANDO MICROSOFT ROBOTICS DEVELOPER STUDIO**

A. MATTA, J. del CERRO y A. BARRIENTOS

Centro de Automática y Robótica UPM-CSIC

El sistema de navegación es una de las principales diferencias entre los robots de interiores y los de exteriores. Habitualmente los sistemas que trabajan en exteriores pueden contar con la información proveniente de sensores GNSS que proporcionan una solución de navegación que puede alcanzar precisiones centimétricas en tiempo real, pero que pueden tener una disponibilidad muy reducida en ciertos entornos como los urbanos. Esta baja disponibilidad obliga a disponer de sistemas adicionales que permitan navegar al robot de una forma segura cuando esta información no está disponible, siendo el más habitual la odometría, que puede aportar soluciones robustas con una precisión razonable en entornos principalmente planos.

Trabajar con robots de exteriores supone en muchas ocasiones sufrir muchas restricciones a la hora de realizar validaciones o experimentos por lo que los sistemas de simulación se convierten en una excelente herramienta para sus desarrolladores. En este capítulo, se presenta cómo se ha desarrollado un sistema de navegación para robots de exteriores utilizando la herramienta *Microsoft Robotics Developer Studio* y cómo esta solución puede ser escalada a un conjunto de robots gracias al uso de las librerías de creación y administración de procesos concurrentes y distribuidos, *CCR* y *DSS*, propias de esta herramienta.

## 1 Introducción

En robótica móvil, un sistema de navegación puede ser considerado como la unión de la teoría y la tecnología necesarias para estimar el estado de un robot considerando como tal la posición, actitud y velocidad de éste en todo momento. Actualmente, los robots terrestres utilizan una gran variedad de sensores para estimar una solución para su sistema de navegación de una forma robusta. Esto presenta una gran ventaja, en particular cuando la calidad de la información de los sensores varíe considerablemente debido a perturbaciones o condiciones ambientales. Basar la información de un sistema de navegación en múltiples sensores mejora la calidad de la solución estimada cuando, ante pobres condiciones, alguno de los sensores pueda fallar. Ante esta situación, los sensores con mejor desempeño adquieren una mayor prioridad para producir una estimación más fiable.

Todo lo anterior conduce a la utilización de técnicas de fusión sensorial para combinar la información que un conjunto de sensores pueda brindar. Dichas técnicas de fusión permiten administrar las diferentes pautas de actualización de cada uno de los sensores mientras se utiliza la información de los mismos dependiendo de su calidad asociada. Además, la fusión sensorial también tiene que tener en cuenta cuándo algún sensor no proporcione ninguna información.

Tanto la simulación de los sensores como los algoritmos de fusión sensorial han sido probados en un escenario simulado en donde un conjunto de robots circula por las calles de un pequeño entorno urbano ubicado sobre un terreno plano. Los robots pueden seguir un diverso número de trayectorias a lo largo de zonas de navegación en donde se define las propiedades de las mismas, tales como disponibilidad de la señal GPS y/o calidad de esta señal.

Para la implementación en software, tanto de los sensores como del entorno de simulación, se utilizó la herramienta *Microsoft Robotics Developer Studio* por diversas características presentes en esta herramienta que resultan atractivas y ventajosas para este tipo de simulaciones.

Entre estas ventajas, caben destacar la de la integración entre el motor de gráficos tridimensionales *XNA* y el motor de dinámica de sólidos *AGEIA Physics Engine* con lo cual se pueden obtener escenarios fotorealistas desde el punto de vista gráfico y de gran precisión en el compartimiento físico de los elementos presentes en el mismo.

Igualmente, como parte de esta herramienta, se encuentra presente la librería de concurrencia y coordinación y la de servicios de software descentralizado, *CCR* y *DSS* respectivamente por sus siglas en inglés, las cuales ofrecen un enfoque novedoso en el momento de crear y administrar proce-

tos concurrentes y distribuidos, en comparación a la programación tradicional de este tipo de procesos, es decir aquella que hace uso de hilos, multiplexores y semáforos (Chen and Bai, 2008), (Jackson, 2007).

En la Sección 2 se hace una descripción de cómo está conformado el sistema de navegación de cada uno de los robots, en la Sección 3 se describe cómo fue utilizada la herramienta *Microsoft Robotics Developer Studio* para la implementación de los sensores y del algoritmo de fusión sensorial, en la Sección 4 se muestran las pruebas y resultados obtenidos a partir de estas pruebas y por último se presentan las conclusiones.

## **2 Descripción del sistema de navegación virtual**

El principal sensor del sistema de navegación está compuesto por una unidad de navegación basada en GPS diferencial con correcciones RTK y un sistema inercial de altas prestaciones. Adicionalmente se dispone de la información proporcionada por el sistema de odometría del robot. Un algoritmo de fusión sensorial se encarga de combinar esta información. A continuación se pasa a describir cada uno de estos ítems.

### **2.1 El sistema sensorial**

El sistema GNSS (*Global Navigation System*) más conocido es el sistema Estadounidense GPS, el cual permite obtener una estimación muy aproximada de la posición del robot a partir de la información brindada por un grupo de satélites pertenecientes a la constelación NAVSTAR. La principal característica de este sistema radica en que brinda una solución absoluta para la posición del robot, y sus errores asociados no aumentan con el paso del tiempo. Sin embargo, en casos en que la señal GPS no se encuentre accesible por algún robot, como suele suceder en áreas urbanas, el sistema GPS no provee solución alguna.

Por otro lado, el sistema de navegación inercial, INS, es un sistema de navegación auto contenido en el robot. Es muy robusto ante efectos ambientales y otras perturbaciones externas. A su vez, tiene algunas limitaciones: la precisión de su salida depende principalmente de la calidad de los sensores (acelerómetros y giróscopos) y el error obtenido de su solución de navegación puede aumentar sin límite con respecto al tiempo por efecto de los errores intrínsecos del sistema y de las derivas.

Estas propiedades complementarias entre el GPS y el INS favorecen su integración. El sistema integrado INS-GPS hace un buen uso de las pro-

iedades de la fusión sensorial en cuanto a estimación y compensación del error se refiere, haciendo de ello un sistema de navegación muy robusto.

Unido a todo esto, está la integración de este sistema INS-GPS con la información proveniente del odómetro presente en el robot. Un odómetro ofrece al sistema de navegación la información concerniente a la velocidad del robot a partir de la velocidad de sus ruedas de una forma segura y estable con lo cual se obtiene una solución aún más robusta y eficiente.

Este sistema integrado INS-GPS-Odómetro puede garantizar una solución precisa, en cuanto a la posición y velocidad del robot se refiere, bajo un gran rango de condiciones que pueden estar presentes en el entorno (Siegwart and Nourbakhsh, 2004). Los parámetros que caracterizan el sistema sensorial han sido obtenidos de sistemas reales comerciales.

## **2.2 Fusión sensorial**

El sistema de navegación del robot está basado en la información obtenida de un sistema INS-GPS integrado y la odometría del robot. Esta información es combinada utilizando un algoritmo de fusión sensorial.

El algoritmo de fusión utiliza los incrementos de posición que provienen del odómetro y los combina con los provenientes del sensor INS-GPS para determinar la estimación final de la posición del robot en cada intervalo de tiempo. Además, la diferencia entre la información inercial y la del odómetro permite estimar la tracción que las ruedas del robot experimentan.

El algoritmo de fusión sensorial utilizado consiste en un filtro extendido de Kalman en cascada tal como se puede observar en la Fig. 1. En este método, la calidad de la señal del GPS es utilizada para estimar el nivel confianza en la estimación.

El sensor virtual INS-GPS contiene un filtro extendido de Kalman con el fin de brindar una solución completa (posición, velocidad y orientación) y una medida de la estimación de la calidad de ésta. Toda esta información se utiliza en el algoritmo de navegación tridimensional para producir una estimación de la posición mejorada.

El paso final consiste en utilizar otro filtro extendido de Kalman para producir un sistema completo de navegación que incluya tanto la posición como la velocidad del robot tomando como base las estimaciones intermedias anteriormente descritas junto con la estimación del GPS y otras estimaciones de calidad para modificar la matriz de ganancia del filtro extendido de Kalman final.

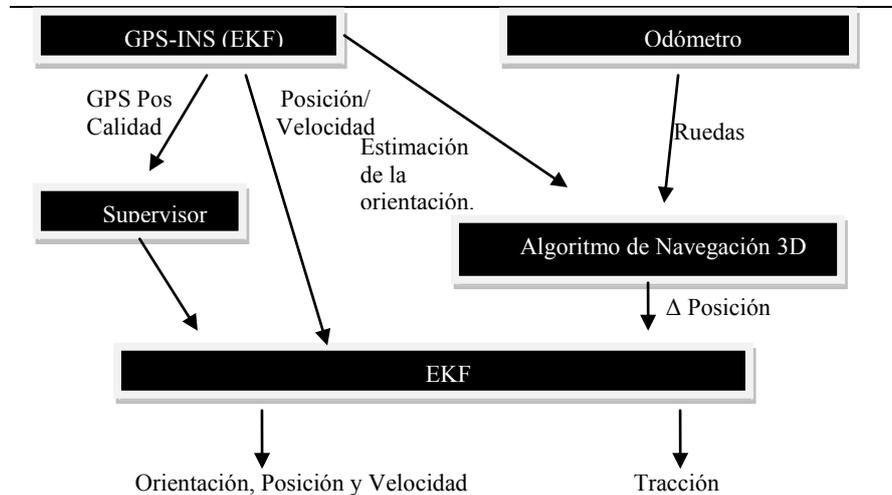


Fig. 1. Estructura de la fusión sensorial

### 3 Implementación del sistema de navegación virtual

El sistema de navegación virtual está basado en la información obtenida a partir del sensor integrado INS-GPS y del odómetro del robot. A nivel de software han sido diseñados e implementados utilizando los criterios de desarrollo presentes en la herramienta de software *Microsoft Robotics Developer Studio* con el fin de que se ajustaran tanto a los requisitos del escenario virtual en donde los robots desempeñan una serie de tareas como a los requisitos propios de esta herramienta con el fin de que el sistema de navegación escale en número al número de robots presentes en el escenario de una forma concurrente y distribuida.

#### 3.1 Implementación de los sensores

En la herramienta *Microsoft Robotics Developer Studio*, los sensores son simulados encapsulando un conjunto de funciones en unas entidades propias de esta herramienta llamadas servicios. Las relaciones existentes entre los servicios que representan los sensores y otros servicios presentes en un robot son coordinadas por otros servicios de tipo administrativo llamados servicios orquesta.

A su vez, la mayoría de los sensores virtuales también necesitan de una representación tridimensional que encapsulen tanto la geometría como la dinámica de dicha representación en un escenario virtual. Por lo tanto, los sensores virtuales fueron diseñados e implementados utilizando dos componentes de software: una entidad para su representación física y un servicio para su funcionalidad.

### **3.1.1 El sistema INS-GPS integrado**

Un sistema INS-GPS esta compuesto de varias partes:

- Tres giróscopos y tres acelerómetros ubicados ortogonalmente.
- Receptor GPS (usualmente entre ocho y doce canales).
- Magnómetro.
- Sensores de temperatura para compensaciones de bias.
- Sistema de procesamiento de datos.

La salida para este tipo de sistema consiste de:

- Estimación de la orientación: La cual se obtiene fusionando la información procedente de los giróscopos, acelerómetros y magnómetros.
- Estimación de la posición y velocidad: Información que se obtiene después de analizar los resultados provenientes de los acelerómetros y la estimación de tanto la orientación como la posición absoluta del GPS

La Fig. 2 despliega el diagrama de bloques principal para el sistema INS-GPS implementado.

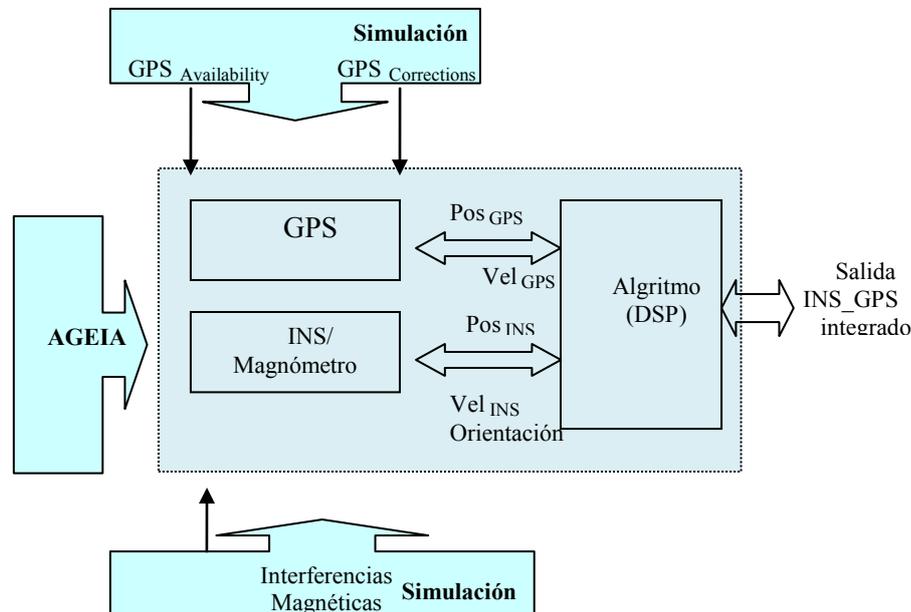


Fig. 2. Arquitectura débil del sensor INS-GPS integrado

### 3.1.1.1 El servicio GPS

El servicio INS-GPS contiene un conjunto de funciones que simulan las funcionalidades propias este tipo de sensores. En la Fig. 3 se puede apreciar como fue diseñado este servicio del cual cabe resaltar lo siguiente:

- La información de entrada es la posición, orientación y velocidad del robot, la cual se obtiene del motor físico *AGEIA Physics Engine*.
- Las perturbaciones se simulan aplicando un ruido gaussiano a la información de entrada. El parámetro  $\sigma$  de la función  $N(0, \sigma)$  se determina en función de estar utilizando un GPS de tipo diferencial o de precisión simple.
- La información de salida es un muestreo en frecuencia que proporciona la posición, orientación y velocidad absoluta del

sensor cuyos errores inherentes son corregidos con un receptor GPS cuando la señal está disponible.

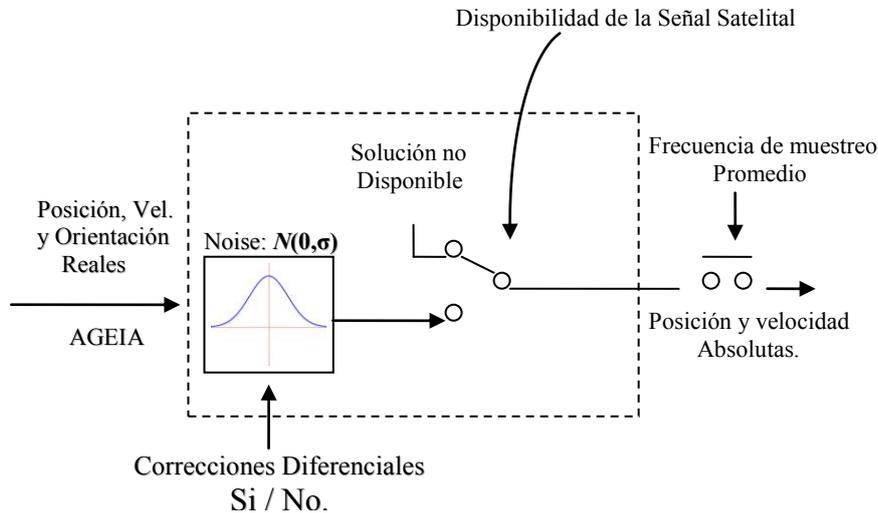


Fig. 3. Diseño del Sensor GPS

### 3.1.2 INS + Compás

Un sensor INS utiliza la aceleración y las velocidades angulares como señales de entrada.

En la Fig. 4 se puede apreciar la forma como fue diseñada la funcionalidad para este sensor. De este diseño cabe resaltar:

- La información de entrada es la orientación y velocidad actual del robot tal cual como se recibe del motor físico AGEIA.
- Bias y muestreo es aplicado a los parámetros de entrada. El parámetro  $\sigma$  de la función  $N(0, \sigma)$  se utiliza para aplicar correcciones de tipo diferencial o de otro tipo presentes en el escenario.
- La información de salida es la orientación y velocidad estimada.
- En sensores INS reales, el efecto bias del Gyro puede ser reducido aplicando medidas provenientes del acelerómetro. Dado que no se provee la estimación de la aceleración, esta se simulará mediante correcciones virtuales.

- La presencia de campos magnéticos en el escenario virtual es considerada por este sensor. Las desviaciones que puedan generar son corregidas con el uso de un compás.

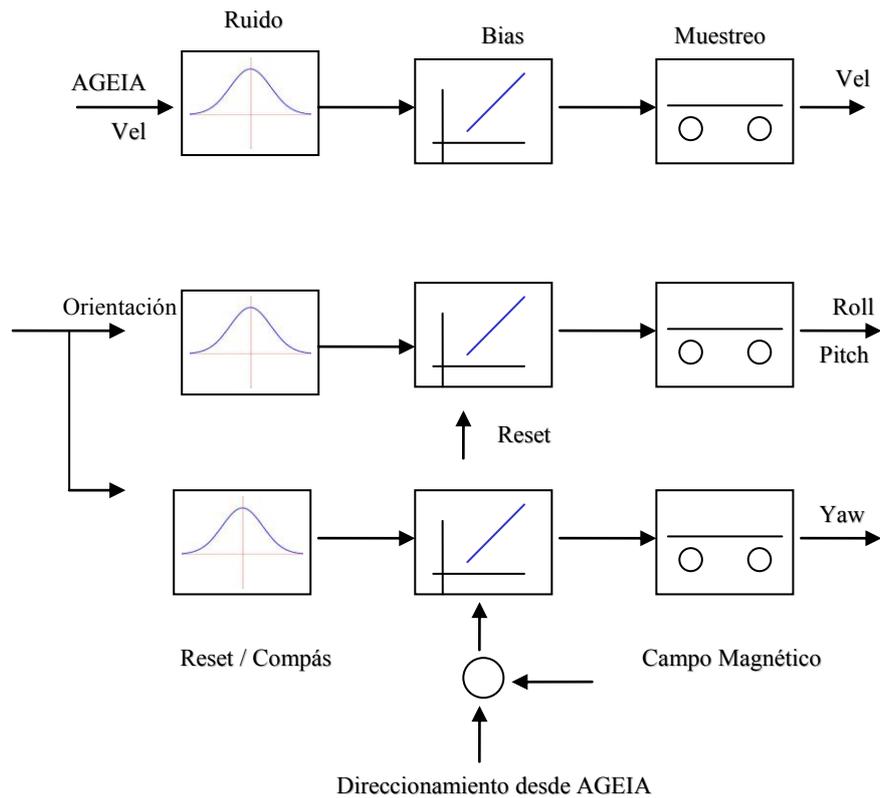


Fig. 4. Modelo del servicio INS + Compás

### 3.1.3 El odómetro

No existe una representación física del odómetro porque este está montado dentro del robot. Se tiene en cuenta que existe un odómetro por cada rueda del robot y la funcionalidad de este sensor se obtiene mediante una serie de funciones que vienen adscritas a cada una de dichas ruedas.

### 3.2 Implementación del filtro extendido de Kalman.

El filtro de Kalman es probablemente el algoritmo más utilizado para fusión sensorial porque es una herramienta muy útil para estimar las variables para un rango muy amplio de procesos y sistemas. Permite estimar los estados de un sistema por lo que es fácil predecir como el sistema evoluciona a través del tiempo (Welch and Bishop, 2001).

Para definir el estado del robot, se ha utilizado una arquitectura débilmente acoplada con la cual se puede estimar la posición del robot, y se expresa de esta manera:

$$x_{(k)} = [px_{(k)}, py_{(k)}, pz_{(k)}]^T \quad (1)$$

y su correspondiente formulación del estado:

$$x_{(k+1)} = x_{(k)} + (R_{(k)}v_{(k)}^b) \cdot T_s \quad (2)$$

Además, se considera los valores de la odometría como entradas. Para pruebas iniciales un simple modelo de la odometría ha sido utilizado, quedando de la siguiente manera:

$$(R_{(k)}v_{(k)}^b) = [\Delta x, \Delta y, \Delta z]^T \quad (3)$$

En donde los incrementos han sido considerados utilizando valores reales provenientes del motor físico AGEIA y corrompidos con ruido blanco.

Y se utiliza como medidas de entrada al filtro extendido de Kalman a la información procedente del sensor INS-GPS.

$$z_{(k)} = [px_{GPS}, py_{GPS}, pz_{GPS}]^T \quad (4)$$

Por lo tanto, la matriz R queda de la siguiente forma:

$$R = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \quad (5)$$

Al igual que el sensor INS-GPS integrado, este algoritmo fue implementado utilizando un servicio del *Microsoft Robotics Developer Studio*. Cada servicio de este tipo queda asociado a cada uno de los robots que puedan

estar presentes en el escenario de simulación, los cuales se ejecutan concurrentemente gracias a las propiedades presentes en la librería de Concurrency y Coordinación (CCR) de esta herramienta de desarrollo (Tsai *et al.*, 2008).

Igualmente, todos los servicios implementados pueden ser ejecutados a lo largo de una red de computadores permitiendo de este modo no sobrecargar un ordenador con todos los servicios que pueda tener una aplicación. De esta forma, en este sistema de navegación virtual, el usuario podría coordinar que el servicio que encapsula las funcionalidades del sensor INS-GPS se ejecute en un ordenador y el servicio encargado de la fusión sensorial se ejecute en otro. Se ha visto que esto es de gran utilidad en el momento de utilizar robots reales porque esta funcionalidad permite que los servicios más costosos, computacionalmente hablando, se ejecuten en los ordenadores más potentes y únicamente los más livianos se ejecuten en el ordenador del robot, que por lo general siempre es el de menor capacidad. Todo esto se puede realizar gracias a la librería DSS que es como se conoce a la librería de servicios de software descentralizado (Chen and Bai, 2008).

## 4 Pruebas y resultados

Para la realización de las pruebas se creó un escenario tridimensional en donde circula el conjunto de robots y posteriormente será utilizado para que dicho conjunto de robots realice labores de vigilancia. Primeramente se hará pequeña descripción de este escenario y posteriormente se mostrarán los resultados obtenidos al dejar transitar un robot por diversas zonas con un tipo de señal GPS distinto.

### 4.1 El escenario de simulación virtual

Para probar el sistema de navegación se hizo uso de un escenario en exteriores que viene como parte de la herramienta *Microsoft Robotics Developer Studio* y de la plataforma robótica *CoroBot* de la compañía *Coroware*, que también viene incluida. La iluminación de esta escena es la propia de un día luminoso, el terreno es de tipo escarpado y su área total es de 512mx512m. En la Fig. 5 se puede ver una toma área de este escenario.

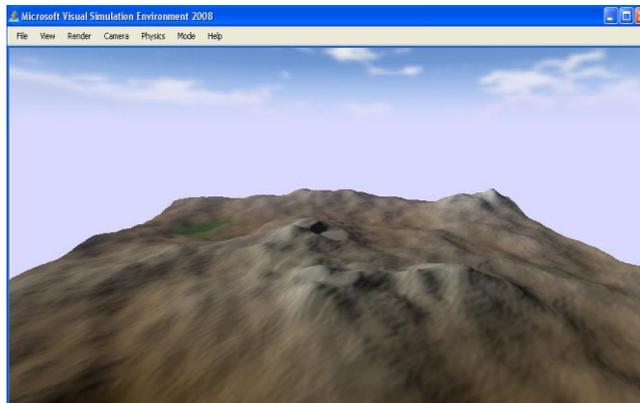


Fig. 5. Vista área del escenario de simulación virtual

Toda la información que describe cómo está constituido un escenario se organiza utilizando archivos XML, en el cual también se puede describir la calidad de la señal GPS presente en diversas zonas del escenario.

#### **4.2 Resultados obtenidos a partir de la simulación**

Las pruebas de navegación se realizaron ubicando al grupo de robots sobre el terreno del escenario, ver Fig. 6, sucesivamente los robots se mueven a lo largo de una línea recta por un área en donde se encuentran diversas zonas de calidad GPS diferente.

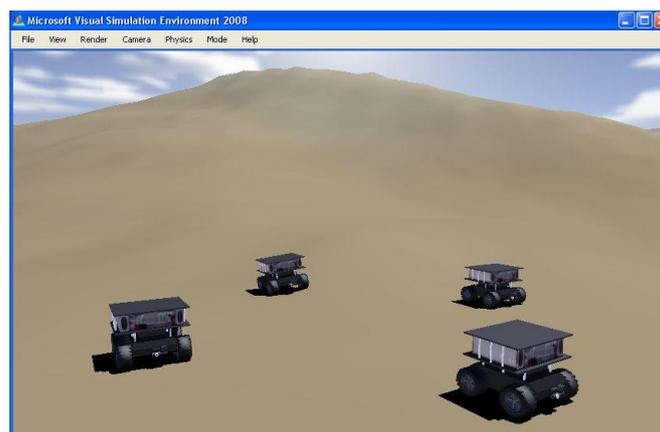


Fig. 6. Conjunto de robots navegando a través de varias zonas GPS

Como resultado, el sensor INS-GPS cambia su señal de salida dependiendo del tipo de zona sobre la cual el robot se esté moviendo y la fusión sensorial estima la posición y velocidad del robot asociado.

Estos resultados se muestran a través de una serie de ficheros tipo Plot, tal como se muestra en la Fig. 7 y en la Fig. 8.

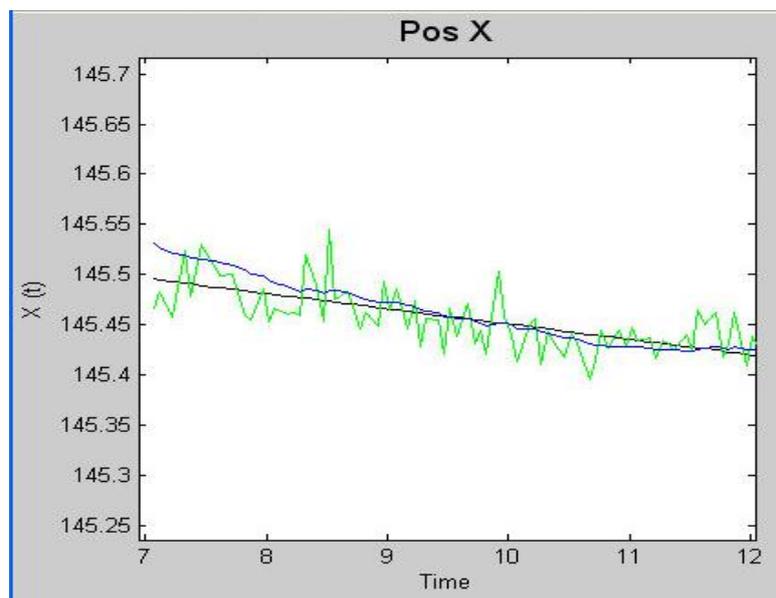


Fig. 7. Posición del robot en la coordenada X cuando la señal GPS está disponible

En esta figura, la posición real del robot está denotada por una línea negra, mientras que la salida del algoritmo de fusión sensorial se aprecia en la línea azul y la línea verde representa la salida del sistema INS-GPS integrado. Por lo tanto, la estimación del desempeño es más precisa de lo esperado (un error menor a 15 cm RMS).

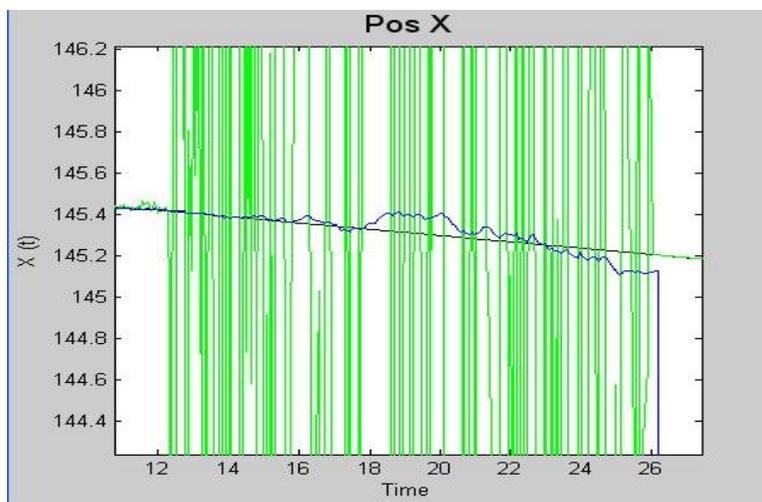


Fig. 8. Estimación de la posición del robot en la coordenada X ante un sensor GPS en modo simple

Al igual que en la figura anterior, la posición real del robot se aprecia en la línea negra y la salida de la fusión sensorial en la línea azul. La línea verde denota la salida del sistema INS-GPS. Por lo tanto, se puede apreciar la gran precisión de la fusión sensorial que muestra un resultado interesante, incluso ante la presencia de un ruido tan alto, como es el que el sistema INS-GPS single maneja.

Cabe resaltar que, aunque solo hemos mostrado los resultados obtenidos para un solo robot, los mismos replican para un conjunto de robots dado el hecho que se crea tanto un servicio INS-GPS como un servicio para la fusión sensorial para cada uno de los robots presentes y que estos servicios se ejecutan de forma concurrente gracias a las propiedades presentes en la librería de concurrencia y coordinación, *CCR*, propia del *Microsoft Robotics Studio*.

## 5 Conclusiones

Este trabajo presenta la implementación de un sistema de navegación tanto desde el punto de vista sensorial como el de la fusión sensorial mediante la utilización de la herramienta *Microsoft Robotics Developer Studio*.

Bajo este esquema, se han podido analizar las ventajas que trae esta herramienta de desarrollo para la realización de simulaciones tridimensionales utilizando uno o varios robots móviles porque fácilmente se pudo diseñar e implementar un sistema de navegación para un robot móvil que fue replicado sin problema alguno para un conjunto de robots tanto en forma concurrente si se utiliza la librería *CCR* como en forma distribuida a lo largo de varias máquinas utilizando la librería *DSS*.

### **Agradecimientos**

Los autores desean expresar su gratitud a la Comunidad de Madrid por su financiación a través del proyecto: ROBOCITY2030 (S-0505/DPI/000235).

### **Referencias**

Borenstein, J., Everett, H.R., Feng, L. 1996. *Where am I? Systems and Methods for Mobile Robot Positioning*, University of Michigan.

Chen, Y., Bai, X. 2008. *On Robotics Applications in Service-Oriented Architecture*. In Proc. 28<sup>th</sup> International Conference on Distributed Computing Systems Workshops. Beijing, China, pp. 551-556.

Jackson, J. 2007. *Microsoft Robotics Studio: A Technical Introduction*. IEEE Robotics & Automation Magazine, pp. 82-87.

Siegwart, R., Nourbakhsh, I. 2004. *Introduction to Autonomous Mobile Robots*. The MIT Press, Cambridge, Massachusetts.

Tsai, W.T., Huang, Q., Sun, X. 2008. *A Collaborative Service-Oriented Simulation Framework with Microsoft Robotics Studio*. In Proc. 41<sup>st</sup> Annual Simulation Symposium., Ottawa, Canada, pp. 263-270.

Welch, G. and Bishop, G. 2001. *An Introduction to the Kalman Filter*. In Proc. SIGGRAPH 2001. Course 8.

