

FILTROS DE PARTÍCULAS

Eva Besada-Portas y Sergey M. Plis

A8.1 INTRODUCCIÓN

A continuación se describe el ejemplo que sirve para ilustrar el funcionamiento de los Filtros de Partículas (*Particle Filter*, PF) que se presentan en el capítulo 8¹.

El problema elegido es la localización de un robot dentro de un entorno conocido. La técnica adoptada para estimar su localización en diferentes instantes de tiempo en base a la información proporcionada por varios sensores es un PF.

En las siguientes secciones se describe el problema, los datos disponibles, los programas² de Matlab utilizados para generar los datos e implementar el PF, y los resultados que se obtienen.

¹ El ejemplo utilizado para explicar el funcionamiento de las cadenas de Markov también presentadas en dicho capítulo se muestra en el apéndice 7.1, junto con el resto de las técnicas de muestreo auxiliar recogidas en el capítulo 7

² Al final de este documento se presenta una tabla explicativa de todos los ficheros de Matlab proporcionados. Aquellos cuyo nombre empieza por “*Main*” se corresponden a los diferentes programas principales explicados a lo largo del documento.

A8.2 DESCRIPCION DEL PROBLEMA

El problema que se presenta en este apéndice es la localización de un robot dentro de un entorno conocido por medio de un PF en base a la información proporcionada por varios sensores.

El robot se mueve sobre una superficie plana en función de una señal de control que le indica el giro que debe realizar y la distancia que debe recorrer en la dirección marcada por la orientación después de realizar el giro. La posición X_t del robot en cada instante de tiempo se encuentra definida por tres elementos: su posición en el plano ($X_{x,t}$ y $X_{y,t}$) y su orientación ($X_{\theta,t}$). La localización inicial del robot ($X_{x,0}$ y $X_{y,0}$) es desconocida, aunque no su orientación ($X_{\theta,0}$).

La información sensorial en cada instante de tiempo y_t está formada por dos elementos: $y_{a,t}$ que representa la distancia, medida con un sensor de ultrasonidos, de la parte superior del robot al techo e $y_{d,t}$ que representa si un sensor fotoeléctrico orientado hacia el techo detecta, o no, un foco de luz situado encima del robot. La distribución de los puntos de luz, que determina si se detecta o no un foco de luz desde los diferentes puntos del espacio, y la altura del techo son elementos conocidos. Por lo tanto, las medidas de los sensores ($y_{a,t}$ e $y_{d,t}$) nos proporcionan información sobre la posición del robot en el plano ($X_{x,t}$ y $X_{y,t}$), hecho que nos permite delimitar probabilísticamente las regiones del espacio en las que se puede encontrar el robot en cada instante.

El comportamiento de este sistema queda definido a través del modelo probabilístico dinámico del robot $p(X_t|X_{t-1}, c_{t-1})$ y del modelo probabilístico sensorial $p(Y_t|X_t, map_a, map_d)$. El modelo dinámico se presenta en la ecuación (A8.1) y se implementa en la función “*ModelRobot.m*”, mientras que el modelo sensorial se recoge en la ecuación (A8.2) y se implementa en la función “*ModelSensor.m*”

El modelo dinámico del robot $p(X_t|X_{t-1}, c_{t-1})$ también depende de la señal de control c_{t-1} aplicada en cada instante del tiempo. La primera componente $c_{l,t-1}$ representa la velocidad lineal del robot, mientras que la segunda $c_{\theta,t-1}$ representa la velocidad angular. La orientación del robot en un instante $X_{\theta,t}$ depende de su orientación $X_{\theta,t-1}$ en el instante anterior, de la velocidad angular

$c_{\theta,t-1}$, de la varianza en la velocidad angular $Q_{\theta,t-1}$ y del tiempo transcurrido Δ_t entre el instante t y el instante $t-1$. La localización en el plano en un instante ($X_{x,t}$ y $X_{y,t}$) depende de la localización ($X_{x,t-1}$ y $X_{y,t-1}$) en el instante anterior, de la orientación $X_{\theta,t}$, de la velocidad lineal $c_{l,t-1}$, de la varianza en la velocidad lineal $Q_{l,t-1}$ y del tiempo transcurrido Δ_t .

$$\begin{aligned}
 p(X_t | X_{t-1}, c_{t-1}) &= p(X_{\theta,t} | X_{\theta,t-1}, c_{\theta,t-1}) \\
 &\quad p(X_{x,t} | X_{\theta,t}, X_{x,t-1}, c_{l,t-1}) \\
 &\quad p(X_{y,t} | X_{\theta,t}, X_{y,t-1}, c_{l,t-1}) \\
 \text{con } X_t &= [X_{x,t}, X_{y,t}, X_{\theta,t}] \\
 c_t &= [c_{l,t}, c_{\theta,t}] \tag{A8.1} \\
 p(X_{\theta,t} | X_{\theta,t-1}, c_{\theta,t-1}) &= \mathcal{N}_{X_{\theta,t}}(X_{\theta,t-1} + \Delta_t c_{\theta,t-1}, \Delta_t^2 Q_{\theta,t-1}) \\
 p(X_{x,t} | X_{\theta,t}, X_{x,t-1}, c_{l,t-1}) &= \mathcal{N}_{X_{x,t}}(X_{x,t-1} + \alpha c_{l,t-1}, \alpha^2 Q_{l,t-1}) \\
 p(X_{y,t} | X_{\theta,t}, X_{y,t-1}, c_{l,t-1}) &= \mathcal{N}_{X_{y,t}}(X_{y,t-1} + \beta c_{l,t-1}, \beta^2 Q_{l,t-1}) \\
 \alpha &= \Delta_t \cos(X_{\theta,t}) \\
 \beta &= \Delta_t \sin(X_{\theta,t}) \\
 \Delta_t &\text{ tiempo transcurrido entre } t \text{ y } t-1
 \end{aligned}$$

El modelo sensorial $p(Y_t | X_t, \text{mapa}_l, \text{mapa}_a)$ depende de la distribución de los puntos de luz mapa_l y de la altura al techo mapa_a . La medida de altura $Y_{a,t}$ depende de la altura al techo en la posición ($X_{x,t}$ y $X_{y,t}$) en la que se encuentra el robot y de la varianza $Q_{a,t}$ asociada al sensor de altura. La medida de detección $Y_{d,t}$ depende de la distancia entre la posición ($X_{x,t}$ y $X_{y,t}$) en la que se encuentra el robot y la posición de todos los focos de luz mapa_l , del factor de decaimiento luminoso r , y de los límites de luminosidad mínimo l_{\min} y máximo l_{\max} de la célula fotoeléctrica.

La expresión utilizada para modelar el sensor de luminosidad $p(Y_{d,t} | X_{x,t}, X_{y,t}, \text{mapa}_l)$ hace que sean puntos equiprobables del espacio, aquéllos en los que se acumula, debido a todos los puntos de luz que tienen a su alrededor, una cantidad de luz suficiente/insuficiente para que el sensor sea capaz de

detectarla o no. La expresión utilizada para modelar al sensor de altura $p(Y_{a,t} | X_{x,t}, X_{y,t}, \text{mapa}_a)$ hace que sean puntos equiprobables del espacio aquellos que se encuentran a la misma distancia de la parte superior del robot

$$\begin{aligned}
 p(Y_t | X_t, \text{mapa}_l, \text{mapa}_a) &= p(Y_{a,t} | X_{x,t}, X_{y,t}, \text{mapa}_a) \\
 &\quad p(Y_{d,t} | X_{x,t}, X_{y,t}, \text{mapa}_d) \\
 \text{con } X_t &= [X_{x,t}, X_{y,t}, X_{\theta,t}] \\
 Y_t &= [Y_{a,t}, Y_{d,t}] \\
 p(Y_{a,t} | X_{x,t}, X_{y,t}, \text{mapa}_a) &= \mathcal{N}_{Y_{a,t}}(A, Q_{a,t}) \\
 A &= \text{Altura}(\text{mapa}_a, X_{x,t}, X_{y,t}) \\
 p(Y_{d,t} = 1 | X_{x,t}, X_{y,t}, \text{mapa}_l) &= f(\delta, l_{\min}, l_{\max}) \\
 f(\delta, l_{\min}, l_{\max}) &= \begin{cases} 0 & \delta < l_{\min} \\ \frac{\delta - l_{\min}}{l_{\max} - l_{\min}} & l_{\min} < \delta < l_{\max} \\ 1 & c.c. \end{cases} \quad (\text{A8.2}) \\
 \delta &= \sum_{i=1}^L g \left((Px(\text{mapa}_l, i) - X_{x,t})^2 + (Py(\text{mapa}_l, i) - X_{y,t})^2 \right) \\
 g(\alpha) &= \exp(-\alpha/r) \\
 L &\text{ número de focos en el } \text{mapa}_l \\
 Px(\text{mapa}_l, i) &\text{ posición } x \text{ del foco } i \text{ en el } \text{mapa}_l \\
 Py(\text{mapa}_l, i) &\text{ posición } y \text{ del foco } i \text{ en el } \text{mapa}_l \\
 \text{Altura}(\text{mapa}_a, X_{x,t}, X_{y,t}) &\text{ altura del } \text{mapa}_a \text{ en la posición} \\
 &\quad X_{x,t}, X_{y,t}
 \end{aligned}$$

Utilizando los modelos matemáticos que definen el comportamiento del sistema realizamos una simulación para obtener la trayectoria real $x_{0:t}$ seguida por el robot y las medidas $y_{0:t}$ proporcionadas por los sensores. A partir de dichos valores, resolveremos el problema de localización del robot, consistente en estimar el valor medio de la trayectoria $X_{0:t}$ utilizando diferentes PFs.

8.1 DATOS DEL PROBLEMA

Los datos utilizados para resolver el problema son los parámetros que nos permiten configurar el comportamiento de los modelos (señal de control en cada instante de tiempo c_{t-1} , su varianza lineal $Q_{l,t-1}$ y angular $Q_{\theta,t-1}$, la diferencia de tiempo Δ_t entre dos instantes de tiempo sucesivos, el mapa de alturas $mapa_a$, la varianza $Q_{a,t}$ del sensor que mide la distancia al techo, el mapa de distribución de los focos de luz $mapa_l$, el factor de decaimiento luminoso r , y los limites de luminosidad mínimo l_{\min} y máximo l_{\max} de la célula fotoeléctrica) y las medidas $y_{0:t}$ obtenidas durante una simulación del comportamiento del sistema a partir de una posición inicial x_0 determinada.

La elección de los valores de todos los parámetros salvo el $mapa_a$ y el $mapa_l$ se realiza en la función de Matlab “*Data.m*”. En ella también se fija el valor de la posición inicial del robot x_0 . Las posiciones de los focos en el $mapa_l$ se definen en la función “*MapLight.m*” y la altura del $mapa_a$ en cualquier punto del espacio, se define en la función “*Height.m*”. Además “*MapLight.m*” y “*Height.m*” utilizan la función “*LimitMap.m*” para definir las dimensiones x e y de la región del espacio en la que se encuentra confinado el robot.

La simulación que genera las medidas $y_{0:t}$ necesarias para estimar la $X_{0:t}$ a partir de los PFs se realiza en la función “*Simulation.m*”.

El programa principal “*MainSimulation.m*” utiliza los valores definidos en “*Data.m*”, “*MapLight.m*”, “*Height.m*” y “*LimitMap.m*” para lanzar la simulación y almacenar los parámetros que la definen y las medidas resultantes de los mismos en el fichero “*SimulaData.mat*”. Su contenido será el punto de partida de una ejecución de un PF. En “*SimulaData.mat*” también se almacena la trayectoria $x_{0:t}$ para poder compararla con la estimada por los PFs.

Debido al comportamiento estocástico de los modelos de transición y medida, dos simulaciones con los mismos parámetros obtienen trayectorias reales $x_{0:t}$ y medidas $y_{0:t}$ diferentes. Esto hace cada ejecución del “*MainSimulation.m*” modificará los valores de $y_{0:t}$ y $x_{0:t}$ almacenados en “*SimulaData.mat*” aunque no se hayan modificado las definiciones de los parámetros definidos en “*Data.m*”, “*MapLight.m*”, “*Height.m*” y “*LimitMap.m*”. Por lo tanto “*MainSimulation.m*” debe ser llamada para generar nuevos datos para los PFs o cuando se modifica el

contenido de las funciones “*Data.m*”, “*MapLight.m*”, “*Height.m*” y “*LimitMap.m*”.

Los datos utilizados en los ejemplos de este apéndice son los que se encuentran actualmente definidos en los ficheros “*Data.m*”, “*MapLight.m*”, “*Height.m*”, “*LimitMap.m*”, “*SimulaData.m*”, y “*SimulaData.mat*”. Una copia de seguridad de todos ellos se suministra en el directorio “*Data*”.

“*MainSimulation.m*” también genera tres figuras que representan gráficamente las características del problema definido. La primera, figura A8.1, muestra la altura del techo y la disposición de los focos (arcos sobre el techo plano). Los cambios de altura del techo se deben a la existencia de dos vigas y una bóveda. La segunda, figura A8.2, representa la trayectoria real $x_{0:t}$ simulada y almacenada en “*SimulaData.mat*”, y la disposición de los puntos de luz (círculos pequeños), vigas (rectas) y bóveda (círculo grande). La tercera y ultima, figura A8.3, muestra las medidas $y_{1:t}$ simuladas y almacenadas en “*SimulaData.mat*”. Estas mismas figuras se presentan e interpretan de forma detallada en el capítulo 8.

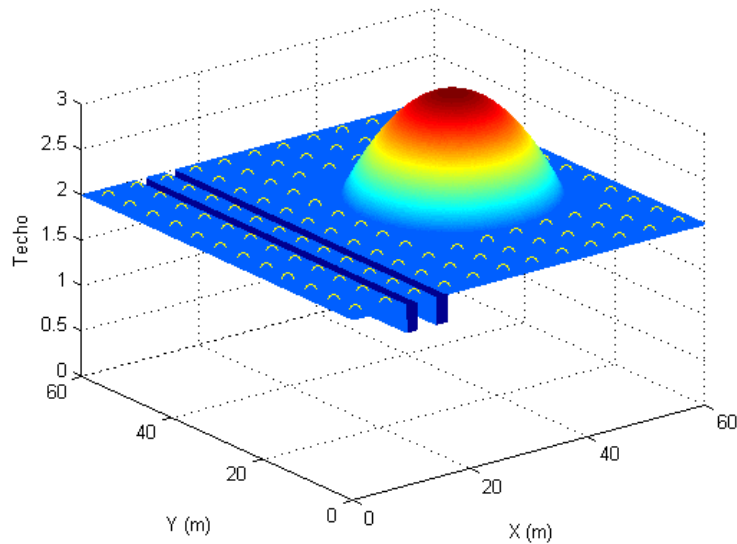


Figura A8.1. Altura del techo y distribución de los puntos de luz.

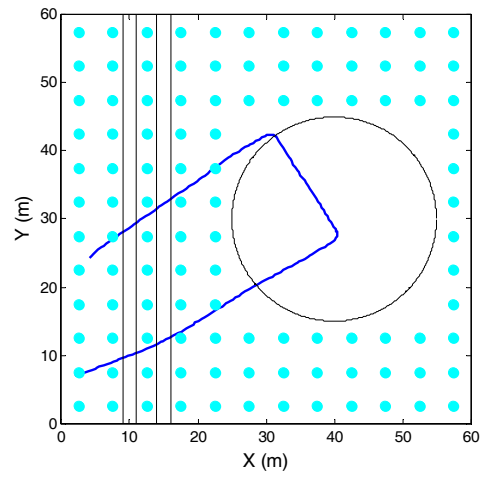


Figura A8.2. Trayectoria real del robot.

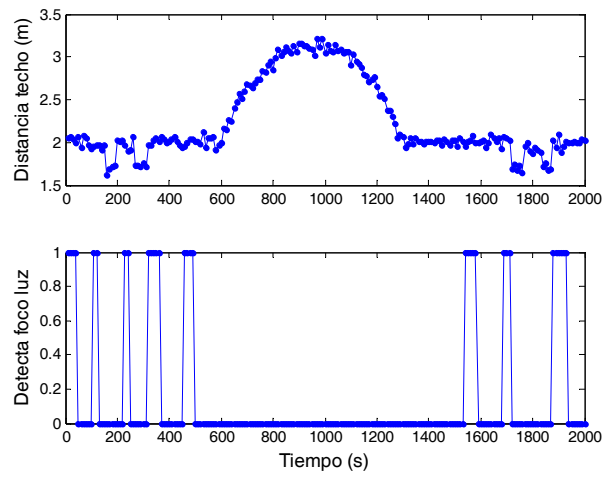


Figura A8.3. Medidas proporcionadas por los sensores

8.2 ALGORIMOS DE PROCESAMIENTO

Los modelos matemáticos y datos presentados en los apartados anteriores son utilizados por 3 PFs para estimar el valor medio de la trayectoria $X_{0:t}$.

Los algoritmos utilizados, “*Sampling Importance Sampling*” (SIS), “*Sampling Importance Resampling*” (SIR), y “*Auxiliary Particle Filter*” (APF), se implementan en una única función “*ParticleFilter.m*”, ya que los 3 PFs elegidos comparten varias etapas de procesamiento. La elección del tipo de PF que se utiliza en cada caso se realiza mediante el parámetro de entrada “*resample*”. Otra característica relevante de “*ParticleFilter.m*” es que permite al usuario especificar por medio del parámetro “*sen*” qué parte del modelo sensorial $p(Y_t | X_t, \text{mapa}_l, \text{mapa}_a)$ debe ser utilizado para calcular las probabilidades de las medidas en las etapas necesarias de los filtros de partículas. Si “*sen=1*” $\rightarrow p(Y_t | X_t, \text{mapa}_l, \text{mapa}_a) = p(Y_{a,t} | X_{x,t}, X_{y,t}, \text{mapa}_a)$, si “*sen=2*” $\rightarrow p(Y_t | X_t, \text{mapa}_l, \text{mapa}_a) = p(Y_{d,t} | X_{x,t}, X_{y,t}, \text{mapa}_l)$ y sin “*sen=[1,2]*” $\rightarrow p(Y_t | X_t, \text{mapa}_l, \text{mapa}_a) = p(Y_{a,t} | X_{x,t}, X_{y,t}, \text{mapa}_a) p(Y_{d,t} | X_{x,t}, X_{y,t}, \text{mapa}_l)$. De esta forma podemos ejecutar los PFs utilizando distinto tipo información sensorial a la hora de estimar el valor medio de la trayectoria $X_{0:t}$. La función “*ParticleFilter.m*” también salva, en el caso en el que se le proporcione el nombre de un fichero en el parámetro “*filename*”, el valor de las partículas $x_{0:k}^{(i)}$ y sus pesos $w(x_{0:k}^{(i)})$ para los distintos valores de k . Esta característica permite analizar el comportamiento de las partículas una vez que el algoritmo finaliza pero requiere de mucha memoria computacional, por lo que en el caso en el que sólo se desee observar el comportamiento del PF mientras está estimando las trayectorias, es conveniente deshabilitarla. Finalmente también se le puede indicar a “*ParticleFilter.m*” que mientras obtiene los valores de las partículas represente en los instantes k elegidos a través del parámetro “*plotsteps*” el valor $x_k^{(i)}$ de la k -ésima componente de todas las partículas en el instante k (rojo), el valor de la trayectoria real $x_{0:k}$ hasta dicho instante k (azul), y el valor medio de la trayectoria $X_{0:k}$ calculado a partir de los valores de las partículas $x_{0:k}^{(i)}$ (verde). Esta representación también puede ser realizada a posteriori, a partir de los resultados almacenados en el fichero de resultados, utilizando la función “*DrawParticleFilterResults.m*”

El programa principal “*MainParticleFilter.m*” carga los resultados de la simulación almacenados en el fichero “*SimulaData.mat*”, define el valor de los parámetros “*resample*”, “*sen*”, “*filename*” y “*plotsteps*”, y ejecuta el PF elegido.

En “*MainParticleFilter.m*” también se define el número de partículas del PF “*NP*” y la región del espacio en el que se pueden generar el valor de las partículas en el instante inicial $x_0^{(i)}$.

El programa principal “*MainDrawParticleFilterResults.m*” representa los resultados obtenidos y almacenados en ficheros por “*MainParticleFilter.m*”. En las siguientes figuras se muestran algunos de estas gráficas cuando “*sen=1*”. Los ficheros de salida utilizados para generarlas no se proporcionan, si bien otros, con un contenido similar, pueden ser generados ejecutando la orden “*MainParticleFilter.m*”.

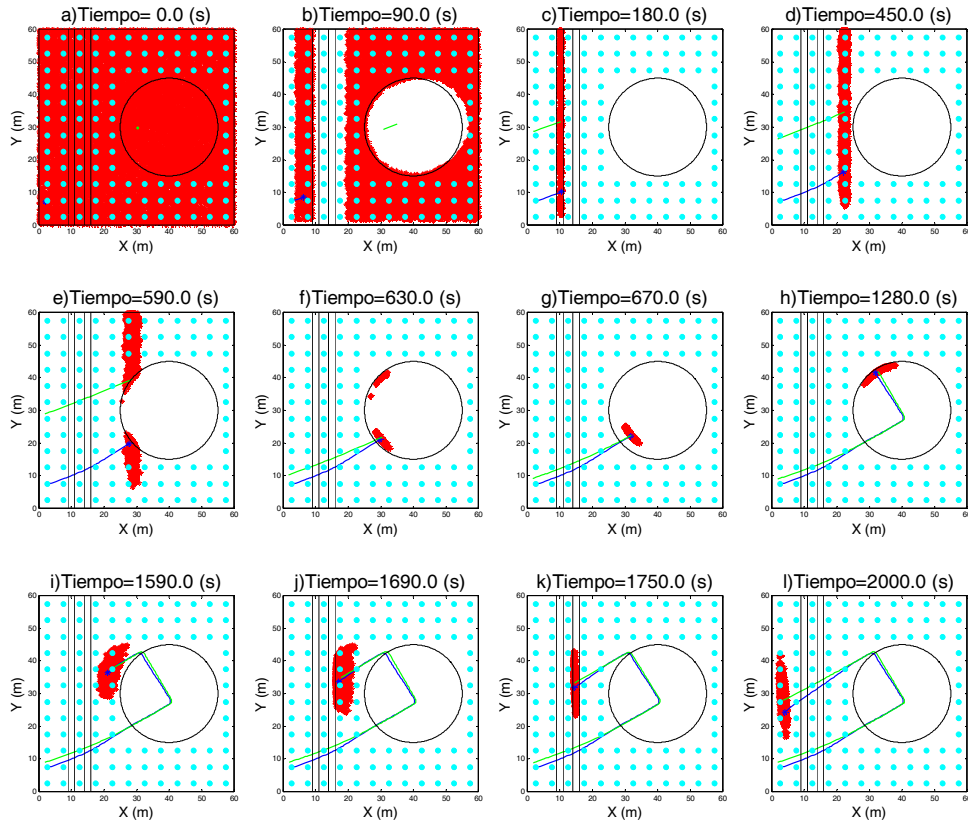


Figura A8.4. Disposición de las partículas usando solo el sensor 1

8.3 LISTADO DE FUNCIONES Y PROGRAMAS

Finalmente se incluye el listado de funciones y programas suministrados, con una breve descripción de su utilidad. Los programas principales han sido resaltados en negrita. En el directorio “Data” se almacena una copia de seguridad con los ficheros de definición de los datos.

| | |
|---|--|
| <i>AngDiff.m</i> | <i>Función que calcula diferencias de ángulos</i> |
| <i>AngNorm.m</i> | <i>Función que convierte cualquier ángulo fuera del rango $[0, 2\pi]$ en un valor dentro de dicho rango</i> |
| <i>Data</i> | <i>Directorio donde se almacena una copia de seguridad de los ficheros de definición de datos</i> |
| <i>Data.m</i> | <i>Fichero de configuración de los modelos</i> |
| <i>DrawParticleFilterResults.m</i> | <i>Función que muestra por pantalla los resultados que se han almacenado en un fichero de salida tras una ejecución de un PF</i> |
| <i>Height.m</i> | <i>Función en la que se define la altura del techo en cualquier posición del espacio</i> |
| <i>MainDrawParticleFilterResults.m</i> | <i>Programa principal que muestra por pantalla los resultados que se han almacenado en un fichero de salida de una ejecución de un PF</i> |
| <i>MainParticleFilter.m</i> | <i>Programa principal que ejecuta un PF</i> |
| <i>MainSimulation.m</i> | <i>Programa principal con el que se realiza una simulación</i> |
| <i>MapLight.m</i> | <i>Función que define la posición de los focos</i> |
| <i>MapLimits.m</i> | <i>Función que define los límites de la región del espacio en los que se mueve el robot</i> |
| <i>ModelRobot.m</i> | <i>Función que implementa el modelo de transición de estado del robot</i> |
| <i>ModelSensor.m</i> | <i>Función que implementa el modelo sensorial del robot</i> |
| <i>ObtainMeanTrajectory.m</i> | <i>Función que calcula la trayectoria media con el valor de las partículas y sus pesos</i> |
| <i>ParticleFilter.m</i> | <i>Función que implementa los PFs</i> |
| <i>ResampleResidual.m</i> | <i>Método de remuestreo³</i> |

³El código de esta función forma parte de la función “*residualR.m*” proporcionado por A. Doucet y N. de Freitas dentro de sus herramientas de implementación de unscented PFs (<http://www.cs.ubc.ca/~nando/software.php>).

| | |
|-----------------------|---|
| <i>SimulaData.mat</i> | <i>Fichero con los resultados de una simulación</i> |
| <i>Simulation.m</i> | <i>Función que realiza una simulación</i> |
| <i>gsamp.m</i> | <i>Función que genera muestras distribuidas de acuerdo con una distribución gaussiana</i> |