

Rastreamento de trajetória de veículo terrestre não linear estocástico multivariável utilizando filtro de partículas^{*}

Hugo Alves Velozo^{*}, Francisco das Chagas Souza^{*}

^{*} Laboratório de Sistemas Adaptativos e Processamento de Sinais-LSAPS, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Maranhão-UFMA, São Luís-MA, Brasil
(e-mails: hugo.velozo@discente.ufma.br; francisco.souza@ufma.br)

Abstract: State estimation of stochastic nonlinear dynamical systems has several applications in many areas of engineering. Among these areas, the system states can be estimated in order to track mobile objects such as, for instance, autonomous vehicles and mobile robots in a given space. A particle filter can be used for this purpose due to its efficiency and precision in systems with generic characteristics. In this article, an application of the particle filter for tracking of a land vehicle with free mobility in a two-dimensional space will be presented. Numerical simulations are carried out considering different scenarios of process and measurement noises. Through these simulations, a comparative analysis between the particle filter and the unscented Kalman filter is performed.

Resumo: A estimação de estados de sistemas dinâmicos não lineares estocásticos tem diversas aplicações em várias áreas da engenharia. Dentre estas áreas, os estados do sistema podem ser estimados com objetivo de rastrear objetos móveis como, por exemplo, veículos autônomos e robôs móveis em um dado espaço. Um filtro de partículas pode ser utilizado para essa finalidade devido à sua eficiência e precisão em sistemas com características genéricas. Neste artigo, uma aplicação do filtro de partículas para o rastreamento de um veículo terrestre com mobilidade livre no espaço bidimensional será apresentada. Simulações numéricas são realizadas considerando diferentes cenários de ruídos de processo e medição. Por meio destas simulações, uma análise comparativa entre o filtro de partículas e o filtro de Kalman *unscented* é realizada.

Keywords: Bayesian estimation; particle filter; state estimation; vehicle trajectory tracking.

Palavras-chave: Estimação bayesiana; filtro de partículas; estimação de estado; rastreamento de trajetória de veículo.

1. INTRODUÇÃO

Conhecer os valores das variáveis de estado para cada instante de tempo de um sistema dinâmico pode ser útil na realização de várias atividades na engenharia, como diagnóstico de falhas, detecção de falhas e supervisão (Isermann, 1997; Pulido et al., 2019; Deng et al., 2020); projeto de algoritmos de controle (Inoue et al., 2015; Sato, 2019); rastreamento de alvo (Wang and Bao, 2021; Xu et al., 2021; Wang et al., 2021); navegação de robôs (Zhibin et al., 2020; Blok et al., 2019), entre outras aplicações. Em muitas situações práticas, as variáveis de estado não podem ser medidas, entre outros motivos, por impossibilidades físicas ou questões de custo na utilização de instrumentos de medição (Monticelli, 2000). Para contornar este problema, uma solução consiste em estimar os estados do sistema por meio de estimadores de estado.

^{*} O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, da Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão (FAPEMA) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Ao longo de anos de pesquisa nesse tema, vários algoritmos foram propostos para realizarem essa tarefa em diferentes contextos. Em sistemas dinâmicos lineares com ruído gaussiano, o filtro de Kalman (KF - Kalman filter) realiza estimações ótimas (Kalman, 1960). Já no contexto de sistemas não lineares com ruídos Gaussianos, variações do filtro de Kalman como, filtro de Kalman estendido (EKF - *extended Kalman filter*), filtro de Kalman *unscented* (UKF - *unscented Kalman filter*) (Julier et al., 1995), filtro de Kalman em quadratura (QKF - *quadrature Kalman filter*) (Arasaratnam et al., 2007) e filtro de Kalman de cubatura (CKF - *cubatura Kalman filter*) (Arasaratnam and Haykin, 2009) obtêm resultados precisos.

Entretanto, os algoritmos citados anteriormente têm seus desempenhos comprometidos quando o sistema dinâmico apresenta um alto grau de não linearidade e ruídos não Gaussianos. Para essas características de sistemas dinâmicos, novos métodos de estimação foram propostos estimulados com o avanço da capacidade computacional por volta de 1940 que possibilitou o desenvolvimento de aproximações numéricas de integrais presentes na formulação de filtragem Bayesiana proposta por Andrey Kolmogorov

e Sydney Chapman (Luengo et al., 2020). Entre essas abordagens está o método Monte Carlo desenvolvido por Stanislaw Ulam e Nicholas Metropolis (Metropolis and Ulam, 1949; Von Neumann and Ulam, 1951; Roger, 1987). Com base no método de Monte Carlo, em 1993, o algoritmo de filtro de partículas foi proposto por Gordon et al. (1993).

O filtro de partículas (PF - *particle filter*) apresentou resultados com alta precisão na estimativa de estados de sistemas dinâmicos não lineares com ruídos Gaussianos e não Gaussianos. Quanto às vantagens do PF, três pontos podem ser destacados. Em primeiro lugar, como a implementação de um PF é independente do modelo do sistema, ele pode ser aplicado em diferentes sistemas não lineares. Em segundo lugar, em comparação com EKF e UKF, o PF pode realizar estimativas de estado mais precisas, particularmente para sistemas altamente não lineares. Terceiro, o PF pode ser aplicado em sistemas com ruídos não gaussianos (Yin and Zhu, 2015).

Dada essas características do filtro de partículas, o mesmo tem sido aplicado na auto-localização de robôs móveis e veículos autônomos em diferentes cenários. A localização é uma parte fundamental em sistemas móveis pois determinam a estratégia locomoção possibilitando aplicações em áreas como agricultura (Qiu et al., 2018; Mahmud et al., 2019) e mineração (Faria et al., 2020), dentre outras.

Tanto os robôs moveis como os veículos autônomos podem ser rastreados com base em uma abordagem probabilística (Panigrahi and Bisoy, 2021). O filtro de Kalman e suas variações (por exemplo, EKF e UKF) podem ser utilizados para realizarem localização probabilística de um objeto móvel (Teslić et al., 2011; Simanek et al., 2014; Bo and Jian, 2016). Entretanto, como comentado anteriormente, estes algoritmos possui limitações quanto o grau de não linearidade do sistema e das características do ruído. Nestas situações, uma abordagem mais eficiente é o método de localização de Monte Carlo (Monte Carlo localization - MCL) (Blanco et al., 2010; Zhang et al., 2019).

Neste artigo, a partir de medições ruidosas, um filtro de partículas é utilizado para rastrear a posição e velocidade de um veículo terrestre descrito por um modelo não linear estocástico e multivariável. Este modelo é discretizado para ser utilizado em um método de localização de Monte Carlo, especificamente um filtro de partículas padrão. Além disso, uma formulação da estimação bayesiana e uma breve introdução sobre filtragem de partículas é apresentada. Finalmente, os resultados computacionais são apresentados e analisados para três cenários de ruídos de processo e medição.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Modelo de Veículo Terrestre

Neste artigo, um veículo terrestre será representado por um modelo simplificado ponto-massa com mobilidade livre no espaço bidimensional, conforme discutido em (Gao, 2014). O objetivo consiste em estimar a posição e velocidade do veículo nos sentidos norte e leste. Esta estimação é realizada utilizando-se medições de posição ruidosas prove-

nientes de sensores como GPS (*global positioning system*).

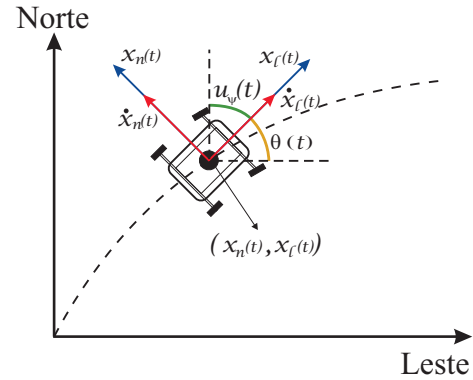


Figura 1. Ilustração de um modelo ponto-massa simples.

Como ilustrado na Figura 1, $x_n(t)$ e $x_e(t)$ são as posições norte e leste do veículo, respectivamente. As velocidades nos sentidos norte e leste são representadas por $\dot{x}_n(t)$ e $\dot{x}_e(t)$, respectivamente. Além disso, a orientação no sentido leste e o ângulo de direção do veículo são representados por $\theta(t)$ e $u_\phi(t)$, respectivamente.

Este modelo pode ser representado matematicamente da seguinte forma:

$$\begin{bmatrix} \dot{x}_l(t) \\ \dot{x}_n(t) \\ \dot{v}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos(\theta(t)) \\ v(t)\sen(\theta(t)) \\ \left(P \frac{u_a(t)}{v(t)} - A C_d v(t)^2 \right) / M \\ v(t)tg(u_\phi(t)) / L \end{bmatrix} + \mathbf{B}\mathbf{w}(t), \quad (1)$$

$$\mathbf{z}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(t) + \mathbf{v}(t), \quad (2)$$

onde P , A , C_d , M , L e $u_a(t)$ são a potência de pico do motor, área frontal, coeficiente de arrasto, massa do veículo, distância entre eixos e posição angular do acelerador, respectivamente. Além disso, o ruído de processo e o ruído de medição são normalmente distribuídos conforme $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$ e $\mathbf{v} \sim \mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$.

O modelo representado nas equações (1) e (2) pode ser discretizado utilizando a aproximação (*forward difference method*) (Chen, 1999):

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}[k+1] - \mathbf{x}[k]}{T} \quad (3)$$

onde T é o período de amostragem e $k \in \{0, 1, 2, 3, \dots, N_T\}$ com N_T sendo o número total de amostras.

Aplicando o método de discretização descrito em (3), obtêm-se o seguinte modelo discreto:

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_{k+1}^{(l)} \\ x_{k+1}^{(n)} \\ x_{k+1}^{(v)} \\ x_{k+1}^{(\theta)} \\ x_{k+1} \end{bmatrix} = \mathbf{f} \left(\mathbf{x}_k, u_k^{(a)}, u_k^{(\psi)} \right) + \mathbf{B}\mathbf{w}_k,$$

$$\mathbf{x}_{k+1} = \begin{bmatrix} T \left(x_k^{(v)} \cos \left(x_k^{(\theta)} \right) \right) + x_k^{(l)} \\ T \left(x_k^{(v)} \sin \left(x_k^{(\theta)} \right) \right) + x_k^{(n)} \\ T \left(\frac{P}{M} \frac{u_k^{(a)}}{x_k^{(v)}} - \frac{AC_d}{M} \left(x_k^{(v)} \right)^2 \right) + x_k^{(v)} \\ T \left(x_k^{(v)} \operatorname{tg} \left(u_k^{(\psi)} \right) / L \right) + \theta_k \end{bmatrix} + \mathbf{B}\mathbf{w}_k, \quad (4)$$

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k. \quad (5)$$

Seja um sistema dinâmico não linear estocástico representado pelo seguinte modelo de espaço de estado:

$$\begin{cases} \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_k) \\ \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \end{cases}, \forall k \in (1, 2, 3, \dots), \quad (6)$$

onde $\mathbf{x}_k \in \mathbb{R}^{n_x}$, $\mathbf{z}_k \in \mathbb{R}^{n_z}$, $\mathbf{u}_k \in \mathbb{R}^{n_u}$, $\mathbf{w}_k \in \mathbb{R}^{n_w}$ e $\mathbf{v}_k \in \mathbb{R}^{n_v}$ representam os vetores de estado, saída, entrada, ruído de processo e ruído de medição, respectivamente.

As funções não lineares $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \mapsto \mathbb{R}^{n_x}$ e $\mathbf{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \mapsto \mathbb{R}^{n_z}$ são as funções de transição de estado e de medição, respectivamente. Além disso, assume-se que \mathbf{w} e \mathbf{v} são processos estocásticos de média nula, mutuamente independentes com distribuições gaussianas $\mathbf{w}_k \sim \mathcal{N}\{0, \sigma_w^2\}$ e $\mathbf{v}_k \sim \mathcal{N}\{0, \sigma_v^2\}$, respectivamente.

O modelo em espaço de estados em (6) pode ser reformulado em termos cadeia de Markov (Naesseth et al., 2019). Assim, as funções não lineares \mathbf{f} e \mathbf{h} correspondem a transição de estado e a distribuição de verossimilhança, portanto $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ e $p(\mathbf{z}_k|\mathbf{x}_k)$, respectivamente. Assim, o modelo de espaço de estado probabilístico pode ser representado como

$$\begin{cases} \mathbf{X}_0 & \sim p(\mathbf{x}_0), \\ \mathbf{X}_k | (\mathbf{X}_{k-1} = \mathbf{x}_{k-1}) & \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}), \\ \mathbf{Z}_k | (\mathbf{X}_k = \mathbf{x}_k) & \sim p(\mathbf{z}_k|\mathbf{x}_k), \end{cases} \quad (7)$$

onde $p(\mathbf{x}_0)$ é a função densidade de probabilidade do primeiro estado, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ é a função densidade de probabilidade associada com a transição para o novo estado $\mathbf{X}_k = \mathbf{x}_k$. Da mesma forma, $p(\mathbf{z}_k|\mathbf{x}_k)$ é a função densidade de verossimilhança de $\mathbf{X}_k = \mathbf{x}_k$ que gera a medida $\mathbf{Z}_k = \mathbf{z}_k$.

O problema da estimação de estado constitui-se em calcular uma estimativa de \mathbf{x}_k , em tempo real, usando uma sequência de medições denotadas por $\mathbf{Z}_{1:k} \equiv \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$.

2.2 Estimação bayesiana

Na abordagem estimação bayesiana, cada estimativa de estado do modelo probabilístico descrito em (7) é resolvida pelo cálculo recursivo do estado a posterior:

$$\mathbf{X}_k | (\mathbf{Z}_{1:k} = \mathbf{z}_{1:k}) \sim p(\mathbf{x}_k|\mathbf{z}_{1:k}), \quad (8)$$

onde $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ é uma função densidade *a posteriori*. Usando o teorema de Bayes, essa função pode ser obtida da seguinte forma

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}, \quad (9)$$

onde $p(\mathbf{z}_k|\mathbf{x}_k)$, $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ e $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ representam as funções densidade de verossimilhança, *a posteriori* e probabilidade marginal, respectivamente.

Considerando que em problemas de estimativa de estado a probabilidade marginal não é conhecida Tulshyan et al. (2016), então nós reescrevemos (9) como uma relação de proporcionalidade com estado anterior e a função densidade de verossimilhança:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1}). \quad (10)$$

Além disso, usando a função de densidade de probabilidade marginal $(\mathbf{z}_k|\mathbf{z}_{1:k-1})$, podemos escrever a função de densidade *a priori* de estado como

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{\mathcal{X}} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}, \quad (11)$$

onde $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ e $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ são as funções densidade de transição e *posteriori* em $k-1$, respectivamente. Resumidamente, a função de densidade *posteriori* (ou função densidade de filtro) no tempo k pode ser calculada usando as seguintes etapas:

1. Atualização de tempo (previsão):

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{\mathcal{X}} \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1})}_{\text{dinâmica}} \underbrace{p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})}_{\text{filtragem}} d\mathbf{x}_{k-1}. \quad (12)$$

2. Atualização de medição (correção):

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \propto \underbrace{p(\mathbf{z}_k|\mathbf{x}_k)}_{\text{medição}} \underbrace{p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}_{\text{predição}}. \quad (13)$$

Com $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ obtido em (13), a estimativa de \mathbf{x}_k é dado por:

$$\hat{\mathbf{x}}_k = \mathbb{E}[\mathbf{X}_k | (\mathbf{Z}_{1:k} = \mathbf{z}_{1:k})] = \int_{\mathcal{X}} \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k, \quad (14)$$

onde $\hat{\mathbf{x}}_k$ é a estimativa de \mathbf{x}_k . O problema de filtragem corresponde a resolver as equações (12) e (13) recursivamente. Os métodos usados para resolver essas equações são chamados de métodos de filtragem Schön and Lindsten (2015); Chen et al. (2003).

Dentre os métodos de filtragem, o método Monte Carlo sequencial (SMC - *sequential Monte Carlo*) é uma classe de métodos de filtragem usados para estimação de estados de sistemas dinâmicos com características gerais. Este método é comumente chamado de filtro de partículas. Os algoritmos de filtragem de partículas mais amplamente usados na literatura são o filtro de reamostragem de importância de amostragem (SIR - *sampling importance resampling*), filtro de reamostragem de importância de amostragem auxiliar (ASIR - *auxiliary sampling importance resampling*) e filtro de partículas Rao-Blackwellized (RBPF - *Rao-Blackwellized particle filter*) Arulampalam et al. (2002); Tulshyan et al. (2016); Chen et al. (2003).

3. FILTRO DE PARTÍCULAS

Estimações $\hat{\mathbf{x}}_k$ do estado \mathbf{x}_k são obtidas calculando a integral (14). Uma aproximação de Monte Carlo dessa integral é obtida gerando amostras perfeitamente distribuídas de acordo com a função de densidade *a posteriori* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. No entanto, em um caso mais geral, essa função de densidade pode ser não gaussiana, portanto amostras

perfeitamente distribuídas não podem ser geradas. Neste caso, pode-se utilizar um método como o SIR para gerar amostras distribuídas de acordo com $p(\mathbf{x}_k | \mathbf{z}_{1:k})$, ou seja,

$$(\mathbf{X}_k = \mathbf{x}_k^{(i)}) \sim p(\mathbf{x}_k^{(i)} | \mathbf{z}_{1:k}), \quad \forall i \in \{1, 2, \dots, N_p\}, \quad (15)$$

onde \mathbf{X}_k , $\{\mathbf{x}_k\}_{i=1}^{N_p}$ e N_p representam uma variável aleatória que indica o estado do sistema em um instante k , conjunto de amostras distribuídas de acordo com $p(\mathbf{x}_k^{(i)} | \mathbf{z}_{1:k})$, e o número total de partículas (amostras), respectivamente. Assim, a atualização da medição é formada por

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \propto p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}), \quad (16)$$

onde $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ e $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ são a função densidade alvo e a função densidade de amostragem, respectivamente. O algoritmo de filtragem de partículas consiste em calcular recursivamente a aproximação de Monte Carlo da densidade posterior $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ por meio de três etapas: propagação, cálculo dos pesos de importância e reamostragem. Na etapa de propagação, uma nova partícula $\mathbf{x}_k^{(i)}$ é amostrada por meio da função densidade $p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)})$. Na etapa de cálculo dos pesos: para cada amostra $\mathbf{x}_k^{(i)}$ é atribuída uma importância $w_{k|k-1}^{(i)}$:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{z}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}, \quad (17)$$

onde $q(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)$ é chamada de função densidade de importância Arulampalam et al. (2002).

Após essa etapa, os pesos de importância são normalizados da seguinte forma

$$\tilde{w}_k^{(i)} = \frac{w_{k|k-1}^{(i)}}{\sum_{i=1}^{N_p} w_{k|k-1}^{(i)}}, \quad (18)$$

onde $\tilde{w}_k^{(i)}$ é o peso de importância normalizado.

A reamostragem pode ser realizada com base em uma aproximação da métrica tamanho de amostra efetiva (ESS - *effective sample size*) (Arulampalam et al., 2002) dada por:

$$\hat{N}_{ESS} = \left(\sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \right)^{-1} \quad (19)$$

onde $\tilde{w}_k^{(i)}$ é o pesos de importância normalizado (18). Se $\hat{N}_{ESS} < \theta N_p$ ($\theta \in [0, 1]$), então com base em $\{\mathbf{x}^{(i)}, \tilde{w}^{(i)}\}_{i=1}^{N_p}$ calcula-se uma aproximação Monte Carlo da função de distribuição cumulativa $\tilde{P}_X(\cdot)$. Desse modo, a etapa de reamostragem gera M novas partículas de acordo com a seguinte probabilidade

$$\tilde{P}r(\mathbf{X}^{(j)} = \mathbf{x}^{(i)}) = \tilde{w}^i, \quad \forall j = 1, \dots, M, \quad (20)$$

onde $\tilde{P}r$ é uma probabilidade obtida da aproximação de Monte Carlo da função de distribuição cumulativa e M é o número de partículas reamostradas ($M = N_p$). Em seguida, as partículas com pesos de importância pequenos

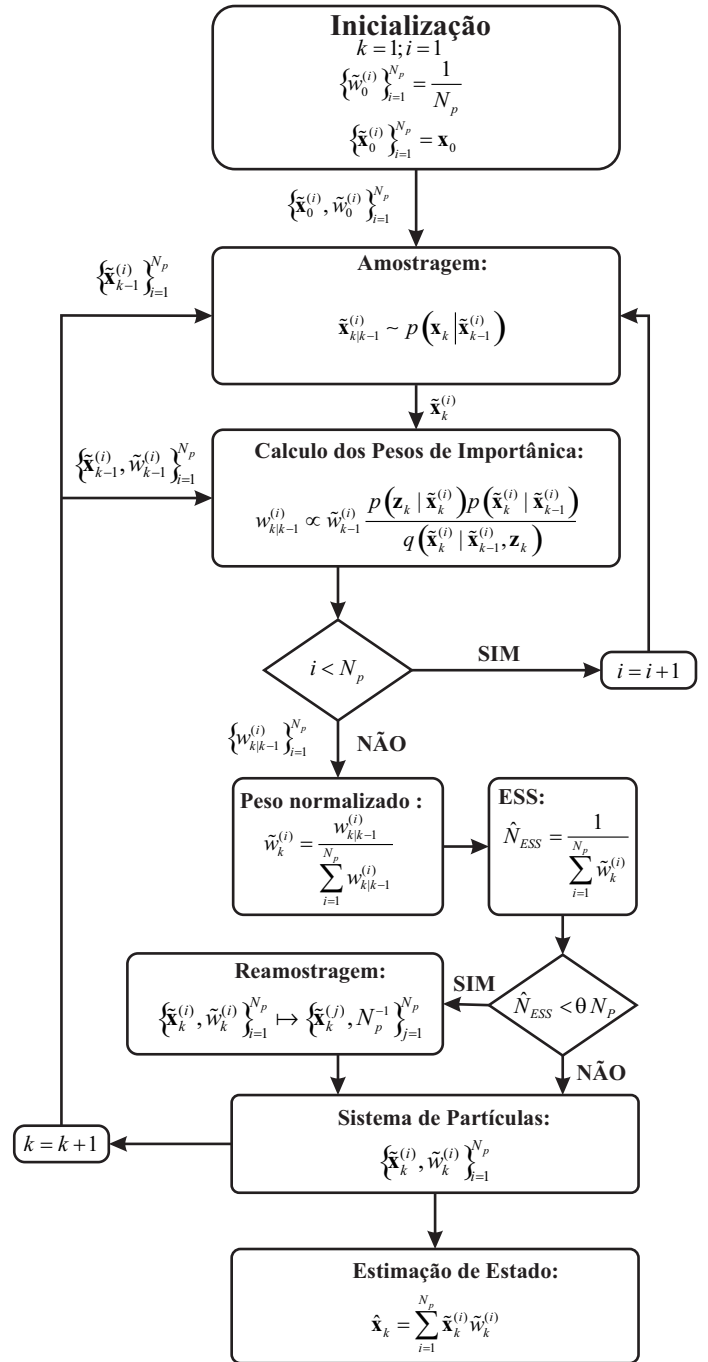


Figura 2. Estimação de estado usando filtro de partículas.

são descartadas e as melhores amostradores são replicadas proporcionalmente de acordo com os pesos de importância.

A Figura 2 ilustra as principais etapas do filtro de partículas na estimação de estados. Inicialmente, o sistema de partícula inicial é formado $\{\mathbf{x}_0^{(i)}, w_0^{(i)}\}_{i=1}^{N_p}$, onde o estado inicial pode ou não ser conhecido *a priori* e os pesos de importância assumem mesmo valor N_p^{-1} . Com as partículas $\{\mathbf{x}_{k-1}^{(i)}\}_{i=1}^{N_p}$ do instante anterior, realiza-se uma amostragem de novas partículas distribuídas de acordo com $\{\mathbf{x}_k^{(i)}\}_{i=1}^{N_p} \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)})$. Com $\mathbf{x}_k^{(i)}$, $\mathbf{x}_{k-1}^{(i)}$ e \mathbf{z}_k é possível obter as funções de densidade e assim

calcular os pesos de importância $w_{k|k-1}^{(i)}$ de acordo com (17) para todas as partículas i que serão normalizadas conforme descrito em (18). Caso a métrica de tamanho de amostra efetiva seja menor que uma certa porcentagem do número de partículas N_p , um método de reamostragem é utilizado para distribuir as amostras de forma uniforme.

Finalmente, com o sistema de partículas $\{\mathbf{x}_0^{(i)}, w_0^{(i)}\}_{i=1}^{N_p}$, realiza-se a estimação $\hat{\mathbf{x}}_k$ do estado verdadeiro.

4. SIMULAÇÕES NUMÉRICAS

Os resultados computacionais apresentados neste artigo foram obtidos tendo como base o modelo não linear estocástico multivariável discreto no tempo apresenta em (4) e (5) com os seguintes parâmetros:

- I Potência de pico do motor: $P = 100.000 \text{ W}$;
- II Área frontal: $A = 1 \text{ m}^2$;
- III Coeficiente de arrasto: $C_d = 0,3$;
- IV Massa do veículo: 1.250 kg ;
- V Distância entre eixos: $L = 2,5 \text{ m}$.

Foi considerado que somente as variáveis de estado posição leste $x_k^{(l)}$ e posição norte $x_k^{(n)}$ sofreriam interferência do ruído de processo \mathbf{w}_k . Com isso a matriz \mathbf{B} e \mathbf{w}_k ficam:

$$\mathbf{B} = \begin{bmatrix} T & 0 \\ 0 & T \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (21)$$

$$\mathbf{w}_k = \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \end{bmatrix}. \quad (22)$$

As características estatísticas do ruído de processo podem ser representadas com:

$$\Sigma_{\mathbf{w}} = \begin{bmatrix} \sigma_{\mathbf{w}}^2 & 0 \\ 0 & \sigma_{\mathbf{w}}^2 \end{bmatrix}, \quad (23)$$

$$\boldsymbol{\mu}_{\mathbf{w}} = [0 \ 0], \quad (24)$$

onde $\Sigma_{\mathbf{w}}$ e $\boldsymbol{\mu}_{\mathbf{w}}$ são a matriz de covariância e vetor média do ruído de processo.

Por sua vez, considerou-se que as três saídas do sistema sofreriam interferência do ruído de medição. Com isso matriz de covariância e o vetor média do ruído de processo são expressos da seguinte forma:

$$\Sigma_{\mathbf{v}} = \begin{bmatrix} \sigma_{\mathbf{v}}^2 & 0 & 0 \\ 0 & \sigma_{\mathbf{v}}^2 & 0 \\ 0 & 0 & \sigma_{\mathbf{v}}^2 \end{bmatrix}, \quad (25)$$

$$\boldsymbol{\mu}_{\mathbf{v}} = [0 \ 0 \ 0]. \quad (26)$$

Na Figura 3 apresenta-se estimação da posição do veículo com relação as coordenadas leste e norte $(x_k^{(l)}, x_k^{(n)})$ para desvio padrão do ruído de processo $\sigma_{\mathbf{w}} = 10$ e ruído de medição $\sigma_v = 15$. Além disso, as médias do ruído de processo e ruído de medição foram consideradas nulas. Pode-se observar que a estimativa obtida pelo filtro com

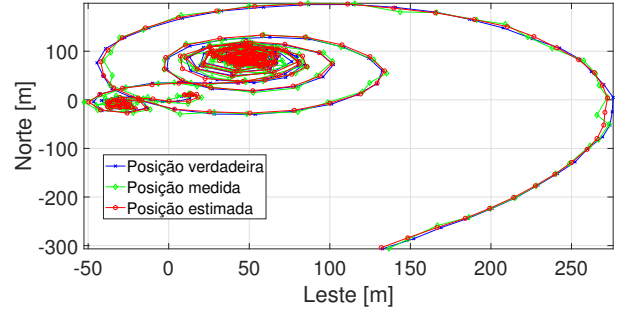


Figura 3. Estimação da posição do veículo usando PF.

300 partículas e método de reamostragem sistemático ficou mais próximo do valor verdade da posição do veículo.

Já as Figuras 4 e 5 apresentam os resultados da estimação das coordenadas leste $x_k^{(l)}$ e $x_k^{(n)}$ para cada amostra k utilizando filtro de partículas e *unscented Kalman filter* (UKF) no cenário de desvio padrão do ruído de processo $\sigma_{\mathbf{w}} = 10$, ruído de medição $\sigma_v = 15$ e médias nulas. Nessas mesmas condições, a Figura 6 apresenta a estimação da variável de estado velocidade $x_k^{(v)}$. Observa-se que devido não existir interferência do ruído de processo nesta variável de estado, as estimativas coincidem com o valor verdadeiro da velocidade mesmo na presença de dados medidos ruidosos.

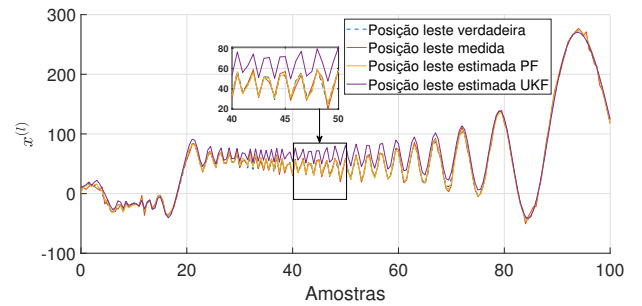


Figura 4. Estimação da posição leste $x^{(l)}$.

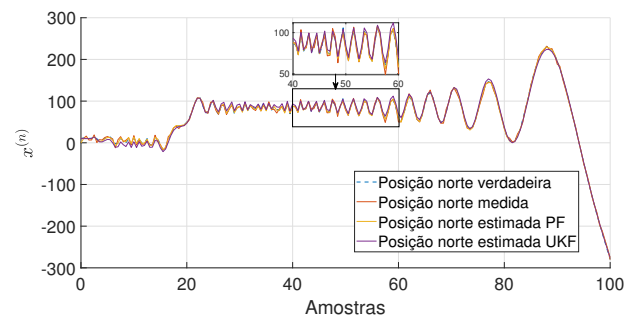


Figura 5. Estimação da posição norte $x^{(n)}$.

Para fazer uma análise do desempenho da filtragem de partículas no rastreamento do veículo, foram feitas 100 realizações de Monte Carlo para diferentes cenários de características de ruído de processo e de medição. Foram utilizadas as medidas de desempenho η , ε_k e $\varepsilon_{\alpha,k}$ baseadas na norma RMSE (*Root-mean-square error*). As seguintes equações representam esses índices de desempenho:

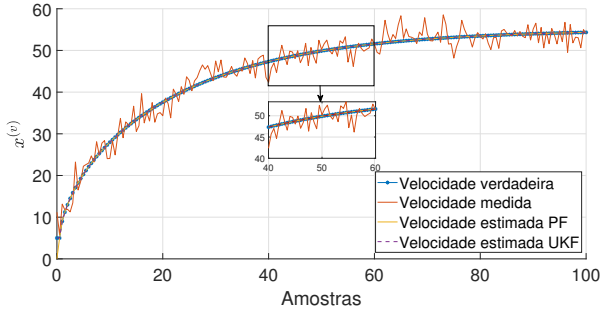


Figura 6. Estimação da variável de estado velocidade $x^{(v)}$.

$$e_{\alpha,k}^r = \hat{x}_{k,\alpha}^r - x_{k,\alpha}^r, \quad (27)$$

$$\varepsilon_k = \frac{1}{N_{MC}} \sum_{r=1}^{N_{MC}} \sqrt{\frac{1}{n_x} \sum_{\alpha=1}^{n_x} (e_{\alpha,k}^r)^2}, \quad (28)$$

$$\eta = \frac{1}{N_{MC}} \sum_{r=1}^{N_{MC}} \frac{1}{N_k} \sum_{k=1}^{N_k} \sqrt{\frac{1}{n_x} \sum_{\alpha=1}^{n_x} (e_{\alpha,k}^r)^2}, \quad (29)$$

$$\varepsilon_{\alpha,k} = \frac{1}{N_{MC}} \sum_{r=1}^{N_{MC}} \sqrt{(e_{\alpha,k}^r)^2} \quad (30)$$

onde $e_{\alpha,k}^r$ e $\varepsilon_{\alpha,k}^r$ representam o erro de estimação de cada variável de estado α para cada amostra k e realização r e o erro médio de cada amostra das variáveis de estado para N_{MC} realizações de simulações de Monte Carlo. Além disso, n_x e N_k representam o número de variáveis de estado e o total de amostras, respectivamente.

Como pode ser observado Figura 7, a medida ε_k para 100 realizações de Monte Carlo o método de filtragem de partículas possui um menor erro médio total para as amostra k em relação ao algoritmo UKF. O mesmo fato pode ser observado no erro médio para cada uma das variáveis de estado $\varepsilon_{1,k}$ (posição leste) e $\varepsilon_{2,k}$ (posição norte), conforme apresentado nas Figuras 8 e 9.

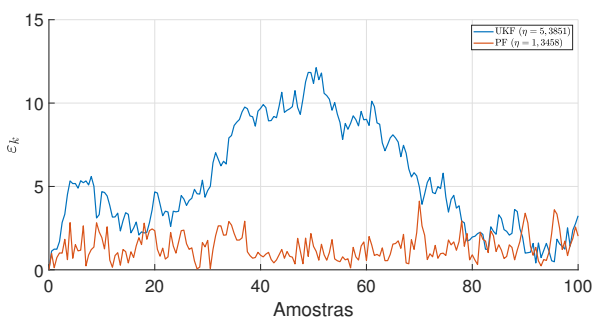


Figura 7. Medida ε_k para 100 realizações de Monte Carlo.

Tabela 1. Valores de η obtidos com simulação de Monte Carlo.

Cenários		η (PF)	η (UKF)
1	$\sigma_w = 5$ e $\sigma_v = 10$	1,0929	4,9375
2	$\sigma_w = 10$ e $\sigma_v = 15$	1,3458	5,3851
3	$\sigma_w = 15$ e $\sigma_v = 20$	1,6726	10,3337

Por meio da Tabela 1, pode-se observar que o filtro de partículas mantém seu desempenho na estimação da trajetória mesmo para aumentos nos desvios padrão do ruído

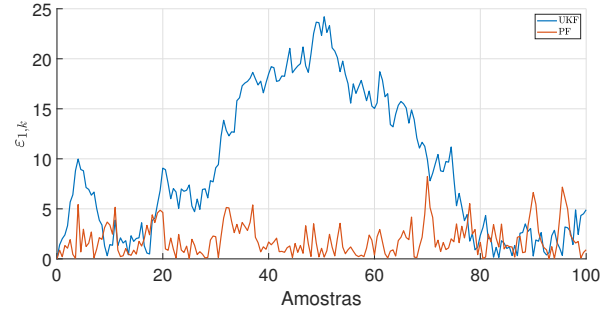


Figura 8. Medida $\varepsilon_{1,k}$ para 100 realizações de Monte Carlo.

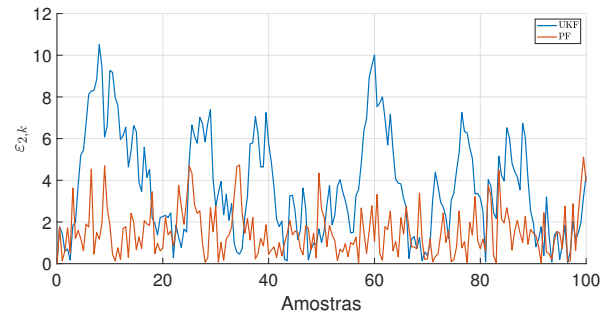


Figura 9. Medida $\varepsilon_{2,k}$ para 100 realizações de Monte Carlo.

de processo e de medição. Em contrapartida, o método UKF tem seu desempenho consideravelmente afetado pelo aumento dos mesmos. Dessa forma, por meio das figuras anteriores e da Tabela 1, certifica-se o desempenho melhor do algoritmo filtro de partículas em relação ao algoritmo UKF no rastreamento de trajetória do veículo modelado no presente artigo.

5. CONCLUSÃO

Neste artigo, foi realizada a estimação da trajetória de um veículo com movimento livre em um espaço bidimensional usando filtragem de partículas. Este veículo foi representado por meio de um modelo não linear estocástico multivariável e discreto no tempo. Através de simulações numéricas em diferentes cenários e uma análise comparativa com o algoritmo UKF, foi possível verificar por meio de medidas de desempenho a eficiência do filtro de partículas no rastreamento de trajetórias de veículos terrestres.

REFERÊNCIAS

- Arasaratnam, I. and Haykin, S. (2009). Cubature kalman filteres. *IEEE Trans. Autom. Control*, 54(6), 1254–1269.
- Arasaratnam, I., Haykin, S., and Elliot, R.J. (2007). Discrete-time nonlinear filtering algorithms using gauss-hermite quadrature. *Proc. IEEE*, 95(5), 953–977.
- Arulampalam, M.S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2), 174–188.
- Blanco, J.L., González, J., and Fernández-Madrigrá, J.A. (2010). Optimal filtering for non-parametric observation models: applications to localization and slam. *The International Journal of Robotics Research*, 29(14), 1726–1742.

- Blok, P.M., van Boheemen, K., van Evert, F.K., IJsselmuident, J., and Kim, G.H. (2019). Robot navigation in orchards with localization based on particle filter and kalman filter. *Computers and Electronics in Agriculture*, 157, 261–269.
- Bo, L. and Jian, Y. (2016). Trajectory-tracking control based on ukf for autonomous underwater vehicle with process and measurement noise. In *2016 IEEE International Conference on Mechatronics and Automation*, 1036–1040. IEEE.
- Chen, C. (1999). *Linear System Theory and Design*. Oxford University Press.
- Chen, Z. et al. (2003). Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1), 1–69.
- Deng, R., Wang, Z., and Fan, Y. (2020). Fault relevant variable selection for fault diagnosis. *IEEE Access*, 8, 23134–23142. doi:10.1109/ACCESS.2020.2970046.
- Faria, H.D., Lizarralde, F., Costa, R.R., Andrade, R.H., Silva, T.H., Pereira, R.F., Barbosa, E.S., Rocha, F., Franca, A., Freitas, G.M., et al. (2020). Rosi: a mobile robot for inspection of belt conveyor. *IFAC-PapersOnLine*, 53(2), 10031–10036.
- Gao, Y. (2014). *Model predictive control for autonomous and semiautonomous vehicles*. University of California, Berkeley.
- Gordon, N.J., Salmond, D.J., and Smith, A.F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, 107–113. IET.
- Inoue, M., Wada, T., Ikeda, M., and Uezato, E. (2015). State-space h controller design for descriptor systems. *Automatica*, 59, 164–170.
- Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods—an introduction. *Control engineering practice*, 5(5), 639–652.
- Julier, S.J., Uhlmann, J.K., and Durrant-Whyte, H.F. (1995). A new approach for filtering nonlinear systems. In *Proc. of the American Control Conference*, 1628–1632. Seattle, Washington.
- Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *ASME Trans. Basic Eng.*, 82, 35–45.
- Luengo, D., Martino, L., Bugallo, M., Elvira, V., and Särkkä, S. (2020). A survey of monte carlo methods for parameter estimation. *EURASIP Journal on Advances in Signal Processing*, 2020, 1–62.
- Mahmud, M.S.A., Abidin, M.S.Z., Mohamed, Z., Abd Rahman, M.K.I., and Iida, M. (2019). Multi-objective path planner for an agricultural mobile robot in a virtual greenhouse environment. *Computers and Electronics in Agriculture*, 157, 488–499.
- Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association*, 44(247), 335–341.
- Monticelli, A. (2000). Electric power system state estimation. *Proceedings of the IEEE*, 88(2), 262–282. doi: 10.1109/5.824004.
- Naesseth, C.A., Lindsten, F., and Schön, T.B. (2019). Elements of sequential monte carlo. *arXiv preprint arXiv:1903.04797*.
- Panigrahi, P.K. and Bisoy, S.K. (2021). Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University-Computer and Information Sciences*.
- Pulido, B., Zamarreño, J.M., Merino, A., and Bregon, A. (2019). State space neural networks and model-decomposition methods for fault diagnosis of complex industrial systems. *Engineering Applications of Artificial Intelligence*, 79, 67–86.
- Qiu, Q., Fan, Z., Meng, Z., Zhang, Q., Cong, Y., Li, B., Wang, N., and Zhao, C. (2018). Extended ackerman steering principle for the coordinated movement control of a four wheel drive agricultural mobile robot. *Computers and Electronics in Agriculture*, 152, 40–50.
- Roger, E. (1987). Stan ulam, john von neumann and the monte carlo method. *Argonne, USA*.
- Sato, M. (2019). Observer-based robust controller design with simultaneous optimization of scaling matrices. *IEEE Transactions on Automatic Control*, 65(2), 861–866.
- Schön, T.B. and Lindsten, F. (2015). Learning of dynamical systems—particle filters and markov chain methods. *Draft available*.
- Simanek, J., Reinstein, M., and Kubelka, V. (2014). Evaluation of the ekf-based estimation architectures for data fusion in mobile robots. *IEEE/ASME Transactions on Mechatronics*, 20(2), 985–990.
- Teslić, L., Škrjanc, I., and Klančar, G. (2011). EKF-based localization of a wheeled mobile robot in structured environments. *Journal of Intelligent & Robotic Systems*, 62(2), 187–203.
- Tulsyan, A., Gopaluni, R.B., and Khare, S.R. (2016). Particle filtering without tears: A primer for beginners. *Computers & Chemical Engineering*, 95, 130–145.
- Von Neumann, J. and Ulam, S. (1951). Monte carlo method. *National Bureau of Standards Applied Mathematics Series*, 12(1951), 36.
- Wang, L., Zhan, R., Liu, S., Zhang, J., and Zhuang, Z. (2021). Joint tracking and classification of multiple extended targets via the phd filter and star-convex rhm. *Digital Signal Processing*, 111, 102961.
- Wang, S. and Bao, Q. (2021). Single target tracking for noncooperative bistatic radar with unknown signal illumination. *Signal Processing*, 183, 107991.
- Xu, C., Wang, X., Duan, S., and Wan, J. (2021). Spatial-temporal constrained particle filter for cooperative target tracking. *Journal of Network and Computer Applications*, 176, 102913.
- Yin, S. and Zhu, X. (2015). Intelligent particle filter and its application to fault detection of nonlinear system. *IEEE Transactions on Industrial Electronics*, 62(6), 3852–3861.
- Zhang, Q.b., Wang, P., and Chen, Z.h. (2019). An improved particle filter for mobile robot localization based on particle swarm optimization. *Expert Systems with Applications*, 135, 181–193.
- Zhibin, Z., Haibo, J., and Jie, Y. (2020). Autonomous optical navigation of mars probe aided by one-way doppler measurements in capture stage. *Journal of Systems Engineering and Electronics*, 31(3), 602–611.