

APLICAÇÃO DE CONJUNTOS ALEATÓRIOS FINITOS NO PROBLEMA DE LOCALIZAÇÃO E MAPEAMENTO DE ROBÔS MÓVEIS

KLEBERSON M. LIMA*, DIEGO PEREIRA-DIAS*, RAMON R. COSTA*

*Universidade Federal do Rio de Janeiro (UFRJ)
Programa de Engenharia Elétrica (PEE/COPPE/UFRJ)
Rio de Janeiro, Rio de Janeiro, Brasil

Emails: kleberonmlima@outlook.com, dpd@metalmat.ufrj.br, ramon@coep.ufrj.br

Abstract— This paper presents an overview on Random Finite Set (RFS) applied to the problem of mapping in robotics. A solution is presented for the RFS-based mapping problem. Initially, this type of approach is presented and the use of its advantages over vector-based techniques is motivated. Conflicts that occur in construction maps are shown when using conventional approaches, such as: maps dependent on the trajectory traversed and the problem of data association. Once justified the use of RFS in the problem of mapping and location of robots, the mathematical framework involved is presented and the main concepts pertinent to this formalism are defined. Afterwards, the RFS is presented within the Bayesian framework, the PHD filter. Finally, through simulations, an application to obtain a map of the environment using the Probability Hypothesis Density (PHD) filter is shown.

Keywords— Robotics, mapping, Random Finite Sets.

Resumo— Este trabalho apresenta uma visão geral sobre Conjuntos Aleatórios Finitos (do inglês *Random Finite Sets*) aplicado ao problema de mapeamento em robótica. Além disso, aplica-se uma solução baseada em RFS para o problema de mapeamento FBRM (*Feature-Based Robot Mapping*). Inicialmente, faz-se uma apresentação desse tipo de abordagem e motiva-se a utilização dessa técnica através de suas vantagens em relação às vetoriais. São mostrados conflitos que acontecem na construção de mapas, quando do uso das abordagens convencionais, tais como: mapas dependentes da trajetória percorrida e o problema da associação de dados. Uma vez que justificado o uso de RFS no problema de mapeamento e localização de robôs, apresenta-se o arcabouço matemático envolvido e define-se os principais conceitos pertinentes a esse formalismo. Em seguida, apresenta-se o RFS dentro do *framework* Bayesiano, o filtro PHD, do inglês *Probability Hypothesis Density*. Por fim, através de simulações, mostra-se uma aplicação para obtenção de um mapa do ambiente com uso do filtro PHD.

Palavras-chave— robótica, mapeamento, Random Finite Sets.

1 Introdução

A importância da robótica na nossa vida cotidiana tem crescido nos últimos anos. Tornou-se frequente, ao se abrir um portal de notícias na *internet*, encontrar manchetes sobre empresas que desejam distribuir seus produtos vendidos *on-line* por meio de um *drone*, robôs utilizados em limpeza doméstica ou sobre sistemas de transportes que operam sem a interferência humana. A autonomia é o ponto comum entre elas.

Quando o mapa já é conhecido e o robô necessita saber (ou estimar) onde se encontra, através de seus diversos sensores, tem-se o que é chamado de **localização**. Uma vez que os processos de localização e mapeamento são solucionados simultaneamente, tem-se o que se chama **SLAM** (*Simultaneous Localization And Mapping*). Segundo (Mullane et al., 2011a), a representação computacional, e incerteza, desses processos é crucial para a navegação autônoma.

Para realizar essas tarefas, o sistema autônomo necessita obter informações sobre o mundo ao seu redor para que possa fornecer subsídios para os sistemas de guiagem do robô. A tarefa de construção de um modelo do ambiente é chamada de **mapeamento**.

O problema de SLAM surgiu na década de 90 e até hoje apresenta-se como um dos maiores desafios em robótica móvel e sistemas autônomos. Tradicionalmente, são utilizadas três técnicas para se resolver esse problema: filtros de Kalman, filtros de partículas e as baseadas em teoria dos grafos (Siciliano and Kha-

tib, 2016).s

As técnicas baseadas em filtros de Kalman e de partículas são as mais amplamente exploradas e aplicadas (Zhang et al., 2017). Apesar dessas técnicas serem bem estabelecidas, elas apresentam uma dificuldade extra em aplicações reais: a associação de dados. De acordo com Siciliano and Khatib (2016), esta é uma dificuldade bastante complexa que deve ser tratada à parte do problema de SLAM, mas que pode afetar diretamente o desempenho da solução. Logo, soluções alternativas, que lidem de maneira mais simples com a essa questão da associação de dados, são de interesse para os problemas de localização e mapeamento.

Nesse contexto, a teoria dos conjuntos aleatórios finitos (do inglês, *Random Finite Sets* - RFS) vem sendo estudada e, de acordo com (Mullane et al., 2011a), é uma ferramenta mais apropriada para a solução do problema de mapeamento (ou SLAM) quando comparada às técnicas convencionais.

Este trabalho apresenta um estudo sobre a aplicação da técnica RFS no problema de mapeamento. A contribuição deste trabalho é fomentar o estudo das técnicas baseadas em RFS em nossa comunidade, através de um caso em ambiente simulado. O trabalho está organizado da seguinte forma: na Seção 2 resume-se o problema de localização, na Seção 3 apresenta-se o ferramental teórico dos RFS, na Seção 4 descrevem-se os resultados obtidos e na Seção 5 são expostas as conclusões e comentários finais.

2 Formulação do Problema

Localização é o problema de descobrir onde os objetos estão, inclusive o próprio robô. O conhecimento disso é o cerne de qualquer interação física bem sucedida com o meio ambiente (Russel and Norvig, 2010).

Segundo (Thrun et al., 2005), **localização**, em **robótica móvel**, é o problema de se determinar a *pose* de um robô em relação a um mapa do ambiente. Para o caso de um robô móvel que se move no plano, temos que a *pose*, em um dado instante t , é definida pelas coordenadas cartesianas e a orientação da parte frontal, ou seja, $x_t = [x \ y \ \theta]^T$.

A localização pode ser vista como um problema de transformação de coordenadas, com o mapa sendo o sistema de coordenadas global. Portanto, a localização é o processo de estabelecer uma relação entre as coordenadas globais e as locais do robô. Contudo, de acordo com (Thrun et al., 2005), em geral a *pose* não pode ser medida diretamente. Ou seja, a maioria dos robôs não possuem um sensor para medi-la. Logo, a *pose* deve ser inferida a partir de dados (por exemplo, por meio de um sensor de distância).

Por conseguinte, no problema de localização, o objetivo é estimar a posição do robô dados: o mapa do ambiente M , as observações z_t e a ação de controle u_t .

Segundo (Thrun et al., 2005), o **mapa** é uma lista de objetos com as respectivas localizações em um ambiente e é definido formalmente por:

$$m = \{m_1, m_2, \dots, m_N\}, \quad (1)$$

onde N é o número total de objetos no ambiente e cada m_i , com $i = 1 : N$, representa as propriedades ou características de um objeto.

Há duas abordagens para a construção de mapas: as baseadas em características (*feature-based*) e as em localização (*location-based*). Em consonância com (Thrun et al., 2005), mapas baseados em localização são volumétricos e carregam a informação não apenas sobre os espaços ocupados por objetos mas também por espaços vazios. No caso planar $m_i = m_{x,y}$, o que evidencia tratar-se de uma coordenada (x,y) . Já nas abordagens baseadas em características, o mapa define a localização de cada objeto e uma (ou mais) característica. Esse estudo concentra-se na abordagem baseada em características (*Feature Based Robotic Mapping – FBRM*).

De acordo com (Thrun et al., 2005), um vetor de características é definido por:

$$f(z_t) = \{f_t^1, f_t^2, \dots\} = \{(r_t^1, \phi_t^1, s_t^1), (r_t^2, \phi_t^2, s_t^2), \dots\},$$

onde r é a distância para a origem de um sistema de coordenadas, ϕ é a orientação em relação à mesma origem e s é a assinatura, a qual pode ser um valor numérico (ex: para caracterizar uma cor) ou um valor multidimensional (ex: altura e cor).

Resumidamente, o problema de mapeamento caracteriza-se por estimar o mapa m através da distribuição de probabilidade $p(m | z_t, x_t)$. Já o

problema de SLAM, caracteriza-se por ser uma combinação dos dois problemas, portanto, estimar o mapa e a localização do robô simultaneamente.

3 Conjuntos Aleatórios Finitos

De acordo com (Mullane et al., 2011a), as técnicas convencionais (baseadas em vetores) utilizadas em FBRM e SLAM sofrem dificuldades quando aplicadas em situações realísticas. Isso acontece em ambientes nos quais um número a priori desconhecido de características deve ser estimado e, na presença de defeitos realistas dos sensores, como detecções não atendidas e falsos alarmes.

Um RFS é uma variável aleatória que toma valores como conjuntos finitos. É definido por uma distribuição discreta que caracteriza o número de elementos no conjunto e uma família de distribuições conjuntas que caracterizam a distribuição dos valores dos elementos, condicionada à cardinalidade (Dames, 2015):

$$p(X) = p(|X| = n)p(X = \{x_1, x_2, \dots, x_n\} | |X| = n). \quad (2)$$

Um exemplo de RFS pode ser visualizado na Figura 1.

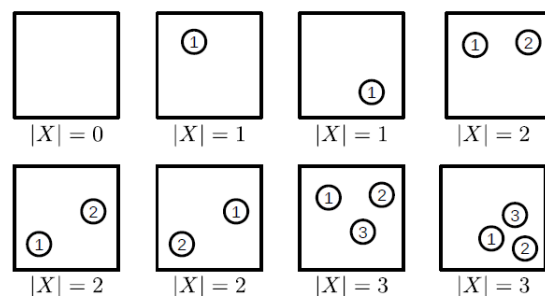


Figura 1: Exemplos de um RFS de 0 a 3 elementos em um ambiente quadrado.

Fonte: Extraído de (Mullane et al., 2011a).

O problema de rastreamento multi-alvos utilizando RFS já é amplamente difundido na comunidade de radares e, recentemente, vem sendo aplicado em robótica móvel, de acordo com (Dames, 2015).

RFS x Técnicas Baseadas em Vetores

Segundo (Dames, 2015), é preciso destacar as diferenças entre as técnicas baseadas em vetores e em conjuntos quando se considera os problemas de FBRM e SLAM (o qual pode ser entendido como um problema *multi-target* quando os alvos são estáticos). Nesse tipo de problema, a quantidade de características (*landmarks*) não é conhecida *a priori* e deve ser descoberta à medida que o robô explora o ambiente. Dois pontos principais, que serão explicados a seguir, surgem nesse tipo de problema:

- Gerenciamento das características (*feature management*) - estimar a posição dos objetos/características no mapa;
- Associação de dados (*data association*) - atrelar as observações aos objetos/características.

Observa-se, na Figura 2, o cenário hipotético em que um robô percorre três trajetórias distintas X_1 , X_2 e X_3 nas vizinhanças dos objetos m_1, \dots, m_7 .

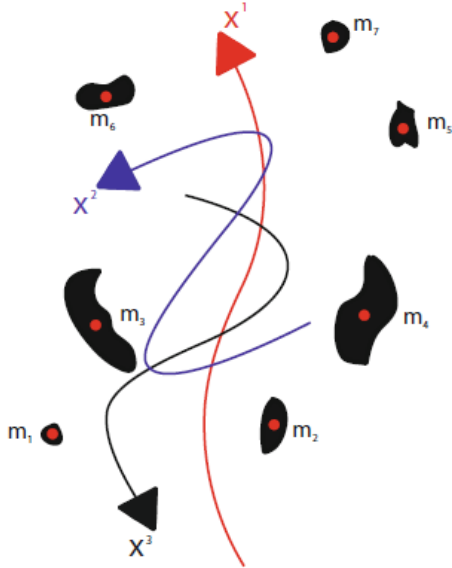


Figura 2: Três trajetórias realizadas por um robô em um cenário hipotético.
Fonte: Extraído de Mullane et al. (2011a).

Caso seja usado um vetor M para representar o mapa estimado, para cada trajetória, teríamos $\hat{M}_1 = [m_1 \ m_2 \ m_3 \ m_4 \ m_5 \ m_6 \ m_7]^T$, $\hat{M}_2 = [m_4 \ m_2 \ m_3 \ m_1 \ m_5 \ m_7 \ m_6]^T$ e $\hat{M}_3 = [m_6 \ m_7 \ m_5 \ m_4 \ m_3 \ m_2 \ m_1]^T$. Como a ordem dos elementos num vetor é importante, ter-se-ia três mapas diferentes, destacando-se o fato de que o mapeamento seria dependente da trajetória.

Contudo, no sentido lógico, o mapeamento deveria ser independente da trajetória percorrida pelo veículo e, matematicamente, qualquer permutação nos elementos dos vetores seria uma representação válida. Por definição, a representação que captura todas as permutações dos elementos dentro de um vetor (e, por conseguinte, todas as características do mapa) é o conjunto finito $\hat{\mathcal{M}} = \{m_1 \ m_2 \ m_3 \ m_4 \ m_5 \ m_6 \ m_7\}$ (Mullane et al., 2011a).

Uma outra motivação para o uso de RFS é o problema de associação entre as observações z e as respectivas características m , como pode ser visualizado na Figura 3. As técnicas de FBRM e SLAM baseadas em vetores exigem um passo de reordenação das observações antes que um filtro Bayesiano (Kalman, EKF, partículas etc.) possa ser aplicado (Mullane et al., 2011a).

Por outro lado, utilizando-se de abordagem baseada em RFS, é possível fazer uma associação di-

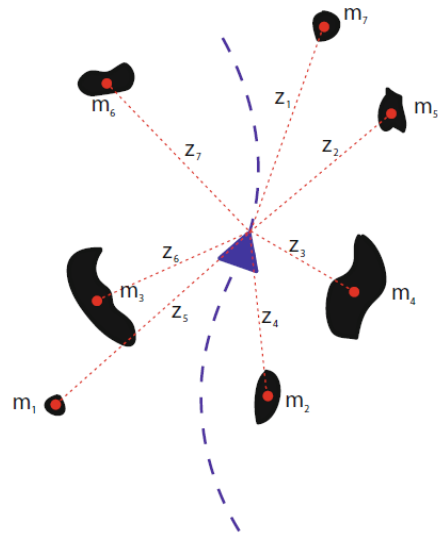


Figura 3: Vetores das observações diferentes a depender da trajetória.
Fonte: Extraído de Mullane et al. (2011a).

reta entre o conjunto \mathcal{M} das características e o \mathcal{Z} das observações.

Um outro problema contornável pelo uso dos RFSs, quando comparado às técnicas baseadas em vetores, é o devido ao uso de sensores não ideais associado ao problema de crescimento do mapa por detecções falsas ou objetos perdidos (fora do campo de detecção). Esse tipo de problema pode ser exemplificado a partir da Figura 4.

Supondo que as características m_1 , m_2 e m_3 já tenham sido detectadas no instante $k-1$, em uma representação baseada em vetores, e que no instante k uma nova característica m_4 é detectada, ter-se-ia:

$$\hat{M}_{k-1} = [m_1 \ m_2 \ m_3]^T \quad (3)$$

e

$$\hat{M}_k = [m_1 \ m_2 \ m_3]^T \text{ “+” } m_4. \quad (4)$$

Segundo (Mullane et al., 2011a), a operação de atribuição de m_4 não é clara do ponto de vista matemático. E, mesmo existindo técnicas que usem um vetor aumentado, com o uso de RFS, bastaria usar a operação de união entre os conjuntos nos instantes $k-1$ e k :

$$\hat{\mathcal{M}}_k = \{m_1 \ m_2 \ m_3\} \cup \{m_4\}. \quad (5)$$

Um outro problema fundamental de qualquer FBRM ou SLAM é a necessidade de se relacionar as observações ao estado estimado (Mullane et al., 2011a).

$$Z_k = h([m_1 \ m_2 \ m_3 \ m_4], X_k) + \text{ruído} \quad (6a)$$

$$\text{i.e. } [z_1 \ z_2 \ z_3 \ z_4 \ z_5] = h([m_1 \ m_2 \ m_3 \ m_4], X_k) + \text{ruído}, \quad (6b)$$

onde Z_k representa o vetor de observações no tempo k (aqui se usa o exemplo da Figura 4), X_k é a **pose** do

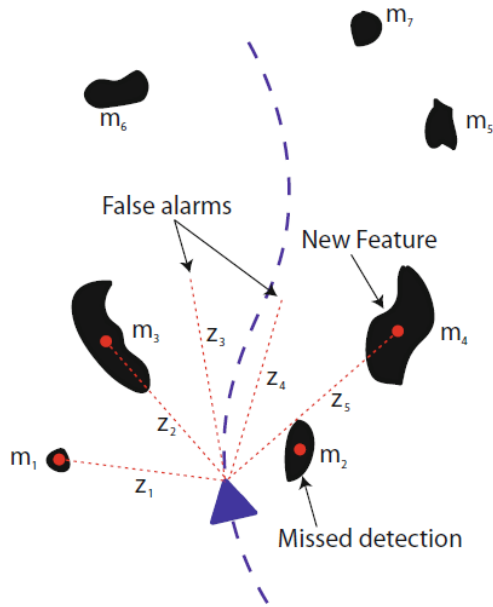


Figura 4: Detecção perdida com sensor realista (não ideal).

Fonte: Extraído de Mullane et al. (2011a).

robô no instante k e h é o mapeamento (tipicamente não linear) entre a pose do robô e as observações. Esse exemplo mostra a dificuldade encontrada para mapear as quatro características (estados) com cinco observações. A observação extra é, possivelmente, devido a uma detecção espúria e a uma característica que foi “perdida”. Numa abordagem convencional, é indefinida a forma como integrar ao equacionamento esses dois “problemas”. É preciso um gerenciamento heurístico do mapa para remover estados em excesso e forçar o funcionamento do mapeamento.

Caso sejam utilizados conjuntos para representar as observações e mapa estimado, tem-se a seguinte relação:

$$\mathcal{Z}_k = \bigcup_{m \in \mathcal{M}_k} \mathcal{D}_k(m, X_k) \cup \mathcal{C}_k(X_k), \quad (7)$$

onde $\mathcal{D}_k(m, X_k)$ é o RFS das medições geradas por uma característica em m e dependente da pose X_k . Além disso, $\mathcal{C}_k(X_k)$ é o RFS das medições espúrias que poderia ser em função da pose. Portanto, $\mathcal{Z}_k = \{z_k^1, z_k^2, \dots, z_k^{z_k}\}$ consiste de um número aleatório, z_k , de medições cuja ordem em que aparecem não tem um significado físico com relação à estimativa das características no mapa. Então, de acordo com (Mullane et al., 2011a), fica claro que, nas técnicas baseadas em vetores, a incerteza no número de características não é modelada diferente de quando se usa RFS (a incerteza é algo intrínseco à técnica).

4 Conceitos Fundamentais em RFS

Segundo (Dames, 2015), apesar de todas as vantagens sobre as abordagens baseadas em vetores, com RFS

há a dificuldade de tratar com matemática não familiar para a maioria dos profissionais e acadêmicos que lidam com robótica. Portanto, para ser possível realizar uma inferência estatística com conjuntos, é preciso definir variáveis aleatórias adequadas e ser capaz de realizar operações como o cálculo de médias dessas variáveis.

4.1 Random Finite Set

De acordo com (Ristic et al., 2016), um RFS é um modelo probabilístico conveniente para a representação de múltiplos sistemas dinâmicos estocástico (objetos ou características) e medições dos sensores.

Suponha que, no instante k , exista n_k objetos com estados $x_{k,1}, \dots, x_{k,n_k}$ assumindo valores do espaço de estados $\mathcal{X} \subseteq \mathcal{R}^{m_k}$. Tanto n_k quanto o número de estados individuais em \mathcal{X} são variáveis aleatórias e variantes no tempo. O estado das múltiplas características, no instante k , é um **estado finito** definido por:

$$\mathbf{X}_k = \{x_{k,1}, \dots, x_{k,n_k}\} \in \mathcal{F}(\mathcal{X}), \quad (8)$$

o qual pode ser modelado como um RFS em \mathcal{X} . Onde $\mathcal{F}(\mathcal{X})$ é o conjunto finito de todos os subconjuntos de \mathcal{X} .

De maneira similar, para as observações teremos:

$$\mathbf{Z}_k = \{z_{k,1}, \dots, z_{k,m_k}\} \in \mathcal{F}(\mathcal{Z}),$$

pode ser modelado como um RFS sobre o estado de observações $\mathcal{Z} \subseteq \mathcal{R}^{m_k}$. Tanto a cardinalidade $|\mathbf{Z}_k|$ quanto os estados individuais em \mathbf{Z}_k são aleatórios e $\mathcal{F}(\mathcal{Z})$ é o conjunto de todos os subconjuntos de \mathcal{Z} .

Um RFS é totalmente caracterizado por sua distribuição de cardinalidade $p(n) = p(|X| = n)$, com $n \in \mathbb{N}$ e por uma família de distribuições conjuntas simétricas $p_n(x_1, \dots, x_n) = p(X = \{x_1, \dots, x_n\} | |X| = n)$ (vide (2)).

4.2 Set integral

Seja $f(\mathbf{X})$ uma função real sobre um conjunto, a *set integral* de $f(\mathbf{X})$ é definida por:

$$\int f(\mathbf{X}) \delta \mathbf{X} = \sum_{n=0}^{\infty} \frac{1}{n!} \int f(\{x_1, \dots, x_n\}) dx_1 \dots dx_n. \quad (9)$$

Essa integral caracteriza a soma sobre a cardinalidade de um conjunto, integrando todos os conjuntos possíveis de acordo com cardinalidade. Além disso, o termo $1/n!$ leva em conta as permutações de um conjunto de tamanho n .

O interesse é quando $f(\mathbf{X})$ é um distribuição de probabilidade!

4.3 Distribuições de Probabilidade Sobre um RFS

Considerando a hipótese de que os elementos do RFS são independentes e identicamente distribuídos

(mesma distribuição de probabilidade), a probabilidade de um RFS é dada por:

$$p(\mathbf{X}) = |\mathbf{X}|! p(|\mathbf{X}|) \prod_{x \in \mathbf{X}} p(X), \quad (10)$$

onde $|\cdot|$ é o operador cardinalidade, $|\mathbf{X}|!$ é o número de permutações dos elementos do conjunto, $p(|\mathbf{X}|)$ é a distribuição de cardinalidade e $p(X)$ é a probabilidade de uma característica (*landmark*) ter o estado x . Para $p(X)$, (10) ser uma distribuição de probabilidade válida, é necessário que a *set integral* seja unitária, a verificação pode ser obtida em (Ristic et al., 2016).

4.4 Probability Hypothesis Density (PHD)

A função intensidade (ou simplesmente PHD) é importante na caracterização do primeiro momento estatístico de uma RFS, segundo (Ristic et al., 2016). Inicialmente, é preciso definir a função *Set Dirac Delta*:

$$\delta_X(X) = \sum_{w \in X} \delta_w(x), \quad (11)$$

onde $\delta_w(x)$ é a função delta de Dirac *standard* concentrada em w . Pode-se, então, definir a cardinalidade de um RFS como:

$$|\mathbf{X}| = \int_{\mathcal{X}} \delta_X dx \quad (12)$$

A função intensidade $D(x)$ pode ser definida de tal forma que a cardinalidade esperada de \mathbf{X} sobre \mathcal{X} é a integral:

$$\{|\mathbf{X}|\} = \int_{\mathcal{X}} D(X) dx. \quad (13)$$

Desenvolvendo a integral é possível obter (Ristic et al., 2016),

$$D_X(x) = \{\delta_X(X)\} = \int_{\mathcal{X}} \delta_X(x) f(X) dx. \quad (14)$$

O valor esperado da PHD corresponde ao número de características (*landmarks*) de uma região e a PHD não se trata de uma distribuição de probabilidade (a área sob a curva não é igual a 1) (Dames, 2015).

4.5 Filtro PHD

A abordagem mais básica usada para estimação com RFS é a qual atualiza a média recursivamente (Dames, 2015). Além disso, é o análogo do Filtro de Kalman, contudo, usa uma distribuição de Poisson, que é definida por

$$p(X) = e^{-\lambda} \prod_{x \in \mathcal{X}} v(x), \quad (15)$$

onde $\lambda = \int v(x) dx$.

Mostra-se o problema do ponto de vista *bayesiano*, o qual possui formulação similar aos filtros bayesianos para o caso vetorial. Ou seja, o problema é resolvido recursivamente e é dividido em duas partes: predição (através do modelo de transição de estados) e filtragem (fusão da informação obtida através das observações ou sensores).

4.6 Mapeamento Baseado em RFS

Como mencionado anteriormente, em contraste com a representação vetorial, o RFS dos estados do mapa \mathcal{M}_k pode encapsular de forma unificada as incertezas de cardinalidade e localização. Assim como o RFS de medidas \mathcal{Z}_k pode encapsular as incertezas de medida e valor.

Como realizado em uma seção anterior deste estudo, o RFS é definido de acordo com a equação (6b). Nessa equação, há um conjunto \mathcal{D}_k obtido a partir das medidas geradas pelas características (*features*) e outra \mathcal{C}_k pelas medidas espúrias, em um instante k , que podem depender da posição \mathcal{X}_k do robô.

De forma geral, \mathcal{Z}_k pode representar uma série de parâmetros medidos, mas, no caso mais simples aqui descrito, é uma medida de distância e ângulo em relação ao sensor do robô. É importante ressaltar que a cardinalidade de \mathcal{Z}_k geralmente difere da de \mathcal{M}_k devido às incertezas nas medições, oclusões, medições espúrias e novos elementos entrando no campo de visão do robô \mathcal{D}_k e \mathcal{C}_k são assumidos como RFS independentes.

O RFS de medidas gerado por uma *feature* m tem densidade de probabilidades igual a $P_{\mathcal{D}}(m, \mathcal{X}_k) g_k(z | m, \mathcal{X}_k)$, onde o primeiro termo é a probabilidade de um sensor detectar uma *feature* m e o segundo é a probabilidade de uma *feature* m gerar uma medida z .

A habilidade de um sensor (ou um algoritmo de detecção de *features*) detectar um dado objeto pode ser altamente influenciada por sua posição relativa. Por exemplo, uma oclusão ou grande distância pode resultar em $P_{\mathcal{D}} = 0$.

A densidade de probabilidade de um sensor produzir uma medida \mathcal{Z}_k , dado um estado do robô (posição e orientação), \mathcal{X}_k , é dada pela convolução:

$$g_k(\mathcal{Z}_k | \mathcal{X}_k, \mathcal{M}_k) = \sum_{\mathcal{W} \subseteq \mathcal{Z}_k} g_{\mathbb{D}}(\mathcal{W} | \mathcal{M}_k, \mathcal{X}_k) g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W}), \quad (16)$$

onde $g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W})$ é a densidade das medições espúrias e $g_{\mathbb{D}}$ é a densidade do RFS \mathcal{D}_k das medições geradas pelas *features* em \mathcal{M}_k (dado estado do veículo). O primeiro termo incorpora as incertezas das medições e ruídos, o segundo termo modela os dados espúrios gerados pelo sensor e é tipicamente definido previamente. É reproduzida abaixo a forma geral da recursão de Bayes para mapeamento baseado em RFS:

$$p_{k|k-1}(\mathcal{M}_k, \mathcal{Z}_{0:k-1}, \mathcal{X}_k) = \int f_{\mathcal{M}}(\mathcal{M}_k | \mathcal{M}_{k-1}, \mathcal{X}_k) p_{k-1}(\mathcal{M}_k, \mathcal{Z}_{0:k-1}, \mathcal{X}_{0:k-1}) \delta_{\mathcal{M}_{k-1}} \quad (17)$$

e

$$p_k(\mathcal{M}_k, \mathcal{Z}_{0:k}, \mathcal{X}_k) = \frac{g_k(\mathcal{Z}_k | \mathcal{X}_k, \mathcal{M}_k) p_{k|k-1}(\mathcal{M}_k, \mathcal{Z}_{0:k-1}, \mathcal{X}_k)}{\int g_k(\mathcal{Z}_k | \mathcal{X}_k, \mathcal{M}_k) p_{k|k-1}(\mathcal{M}_k, \mathcal{Z}_{0:k-1}, \mathcal{X}_k) \delta_{\mathcal{M}_k}}, \quad (18)$$

sendo δ a indicação de integral de um conjunto e a integral é definida de acordo com a equação (9).

Esse mapa baseado em *features* encapsula as incertezas inerentes no número de características introduzidas na detecção, medidas espúrias e movimentos do robô, assim como as incertezas de posição introduzidas pelo ruído de medição.

Como fica evidente pelas equações acima, a aplicação da forma geral da recursão de Bayes é matematicamente intratável. Uma aproximação baseada em filtro PHD é proposta em (Mullane et al., 2011b). O objetivo é propagar a intensidade v_k do mapa no lugar de toda a densidade multidimensional (vide (18)).

$$v_{k|k-1}(m|X_k) = v_{k-1}(m|X_{k-1}) + b_k(m|X_k), \quad (19)$$

onde b_k é o PHD do RFS das últimas *features* detectadas $\mathcal{B}(\mathcal{X}_k)$.

A correção do filtro PHD, que tem forma similar ao Filtro de Bayes usual, é dada por:

$$v_k(m|X_k) = v_{k|k-1}(m|X_k) \times \left[1 - P_{\mathcal{D}}(m|X_k) + \sum_{z \in \mathcal{Z}_k} \frac{P_{\mathcal{D}} g_k(z|m, X_k)}{c_k(z|X_k) + \int P_{\mathcal{D}}(\xi|X_k) g_k(z|\xi, X_k) v_{k|k-1}(\xi|X_k) d\xi} \right], \quad (20)$$

onde $v_k(m|X_k)$ é a predição para a função de intensidade, $P_{\mathcal{D}}(m|X_k)$ a probabilidade de uma *feature* m ser detectada com o robô na posição X_k , $g_k(z|m, X_k)$ o modelo de medição do sensor no instante k e $c_k(z|X_k)$ a intensidade de medições espúrias no instante k .

5 Simulação e Resultados

Para a simulação dos resultados, foram utilizados como base os dados do ambiente e dos sensores disponibilizados para download em (Stachniss, 2018). Tal *framework* foi adaptado para receber o algoritmo implementado, o qual foi baseado nas referências (VO and MA, 2006), (Clark et al., 2006) e (Mullane et al., 2011a), bem como nos códigos disponibilizados em (Vo; Clarke) e (Sola, 2018). O mapa *true* do ambiente, com escala em metros, pode ser visualizado na Figura 5. As *landmarks* estão marcadas em azul, caminhos que não devem ser cruzados em preto e a trajetória desejada para o robô está em vermelho.

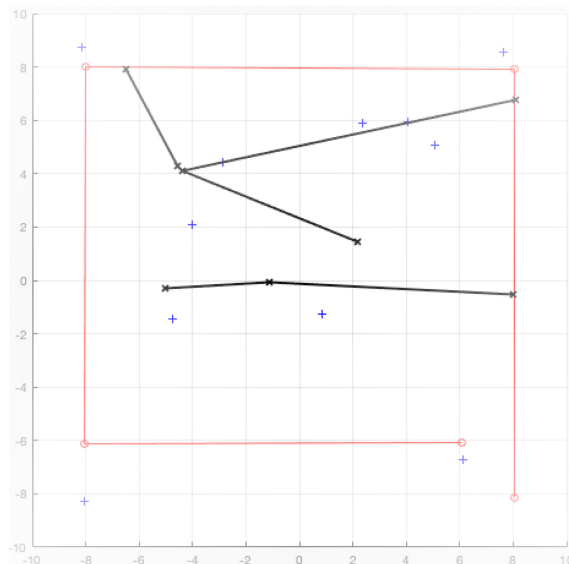


Figura 5: Mapa do ambiente com as posições reais das *features* em azul e trajetória planejada para o robô em vermelho.

O resultado obtido, após o seguimento da trajetória pelo robô, pode ser observado na Figura 6. A trajetória estimada é apresentada em azul, a trajetória executada pelo veículo em vermelho. Já as *features* estimadas estão em verde, sendo os círculos azuis as *features true*.

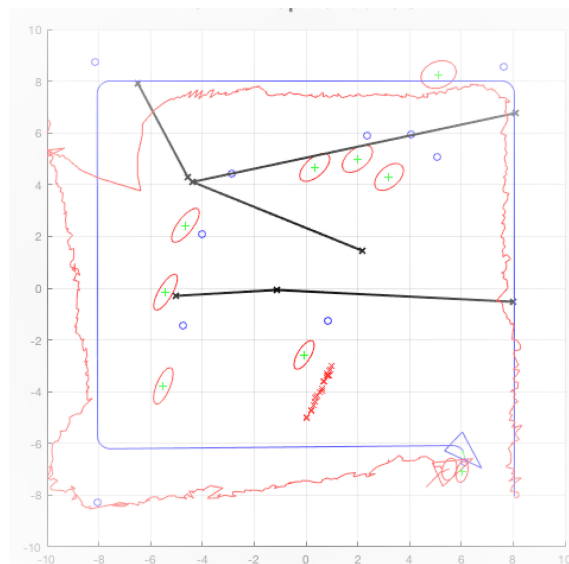


Figura 6: Posições estimadas das *features* x posições reais.

Neste estudo foi considerado apenas o problema de mapeamento. Mesmo no RFS SLAM, para um melhor desempenho computacional, há uma separação do problema, sendo a localização normalmente realizada por algum algoritmo baseado em Kalman ou filtro de partículas, conforme (Ristic et al., 2016).

6 Conclusão

Inicialmente, pode-se concluir que o uso de *Random Finite Sets* é adequado à aplicação no problema de mapeamento em robótica. Tal resultado já era esperado, dado que fundamentalmente foi pensado para o problema de *multi-target tracking* oriundo da comunidade de radares.

Também é possível verificar que, apesar de todas as vantagens apresentadas, o arcabouço matemático diferente do usual, do utilizado pela comunidade e estudiosos em robótica, pode tornar-se um entrave.

Uma vez superada tal dificuldade, a aplicação do filtro PHD apresenta resultados consistentes e há uma associação direta entre a integral da função intensidade e número de *features* ou *landmarks* existente no ambiente. Além disso, torna-se uma tarefa simples estimar a posição desses marcos, já que os valores de máximo locais da mistura Gaussiana correspondem à posição deles. E, em posse dessa informação, é possível obter-se o vetor das médias e, portanto, as coordenadas x e y de cada *feature*.

Dessa forma, a principal colaboração desse trabalho é fomentar o estudo das técnicas baseadas em RFS em nossa comunidade. Além disso, este trabalho abre a possibilidade para aplicação de RFS em problemas reais nos robôs pertencentes ao Laboratório de Controle (LabCON/PEE) da Universidade Federal do Rio de Janeiro. Portanto, espera-se apresentar trabalhos futuros dessa forma, além de uma evolução para o caso mais complexo SLAM e elaboração de um ambiente simulado via ROS.

Referências

- D. Clark, K. Panta, and B.-n. Vo. The gm-phd filter multiple target tracker. In *Proceedings of 9th International Conference on Information Fusion*, 2006. doi: 10.1109/ICIF.2006.301809.
- B. Clarke. Gaussian mixture probability hypothesis density filter (gm-phd). Disponível em <<https://goo.gl/Rj5hVo>>. Acesso em 29 de janeiro de 2017.
- P. M. Dames. *Multi-Robot Active Information Gathering Using Random Finite*. PhD thesis, Philip M. Dames, 2015.
- J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo. Mobile robotics in a random finite set framework. In *Lecture Notes in Computer Science*, pages 519–528. Springer Nature, 2011a. doi: 10.1007/978-3-642-21524-7_64.
- J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo. A random-finite-set approach to bayesian SLAM. *IEEE Transactions on Robotics*, 27(2):268–282, apr 2011b. doi: 10.1109/tro.2010.2101370.
- B. Ristic, M. Beard, and C. Fantacci. An overview of particle methods for random finite set mo-

dels. *Information Fusion*, 31(C):110–126, September 2016.

- S. J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.
- B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer, 2nd edition, 2016. Chapter 46.
- J. Sola. Slam toolbox for matlab. Disponível em <<https://github.com/joansola/slamtb>>. Acesso em 1 de fevereiro de 2018., 2018.
- C. Stachniss. Codes - framework to slam. Disponível em <<https://goo.gl/cgMCFu>>. Acesso em 22 de março de 2018., 2018.
- S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agent. MIT Press, August 2005.
- B. Vo. Codes for matlab. Disponível em <<http://ba-tuong.vo-au.com/codes.html>>. Acesso em 29 de janeiro de 2017.
- B. VO and W. MA. The gaussian mixture probability hypothesis density filter. *IEEE Transactions Signal Processing*, 54(11):4091–4104, September 2006. doi: 10.1109/TSP.2006.881190.
- F. Zhang, S. Li, S. Yuan, E. Sun, and L. Zhao. Algorithms analysis of mobile robot slam based on kalman and particle filter. In *Proceedings of 9th International Conference on Modelling, Identification and Control (ICMIC)*, Kunming, China, July 2017.