




# Simulación del control de un robot móvil tipo rover basado en fusión de sensores mediante filtro de partículas

## Control of a rover-type mobile robot using sensor fusion through particle filtering

J. D. Diaz-Ortega <sup>a,\*</sup>, O. Gutiérrez-Frías <sup>a</sup>, J. A. Aguirre-Anaya <sup>b</sup>

<sup>a</sup> Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, Instituto Politécnico Nacional, Ciudad de México.

<sup>b</sup> Unidad Profesional Interdisciplinaria de energía y movilidad, Instituto Politécnico Nacional, Ciudad de México.

### Resumen

Los algoritmos de navegación de robots móviles usan sistemas de control de posición para ejecutar tareas como la exploración, seguimiento de trayectorias, búsqueda y rescate, entre otras. Sin embargo, la implementación de estos algoritmos involucra sensores como los GPS, cuya precisión depende de factores externos como los climáticos, así como su funcionamiento puede verse afectado por la disposición de satélites, generando errores de posición. Como solución a lo anterior, en este trabajo se propone un sistema de control basado en fusión de sensores mediante el filtro de partículas que permita mejorar la navegación de un robot móvil tipo rover, consiguiendo una mayor precisión en la navegación, reduciendo la vulnerabilidad del sistema, incrementando la confiabilidad y la tolerancia a fallos y un aumento en la confianza en las mediciones. Finalmente, se muestran los resultados de la fusión de 5 sensores, el modelo cinemático del robot y el control PI de posición para un recorrido desde una posición inicial a una final.

**Palabras Clave:** Fusión de sensores, Filtro de partículas, Control PI, Robot rover.

### Abstract

Robot mobile navigation algorithms use position control systems to perform tasks such as exploration, trajectory tracking, search, and rescue, among others. However, the implementation of these algorithms involves sensors such as GPS, whose accuracy depends on external factors such as weather, as well as its operation can be affected by the position of satellites, discovering position errors. As a solution to the above, this work proposes a control system based on sensor fusion through the particle filter that allows improving the navigation of a rover-type mobile robot, achieving greater precision in navigation, reducing the vulnerability of the system, increasing reliability and fault tolerance, and increased confidence in measurements. Finally, the results of the fusion of 5 sensors, the kinematic model of the robot and the PI position control for a path from an initial to a final position are shown.

**Keywords:** Sensor fusion, Particle filter, PI control, rover robot.

## 1. Introducción

En años recientes se ha incrementado el interés por los robots tipo rover, pues han sido fundamentales en importantes misiones como las presentadas en (Von Ehrenfried, 2022), (Ding et al., 2022), donde su propósito general es la exploración de la superficie marciana. Los rover son vehículos robóticos terrestres que tienen la capacidad de interactuar con terrenos hostiles; su principal objetivo es navegar sobre dichos terrenos, reconocer y sortear obstáculos, así como efectuar una tarea específica (Ellery, 2004). Por otra parte, es común encontrar en la literatura el uso del EKF (Extended Kalman Filter) para fusión de sensores en sistemas no lineales, sin

embargo, la linealización de primer orden puede introducir errores de estimación. Como alternativa al EKF, se propone el filtro de partículas, el cual es un método secuencial basado en la técnica de Monte Carlo, que consiste en una representación de la distribución de estados usando el muestreo secuencial y el remuestreo (Yang et al., 2005).

Desde hace varios años, diversos trabajos han aprovechado las ventajas del filtro de partículas y reportado su uso para la fusión de sensores con múltiples aplicaciones, donde las tareas de seguimiento son las más comunes; algunos ejemplos de lo anterior son: el seguimiento de objetos en movimiento aleatorio (Vadakkepat & Jing, 2006), seguimiento de peatones (Tian et al., 2020), el seguimiento en tiempo real de voz para

\*Autor para la correspondencia: [jheisondz@gmail.com](mailto:jheisondz@gmail.com) - [jdiaz01800@alumno.ipn.mx](mailto:jdiaz01800@alumno.ipn.mx)

**Correo electrónico:** [jheisondz@gmail.com](mailto:jheisondz@gmail.com) (Jheison Duvier Diaz-Ortega), [ogutierrezf@ipn.mx](mailto:ogutierrezf@ipn.mx) (Octavio Gutiérrez-Frías), [jaguirrea@ipn.mx](mailto:jaguirrea@ipn.mx) (J. Alejandro Aguirre-Anaya)

localizar el emisor (Chen & Rui, 2004), seguimiento de objetivos en una red colaborativa de sensores acústicos (Zhai et al., 2008), entre otros. No obstante, existen muchos otros estudios que presentan otras aplicaciones del filtro de partículas como serían: la estimación tridimensional híbrida de la postura de cámaras (Ababsa & Mallem, 2012), el ensamblaje de robots (Thomas et al., 2007), o la monitorización robusta del ritmo cardiaco mediante sensores portátiles (Nathan & Jafari, 2018).

Así mismo, existen trabajos reportados que examinan, proponen y modifican distintas estrategias de fusión de sensores mediante el filtro de partículas; para la navegación (Yang et al., 2005), control de movimiento (Rigatos, 2010), posicionamiento (Bassiri et al., 2018) y localización (Zaidner & Shapiro, 2016), (Agarwal et al., 2019) de robots móviles.

Por lo tanto, es imprescindible considerar distintas metodologías de fusión, y estudiar diversas alternativas de modelado y estrategias de control, que posibiliten el aumento de las aplicaciones para los robots tipo rover; entre otras cosas, porque la navegación de este tipo de robots requiere de sistemas de control de posición precisos, consistentes y confiables. Para tales propósitos, se propone en este trabajo, un algoritmo de control basado en la fusión de sensores; donde la localización del rover, es estimada mediante un filtro de partículas, con el fin de eliminar algunos posibles errores de los sensores, además de que las características de navegación mejoran respecto a las de un sistema basado en un solo sensor, permitiendo incrementar la confiabilidad, robustez y exactitud del sistema.

La organización del artículo es la siguiente: En la sección 2, se describe la configuración del robot móvil tipo rover y su modelo matemático. En la sección 3, se presenta la estrategia de control basada en fusión de sensores, y se expone el diagrama general de la propuesta. Los detalles y resultados de las pruebas realizadas mediante simulación numérica se muestran en la sección 4; incluyendo la descripción de las funciones de transición de estados y de probabilidad de medición requeridas por los bloques de Matlab-Simulink. Finalmente, en la sección 5 se presentan las conclusiones.

## 2. Robot móvil tipo rover



Figura 1: Robot rover

El robot rover empleado para el desarrollo de la estrategia se muestra en la Figura 1 y fue implementado en la Sección de Estudios de Posgrado e Investigación, de la Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas del

Instituto Politécnico Nacional y algunas características se muestran en la Tabla 1.

Características	Descripción
Ancho	50.5 cm
Largo	75 cm
Alto	60 cm
Peso	9.5 kg
Número de ruedas	6
Tipo de suspensión	Pasiva
Sistema de tracción	Cada rueda está actuada

El prototipo dispone de una configuración geométrica de manera que la carga se distribuye equitativamente; contando con dos ruedas situadas en cada lado unidas por eslabones en forma de paralelogramo, así como una rueda sin suspensión en la parte trasera, y una rueda en la parte frontal unida al chasis mediante un mecanismo de cuatro barras.

Además, su sistema de dirección se compone por las ruedas frontal y posterior actuadas independientemente, otorgando la capacidad de realizar giros de radio pequeño sobre su propio eje sin el deslizamiento de las ruedas; consecuentemente, el rover es un sistema capaz de transitar terrenos agrestes, y superar obstáculos de hasta el doble del diámetro de sus ruedas.

### 2.1. Modelo matemático

El modelo matemático del robot rover se obtiene mediante el análisis global de la cinemática usando como referencia el análisis de un robot diferencial; obteniendo un modelo cinemático aproximado, que considera un movimiento planar sin deslizamiento de las ruedas a partir de la vista superior del rover y de la mitad posterior como se muestra en la Figura 2.

Lo anterior, permite modelar y controlar solo la mitad posterior y debido a la simetría en la construcción del robot, se puede utilizar este resultado para controlar la mitad frontal.

El modelado se realiza considerando la posición y velocidad del punto de control situado en el centro de gravedad del rover, señalado con el punto de coordenadas  $(x, y)$  y se expresa de la siguiente manera: (Baturone, 2005).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{R \cos \varphi}{2} + \frac{LR \sin \varphi}{b} & \frac{R \cos \varphi}{2} - \frac{LR \sin \varphi}{b} \\ \frac{R \sin \varphi}{2} - \frac{LR \cos \varphi}{b} & \frac{R \sin \varphi}{2} + \frac{LR \cos \varphi}{b} \\ -\frac{R}{b} & \frac{R}{b} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \quad (1)$$

donde:

- $x$ : Posición del rover en el eje  $x$
- $y$ : Posición del rover en el eje  $y$
- $\varphi$ : Ángulo de orientación del rover respecto al eje  $x$
- $\dot{x}$ : Velocidad traslacional en el eje  $x$
- $\dot{y}$ : Velocidad traslacional en el eje  $y$
- $\dot{\varphi}$ : Velocidad de rotación u orientación del rover
- $L$ : Distancia perpendicular desde la mitad del eje al centro de gravedad
- $L_1$ : Distancia perpendicular desde la mitad del eje a la rueda de dirección
- $b$ : Vía del rover o distancia entre las ruedas laterales
- $R$ : Radio de las ruedas
- $\omega_1$ : Velocidad angular de la rueda izquierda
- $\omega_2$ : Velocidad angular de la rueda derecha

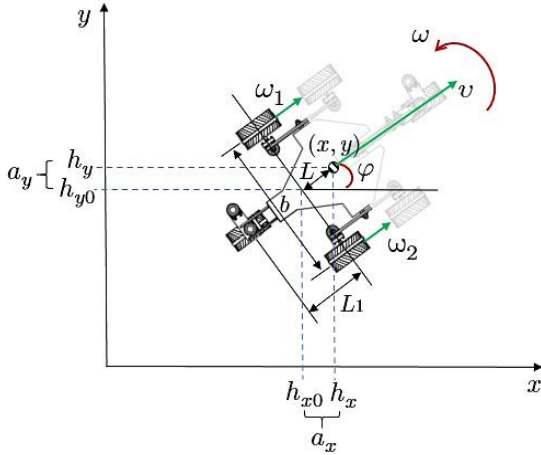


Figura 2: Esquema para el modelado del rover

### 3. Estrategia de control

Dependiendo de la aplicación, los robots rover tendrán requerimientos de navegación diferentes, que pueden variar de acuerdo con las características del terreno, el espacio de trabajo (interior o exterior), el tipo y tamaño de los obstáculos, entre otros; por tales razones, surge la necesidad de plantear diferentes estrategias de control. Así entonces, en el presente trabajo se propone una ley de control de posición basada en el modelo cinemático del robot.

El controlador propuesto es de tipo PI (Proporcional Integral) el cual puede ser expresado respecto al tiempo como  $u = k_p e(t) + k_i \int_0^t e(\tau) d\tau + \dot{q}^*$ , donde  $u$  es la acción de control,  $e(t)$  es el error de posición del robot, es decir es la diferencia entre la localización deseada y la localización del rover en cada eje coordenado.  $\dot{q}^*$  es la velocidad traslacional deseada. Consecuentemente, la ley de control basada en el modelo matemático es la presentada en (2).

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos \varphi & -L \sin \varphi \\ \sin \varphi & L \cos \varphi \end{bmatrix}^{-1} \begin{bmatrix} k_{px} e_x(t) + k_{ix} \int_0^t e_x(\tau) d\tau + \dot{x}^* \\ k_{py} e_y(t) + k_{iy} \int_0^t e_y(\tau) d\tau + \dot{y}^* \end{bmatrix} \quad (2)$$

donde  $k_{px}, k_{ix}, k_{py}$ , y  $k_{iy}$  son las constantes proporcional e integral con las cuales se sintoniza el controlador en cada eje coordenado,  $e_x$  y  $e_y$  son los errores de posición en cada coordenada,  $\dot{x}^*$  y  $\dot{y}^*$  son las componentes de la velocidad traslacional deseada.

Por otro lado,  $u_1$  y  $u_2$  se relacionan con la velocidad traslacional  $v$  y rotacional  $\omega$  respectivamente, que debe tener el robot para alcanzar la posición deseada (ver Figura 2). Sin embargo, para controlar al robot se requieren las velocidades angulares de las ruedas  $\omega_1$  y  $\omega_2$  las cuales, son las entradas de control del rover y se determinan a partir de las salidas de los controladores mediante la siguiente expresión.

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ -\frac{R}{b} & \frac{R}{b} \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3)$$

Reemplazando (2) en (3) y sustituyendo ese resultado en la ecuación (1), se obtienen las expresiones (4), (5) y (6), a partir de las cuales se puede analizar el comportamiento del controlador.

$$\dot{x} = \sigma_1 \quad (4)$$

$$\dot{y} = \sigma_2 \quad (5)$$

$$\dot{\varphi} = \frac{\sigma_2 \cos(\varphi) - \sigma_1 \sin(\varphi)}{L} \quad (6)$$

donde  $\sigma_1 = k_{px} e_x(t) + k_{ix} \int_0^t e_x(\tau) d\tau + \dot{x}^*$  y  $\sigma_2 = k_{py} e_y(t) + k_{iy} \int_0^t e_y(\tau) d\tau + \dot{y}^*$ . Las ecuaciones (4), (5) y (6) representan el comportamiento del sistema en lazo cerrado utilizando el controlador propuesto. La variable de estado  $\dot{\varphi}$  (velocidad rotacional del robot) no necesita ser controlada, debido a que el robot puede alcanzar la localización deseada con una orientación arbitraria; de manera que el análisis de estabilidad se reduce a las variables de posición.

Al despejar las ecuaciones de la velocidad traslacional de (4) y (5) y presentarlas en el dominio de Laplace, se obtienen las dinámicas del error mostradas en (7) y (8).

$$e_x(S)S^2 + k_{px} e_x(S)S + k_{ix} e_x(S) = 0 \quad (7)$$

$$e_y(S)S^2 + k_{py} e_y(S)S + k_{iy} e_y(S) = 0 \quad (8)$$

Para garantizar la estabilidad del sistema las constantes  $k_{px}, k_{ix}, k_{py}$ , y  $k_{iy}$  deben ser ajustadas de tal modo que los polinomios (7) y (8) sean Hurwitz, de este modo  $e \rightarrow 0$ , cuando  $t \rightarrow \infty$ .

#### 3.1. Fusión de sensores

La fusión de datos de sensores es el proceso de combinar piezas de información incompletas e imperfectas procedentes de sensores mutuamente complementarios, de manera que se logre una mejor comprensión de un fenómeno subyacente del mundo real (Koch, 2016). Paralelamente, un sistema que incorpora fusión de sensores supera en precisión, robustez, confiabilidad, entre otros aspectos, a un sistema que utilice para sus mediciones un solo sensor (Ávila & Arancibia, 2015); por estas razones su uso en la robótica móvil es sobresaliente. En el presente trabajo, se propone un algoritmo basado en el filtro de partículas, debido a que el modelo cinemático del robot tiene no linealidades que lo hacen complejo.

Por lo anterior, se plantea un algoritmo de fusión de sensores que tiene como propósito mejorar la localización del robot rover, el cual utiliza la información de 5 sensores que operan a diferentes frecuencias de muestreo, y de los cuales 3 son sensores de orientación y 2 son sensores que entregan la posición.

El esquema completo de control basado en fusión de sensores se muestra en la Figura 3; el cual cuenta con el bloque de control PI realimentado con el vector de estados estimado ( $\hat{x}, \hat{y}, \hat{\varphi}$ ) obtenido mediante el filtro de partículas. Además, el bloque de control recibe la señal de la localización deseada ( $x^*, y^*$ ) y entrega la señal de control  $u$  aplicada al modelo y al filtro. En el esquema de la figura 3 se presentan los modelos de medición en dos grupos, el primero mide la orientación del rover y el segundo mide la posición en las coordenadas  $(x, y)$  y a partir de estas señales, el filtro de partículas realiza la fusión.

### 4. Simulación y resultados

La simulación numérica se realiza con ayuda del software Matlab-Simulink. Y la idea es evaluar recursivamente la estimación de los estados del sistema en tiempo discreto y realizar una comparación entre los estados, las mediciones y estimaciones.

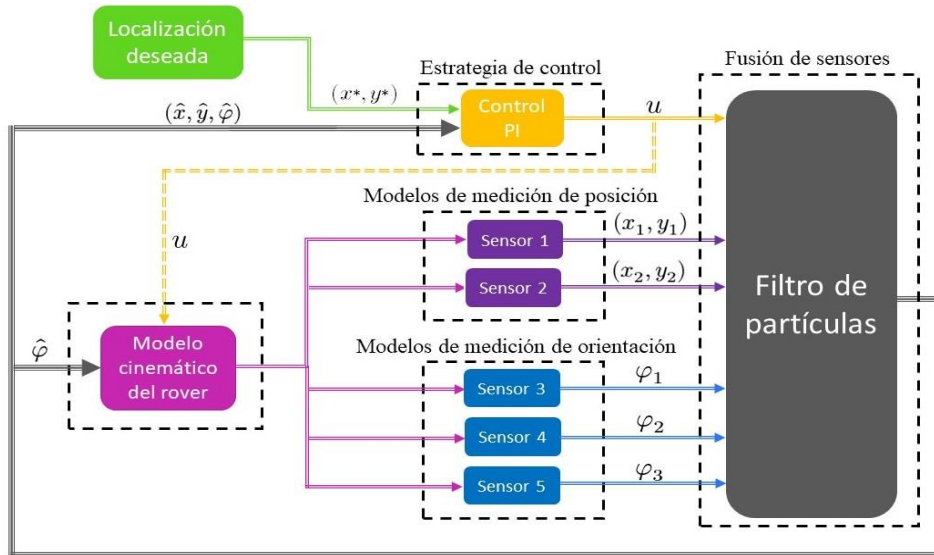


Figura 3: Esquema de control basado en fusión de sensores.

Tabla 2: Parámetros de simulación

	Parámetro	Descripción
<b>Disposición del entorno</b>	Tiempo de simulación	500 s
	Tipo de paso	Paso variable
	Solver	Ode45 (Dormand-Prince)
<b>Filtro de partículas</b>	Función de transición de estados	transicionEstadosRoverShrimpPF
	Función de probabilidad de medición de posición	probabilidadMedicionPosicionXY
	Función de probabilidad de medición de orientación	probabilidadMedicionOrientacion
	Media inicial	[0;0;0]
	Covarianza inicial	diag([1e-6 1e-6 1e-6])
	Número de partículas	1000
<b>Constantes del Controlador</b>	$k_{px}, k_{ix}$	0.4, 0.001
	$k_{py}, k_{iy}$	0.4, 0.001
<b>Modelo del rover</b>	$R$	0.062 m
	$L$	0.105 m
	$b$	0.445 m

#### 4.1. Función de transición de estados

Uno de los elementos que requieren los algoritmos de estimación de estados, es la función de transición de estados (modelo del rover), la cual describa su evolución en el tiempo. Para el caso del filtro de partículas empleado, se tiene la siguiente función.

$$X[k + 1] = f_{pf}(X[k], u[k]) \quad (9)$$

donde  $X[k]$  es el vector de estados de tamaño  $[N_s \ N_p]$ ; con  $N_s$  igual al número de estados y  $N_p$  el número de partículas,  $u[k]$  es el vector de entradas del sistema que, en este caso, son las salidas del controlador. La sintaxis de la ecuación (9) se debe a que el filtro de partículas funciona siguiendo las trayectorias de muchas hipótesis de estado (partículas). Además, el impacto del ruido del proceso  $w$ , sobre las hipótesis es calculado en la función  $f_{pf}(X[k], u[k])$  y no se realizan suposiciones sobre la distribución de probabilidad del ruido  $w$ , por lo que no necesita ser declarado. La función de transición de estados se declara en un script de Matlab a partir del modelo cinemático del rover y el código se presenta en el apéndice A.

#### 4.2. Funciones de probabilidad de medición

El filtro de partículas también requiere las funciones de medición, que calculen la probabilidad de cada hipótesis de estado. Estas funciones tienen la siguiente forma:

$$L[k] = h_{pf}(X[k], y(k), u(k)) \quad (10)$$

donde  $L[k]$  es un vector de  $N_p$  elementos, con  $N_p$  igual al número de partículas elegidas. El  $n$ -ésimo elemento de  $L[k]$  es la probabilidad de la partícula  $n$ -ésima (columna) en  $X[k]$ ,  $y[k]$  es la medición del sensor,  $u[k]$  es un parámetro de entrada opcional, que puede diferir de las entradas de la función de transición de estados. En este caso, los sensores miden estados diferentes, los sensores de posición miden los dos primeros estados (posición en  $x, y$ ) y los de orientación miden el tercer estado; por lo que son necesarias dos funciones de probabilidad de medición diferentes. Los errores entre las mediciones reales y predichas se asumen con una distribución gaussiana; sin embargo, se puede utilizar cualquier otro tipo de distribución para calcular las probabilidades, lo cual es una de las ventajas del filtro de partículas. Las funciones  $h_{pf}(X[k], y(k), u(k))$  de cada sensor se crean en scripts separados; el código se presenta en los apéndices B y C.



### 4.3. Configuración de la simulación

Una vez construidas las funciones requeridas por del filtro de partículas de Simulink, se crean los bloques del modelo del rover, la estrategia de control y los 5 sensores, con el fin de ejecutar la simulación.

Tabla 3 Características de los sensores

Sensor	Varianza del ruido (Precisión)	Tiempo de muestreo
Sensor 1	16	1 s
Sensor 2	6.25	0.2 s
Sensor 3	$9.2793 \times 10^{-12}$	0.01 s
Sensor 4	$37.21 \times 10^{-6}$	0.01 s
Sensor 5	$1.9097 \times 10^{-3}$	0.01 s

Por otra parte, los sensores se simulan mediante la obtención de los estados y la adjunción del ruido de medición, que en este caso, es simulado como ruido blanco de tipo con valores similares a los de unidades de medición inercial y GPS comerciales, cuyos valores se presentan en la Tabla 2. Los bloques de los subsistemas de medición (sensores del 1 al 5 de la Figura 3) están representados en la Figura 4, donde se bosqueja la forma en que se simulan las mediciones de los distintos sensores; para cada medición se toma la señal real de posición u orientación y se le suma el ruido de acuerdo con la varianza de la Tabla 2, obteniendo una señal similar a la entregada por cada sensor físico en cuanto a precisión.

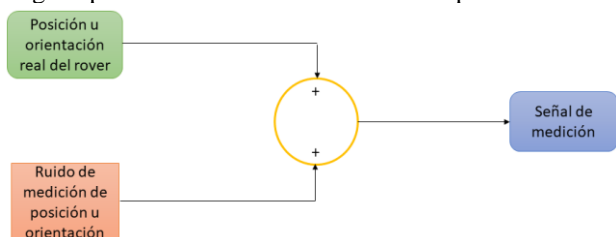


Figura 4: Diagrama de la simulación de las mediciones

Finalmente, en la Tabla 3 se presentan los demás detalles concernientes a los parámetros de la simulación.

### 4.4. Resultados

Con la finalidad de verificar el comportamiento del esquema de control basado en fusión de sensores, se propone que el robot alcance la posición  $x = y = 120$  m en un movimiento en línea recta recorriendo aproximadamente 170 m. Para la simulación del recorrido se considera una velocidad angular máxima de las ruedas de 8 rad/s. Además, las condiciones iniciales de los 3 estados son 0 y las velocidades traslacionales deseadas también son consideradas 0, debido a que la propuesta contempla únicamente coordenadas deseadas constantes, invariantes en tiempo; es decir, la trayectoria deseada se considera como una referencia de tipo escalón, en cuyo caso se espera que la velocidad del rover sea cero al alcanzar la localización deseada.

En la Figura 5 se muestra el recorrido del rover hasta la localización deseada, mostrando la comparación entre las señales: real, estimada y de medición; junto con una vista detallada de los resultados del recorrido. Además, se puede ver que la señal de estimación tiene un comportamiento mejor que la señal de medición de un solo sensor, así como un comportamiento similar a la señal real. Asimismo, se puede

observar que la señal de estimación se encuentra acotada por la región de las mediciones. De lo anterior, se puede deducir que, el control del rover basado en fusión de sensores mediante el filtro de partículas consigue resultados de seguimiento adecuados.

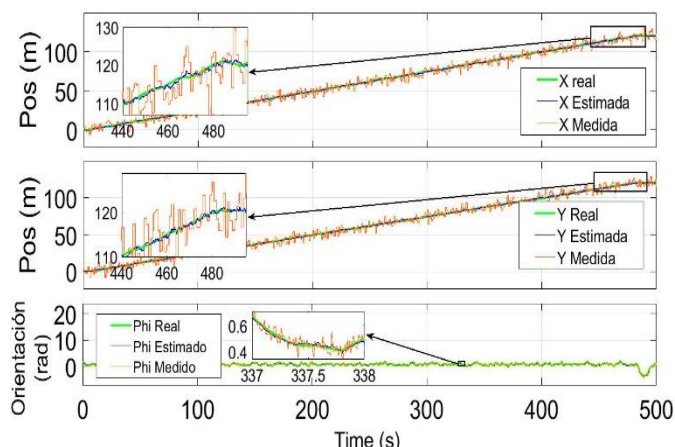


Figura 5: Comportamiento del recorrido del robot

En la Figura 6 se muestra el valor RMSE (sigla en inglés de error cuadrático medio) de estimación; dicho error es la diferencia entre la señal estimada por el filtro de partículas y la señal de posición y orientación entregadas por el modelo cinemático; en la figura se puede ver el comportamiento del RMSE, el cual converge a 0.6 antes de los primeros 100 s. La convergencia del error es un criterio importante en los análisis de consistencia de algunos métodos de filtrado, por lo que el resultado indica una correcta estimación.

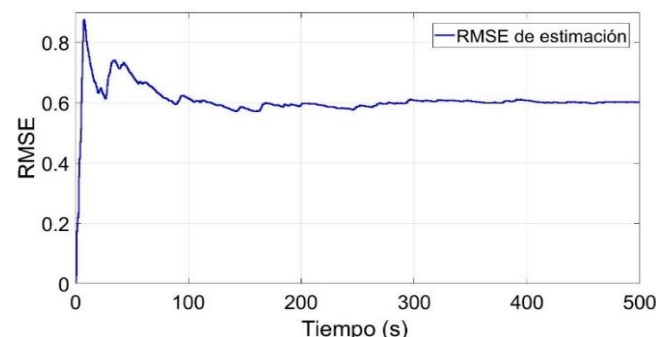


Figura 6: RMSE de estimación

En la Figura 7 se muestra una comparación del valor RMSE de posición obtenido mediante estimación y medición. En la figura es posible apreciar como en los dos casos el error converge a cero a medida que el robot se aproxima a la posición deseada. En el caso del RMSE de medición (RMSE de posición sin la fusión), se simula el mismo recorrido del rover, pero utilizando las mediciones de un solo sensor para cada variable de estado (el más preciso). Además, también es posible apreciar en la figura, en la vista con acercamiento, como el valor RMSE de medición es mayor que el de estimación, a su vez que incrementa cuando el robot ha alcanzado la posición deseada. Finalmente, se puede mencionar que el valor de RMSE es 0.71 y 1.762 para la estimación y medición respectivamente, representando una mejora del 40.3% en la precisión del robot al utilizar la fusión de sensores.

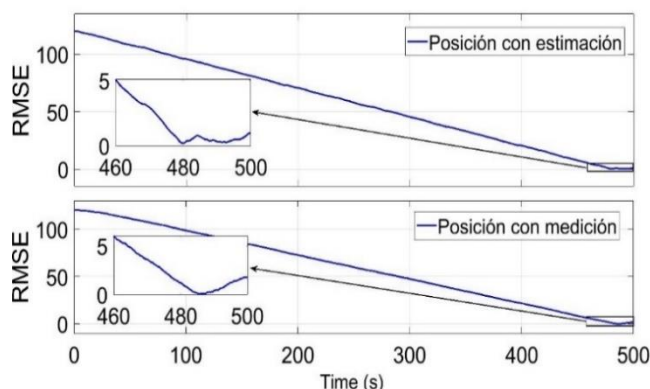


Figura 7: RMSE de posición obtenida con estimación y con medición

## 5. Conclusiones

En este trabajo, se presenta un esquema de control basado en fusión de sensores aplicado a un robot tipo rover, teniendo un comportamiento adecuado frente a un sistema que utilice solo un dispositivo de medición. Esto se debe a que el controlador utiliza la posición filtrada proveniente de la fusión de 5 sensores, que es más precisa que cualquiera de las mediciones por sí solas para llevar el robot de un punto inicial al final, obteniendo que la señal de estimación tiene un comportamiento mejor que la señal de medición de un solo sensor, así como un comportamiento similar a la señal real. Como trabajo futuro se tiene la implementación del esquema propuesto en el prototipo de robot tipo rover presentado.

## Agradecimientos

Este trabajo está financiado por la SIP-IPN, con el número de registro 20220632. Los autores agradecen el apoyo otorgado en la convocatoria de proyectos de desarrollo tecnológico o de innovación para estudiantes del IPN 2021.

## Referencias

- Ababsa, F. E., & Mallem, M. (2012). Hybrid three-dimensional camera pose estimation using particle filter sensor fusion. *Http://Dx.Doi.Org/10.1163/156855307779293689*, 21(1–2), 165–181. <https://doi.org/10.1163/156855307779293689>
- Agarwal, H., Tiwari, P., & Tiwari, R. G. (2019). Exploiting Sensor Fusion for Mobile Robot Localization. *Proceedings of the 3rd International Conference on I-SMAC IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2019*, 463–466. <https://doi.org/10.1109/I-SMAC47947.2019.9032653>
- Ávila, M. A. O., & Arancibia, J. A. G. (2015). Sistema de localización autónoma para robots móviles basado en fusión de sensores propioceptivos. *Revista Politécnica*, 11(21), 75–84.
- Bassiri, A., Asghari Oskoei, M., Basiri, A., & Li, L. L. (2018). Particle filter and finite impulse response filter fusion and hector SLAM to improve the performance of robot positioning. *Journal of Robotics*, 2018. <https://doi.org/10.1155/2018/7806854>
- Baturone, A. O. (2005). *Robótica: manipuladores y robots móviles*. Marcombo.
- Chen, Y., & Rui, Y. (2004). Real-time speaker tracking using particle filter sensor fusion. *Proceedings of the IEEE*, 92(3), 485–494. <https://doi.org/10.1109/JPROC.2003.823146>
- Ding, L., Zhou, R., Yu, T., Gao, H., Yang, H., Li, J., Yuan, Y., Liu, C., Wang, J., Zhao, Y.-Y. S., Wang, Z., Wang, X., Bao, G., Deng, Z., Huang, L., Li, N., Cui, X., He, X., Jia, Y., ... Di, K. (2022). Surface characteristics of the Zhurong Mars rover traverse at Utopia Planitia. *Nature Geoscience*, 15(3), 171–176. <https://doi.org/10.1038/s41561-022-00905-6>
- Ellery, A. (2004). Space Robotics: Part 3: Robotic Rovers for Planetary Exploration. *Https://Doi.Org/10.5772/5809*, 1(4), 31. <https://doi.org/10.5772/5809>
- Koch, W. (2016). *Tracking and sensor data fusion*. Springer.
- Nathan, V., & Jafari, R. (2018). Particle Filtering and Sensor Fusion for Robust Heart Rate Monitoring Using Wearable Sensors. *IEEE Journal of Biomedical and Health Informatics*, 22(6), 1834–1846. <https://doi.org/10.1109/JBHI.2017.2783758>
- Rigatos, G. G. (2010). Extended Kalman and Particle Filtering for sensor fusion in motion control of mobile robots. *Mathematics and Computers in Simulation*, 81(3), 590–607. <https://doi.org/10.1016/J.MATCOM.2010.05.003>
- Thomas, U., Molkenstruck, S., Iser, R., & Wahl, F. M. (2007). Multi sensor fusion in robot assembly using particle filters. *Proceedings - IEEE International Conference on Robotics and Automation*, 3837–3843. <https://doi.org/10.1109/ROBOT.2007.364067>
- Tian, Q., Wang, K. I. K., & Salcic, Z. (2020). A Resetting Approach for INS and UWB Sensor Fusion Using Particle Filter for Pedestrian Tracking. *IEEE Transactions on Instrumentation and Measurement*, 69(8), 5914–5921. <https://doi.org/10.1109/TIM.2019.2958471>
- Vadakkepat, P., & Jing, L. (2006). Improved particle filter in sensor fusion for tracking randomly moving object. *IEEE Transactions on Instrumentation and Measurement*, 55(5), 1823–1832. <https://doi.org/10.1109/TIM.2006.881569>
- Von Ehrenfried, M. “Dutch.” (2022). Perseverance and the Mars 2020 Mission. *Perseverance and the Mars 2020 Mission*. <https://doi.org/10.1007/978-3-030-92118-7>
- Yang, N., Tian, W. F., Jin, Z. H., & Zhang, C. bin. (2005). Particle filter for sensor fusion in a land vehicle navigation system. *Measurement Science and Technology*, 16(3), 677. <https://doi.org/10.1088/0957-0233/16/3/008>
- Zaidner, G., & Shapiro, A. (2016). A novel data fusion algorithm for low-cost localisation and navigation of autonomous vineyard sprayer robots. *Biosystems Engineering*, 146, 133–148. <https://doi.org/10.1016/J.BIOSYSTEMSENG.2016.05.002>
- Zhai, Y., Yeary, M. B., Havlicek, J. P., & Fan, G. (2008). A new centralized sensor fusion-tracking methodology based on particle filtering for power-aware systems. *IEEE Transactions on Instrumentation and Measurement*, 57(10), 2377–2387. <https://doi.org/10.1109/TIM.2008.919009>

## Apéndice A. Función de transición de estados

```
function x =
transicionEstadosRoverShrimpPF(x,u)

% Datos reales del rover
R = 0.062;
L = 0.105;
b = 0.425;

% Tamaño del paso
dt = 10e-3;

% Modelo
xt = zeros(3,length(x));
xt(1,1:length(x)) =
((R/2)*cos(x(3,:))+(L*R/b)*sin(x(3,:)))*u
(1,1)+((R/2)*cos(x(3,:))-
(L*R/b)*sin(x(3,:)))*u(2,1);
xt(2,1:length(x)) = ((R/2)*sin(x(3,:))-
(L*R/b)*cos(x(3,:)))*u(1,1)+((R/2)*sin(x(
3,:))+(L*R/b)*cos(x(3,:)))*u(2,1);
xt(3,1:length(x)) = (-
R/b*ones(1,length(x))).*u(1,1)+(R/b*ones(
1,length(x))).*u(2,1);

% Evolución de los estados
x = (x+xt*dt)+bsxfun(@times,[2.7e-2;
2.7e-2; 8e-2],randn(size(x)));
end
```

**Apéndice B. Función de probabilidad de medición de posición**

```
function probabilidad =  
probabilidadMedicionPosicionXY(particulas  
, medicion)  
  
yHat = particulas(1:2,:);  
e = bsxfun(@minus, yHat,  
medicion(1:2,:)); % Error  
numberOfMeasurements = 2;  
mu = 0; % Media  
Sigma = eye(numberOfMeasurements); %  
Varianza  
mediciontErrorProd = dot((e-mu), Sigma \  
(e-mu), 1);  
c = 1/sqrt((2*pi)^numberOfMeasurements *  
det(Sigma));  
probabilidad = c * exp(-0.5 *  
mediciontErrorProd);  
end
```

**Apéndice C. Función de probabilidad de medición de orientación**

```
function probabilidad =  
probabilidadMedicionOrientacion(particula  
s, medicion)  
  
yHat = particulas(3,:);  
e = bsxfun(@minus, yHat, medicion(:)); %  
Error  
numberOfMeasurements = 1;  
mu = 0; % Media  
Sigma = eye(numberOfMeasurements); %  
Varianza  
mediciontErrorProd = dot((e-mu), Sigma \  
(e-mu), 1);  
c = 1/sqrt((2*pi)^numberOfMeasurements *  
det(Sigma));  
probabilidad = c * exp(-0.5 *  
mediciontErrorProd);  
end
```