

Localización basada en lógica difusa y filtros de Kalman para robots con patas

Francisco Martín, Vicente Matellán, Pablo Barrera, Jose María Cañas

Grupo de Robótica, Universidad Rey Juan Carlos,
C/ Tulipán s/n 28933 Móstoles (Madrid), España.
{fmartin,vmo,barrera,jmplaza}@gsyc.escet.urjc.es *

Resumen En la liga de 4 patas de la Robocup, equipos de 4 robots AIBO autónomos se enfrentan entre si en partidos de fútbol. El objetivo de esta competición es presentar un entorno desafiante en que se han de resolver varios problemas relacionados con la robótica. En particular, la localización. Cada uno de los jugadores ha de estar localizado durante los partidos para que su comportamiento sea coherente. En este artículo proponemos un método de localización que combina técnicas de lógica difusa y filtros de Kalman para conseguir una localización más robusta, fiable y ligera.

Palabras Clave: Robótica móvil, fútbol robótico, localización, fuzzy, kalman

1. Introducción

Una de las habilidades más básicas que ha de ser capaz de llevar a cabo un robot móvil es la capacidad de auto-localización [4]. Esta capacidad se puede definir como la habilidad de un robot de determinar su posición en el mundo usando sus propios sensores. Las técnicas que se usan para resolver este problema varían enormemente dependiendo de los sensores disponibles en cada tipo de robot. En muchos trabajos previos, por ejemplo [9][8][6][5][2], se han propuesto soluciones a la localización de robots móviles con ruedas equipados de sensores de sonar y láser. La diferencia principal de estos trabajos con nuestra propuesta es que este tipo de robots dispone de información odométrica muy precisa y una abundante información del entorno de 360°. Estas soluciones pueden no ser aplicables en otros robots cuyos sensores sean diferentes.

En este trabajo nos centramos en el robot con patas AIBO (parte derecha de la figura 1). Este robot tiene como sensor principal una cámara situada en su cabeza. Las imágenes que se obtienen de la cámara han de ser procesadas para obtener información de ella. Otra característica de este robot es la de tener como principales actuadores de locomoción cuatro patas, lo que hace difícil obtener una información odométrica precisa.

* Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología, en el proyecto ACRAE: DPI2004-07993-C03-01 y la Comunidad de Madrid en el proyecto RoboCity 2030: S-0505/DPI/0176

Este modelo de robot se usa en la competición de la RoboCup¹, en la categoría de cuatro patas² (parte izquierda de la figura 1). La competición RoboCup es una iniciativa de investigación y educación en el ámbito internacional, que pretende fomentar el campo de la inteligencia artificial y de la robótica a base de proporcionar a los investigadores un problema estándar, donde puedan probar sus trabajos y evaluarlos.

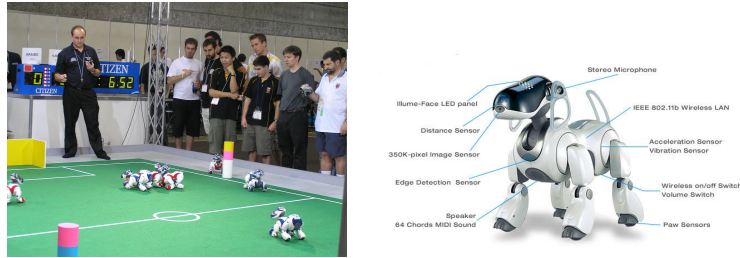


Figura 1. El TeamChaos en la RoboCup 2006 (Osaka, Japón). A la derecha el robot AIBO ERS-7

En nuestro entorno es importante que los robots estén localizados en todo momento para que su comportamiento sea coherente. Los robots no deben de salirse de los límites del campo y deben colocarse, por ejemplo, en una posición inicial conocida al iniciar el partido y después de cada gol. Así mismo, para saber hacia dónde han de dirigir la pelota y poder generar una estrategia común entre los miembros de un equipo, los robots han de estar localizados. Los métodos usados por cada uno de los equipos son muy variados, quedando reflejados en la tabla 1 los usados en la edición del 2004, última para la que está publicada la documentación de todos los equipos. En esta tabla puede apreciarse como la mayor parte de los equipos usan métodos basados en Filtros de Partículas (Localización de Monte Carlo). Varios equipos utilizan también el Filtro de Kalman Extendido, algunos de ellos combinados con Monte Carlo. Otros simplemente utilizan triangulación. nuestro equipo³ hasta la fecha ha usado un método de lógica difusa [1] [7].

En una competición tan exigente como la RoboCup, en la que se tiene que realizar una toma de decisiones en tiempo real, el tiempo de proceso de cada uno de sus módulos es crítico. Si se pretenden realizar comportamientos complejos de alto nivel, estrategias sobre todo, el procesado de tareas de bajo nivel, como filtrado de imágenes o localización, han de consumir el mínimo tiempo de proceso posible. El método de localización de lógica difusa, que hemos estado usando hasta ahora, consume una gran cantidad de recursos del robot si se desea una precisión de localización aceptable. Esto ha motivado que se decidiera estudiar

¹ <http://www.robocup.org>

² <http://www.tzi.de/4legged/bin/view/Website/WebHome>

³ <http://veo.dat.escet.urjc.es/dipta/teamchaos.html>

Método	Nº equipos
Localización de Monte Carlo	10
Filtro Extendido de Kalman	2
Monte Carlo + Filtro Extendido de Kalman	2
Triangulación	2

Cuadro 1. Métodos de localización utilizados en la edición de la RoboCup 2004

la combinación de este método con un Filtro de Kalman, que es un estimador de estados óptimo que consume pocos recursos computacionales. Este artículo mostrará el método usado y los resultados obtenidos, demostrando que es posible obtener una localización fiable y robusta, que es aportada por el método de lógica difusa, y una condiciones aceptables de tiempo de proceso, aportadas por el Filtro de Kalman.

La sección 2 mostrará el entorno en el que el robot se ha de localizar. En la sección 3 mostraremos el método de lógica difusa empleado en hasta ahora. En la sección 4 expondremos el método de Filtro de Kalman diseñado para este entorno, para explicar en la sección 5 cómo combinados los dos anteriores. En la sección 6 pondremos a prueba la aproximación implementada, que será discutida finalmente en la sección 7, en la que abordaremos también los posibles trabajos futuros.

2. Entorno

El entorno donde debe localizarse el robot está diseñado para que el robot disponga de una serie de marcas visuales que el robot puede usar para localizarse. Las dimensiones del campo y las posiciones de las marcas visuales son conocidas *a priori*, como puede observarse en la figura 2. Estas marcas visuales son las dos porterías y cuatro balizas de colores situadas en las bandas. Nada impide que uses otras marcas del campo, como son las líneas, para localizarte en él. Las condiciones de luz son también controladas para que sean constantes en todo momento. A pesar de estas facilidades, los algoritmos desarrollados para la localización deben tener en cuenta que las marcas pueden ser tapadas por otros robots, y que la información que se extrae de las imágenes puede ser errónea, debido al continuo movimiento de la cámara, que se encuentra situada en la cabeza del robot, las colisiones entre ellos, etc.

3. Método de localización usando lógica difusa

El objetivo de desarrollar este método de localización fue la de dotar al robot de una forma robusta de representar la incertidumbre de su posición. También debía de poder recuperarse de situaciones en las que el robot era desplazado de un lugar a otro del campo. Esta situación se puede producir cuando el robot es penalizado por el arbitro y retirado unos minutos del terreno de juego, o cuando

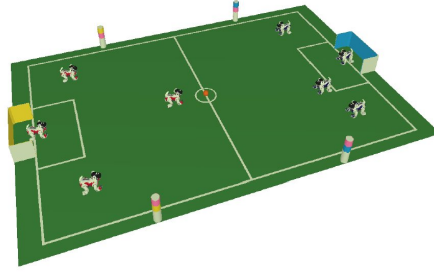


Figura 2. Campo de juego

es empujado por otros robots. Su descripción completa puede encontrarse en [1] y [3]. A continuación resumimos este trabajo previo como base de la mejora propuesta en este artículo.

El campo de juego se divide en una cuadrícula G_t tal que $G_t(x, y)$ representa la probabilidad, en $[0, 1]$, de que el robot se encuentre en la posición (x, y) . Cada una de las posiciones de esta cuadrícula es una celda de dimensión configurable. Cada una de las celdas, que definiremos en adelante como $fcell$, contiene información sobre la probabilidad de que el robot esté en esa celda, e información sobre cual es el rango de orientaciones más probables para el robot, es decir, se trata realmente de una cuadrícula de $2\frac{1}{2}D$

Esta información se representa por medio de un trapezoide difuso. En la figura 3 podemos ver este trapezoide, definido por la tupla

$$\langle \theta, \Delta, \alpha, h, b \rangle$$

Intuitivamente, si h es bajo, la probabilidad de estar en esta celda es baja. Si h es alto, es muy probable que el robot esté en esta posición. Si el trapezoide es ancho, existe gran incertidumbre sobre la orientación del robot. Si el trapezoide es estrecho, o tiene incluso forma de triángulo (porque Δ es prácticamente nulo), la incertidumbre de orientación es tan baja que podemos afirmar que es θ .

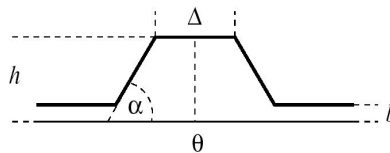


Figura 3. $fcell$ representando el ángulo θ

El proceso de localización usando en este método es iterativo, teniendo cada ciclo un paso de predicción y otro de actualización. La fase de predicción se realiza cada vez que se realiza un movimiento difuminando la probabilidad en la

dirección del movimiento. En la fase de actualización se incorpora la información visual. Cada observación de una marca visual se compone de una distancia y un ángulo a cada una de las visibles. Para codificar la información de una marca visual conocida en el instante t , construimos la distribución de probabilidad $S_t(\cdot|r)$, tal que $S_t(x, y|r)$ es la posibilidad de que el robot se encuentre en la posición (x, y) , siendo la distancia a una marca visual determinada r .

Un ejemplo de la aplicación secuencial de estas dos operaciones para la localización de un robot se puede observar en la figura 4. El robot parte de un estado de total incertidumbre, y mediante la información odométrica y la información de la posición relativa de la portería y de la baliza superior derecha, termina localizándose correctamente.

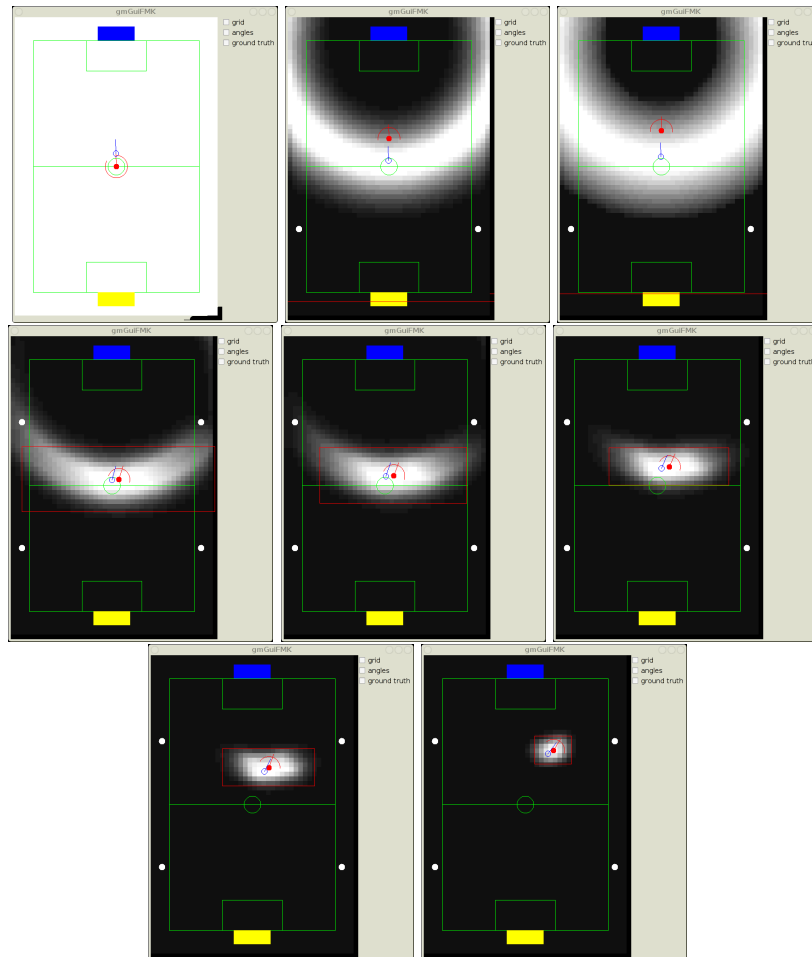


Figura 4. Proceso de localización del robot mediante grid borroso

El proceso anterior nos proporciona la probabilidad de estar en una posición de la cuadrícula, pero no nos aporta información angular. Podría ser natural considerar un cubo 3D $G_t(x, y, \theta)$, pero el tiempo de computación de todas las posibles posiciones del robot haría que el algoritmo no fuera abordable computacionalmente. En lugar de eso se mantiene una cuadrícula 2D de $fcell$, que es una forma compacta de representar información angular. Con el procedimiento descrito anteriormente obtenemos la componente h de esta $fcell$, pero aún es necesario describir como obtener el resto de las componentes cuando se produce una observación. Esta operación es muy ligera y calcula las nuevas probabilidades de orientación a partir de una estimación anterior y una nueva observación.

4. Método de localización usando filtro de Kalman

Para implementar un Filtro de Kalman hemos definido la posición de un robot como el vector de estado $s \in \mathbb{R}^3$, que se compone de :

$$\mathbf{s} = (x_{robot} \ y_{robot} \ \theta_{robot}) \quad (1)$$

El proceso de evolución de esta estimación de la posición del robot estará guiado por dos funciones no lineales, f y h (ésta última será definida en la sección 4.2 para mayor claridad). La primera, f (obtenida a partir de la figura 4, relaciona el estado anterior s_{t-1} , la odometría u_{t-1} y un ruido en el proceso gaussiano w_{t-1} con el estado actual s_t :

$$s_t = f(s_{t-1}, u_{t-1}, w_{t-1}) \quad (2)$$

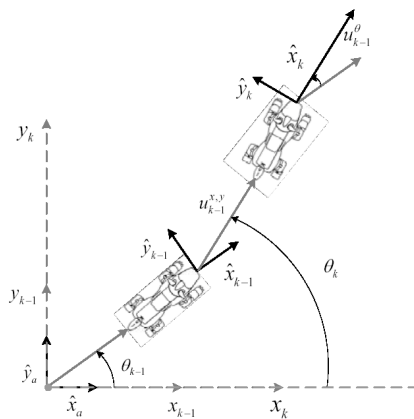


Figura 5. Modelo de movimiento basado en la odometría

$$f_1 = x_t^- = x_{t-1}^- + (u_{t-1}^x + w_{t-1}^x)\cos\theta_{t-1} - (u_{t-1}^y + w_{t-1}^y)\sen\theta_{t-1} \quad (3)$$

$$f_2 = y_t^- = y_{t-1}^- + (u_{t-1}^x + w_{t-1}^x)\sen\theta_{t-1} + (u_{t-1}^y + w_{t-1}^y)\cos\theta_{t-1} \quad (4)$$

$$f_3 = \theta_t^- = \theta_{t-1}^- + u_{t-1}^\theta + w_{t-1}^\theta \quad (5)$$

4.1. Fase de predicción

En esta fase calcularemos $s_t^- \in \mathfrak{R}^3$ que es la estimación de la posición a priori, es decir, la posición en la que se encontrará el robot según la información de la odometría y aplicando el modelo de movimiento de la figura 4 en el proceso. También calcularemos $P_t^- \in \mathfrak{R}^{3 \times 3}$, que es la covarianza del error del sistema a priori. Esta matriz será la que almacene el error general del sistema. Si tuviéramos certeza absoluta de la posición inicial $P_0 = 0$, pero este no es el caso. Arbitrariamente elegimos una matriz no nula, $P_0 = I_3$, ya que esta matriz identidad es capaz de converger rápidamente a valores correctos.

$$s_t^- = f(s_{t-1}, u_{t-1}, 0) \quad (6)$$

$$P_t^- = A_t P_{t-1} A_t^T + W_t Q_{t-1} W_t^T \quad (7)$$

A continuación vamos a describir los principales elementos que intervienen en la fase de predicción y el significado de sus valores:

Q_t es la covarianza del error en la fase de predicción viene definido por $Q_t \in \mathfrak{R}^{3 \times 3}$ ($Q_t = E[w_k w_k^T]$), y los valores que contiene son experimentales. Indica el error que se puede producir al aplicar la odometría.

$$Q_t = \begin{pmatrix} 0,1|u_{t-1}^x| & 0 & 0 \\ 0 & 0,1|u_{t-1}^y| & 0 \\ 0 & 0 & 0,1|u_{t-1}^\theta| + 0,001\sqrt{(u_{t-1}^x)^2 + (u_{t-1}^y)^2} \end{pmatrix} \quad (8)$$

A_t es el jacobiano de f con respecto a s . De las ecuaciones 3, 4 y 5 obtenemos:

$$A_t = \frac{\partial f}{\partial s} = \begin{pmatrix} 1 & 0 & -u_{t-1}^y \cos\theta_{t-1} - u_{t-1}^x \sen\theta_{t-1} \\ 0 & 1 & u_{t-1}^x \cos\theta_{t-1} - u_{t-1}^y \sen\theta_{t-1} \\ 0 & 0 & 1 \end{pmatrix} \quad (9)$$

W_t es el jacobiano de f con respecto al error de odometría w . Al igual que en 9, de las ecuaciones 3, 4 y 5 obtenemos:

$$W_t = \frac{\partial f}{\partial w} = \begin{pmatrix} \cos\theta_{t-1} & -\sen\theta_{t-1} & 0 \\ \sen\theta_{t-1} & \cos\theta_{t-1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (10)$$

4.2. Fase de corrección

En la fase de corrección, la posición del robot se corrige con la información sensorial que percibe. Esta información sensorial percibida, $\hat{z}_t \in \mathfrak{R}^{2m}$, tiene dimensión variable, pues depende del número de elementos m del que tiene información. La medida para cada elemento está formada por un ángulo (α_i) y una distancia (ρ_i) a dicho elemento. De esta manera, tenemos:

$$\hat{z}_t = (\rho_1 \alpha_1 \rho_2 \alpha_2 \cdots \rho_m \alpha_m)^T \quad (11)$$

En esta fase hallaremos la posición del robot una vez corregida con las medidas s_t y la matriz de covarianza P_t que almacena el error del sistema:

$$s_t = s_t^- + K_t(\hat{z}_t - h(s_t^-, 0)) \quad (12)$$

$$P_t = (I - K_t H_t) P_t^- \quad (13)$$

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + V_t R_t V_t^T)^{-1} \quad (14)$$

La función $h(s_t^-, 0)$ relaciona la posición s_t^- con la medida teórica z_t que debería obtenerse en esa posición. Usamos un modelo geométrico para calcularlo:

$$z_t = h(s_t^-, 0) \quad (15)$$

$$\begin{pmatrix} \rho_1 \\ \alpha_1 \\ \vdots \\ \rho_m \\ \alpha_m \end{pmatrix} = \begin{pmatrix} h_1(s_t^-, 0) \\ h_2(s_t^-, 0) \\ \vdots \\ h_{2m-1}(s_t^-, 0) \\ h_{2m}(s_t^-, 0) \end{pmatrix} = \begin{pmatrix} distancia(s_t^-, lm_1) \\ angulo(s_t^-, lm_1) \\ \vdots \\ distancia(s_t^-, lm_m) \\ angulo(s_t^-, lm_m) \end{pmatrix} \quad (16)$$

A continuación vamos a describir los principales elementos que intervienen en la fase de corrección y el significado de los valores aplicados:

R_t Las medidas tienen asociadas una matriz, $R_t \in \mathfrak{R}^{2m \times 2m}$, $R_t = E[v_k v_k^T]$, que representa el error en la medida. Esta matriz es una matriz diagonal cuyos valores son empíricos. En el caso de las distancias, el valor depende del tipo de elemento y de la distancia a él, y representa el error que puede darse en esas condiciones. En caso del ángulo, el error es muy pequeño (2°) si el elemento está completo en la imagen, y mayor si no está completo (18°).

V_t La matriz $V_t \in \mathfrak{R}^{2m \times 2m}$ es la matriz del jacobiano de h con respecto a v , que es la matriz identidad I^{2m} en nuestro caso.

H_t la matriz $H_t \in \mathfrak{R}^{2m \times 3}$ es la matriz del jacobiano de h con respecto a s ,

$$H_t = \frac{\partial h_t}{\partial s^-} = \begin{pmatrix} \frac{\partial distancia(s_i^-, lm_1)}{\partial x} & \frac{\partial distancia(s_i^-, lm_1)}{\partial y} & \frac{\partial distancia(s_i^-, lm_1)}{\partial \theta} \\ \frac{\partial angulo(s_i^-, lm_1)}{\partial x} & \frac{\partial angulo(s_i^-, lm_1)}{\partial y} & \frac{\partial angulo(s_i^-, lm_1)}{\partial \theta} \\ \vdots & \vdots & \vdots \\ \frac{\partial distancia(s_i^-, lm_{2m-1})}{\partial x} & \frac{\partial distancia(s_i^-, lm_{2m-1})}{\partial y} & \frac{\partial distancia(s_i^-, lm_{2m-1})}{\partial \theta} \\ \frac{\partial angulo(s_i^-, lm_{2m})}{\partial x} & \frac{\partial angulo(s_i^-, lm_{2m})}{\partial y} & \frac{\partial angulo(s_i^-, lm_{2m})}{\partial \theta} \end{pmatrix} \quad (17)$$

$$H_{[i,j],t} = \frac{\partial h_{t,i}}{\partial s_j} = \frac{h_i(s_{j,t}^- + \Delta, 0) - h_i(s_{j,t}^- - \Delta, 0)}{2\Delta} \quad (18)$$

5. Combinación de lógica difusa y filtro de Kalman

La combinación de los dos métodos de localización combina las mejores cualidades de cada uno. Por una parte, el método de localización basado en lógica difusa permite una localización robusta, capaz de recuperarse de situaciones de completa incertidumbre.

Este método, aunque use una manera compacta de representar la información angular de cada casilla para hacerlo más ligero, consume muchos recursos computacionales. Más aún cuando para la edición del 2005 de la RoboCup se decidió aumentar considerablemente la resolución⁴. Una posible solución para que el método fuera usable sería aumentar el tamaño de cada cuadrícula en la que se divide el campo, pero esto daba lugar a una significativa pérdida de precisión en la estimación de la posición del robot.

El método de localización basado en el Filtro de Kalman es, por contra, computacionalmente ligero. El problema de este método es que no es capaz de localizarse partiendo de un escenario de total incertidumbre, y no es capaz de recuperarse de situaciones de alto error en la estimación o de cambio manual de la posición del robot en el terreno de juego.

La estrategia propuesta consiste en intercambiar el método de localización alternativamente. Al iniciarse el robot, éste usa el método de localización basado en lógica difusa hasta que la calidad de esta localización es suficientemente aceptable para asegurar que el robot está perfectamente localizado. En ese momento el robot cambia su algoritmo de localización al Filtro de Kalman. Si en algún momento la matriz P del Filtro de Kalman refleja una incertidumbre excesiva para considerar al robot localizado, el algoritmo de localización del robot cambia al método de lógica difusa. El proceso se repite alternativamente dependiendo de la calidad de las estimaciones.

⁴ <http://www.tzi.de/4legged/pub/Website/Downloads/Rules2005.pdf>

6. Experimentos

Los experimentos llevados a cabo en este proyecto se centran en validar la robustez del método del Filtro de Kalman. Los resultados experimentales del método de lógica difusa pueden encontrarse en [1][3].

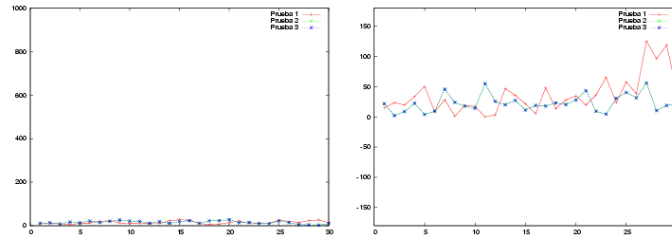


Figura 6. Error en la estimación de la posición (x, y) y en la estimación de la orientación θ en el experimento 1

Los experimentos llevados a cabo para validar el método del Filtro de Kalman se llevaron a cabo en un simulador que permite configurar los valores máximos de un ruido que es generado aleatoriamente, tanto para la información odométrica, como para las medidas de distancia y ángulo con respecto a cada una de las marcas visuales.

Las figuras 6,7 y 8 muestran los resultados de tres experimentos diferentes donde el robot ha realizado un desplazamiento por el terreno de juego. Las figuras muestran el error en la estimación de la posición (x, y) y en la estimación de la orientación θ con valores de ruido similares a los que realmente existen en la realidad.

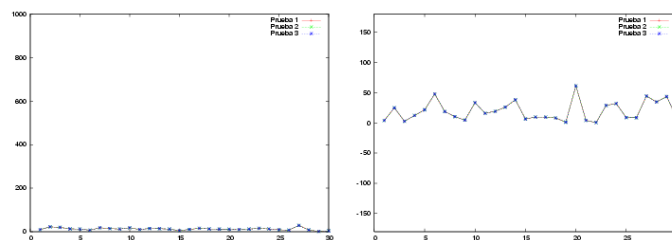


Figura 7. Error en la estimación de la posición (x, y) y en la estimación de la orientación θ en el experimento 2

En estos experimentos se muestra como el error en estimación de la posición (x, y) no supera en ningún momento los 7 cm. La estimación de orientación es

la que más error acumula, aunque nunca llega a superar los 60 grados. Además, se mantiene bajo a lo largo de la mayoría del proceso.

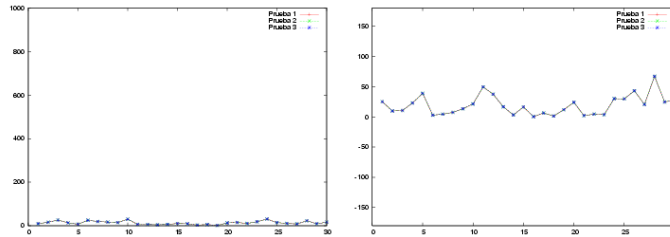


Figura 8. Error en la estimación de la posición (x, y) y en la estimación de la orientación θ en el experimento 3

Hay que hacer constar que no se produce un cambio continuo de métodos de localización, ya que el método de lógica difusa converge rápidamente, y el Filtro de Kalman estima correctamente la posición en la mayor parte las situaciones. Generalmente las únicas situaciones en las que hay que volver a localizarse totalmente con el método de lógica difusa son al principio del funcionamiento del robot, cuando éste es “secuestrado”, o cuando hay un error grave y continuo de la información extraída de las imágenes.

7. Conclusiones y trabajo futuro

Se ha desarrollado en este trabajo un método de localización combinando la robustez y capacidad de recuperación del método de localización difuso implementado previamente, con un método que usa un Filtro de Kalman, cuyo punto fuerte es el escaso tiempo de procesamiento que necesita debido a las operaciones que ha de realizar en cada ciclo. El Filtro de Kalman desarrollado es capaz de estimar correctamente la posición del robot con los niveles de ruido presentes en el escenario donde se ha de localizar. La combinación de ambos métodos es simple. Se activa en cada momento el método adecuado dependiendo de las necesidades de localización y de la estimación actual. La mayor parte del tiempo usa el método del Filtro de Kalman, que consume pocos recursos, y el método de localización basado en lógica difusa se activa puntualmente para recuperarse rápidamente en caso de que la estimación del error aumente mucho.

El tiempo de procesamiento total del módulo de localización se ha optimizado, resultado que era necesario para la viabilidad de la totalidad del código desarrollado para el robot. Si este tiempo fuera superior, tareas como la locomoción o la coordinación del movimiento con la estimación de la pelota fallarían.

Los trabajos futuros se centran en realizar un combinación óptima de ambos algoritmos. también el desarrollo de otros métodos, como puedes ser Filtros de Partículas, para reiniciar el Filtro de Kalman en ciertas situaciones puede ser útil.

Referencias

- [1] P. Buschka, A. Saffiotti, and Z. Wasik. Fuzzy landmark-based localization for a legged robot. In *Proceedings of the International Conference on Intelligent Robots and Systems 2000*, Takamatsu, Japan, October 2000.
- [2] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.
- [3] D. Herrero-Pérez, H. Martínez-Barberá, and A. Saffiotti. Fuzzy self-localization using natural features in the four-legged league. *Lecture Notes in Computer Science. Robocup 2004*, 3276, 2005.
- [4] J. Borenstein, B. Everett, and L. Feng. *Navigating mobile robots: Systems and techniques*. Ltd. Wesley, MA, 1996.
- [5] Jana Kosecká and Fayin li. Vision based topological markov localization. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, Barcelona (Spain), April 2004.
- [6] María E. López, Luis Miguel Bergasa, and M.S. Escudero. Visually augmented POMDP for indoor robot navigation. *Applied Informatics*, pages 183–187, 2003.
- [7] Humberto Martínez, Vicente Matellán, and Miguel Cazorla. Teamchaos technical report. Technical report, TeamChaos, 2006.
- [8] Dandapani Radhakrishnan and Illah Nourbakhsh. Topological localization by training a vision-based transition detector. In *Proceedings of IROS 1999*, volume 1, pages 468 – 473, October 1999.
- [9] Reid Simmons and Sven Koenig. Probabilistic navigation in partially observable environments. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, pages 1080–1087, Montreal (Canada), July 1995.