

Capítulo 3 – Localización Monte Carlo.

3.1 Introducción.

En este capítulo se estudia el método probabilístico de localización Monte Carlo. También se presentan tres variantes de este método: Monte Carlo Dual, Mixture Monte Carlo y Adaptive Monte Carlo. Estas variantes han sido propuestas por distintos autores y tienen la finalidad de mejorar aspectos de Monte Carlo como la velocidad de ejecución, el tiempo de convergencia y la exactitud del mismo.

3.2 Monte Carlo.

Los métodos de Monte Carlo son un conjunto de técnicas de muestreo estadístico, las cuales investigadores y científicos han desarrollado durante los últimos 50 años para representar cualquier distribución con un número reducido de muestras sin perder representatividad [18]. Estos métodos fueron introducidos en los setentas [15] y recientemente redescubiertos independientemente en seguimiento de objetivos (target-tracking) [21], estadística [10] y visión por computadora [20].

La localización de Monte Carlo (MCL), es un método probabilístico muy adecuado en mapas de interiores, ya que incorpora incertidumbre en acciones y observaciones, lo cual concuerda con la incertidumbre (errores de odometría) de las estimaciones de los sensores [22]. La localización probabilística consiste en el proceso de determinar la probabilidad de que el robot se encuentre en una determinada pose dada una historia de las lecturas tomadas por los sensores del robot y de una serie de acciones ejecutadas por el mismo. A cada

posible pose del robot se le asocia una probabilidad que refleja la verosimilitud de que sea la pose actual. Las estimaciones se actualizan con la incorporación de nuevas observaciones y nuevos movimientos ejecutados por el robot. Esto permite al robot poder localizarse dentro del ambiente aun sin conocer su posición inicial, toma en cuenta los errores de odometría y además permite la representación de situaciones ambiguas que se resuelven posteriormente, la eficiencia de este método depende del tamaño del mapa en el que se realiza la localización del agente autónomo [18].

3.2.1 Filtros de Bayes.

El método de localización Monte Carlo corresponde a un filtro de Bayes recursivo que estima la distribución a posteriori del estado de un sistema, condicionada en función de los datos. Los filtros de Bayes apuntan al problema de estimar el estado x de un sistema dinámico, es decir, variante en el tiempo. En el caso de la robótica móvil, el sistema dinámico es el robot móvil y su entorno, mientras que el estado es su pose. La idea central de los filtros de Bayes es estimar una densidad de probabilidad sobre el espacio de los estados, condicionada a los datos. Esta densidad de probabilidad generalmente es llamada la creencia (*belief*) [22][6].

$$Bel(x_t) = p(x_t | o_t, a_{t-1}, o_{t-1}, a_{t-2}, \dots, o_0)$$

Donde x_t es el estado en el tiempo t , o_t corresponde a los datos perceptuales obtenidos en el instante t , y a_t corresponde a los comandos de movimiento enviados al

robot en el instante t , los cuales se usan como datos odométricos. Usando la regla de Bayes y asumiendo que el proceso es de tipo Markoviano, se puede derivar la siguiente ecuación recursiva.

$$Bel(x_t) = \eta \cdot p(o_t | x_t) \int p(x_t | x_{t-1}, a_{t-1}) Bel(x_{t-1}) dx_{t-1} \quad (1)$$

Donde η corresponde a una constante de normalización. Esta es la ecuación de actualización recursiva para filtros de Bayes. Para implementarla, es necesario conocer dos densidades de probabilidad: $p(x_t | x_{t-1}, a_{t-1})$ la cual es usualmente llamada modelo de movimiento y $p(o_t | x_t)$ que recibe comúnmente el nombre de modelo de observación. Ambos modelos son estacionarios por lo que pueden ser escritos como $p(x', | x, a)$ y $p(o | x)$ respectivamente. El método Monte Carlo no requiere una definición analítica del modelo de movimiento; basta con tener un modelo de muestreo que sea compatible con $p(x', | x, a)$. Un modelo de muestreo puede ser definido como una función que recibe x y a como parámetros y entrega posiciones aleatorias x' distribuidas de acuerdo con $p(x', | x, a)$ [22][14].

3.2.2 Filtros de partículas.

Dado que la localización de robots móviles es un problema de estados continuos, implementar la ecuación (1) correspondiente a los filtros de Bayes no es trivial. Por lo tanto el cálculo de $Bel(x_t)$ es muy costoso debido a que para el cálculo de la integral se deben evaluar $p(x_t | x_{t-1}, a_{t-1})$ y $Bel(x_t)$ en todo el espacio de estados [22].

El método de localización Monte Carlo es una versión de samplig/importance resampling (SIR) [5]. Alternativamente también es conocido como filtro bootstrap [21], filtro Monte Carlo [10], y Condensation algorithm [20]. Todos estos métodos son generalmente conocidos como filtros de partículas [4].

El principio de los filtros de partículas, consiste en representar la creencia posterior $Bel(x_t)$ por un conjunto de N muestras ponderadas, distribuidas de acuerdo con $Bel(x_t)$ también llamadas partículas o individuos. Sus ponderaciones $\omega_t^{(i)}$ son también llamados pesos o puntajes normalizados, mientras que el conjunto Φ_t de muestras es llamado población y está definido por:

$$\Phi_t = \left\{ \langle X_t^{(i)}, \omega^{(i)} \rangle \mid i = 1, \dots, n \right\}$$

Donde n corresponde al tamaño de la población.

El conjunto de muestras constituye una aproximación discretizada de la distribución de probabilidad.

Las partíuclas en MCL son del tipo:

$$\langle \langle x, y, \theta \rangle, p \rangle$$

Donde $\langle x, y, \theta \rangle$ denota una posición del robot, y $p \geq 0$ es un factor numérico de puntaje (también llamado peso), análogo a la probabilidad discretizada.

Por consistencia se asume que $\sum_{n=1}^N p_n = 1$.

La correspondencia entre los filtros de Bayes y la aproximación utilizando partículas esta dado por:

$$Bel(x_t) \approx \Phi_t$$

Entonces a partir de la ecuación (1) podemos obtener:

$$Bel(X_t^{(i)}) = \frac{p(o_t | X_t^{(i)})}{\sum_{\Phi_t} p(o_t | X_t^{(j)})} \left\{ \sum_{\Phi_{t-1}} p(X_t^{(i)} | X_{t-1}^{(j)}, a_{t-1}) Bel(X_{t-1}^{(j)}) \right\} \quad (2)$$

Definiéndose de esta forma el puntaje como:

$$\omega_t^{(i)} = \frac{p(o_t | X_t^{(i)})}{\sum_{\Phi_t} p(o_t | X_t^{(j)})}$$

Así, el cálculo de $Bel(x_t)$ se reduce a encontrar Φ_t en cada momento, por lo cual es necesario encontrar los valores de todos los $X_t^{(i)}$ y ω_t^i

El método de localización Monte Carlo consta de dos fases [6][4]:

Fase de predicción. En esta primera fase se usa un modelo de movimiento para predecir la posición actual del robot en forma de función de probabilidad de densidad (PDF). Se parte del conjunto de partículas Φ_{t-1} generado en la iteración previa; a cada una de las partículas $X_{t-1}^{(i)}$ se le aplica el modelo de movimiento y con esto se obtiene un nuevo conjunto de muestras Φ'_t a partir de la densidad $p(X_t^{(i)} | X_{t-1}^{(j)}, a_{t-1})$:

(i) Para cada partícula $X_{t-1}^{(i)}$:

Generar una partícula $\tilde{X}_t^{(i)}$ desde $p(X_t^{(i)} | X_{t-1}^{(j)}, a_{t-1})$.

En Φ'_t no se han incorporado ninguna medición de los sensores en el tiempo t .

Fase de actualización. La fase de actualización puede ser dividida a su vez en dos fases, en la primera se toman mediciones o_t desde los sensores, y el puntaje de las muestras en Φ'_t es actualizado con el puntaje $\omega_t^{(i)} = p(o_t | X_t^{(i)})$. En la segunda fase se normalizan el puntaje de cada una de las muestras $\omega_t^{(i)} = \frac{p(o_t | X_t^{(i)})}{\sum_{\Phi_t} p(o_t | X_t^{(j)})}$. La nueva población Φ_t se obtiene por remuestreo a partir de Φ'_t , una vez actualizados los puntajes.

(ii) Para $j = 1..N$:

Generar una muestra $X_t^{(j)}$ a partir de $p(o_t | X_t^{(i)})$.

El remuestreo (resampling) selecciona muestras $\tilde{X}_t^{(i)}$ con alta probabilidad y de esta forma se tiene una probabilidad alta asociada a Φ_t . Un algoritmo eficiente para mejorar el rendimiento del proceso de remuestreo en un tiempo $O(N)$ se puede encontrar en [13].

Después de la fase de actualización los pasos (i) y (ii) son repetidos recursivamente.

Algorítmicamente MCL puede verse como [22]:

1. **Resamplig.** Elegir N nuevos $\tilde{X}_t^{(i)}$ a partir de los $X_{t-1}^{(j)}$, en base a su $\omega_{t-1}^{(j)}$. Algunos $X_{t-1}^{(j)}$ pueden desaparecer y otros multiplicarse.

2. **Samplig.** $X_t^{(i)} = \tilde{X}_t^{(i)} + \Delta X_{t-1}^{(i)}$

Donde $\Delta X_{t-1}^{(i)}$ depende de a_{t-1} y sigue una distribución de probabilidad normal que se deriva de $p(X_t^{(i)} | X_{t-1}^{(j)}, a_{t-1})$.

3. **Importance Samplig.** $\omega_t^{(i)} = \frac{p(o_t | X_t^{(i)})}{\sum_{\Phi_t} p(o_t | X_t^{(j)})}$

Donde o_t corresponde a las observaciones.

MCL representa la probabilidad de densidad para la pose del robot como un conjunto de muestras; dado que cada muestra es un punto exacto en el espacio local, las actualizaciones de las muestras son fáciles de implementar [23][24]. Al utilizar una representación basada en muestras MCL tiene algunas ventajas [6][7]:

1. En contraste con las técnicas basadas en los filtros de Kalman, MCL es capaz de representar distribuciones multi-modales y también tiene la capacidad de localizar al robot cuando no conoce su posición inicial (*global localization*).
2. Reduce drásticamente la cantidad de memoria requerida comparado con la localización Markoviana (Markov localization) y puede integrar medidas con una frecuencia considerablemente alta.
3. Es más exacto que la localización Markoviana ya que la representación de las muestras no está discretizada.

4. Es fácil de implementar.
5. Enfoca los recursos computacionales en las zonas donde es necesario incrementar la resolución.

3.3 Monte Carlo Dual.

Este es una versión alternativa al método Monte Carlo. El principio de este método es invertir el proceso de muestreo. Monte Carlo genera las nuevas partículas según la observación más reciente y luego ajusta los factores de importancia según la creencia anterior $Bel(x_{t-1})$. Por esta razón el algoritmo dual tiene fortalezas y debilidades complementarias a las de Monte Carlo, es ideal para sensores altamente precisos pero es muy sensible al ruido en las mediciones. En esta aproximación, la ecuación recursiva de los filtros de Bayes es interpretada de la siguiente manera:

$$p(X_t^{(i)} | \Phi_{t-1}, a_{t-1}, o_t) = \frac{p(o_t | X_t^{(i)})}{\sum_{\Phi_t} p(o_t | X_t^{(i)})} \quad (3)$$

$$\omega_t^{(i)} = \sum_{\Phi_{t-1}} p(X_t^{(i)} | X_{t-1}^{(j)}, a_{t-1}) \quad (4)$$

En este método, $p(X_t^{(i)} | X_{t-1}^{(j)}, a_{t-1})$ necesita ser calculado explícitamente, a diferencia de Monte Carlo donde sólo se necesita un método de muestreo que siguiera $p(X_t^{(i)} | X_{t-1}^{(j)}, a_{t-1})$. Este cálculo es realizando proyectando $X_t^{(i)}$ a un posible predecesor $\tilde{X}_t^{(i)}$ usando la información de odometría a_{t-1} . Después se calcula $\omega_t^{(i)} = Bel(\tilde{X}_t^{(i)})$

usando Φ_{t-1} . Para llevar la recursión planteada en las ecuaciones (3) y (4), Monte Carlo Dual aplica a cada individuo $X_{t-1}^{(i)}$ el algoritmo recursivo para Monte Carlo [22].

Este método no es de utilidad por sí solo porque es muy ruidoso y no se puede aplicar cuando no hay información observacional. Usado en conjunto con el Monte Carlo se obtiene un método más poderoso llamado Mixure – Monte Carlo.

3.4 Mixure - Monte Carlo.

Este algoritmo pretende heredar las fortalezas de Monte Carlo y Monte Carlo Dual haciendo un algoritmo híbrido. La idea central es que cada muestra sea generada usando uno de los dos algoritmos, siendo aleatoria la decisión acerca de cuál de ellos se usa para generar la muestra. Se usa una taza de mezcla $\phi(0 \leq \phi \leq 1)$ constante, de manera que cada muestra es generada con probabilidad $1 - \phi$ usando Monte Carlo y con probabilidad ϕ usando Monte Carlo Dual [22] [25].

A continuación se muestra el algoritmo recursivo para Monte Carlo Dual.

1. $X_t^{(i)}$ es generado aleatoriamente usando la distribución $\frac{p(o_t | X_t^{(i)})}{\sum_{\Phi_t} p(o_t | X_t^{(i)})}$

2. $\tilde{X}_{t-1}^{(i)} = X_t^{(i)} - \Delta X_{t-1}^{(i)}$

Donde $\Delta X_{t-1}^{(i)}$ sigue una distribución de probabilidades que se deriva de $p(X_t^{(i)} | X_{t-1}^{(i)}, a_{t-1})$.

3. $\omega_t^{(i)} = \sum_{\Phi_{t-1}} p(X_t^{(i)} | X_{t-1}^{(i)}, a_{t-1})$.

3.5 Adaptive – Monte Carlo.

Este método fue desarrollado por el equipo de fútbol robótico de la Universidad de Washington para la RoboCup 2002 [17]. La idea central de este algoritmo es usar una combinación de dos estimadores de la probabilidad observacional \tilde{p} . La probabilidad observacional es definida como la probabilidad promedio de los individuos de la población según el modelo observacional. Esto nos dice que tan bien se ajustan los individuos a los datos obtenidos por la cámara.

$$\tilde{p} = \frac{\sum p(o_t | x_t^{(i)})}{N}$$

Se pueden definir dos estimadores de la probabilidad observacional. El primer estimador \bar{p}_l , es llamado promedio a largo plazo de la probabilidad observacional, y el segundo \bar{p}_s , es llamado promedio a corto plazo de ella. Mientras \bar{p}_l estima el nivel de ruido en el ambiente y los sensores, \bar{p}_s es usado para estimar cambios rápidos en la probabilidad debidos a fallas en la estimación de la posición. El cálculo de los estimadores \bar{p}_l y \bar{p}_s , es recursivo y se actualiza en cada iteración de la siguiente manera:

$$\bar{p}_s' = \bar{p}_s + \eta_s(\tilde{p} - \bar{p}_s)$$

$$\bar{p}_l' = \bar{p}_l + \eta_l(\tilde{p} - \bar{p}_l)$$

La única diferencia entre \bar{p}_l y \bar{p}_s radica en los factores η_l y η_s , $0 \leq \eta_l \ll \eta_s \leq 1$. La diferencia de este método con Mixure – Monte Carlo es que la probabilidad ϕ de usar el algoritmo Monte Carlo Dual es ahora variable [22] y se calcula como:

$$\phi = \max\left(1 - v \frac{\bar{p}_s}{\bar{p}_l}, 0\right)$$

Es importante notar que para que ϕ sea positivo, se debe cumplir que $\bar{p}_l > v\bar{p}_s$. Es decir el parámetro v permite ajustar el umbral para $\frac{\bar{p}_s}{\bar{p}_l}$ sobre el cual se empieza a usar Monte Carlo Dual.

3.6 Conclusiones del capítulo.

En este capítulo se mostraron las bases teóricas del método de localización Monte Carlo. Además, se mostraron las variantes de este método que diferentes autores han propuesto: Monte Carlo Dual, Mixture Monte Carlo y Adaptative Monte Carlo. El estudio mostrado en este capítulo, tiene la finalidad de encontrar los puntos débiles del método Monte Carlo para posteriormente proponer algunas modificaciones de éste y mejorar algunos aspectos del rendimiento del método como reducción en el tiempo de convergencia, utilizar una menor cantidad de muestras y reducir el error promedio en cada una de las variables que definen la pose del robot ($pose = (x, y, \theta)$).