Estado del arte del SLAM

Indice

Estado del arte del SLAM....1
Objetivo del documento....2
Sección I – El problema de SLAM....3
Motivación....3
Objetivos del SLAM....3
Principales desafíos....3
Incertidumbre total....3
Ruido en los sensores....4
Incertidumbre en las medidas odométricas....4
Identificación de las características....4
Cerrado de ciclos (loop-closure)....4
Sección II - Conocimientos previos necesarios....5
Modelos Bayesianos....5
Redes Bayesians 5
Factorización....6
Separación-d....6
Filtro de Kalman....6
Tratamiento de señales....6
Modelado de evolución de Kalman....7
Representación bayesiana....7
Estimador óptimo....8
Dinámica del algoritmo....8
Paso de predicción....8
Paso de actualización....9
Conclusiones....10
Referencias....12

Objetivo del documento

El objetivo de este documento es presentar el problema de localización y confección de mapas de forma simultanea (Simultaneous Location and Mapping – SLAM).

En la primera sección presentaremos el problema del SLAM, sus objetivos y sus principales dificultades. Esta sección tendrá como objetivo introducir el problema en cuestión orientado a un público general sin conocimientos previos de robótica.

Luego, dedicaremos la siguiente sección a la introducción de algunos conceptos importantes para comprender el marco teórico del problema y las soluciones propuestas.

La sección III estará dedicada a la revisión de las principales corrientes de solución del problema del SLAM que conviven actualmente.

Sección I – El problema de SLAM

Motivación

El estudio de posibles soluciones robóticas para algunos problemas como producción industrial, rescates en zonas peligrosas y exploración espacial han detectado la necesidad de robots capaces de navegar de forma autónoma en terrenos inhóspitos.

Para navegar efectivamente por un terreno conocido a partir de un mapa conocido a priori, un robot deberá ubicar en el espacio ciertas características distinguibles y conocidas (llamadas 'features' en la literatura anglosajona). El agente utilizará estas características para determinar su posición relativa en el mapa conocido.

Sin embargo, acorde a la definición de robot autónomo [Ref], un robot deberá basarse principalmente en el conocimiento adquirido durante la ejecución para desempeñar sus tareas. La ventaja de un robot autónomo es que este es capaz de realizar sus tareas en ambientes para los que la construcción de un mapa previamente a la ejecución de las tareas del robot es muy costosa o imposible.

Por lo tanto, la comunidad de investigadores se ha volcado al estudio de algoritmos para la confección de mapas en linea [Ref], es decir, la confección de mapas por parte del robot mientras este realiza sus tareas.

Esta confección de mapas en linea presenta una paradoja que radica en que un agente deberá conocer su posición para poder construir un mapa de su entorno y, como hemos visto, precisa de un mapa de su entorno para poder ubicarse a si mismo.

Objetivos del SLAM

Los algoritmos del SLAM pretenden resolver esta paradoja, mediante el refinamiento iterativo de la posición del robot y de las características distinguibles de su entorno en base a la información sensada y la información interna de control (noción de movimiento propio).

En concreto, este tipo de soluciones pretende obtener una estimación de la posición del robot relativa a un conjunto de características distinguibles del entorno (localización) basándose en información adquirida sobre la ubicación estimada de estas características en el pasado (confección de mapas).

Principales desafíos

La localización y confección de mapas simultanea presenta varios desafios complejos de resolver. A continuación presentaremos los principales problemas a los que se enfrentan los investigadores a la hora de resolver el problema de SLAM.

Incertidumbre total

Como ya se comentó más arriba, la determinación de la posición exacta del agente requiere conocimiento sobre la posición exacta de las características de su entorno, y la determinación de la posición exacta de estas características requiere del conocimiento de la posición exacta del agente.

Dado que el agente no posee inicialmente ninguno de estos datos, y dado que la incertidumbre de su propia posición aumenta con su movimiento (como veremos más adelante), los algoritmos del SLAM deben ser capaces de manejar un cierto error en los datos que son computados.

Esta incertidumbre debe ser manejada de forma tal que el error en las estimaciones no crezca (de forma de evitar una divergencia en la estimación de la posición del agente y de las características de su entorno).

Ruido en los sensores

Las medidas realizadas por los sensores no son precisas, si no que contienen un cierto ruido estocástico que puede representarse como

$$\tilde{z} = z + R$$

donde z corresponde a la medida real y R es el ruido. Este ruido suele ser modelado como sonido blanco (white noise) en el sentido que su valor esperado es 0. En casos particulares como los filtros de Kalman (que veremos más adelante), el ruido es considerado como una variable estocástica de distribución normal centrada en 0 y varianza v.

$$R = N(0, v)$$

Incertidumbre en las medidas odométricas

Debido a las condiciónes potencialmente inhóspitas en los que se desempeñan los agentes, el desplazamiento realmente realizado puede diferir del esperado en función de los comandos emitidos a los sistemas de desplazamiento. Por ejemplo, si la superficie sobre la que se desplaza el robot es resbaladiza puede ocasionar que las ruedas patinen en vez de impulsar al agente provocando que el robot estime que se movió una distancia x cuando en realidad se movió una distancia menor o no consiguió moverse. En algunos casos, este error también es modelado como una variable estocástica de distribución normal con valor esperado 0 y varianza w.

$$x = N(0, w)$$

Identificación de las características

Como hemos explicado, el agente necesita de ciertas características distinguibles del terreno para poder ubicarse a si mismo. Estas características pueden ser artificiales (colocadas artificialmente con el fin de permitir que el robot se ubique) o naturales (que pertenecen naturalmente al paisaje).

Para utilizar la ubicación de estas características, el robot debe poder identificarlas en el entorno. Sin embargo, esto no siempre resulta una tarea simple y es más dificultosa a medida que el error en la estimación de la ubicación de las características de un entorno en particular aumenta debido a que el robot no visita este entorno. Además en la mayoría de los casos la tarea de reconomiento de marcas (naturales, artificiales o amgas) se realiza con el robot en movimiento lo cual introduce aún mas ruido en las lecturas de los sensores.

Cerrado de ciclos (loop-closure)

Cuando un agente retorna nuevamenta a un lugar ya conocido, para el que se confeccionó un mapa del entorno, este deber reconocer que ha retornado a este lugar, evitando catalogarlo como un nuevo entorno.

En ciertos casos, la información de odometría puede contener errores que llevan a un agente a creer que ha llegado a un nuevo entorno cuando está volviendo a uno conocido. Si en estos casos, la información sensada sobre las características distinguibles del entorno no permiten al agente corregir este error, el mapa finalmente construido perderá precisión drásticamente, debido a que el mapa contendrá réplicas diferentes para un mismo lugar, causados por ese desfasaje inicial.

Por lo tanto, es muy importante que los algoritmos del SLAM sean capaces de reconocer cuando han llegado a un entorno conocido, realizando un buen cerrado de los ciclos en el mapa.

Sección II - Conocimientos previos necesarios

Actualmente las soluciones del SLAM se concentran en dos modelos principales:

- Bayesianos
- BioInspirados

A fin de poder comprender a fondo los modelos probabilistas bayesianos vamos a introducir a continuación algunos conceptos relacionados con estos modelos.

Modelos Bayesianos

Redes Bayesianas

Las redes bayesianas son una representación gráfica de la interdependencia de variables estocásticas.

Una red bayesiana consta de un grafo dirigido acíclico en donde los nodos representan variables estocásticas y las aristas representan relaciones de dependencia entre las variables. Esto quiere decir que una arista entre un nodo X y otro nodo Y significa que las variables X e Y no son independientes, o de otra forma:

$$p(X \vee Y) \neq p(X).\, p(Y)$$

Estas redes son utilizadas para modelar sistemas de interdependencia probabilisticas y comunicarlo de manera clara. Por ejemplo, el diagrama en la Fig. 1, representa una relación causa efecto entre el posible hundimiento de un barco y la presencia de una tormenta o un desperfecto técnico. A su vez, se muestran algunos factores que inciden en estas posibles causas de tormentas y desperfectos técnicos.

Es importante notar que la ausencia de aristas entre dos nodos indica que las variables representadas por estos nodos son independientes.

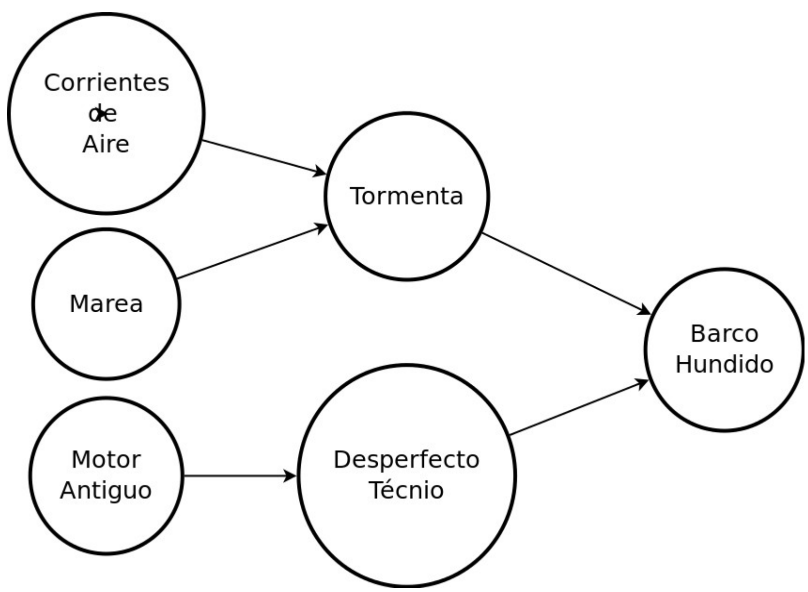

Fig. 1

Factorización

Una definición más rigurosa de las redes bayesianas establece que un vector estocástico X de N variables es una red bayesiana con respecto a un grafo G si

$$p(X=x)=\prod_{1}^{N} p(x_i | x_{pa}(i))$$

donde pa(i) representa a los los padres del nodo i en el grafo G.

Por lo tanto, la probabilidad de BarcoHundido puede calcularse como

$$p(BarcoHundido)=p(BarcoHundido|Tormenta, DesperfectoTecnico)$$
$$*\, p(Tormenta|CorrientesDeAire, Marea)$$
$$*\, p(DesperfectoTécnico|MotorAntiguo)$$

Separación-d

Este modelo permite también establecer pautas sobre la independencia entre variables estocásticas dado que los valores de otras variables relacionadas son conocidos.

Para esto, se define el concepto de markov-blanquet de un nodo X como el conjunto de nodos del grafo que son padres de X o hijos de X o padres de los hijos de X. Por ejemplo, el markov-blanquet de la variable BarcoHundido esta conformado por DesperfectoTecnico y Tormenta, mientras que el de Tormenta está conformado por BarcoHundido, DesperfectoTecnico, Marea y CorrientesDeAire.

Luego, establecemos que una variable estocástica será independiente de todas las demás si se conocen los valores de todas las variables de su markov-blanquet. Ejemplificando:

$$p(BarcoHundido|Tormenta, DesperfectoTecnico, Marea, CorrientesDeAire, MotorAntiguo)$$
$$=p(BarcoHundido|Tormenta, DesperfectoTecnico)$$

pues Tormenta y DesperfectoTecnico conforman su markov-blanquet. Esto resulta bastante coherente tomando en cuenta nuestro modelo simplificado, pues es razonable que el hecho de poseer un motor antiguo no afecte la probabilidad de sufrir un naufragio, si por ejemplo ya sabemos que este no será causado por un desperfecto técnico.

Filtro de Kalman

Tratamiento de señales

De la literatura del procesamiento de señales con ruido es que viene el término de filtro. Estas herramientas intentan solucionar el problema de estimar una señal original x(t) cuando solo es posible observar

$$y(t)=x(t)+r(t)$$

donde r(t) es una señal de ruido no conocida.

Muchas soluciones a este problema buscan encontrar el x*(t) óptimo tal de maximizar la probabilidad

$$p(x(t)|y(0)..y(h))$$

El nombre que recibe el problema a tratar varía según la relación entre h (los datos a tomar en cuenta) y t (el momento en el que se desea estimar el valor de la señal). Se diferencian tres grandes grupos de problemas

- predicciones: estimar x(t) con datos estrictamente anteriores (h < t)
- filtros: estimar x(t) con datos conocidos hasta el momento (h = t)
- interpolación: estimar x(t) con datos estrictamente posteriores (h > t)

Nosotros nos concentraremos en los filtros pues son los que luego aplicaremos en la soluciones del problema del SLAM.

Modelado de evolución de Kalman

Para la solución de este problema, Kalman propone un esquema particular [1] que luego resultará muy propicio para la solución del problema de localización y construcción de mapas en simultaneo. Kalman elige representar a la componente oculta como un vector que representan de manera completa el estado de un sistema.

El esquema propuesto establece una relación lineal en la evolución del sistema

$$x(t)=F.x(t-1)+r(t)$$

donde x(t) es un vector que representa el estado del sistema, F es una matriz que representa la función de transición de un estado a otro y r(t) representa un ruido en la transición del sistema.

A su vez las observaciones son modeladas como una función lineal del estado y un ruido adicional

$$y(t)=H.x(t)+w(t)$$

donde H es una matriz que representa la función que determina la observación dado el estado y w(t) es el ruido asociado a este proceso.

Representación bayesiana

Una gran ventaja de este modelo es su recursividad. Dado que el valor del estado depende tan solo del estado anterior se torna innecesario utilizar los valores anteriores del estado, disminuyendo el orden computacional del cálculo y el espacio necesario para correr el algoritmo de estimación.Si además observamos que el tiempo es tratado en forma discreta, podemos modelar la evolución del sistema según la red bayesiana de la Fig. 2.

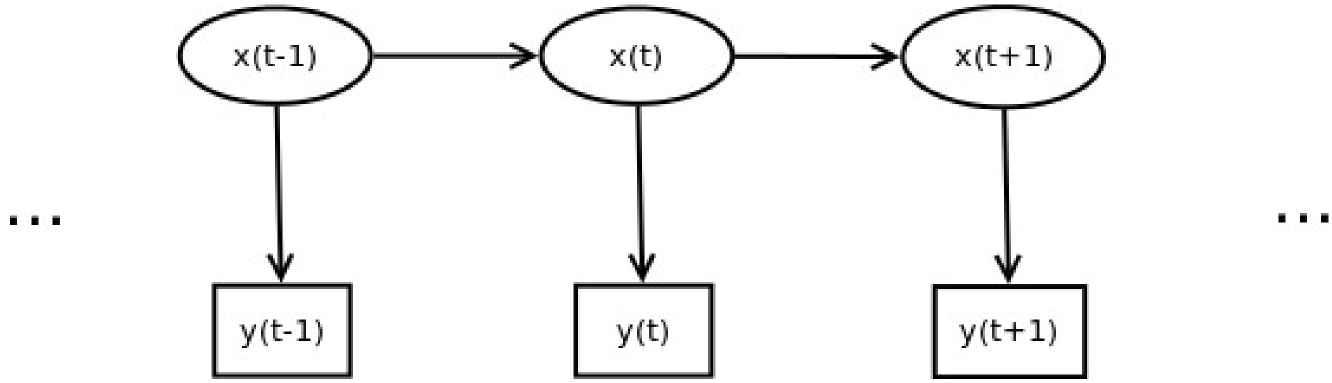

Fig. 2

En esta red bayesiana se puede observar un hecho adicional que está implicito en las ecuaciones de actualización. Las observaciones realizadas serán independientes entre sí. Esto es importante pues es utilizado en las derivaciones de las formulas de actualización del estado.

También podemos observar como dado el valor de un estado del sistema x(t), los valores posteriores del sistema serán independientes de todos los valores de estado y observaciones previas, reafirmando lo establecido sobre la recursividad de este modelo.

Estimador óptimo

Presentado el modelo de evolución del sistema, solo resta encontrar un estimador que nos permita obtener un valor del sistema en un tiempo t dado que sea óptimo según algún criterio. Este criterio podría ser minimizar el costo computacional contemplando un error dado, o minimizar la suma del cuadrado del error en la estimación.

Este último criterio es precisamente el que utiliza el filtro de Kalman. Si suponemos que en el timpo t+1 tenemos un estimador no sesgado del estado en el tiempo t ($\tilde{x}(t)$), podemos definir el error como

$$e(t)=x(t)-\tilde{x}(t)$$

Este error es en sí una variable estocástica que tendrá valor esperado nulo debido a que el estimador $\tilde{x}(t)$ es un estimador sin sezgo.

$$E(\tilde{x}(t)-x(t))=E(e(t))=0$$

Definidos estas variables, el filtro intentará encontrar un estimador no sezgado del sistema para el tiempo t + 1 tal que presente el menor error cuadrático de estimación. Es decir que se buscará un $\tilde{x}(t+1)$ tal que

$$E(\tilde{x}(t+1)-x(t+1))=E(e(t+1))=0$$

$$E((\tilde{x}(t+1)-x(t+1))^2)=E(e(t+1)^2)=E(e(t+1)^2-E(e(t+1)))=var(e(t+1))$$

Dado que el $E(e(t+1))$ es nulo, encontramos que minimizar el error cuadrático es igual a encontrar un estimador tal que tenga la menor varianza posible.

Dinámica del algoritmo

Para encontrar el estimador no sezgado de menor varianza, el algoritmo propuesto realiza dos pasos, correspondientes a cada una de las ecuaciones que describen la evolución del sistema:

- En primera instancia se realiza un paso de predicción o extrapolación del estado del sistema para el tiempo t+1.
- Luego, la información sensada es utilizada para hacer una actualización del estimado del estado del sistema.

Además, en cada uno de los pasos se actualiza la varianza del estimador.

En esta sección omitiremos la derivación de las fórmulas aplicables a cada paso pero pasaremos a dar pautas generales de los principios que los subyacen. Para una explicación detallada de las derivaciones de las fórmulas puede leerse [2], y para una explicación conceptual de las ideas que se manejan en el filtro puede leerse [3].

Paso de predicción

En el paso de predicción, el estado estimado del sistema en el tiempo t es utilizada para estimar el estado en el tiempo t + 1.

Dado que el sistema evoluciona de forma lineal, esta estimación se realiza calculando

$$x(t+1)=F.x(t)$$

La varianza de este estimador se representa por la matriz P(t). Esta varianza es actualizada según la ecuación:

$$P(t+1)=F\,P(t)\,F^t+Q$$

donde Q es la matriz que representa la varianza del ruido r(t) explicado en el modelo. Esta ecuación puede ser entendida como la suma de dos varianzas

$$var(r(t))=Q$$

$$var(F.x(t))=F.var(x(t)).\,F^t=F.P(t).\,F^t$$

Es importante observar que en este paso la varianza del estimador aumenta con respecto al estimador antes del paso de predicción. Esto es razonable, pues la incertidumbre aumenta en la medida que el sistema cambia debido al ruido presente en el proceso de transición. Si este ruido no existiera, en el mejor de los casos, la varianza P no crecería con el tiempo, pero tampoco disminuiría.

Dado que nuestra incertidumbre del sistema aumenta en el paso de predicción, es de esperar que disminuya en el paso siguiente para que nuestro estimador no diverja con el paso del tiempo. A continuación mostraremos que esto efectivamente ocurre.

Paso de actualización

Durante el paso de actualización, la información sensada es comparada con la predicción de la información a sensar basada en el estado actual del sistema. Por lo tanto calcularemos, en primer lugar, esta diferencia

$$\tilde{y}(t)=z(t)-H.x(t)$$

donde z(t) es la observación efectuada en el tiempo t y H es la matriz que relaciona el estado del sistema y las observaciones realizadas.

De manera similar podemos expresar la varianza de este estimador como

$$S(t+1)=H.P(t).H^t+Q$$

donde Q representa la matriz del ruido w(t) presentado con el modelo.

Luego calcularemos la ganancia de información asociada a esta diferencia. El concepto de ganancia representa una medida de el aporte de información que realizan estos nuevos datos sensados. También puede verse como una relación entre la confianza en la información sensada y la información derivada de la etapa de predicción.

Esta ganancia se calcula

$$K(t+1)=P(t).H(t)^t.S(t)^{-1}$$

Finalmente, se tienen todos los datos para realizar el paso de actualización

$$x(t+1)=x(t+1)+K(t+1).\tilde{y}(t)$$

Al observar esta ecuación se puede arribar a algunas conclusiones preliminares

- a menor ganancia (K), habrá un menor cambio en el estimado del sistema con respecto a lo extrapolado en el paso anterior
- si la diferencia $\tilde{y}(t)$ (también denominada innovación) es muy pequeña, el cambio en el sistema no será grande
- una alta varianza P(t) para el paso anterior, que implica una alta incertidumbre en el valor estimado, implicará una gran ganancia que determinará una actualización más significativa
- una alta varianza en las observaciones realizadas (Q), que implica una alta incertidumbre en los datos sensados, determinará una varianza S grande y por lo tanto un K pequeño, haciendo que la actualización no realice grandes cambios en el valor extrapolado en el paso anterior.

Luego resta calcular la nueva covarianza del estimador

$$P(t+1)=P(t+1)+K(t+1).H(t+1)P(t+1)$$

Conclusiones

Luego de todo lo expuesto, tenemos un algoritmo que nos permite actualizar de manera iterativa la estimación de un sistema dado bajo las hipotesis (bastante fuertes) de que su transformación es lineal con el tiempo y también lo son las observaciones con respecto al estado del sistema.

Si nosotros modelamos la posición de un robot y la de todas las características distinguibles del mapa como el estado del sistema, la aplicación del filtro de Kalman iterativo nos proporciona un método de realizar localización (actulización de la posición del robot) y confeccion de mapas (actualización de la posición de las características) de forma simultanea.

Nota para los tutores:

Este documento no pretener ser una primera versión, pero sí muestra el rumbo iniciado por el equipo en lo que refiere al estado del arte y como presentarlo.

Algunas de las tareas pendientes para seguir desarrollando el documento son :

Agregar que el ruido es gausiano

- Agregar que la varianza puede verse como la incertidumbre
- Agregar derivación de formulas
- Agregar apendice de probabilidades en vectores
- Agregar cuestion de error de observaciones esta ligado
- Cambiar x(t+1) del update por x-(t+1)
- Kalman extendido
- Costo computacional de kalman

Referencias

[1] R.E. Kalman, “A New Approach to Linear Filtering and Prediction Problems”
[2] “Kalman Filter Derivation”, http://www.aticourses.com/kalman_filter.pdf
[3] Bradley Hiebert - Treuer, “An Introduction to Robot SLAM (Simultaneous Localization And Mapping)”, http://dspace.nitle.org/bitstream/handle/10090/782/s10csi2007hieberttreuer.pdf?sequence=1