

# Sumário

<b>1</b>	<b>Plano Inicial</b>	<b>2</b>
<b>2</b>	<b>Atividades realizadas</b>	<b>2</b>
<b>3</b>	<b>Descrição do Problema</b>	<b>3</b>
<b>4</b>	<b>Resultados</b>	<b>4</b>
4.1	Introdução . . . . .	4
4.2	Filtros Bayesianos . . . . .	5
4.3	Projeto e alterações . . . . .	8
4.4	Modelo Matemático . . . . .	9
4.5	Simulações e Estatísticas . . . . .	10
<b>5</b>	<b>Conclusão</b>	<b>14</b>
<b>6</b>	<b>Plano de Trabalho e Cronograma Futuro</b>	<b>14</b>
<b>7</b>	<b>Agradecimentos</b>	<b>15</b>
<b>8</b>	<b>Bibliografia</b>	<b>15</b>

# Tracking e Localização para Robôs Diferenciais

## Jogadores de Futebol

Luis Guilherme Gomes Aguiar, Takashi Yoneyama

Instituto Tecnológico de Aeronáutica

São José dos Campos, SP, Brazil

2 de agosto de 2017

### 1 Plano Inicial

#### 1º bimestre (ago/set)

- Estudo de técnicas de filtragem estatística: Filtro de Kalman, Filtro de Kalman Extendido, Unscented Kalman Filter e Filtro de Partículas.
- Implementação dessas técnicas para um problema de localização genérico, utilizando linguagem de programação de alto nível.

#### 2º bimestre (out/nov)

- Estudo do modelo dinâmico do robô e dos sensores para modelar os sensores e a odometria.

#### 3º bimestre (dez/jan)

- Projeto de filtro estocástico

### 2 Atividades realizadas

#### (ago/set/out)

- Estudo das técnicas de filtragem bayesianas Filtro de Kalman, Filtro de Kalman Extendido, Unscented Kalman Filter e Filtro de Partículas.
- Implementação dessas técnicas em problemas de localização genéricos, utilizando o software MATLAB, a fim de aplicar, compreender e comparar os algoritmos.

**(novembro)**

- Modelamento estocástico do problema em questão, a partir da análise dos sensores e da dinâmica do robô.

**(dez/jan)**

- Projeto de filtro estocástico para localização e tracking de bola e robôs no IEEE's Very Small Size Soccer.
- Implementação em MATLAB
- Simulações, obtenção de estatísticas e avaliação de resultados e performance

### **3 Descrição do Problema**

Robótica móvel tem se mostrado uma área de grande destaque no contexto atual, tanto pela abrangência e profundidade das pesquisas envolvidas quanto pelo rápido crescimento na indústria atualmente. A fim de incentivar e estimular pesquisas na área, foram criadas diversas competições nacionais e internacionais, como a RoboCup, mais famosa competição mundial criada em 1997 que tem como objetivo fomentar o avanço da robótica e inteligência artificial com o desafio de até 2050 criar um time de robôs humanoides capazes de ganhar de um time de humanos em uma partida de futebol.

Da mesma forma, vários desafios e competições foram criados com o objetivo de promover o avanço tecnológico em cada campo da robótica, como é o caso do IEEE's Very Small Size Soccer (VSS). O VSS é uma liga de futebol de robôs autônomos mantida pela IEEE Robotics and Automation Society que consiste em uma partida de futebol disputada por três robôs diferenciais com dimensões máximas de 7,5x7,5x7,5cm. As informações da partida são captadas através de uma câmera acima do campo e processadas autonomamente por um computador, que a partir desses dados toma decisões de movimento aos robôs, enviadas por rádio, de forma a ganhar a partida.

Dessa forma, o problema deve ser modelado matematicamente e cada etapa processada, como a visão computacional, o controle, o cálculo de rotas, a estratégia a ser tomada e, inclusive, a localização do robô do nosso time, do time adversário, e da bola da partida.

Localização de robôs móveis é uma etapa fundamental no desenvolvimento de robótica. A dificuldade surge a partir do fato do mundo real não ser determinístico, mas ruidoso. Um exemplo disso são os dados dos sensores empregados. Nesse caso em particular fala-se da imagem da

câmera; ou o modelo de movimento, que inclui uma diferença entre a ação comandada e a realizada pelos motores. Saber com acurácia e precisão a posição e a velocidade de robôs do próprio time e do time adversário é essencial nesse desafio, como requisitos para o desenvolvimento da estratégia de alto nível.

A abordagem estatística é amplamente aceita como a mais bem-sucedida para esse problema. Dessa forma, a observação e movimento são modelados como um processo de Markov. Os ruídos são variáveis aleatórias e os dados da odometria e visão são processados de forma a alcançar uma estimativa ótima da posição real do robô.

## 4 Resultados

### 4.1 Introdução

Como definido, o desafio lidado é o de se obter a posição e as velocidades linear e angular do robô diferencial na competição Very Small Size. A abordagem probabilística para esse problema consiste em tratar a posição do robô no campo em um dado tempo  $t$  como um vetor de variáveis aleatórias  $x_t$ . O objetivo, portanto, é obter sua função distribuição de probabilidade ao longo de todo o espaço em questão.

O estado do robô  $x_t$  deve englobar todas as grandezas necessárias para a sua localização e que tem impacto em seu estado futuro. E ainda, como desejado nesse problema, deve conter variáveis cinemáticas, como velocidades lineares e angulares.

Um robô interage com o ambiente através de sua percepção por meio de sensores e atuadores. Através de comandos de controle, incluindo sinais de referência tais como a velocidade linear, por exemplo, o robô se move, posiciona, e alcança a configuração de estado desejado.

A fim de empregar as técnicas de filtragem bayesianas amplamente usadas ([1], [2]), como o filtro de Kalman ou o filtro de partículas, a dinâmica do robô é modelada através de duas equações estocásticas: a equação de movimento, que descreve como o estado varia no tempo como efeito de sua cinemática e controle (equação (1)), e a equação de observação, que descreve como se dá a observação obtida pelos sensores em função do estado (equação (2)).

$$x_{t+1} = f(x_t, \epsilon_t) + u_t \quad (1)$$

$$z_t = h(x_t) + v_t \quad (2)$$

Na equação (1),  $\epsilon_t$  é o vetor de entradas de controle e  $u_t$  um ruído de estado. Na equação (2),  $v_t$  é também um ruído aleatório comum ao processo de observação pelo sensor.

Considerando as funções distribuição de probabilidade, a probabilidade do estado estar em  $x_t$ , condicionado ao fato de estar em  $x_{t-1}$  no instante  $t - 1$ , é dada por  $p(x_t|x_{t-1})$ , enquanto a da observação é dada por  $p(z_t|x_t)$ . A ideia central por trás dos filtros bayesianos é obter uma distribuição a posteriori para o estado do robô, mesclando as informações dos passos de movimento e observação através de um algoritmo de filtragem. A distribuição desejada, incluindo a observação no instante, seria, portanto,  $p(x_t|z_{1:t})$ . As equações (3) e (4), presentes em [1] e [3], descrevem o filtro de Bayes para as hipóteses assumidas nas equações (1) e (2).

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1}) dx_{t-1} \quad (3)$$

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{\int p(z_t|x_t)p(x_t|z_{1:t-1}) dx_{t-1}} \quad (4)$$

## 4.2 Filtros Bayesianos

Para sistemas não lineares, como descritos pelas equações (1) e (2), o filtro de Bayes, em geral, é de difícil aplicação prática. Como observa-se na equação (3) a integral descrita pode ser de difícil cálculo ou até mesmo nem ter uma expressão analítica fechada. Contudo, existem diferentes técnicas e algoritmos que fornecem aproximações satisfatórias para esse filtro, com mudanças características de hipóteses, eficiência computacional, precisão da aproximação e facilidade de implementação. Cada técnica resulta em diferentes distribuições de probabilidade finais para o estado calculado, dependendo das hipóteses assumidas para cada implementação. Com isso, foi realizado o estudo das técnicas mundialmente mais conhecidas e empregadas em localização e tracking de robô móvel, como o filtro de Kalman, e suas extensões não lineares, como o EKF e UKF, além do filtro não paramétrico mais empregado: filtro de partículas.

O filtro de Kalman é a técnica mais conhecida e já estudada de filtragem estocástica ([1] e [4]). Ele é uma solução ótima para o cálculo da distribuição posteriori em sistemas lineares sujeitos a ruídos brancos gaussianos. Para esses sistemas, a resposta a uma entrada gaussiana (ou normal) é também gaussiana. Logo, nesse algoritmo, pode-se representar a distribuição de cada estado por seus parâmetros média ( $\mu_t$ ) e covariância ( $\Sigma_t$ ). Sendo  $u_t \sim N(0, Q_t)$  e  $v_t \sim (0, R_t)$ .

$$x_t = A_t x_{t-1} + B_t \varepsilon_t + u_t \quad (5)$$

$$z_t = C_t x_t + v_t \quad (6)$$

O filtro de Kalman possui um custo computacional muito baixo, sendo, portanto, muito rápido e de boa aplicação para sistemas que demandam filtragem em tempo real. Sua desvan-

---

**Algorithm 1** Filtro de Kalman, extraído de [[1]p.42]

---

```

1: function FILTRO DE KALMAN( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ )
2:   Predição
3:    $\bar{\mu}_t = A_t \mu_{t-1} + B_t \varepsilon_t + u_t$ 
4:    $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$ 
5:   Atualização
6:    $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$ 
7:    $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$ 
8:    $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t (I - K_t C_t)^T + K_t R_t K_t^T$ 
9: end function

```

---

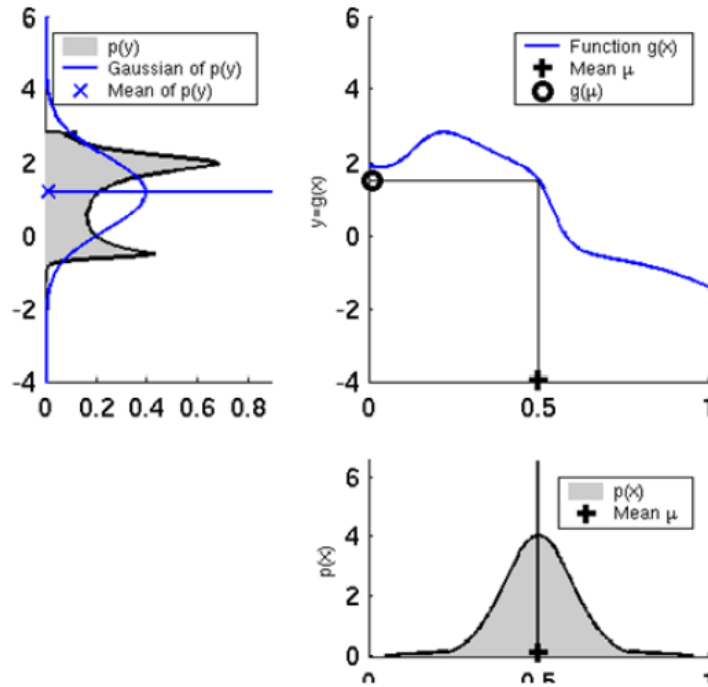


Figura 1: Linearização pelo EKF

tagem é sua limitação, sendo restrito a sistemas lineares, o que constitui casos bem restritos em problemas práticos de mundo real.

O filtro de Kalman estendido [4] é uma versão não linear do filtro de Kalman que se baseia na linearização das equações de atualização e de medida em torno do ponto de maior probabilidade, e a partir dessas linearizações aplica-se o filtro de Kalman convencional.

A partir de um sinal gaussiano para  $x_{t-1}$ , aplica-se as equações não lineares, como descritas em (1) e (2). O sinal seria então deformado por essa propagação. Aplica-se uma expansão de primeira ordem do polinômio de Taylor no ponto de concentração de maior probabilidade, e a seguir, a partir da função linear obtida, o filtro de Kalman. O processo é ilustrado na figura 1.

Observa-se que a distribuição posteriori é uma gaussiana, apesar da forma da distribuição final desejada,  $p(x_t|z_{1:t})$ , não ser necessariamente uma gaussiana. O resultado obtido é então uma aproximação na forma de gaussiana. Como pode ser observado pela matemática da técnica, a eficiência dessa aproximação pode ser limitada pelo fato do sistema e as equações de atualização ou medida serem altamente não lineares, ou ainda por uma original alta variância nos sinais filtrados. O custo computacional do algoritmo é muito baixo, tal como o filtro de Kalman.

O algoritmo do filtro de Kalman pode ser adaptado para essa linearização via expansão de Taylor. Com isso, o algoritmo seria o mesmo, com a particularidade de que  $A_t$  seria a jacobiana de  $f$  calculada no ponto  $\mu_{t-1}$ , enquanto  $B_t$  seria a jacobiana de  $h$  calculada no ponto  $\bar{\mu}_t$ .

$$A_t = \frac{\partial f}{\partial x}(\mu_{t-1}) \quad (7)$$

$$C_t = \frac{\partial h}{\partial x}(\bar{\mu}_t) \quad (8)$$

Como abordado em [5] e [6], o *Unscented Kalman Filter*, ou UKF, é outra extensão não linear do filtro de Kalman com o mesmo princípio de linearizar uma transformação de um vetor de variáveis aleatórias gaussianas. A aproximação utilizada é baseada na *unscented transformation*, e consiste em extrair deterministicamente pontos específicos, chamados *sigma points*, da entrada gaussiana e então passá-los pela função de atualização não linear  $f$ . As médias e covariâncias são extraídas com pesos em relação a esses *sigma points*.

O UKF, apesar de ser um algoritmo de compreensão complexa, possui algumas vantagens em relação ao EKF. A primeira e imediata é a ausência da necessidade de se calcular as derivadas parciais para as jacobianas presentes no EKF, o que pode ser difícil de determinar analiticamente para alguns casos. A segunda vantagem é o desempenho em precisão ser melhor ou igual ao do EKF, principalmente nos casos de alta não linearidade e variância alta na entrada. No geral o UKF é ligeiramente mais lento do que o EKF, mas ainda assim possui um custo computacional baixo. Vale lembrar que o UKF, assim como o filtro de Kalman estendido, representa as distribuições posteriores como gaussianas que melhor aproximam a distribuição pretendida  $p(x_t|z_{1:t})$ , tendo, portanto, dificuldades em representar distribuições multimodais.

O filtro de partículas é, dessa forma, uma solução a essa limitação pela sua natureza não paramétrica ([7] e [8]). O filtro busca, primeiramente, estimar a distribuição de probabilidade  $p(x_t|x_{t-1})$  propagando várias amostras, denominadas partículas, do estado  $x_{t-1}$  através da equação de estado. Após a obtenção de várias partículas propagadas, a nova etapa, denominada de filtragem, está em corrigir as posições de tais amostras com os pesos dados pela equação (9). O processo é repetido gerando-se novas partículas através de reamostragem. A ideia central do filtro é explorar a dualidade entre distribuições de probabilidade e um conjunto de amostras ge-

radas por essa função de distribuição. Além disso, qualquer expectativa da distribuição sobre o espaço pode ser calculado pelo método de Monte Carlo, que segue na equação (10). Na equação (9),  $\eta$  é uma constante de proporcionalidade usada para normalizar o valor dos pesos.

$$w_t^{[m]} = \eta w_{t-1}^{[m]} p(z_t | x_t) \quad (9)$$

$$\left[ \sum_{m=1}^M w_t^{[m]} \right]^{-1} \sum_{m=1}^M g(x_t^{[m]}) w_t^{[m]} \xrightarrow{M \rightarrow \infty} \int_A g(x_t) p(x_t | z_{1:t}) dx_t \quad (10)$$

Nas equações,  $w_t^{[m]}$  é o peso para a  $m$ -ésima partícula no instante  $t$ ,  $M$  é a quantidade de partículas, e  $g(x)$  uma função mensurável qualquer. Com o filtro de partículas, a distribuição posteriori desejada pode ser bem aproximada mesmo que seja multimodal, desde que tenhamos um número de partículas suficientemente grande. Essa é a principal vantagem do filtro de partículas. Sua maior desvantagem é o custo computacional, que pode crescer significativamente com o aumento da quantidade de partículas.

### 4.3 Projeto e alterações

A ideia inicialmente pretendida para o projeto era a de estender o estado na filtragem da posição dos robôs do nosso time de VSS, acrescentando suas velocidades linear e angular. Um modelo de filtro e sua implementação no MATLAB já foram feitos antes por outro integrante do grupo, contudo só incluía as entradas necessárias para sua localização espacial, coordenadas planas  $x$  e  $y$  e ângulo de miragem  $\psi$  do robô. Os dados de velocidades linear e angular ainda eram ruidosas, e a ideia era incluir tais grandezas no filtro estocástico. Contudo, o filtro não seria tão eficiente para essa finalidade, e a solução para diminuir o ruído nos valores de controle  $v$  e  $w$  foi um filtro passa baixas.

Outra ideia para minimizar tal incerteza, ainda a ser implementada, e que é muito usada em robótica, é ajustar os valores de controle linearmente. Dessa forma, se  $[v, w]$  são os valores de controle comandados, procura-se determinar empiricamente parâmetros  $\xi$  e  $\eta$  de forma que  $[\xi v, \eta w]$  se aproxima mais aos reais valores de controle.

O projeto focou-se então em implementar os algoritmos de tracking dos robôs adversários e da bola, utilizando os filtros bayesianos estudados e incluindo as velocidades linear e angular no estado dinâmico. Nesse caso, o filtro ofereceria uma boa estimativa sobre essas variáveis, visto que as entradas de controle do adversário são desconhecidas, e a priori não temos nenhuma informação sobre suas variáveis dinâmicas  $v$  e  $w$ .



#### 4.4 Modelo Matemático

No tracking do robô, uma vez que as entradas de controle são desconhecidas, as equações de atualização e medida serão da forma:

$$x_t = f(x_{t-1}) + M_t u_t \quad (11)$$

$$z_t = h(x_t) + \vartheta_t \quad (12)$$

O estado completo do robô diferencial pode ser descrito por suas coordenadas cartesianas planas em função do tempo,  $x_t$  e  $y_t$ , por seu ângulo de orientação em função do tempo  $\psi_t$ , definido a partir do semi-eixo positivo das abcissas com sentido trigonométrico, e pelas velocidades linear  $v_t$  e angular  $w_t$ . Dessa forma, o modelo contínuo de atualização na posição é dado pela equação (13).

$$\begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\psi}_t \\ \dot{v}_t \\ \dot{w}_t \end{bmatrix} = \begin{bmatrix} v_t \cos \psi_t \\ v_t \sin \psi_t \\ w_t \\ a \\ \alpha \end{bmatrix} \quad (13)$$

O modelo discreto de movimento é obtido a partir da integração da equação (13) no intervalo  $[(t-1), t]$ . O intervalo de tempo entre dois instantes consecutivos é constante  $T$ . Para as atualizações de  $x$  e  $y$  considera-se um movimento uniforme de velocidades constantes  $v_{t-1}$  e  $w_{t-1}$ , já que a integração com acelerações linear e angular não possui uma expressão analítica simples, não sendo, portanto, adequada para ser incluída na formulação do EKF, e além disso os ruídos aditivos atenuam parcialmente essa falha de modelo. Por último, as acelerações serão inseridas no modelo dinâmico como valores aleatórios inclusos no ruído gaussiano aditivo  $u_t \sim N(0, Q)$ . No ruído da observação temos também  $v_t \sim N(0, R)$ .

$$\begin{bmatrix} x_t \\ y_t \\ \psi_t \\ v_t \\ w_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + \frac{2v_{t-1}}{w_{t-1}} \sin\left(\frac{w_{t-1}T}{2}\right) \cos\left(\psi_{t-1} + \frac{w_{t-1}T}{2}\right) \\ y_{t-1} + \frac{2v_{t-1}}{w_{t-1}} \sin\left(\frac{w_{t-1}T}{2}\right) \sin\left(\psi_{t-1} + \frac{w_{t-1}T}{2}\right) \\ \psi_{t-1} + w_{t-1}T \\ v_{t-1} \\ w_{t-1} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \frac{T^2}{2} \\ 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & T \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_\psi \\ a \\ \alpha \end{bmatrix} \quad (14)$$

Tabela 1: Covariâncias - Tracking adversário

$$Q = \begin{bmatrix} 0.15 & 0 & 0 & 0 & 0 \\ 0 & 0.15 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.15 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

$$z_t = \begin{bmatrix} x_t \\ y_t \\ \psi_t \end{bmatrix} + \begin{bmatrix} \vartheta_x \\ \vartheta_y \\ \vartheta_\psi \end{bmatrix} \quad (15)$$

O modelo para o tracking da bola segue a mesma lógica, contudo não temos mais uma orientação. A bola possui liberdades de movimento independentes em  $x$  e  $y$ .

$$\begin{bmatrix} x_t \\ y_t \\ v_{x,t} \\ v_{y,t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ v_{x,t-1} \\ v_{y,t-1} \end{bmatrix} + \begin{bmatrix} 1 & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & \frac{T^2}{2} \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & T \end{bmatrix} \begin{bmatrix} \beta_x \\ \beta_y \\ a_x \\ a_y \end{bmatrix} \quad (16)$$

$$z_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} \quad (17)$$

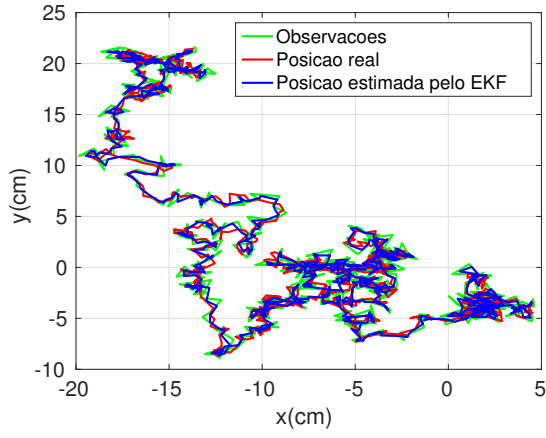
#### 4.5 Simulações e Estatísticas

Para implementar os modelos de tracking explicitados, foi empregado o filtro de Kalman extendido. A técnica serviu bem aos modelos, já que não temos nenhuma equação de atualização altamente não linear que prejudique a aproximação do filtro, e além disso o EKF é relativamente fácil de aplicar nesse caso e tem o custo computacional conhecidamente muito baixo.

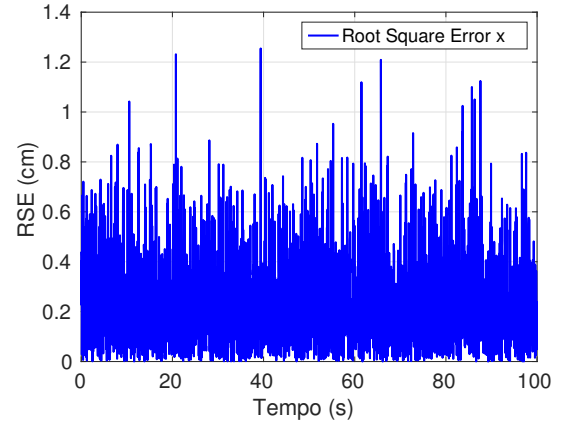
Por enquanto, o filtro foi implementado apenas em MATLAB e extraímos a simulação considerando tempo total de 100s e intervalo de  $\frac{1}{30}s$  entre cada iteração. Os valores empregados para  $Q$  e  $R$  seguem na tabela 1.

Os resultados gráficos para a simulação do tracking do robô adversário podem ser vistos na figura 2. A figura 2a mostra as trajetórias real, estimada e a calculada pela observação apenas.

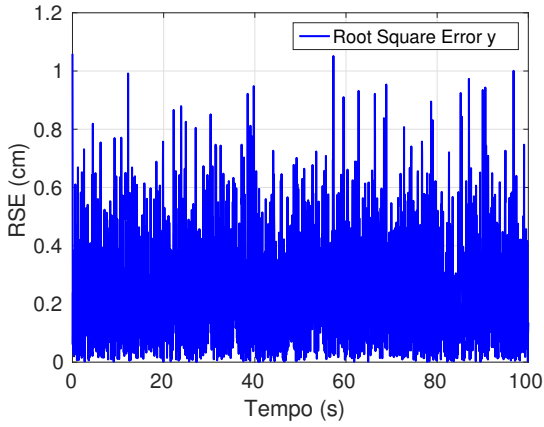
As figuras 2b, 2c, 2d, 2e e 2f mostram o erro quadrático da estimativa em relação ao real ao longo da simulação para os valores de  $x$ ,  $y$ ,  $\psi$ ,  $v$  e  $w$ , respectivamente.



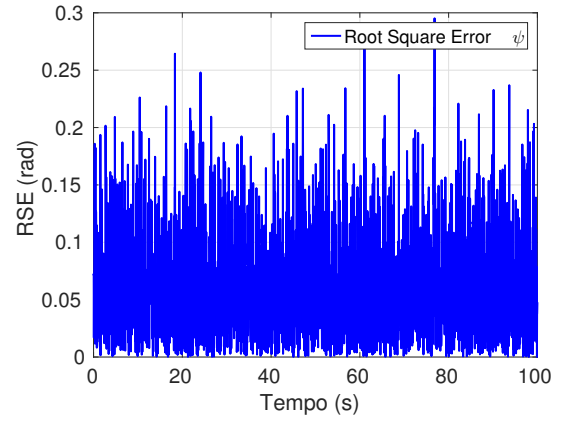
(a) Trajetórias - Simulação



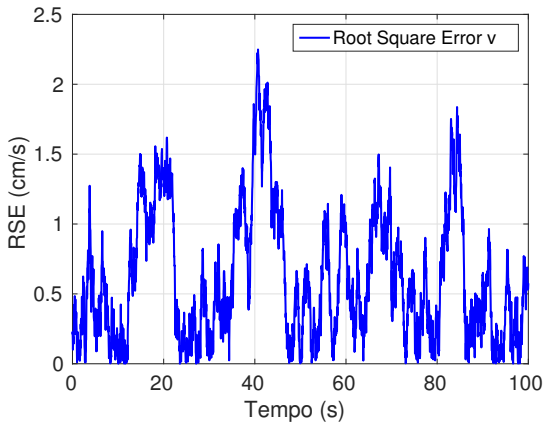
(b) RSE- $x$



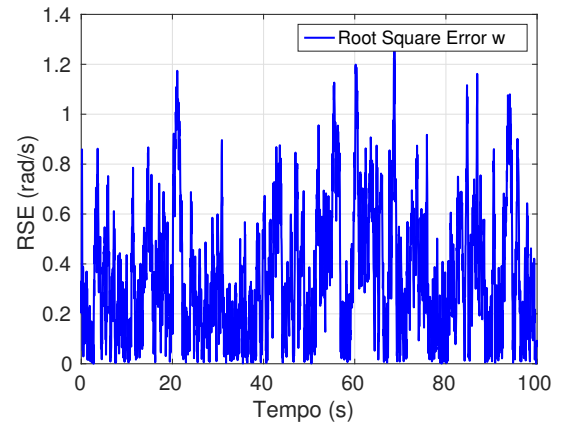
(c) RSE- $y$



(d) RSE- $\psi$



(e) RSE- $v$



(f) RSE- $w$

Figura 2: Trajetórias e Root Square Error em  $x$ ,  $y$ ,  $\psi$ ,  $v$  e  $w$  - Tracking adversário

Para o cálculo dos erros quadráticos médios da estimativa e da observação para o tracking do robô adversário, a fim de comparação entre eles e avaliação do desempenho do filtro, a simulação

Tabela 2: RMSE - Tracking adversário

Estimativa		Observação	
$x$	0.2433	$x$	0.3093
$y$	0.2431	$y$	0.3089
$\psi$	0.0631	$\psi$	0.0798
$v$	0.6468	$v$	19.9163
$w$	0.3795	$w$	4.1417

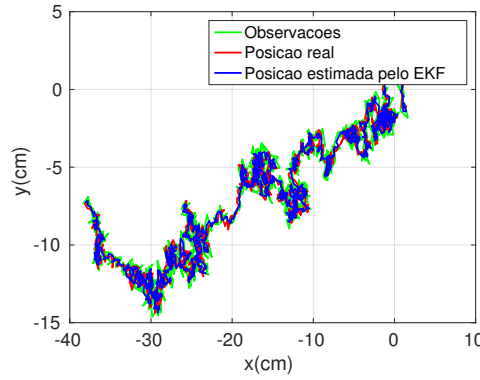


Figura 3: Trajetórias simulação - Tracking bola

Tabela 3: Covariâncias - Tracking bola

$$Q = \begin{bmatrix} 0.05 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

descrita acima foi repetida 500 vezes, computando-se então a média dos erros médios das 500 simulações. Os resultados seguem na Tabela 2. As velocidades linear e angular segundo os dados da visão foram calculadas por derivação numérica, empregando os valores de  $x$  e  $y$  observados em duas iterações consecutivas.

Para o tracking da bola, também foi empregada uma simulação de 100s e passo de  $\frac{1}{30}s$ . Os erros quadráticos médios também foram calculados considerando uma média com 500 simulações. Os resultados seguem na tabela 4 e os valores usados para  $Q$  e  $R$  seguem na tabela 3. Os resultados gráficos para uma simulação estão presentes nas figuras 3 e 4.

Os valores empregados para as matrizes de covariância são apenas estimativas por enquanto,

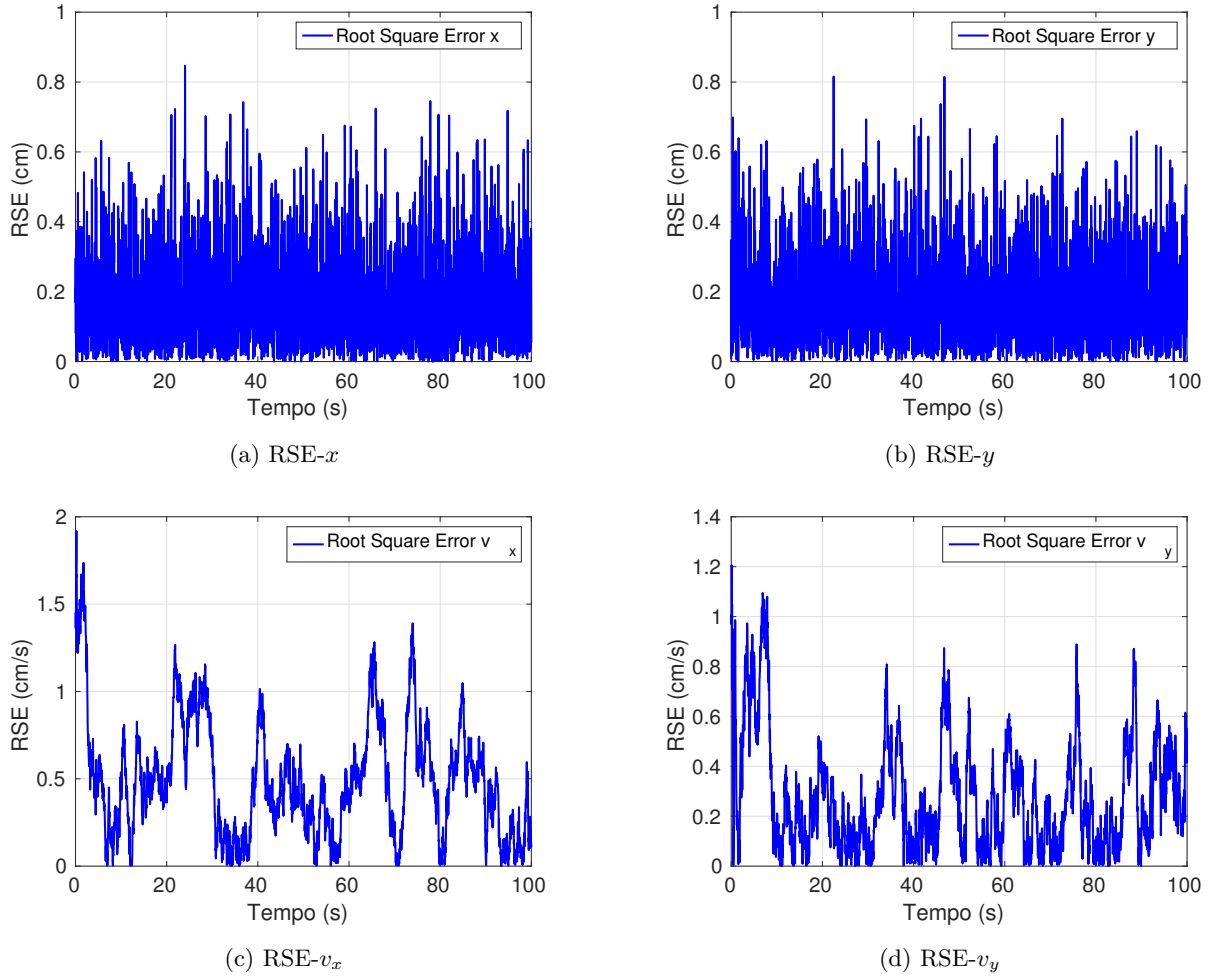


Figura 4: Root Square Error em  $x$ ,  $y$ ,  $v_x$  e  $v_y$  - Tracking bola

Tabela 4: RMSE - Tracking bola

Estimativa		Observação	
$x$	0.1788	$x$	0.2523
$y$	0.1788	$y$	0.2521
$v_x$	0.3915	$v_x$	11.9652
$v_y$	0.3863	$v_y$	11.9559

posteriormente planeja-se tirar um modelo mais preciso dos erros do movimento e da observação da câmera. A estimativa foi feita supondo um erro médio de aproximadamente  $0.3cm$  em  $x$  e  $y$  e  $5$  graus em  $\psi$  para os dados da visão, além de considerar uma variância bem alta para as acelerações.

Analisando os resultados, é possível constatar que o filtro de Kalman ofereceu um bom ganho de precisão em relação à observação. O erro médio da estimativa é de aproximadamente 78% do

erro da visão, para o tracking do adversário, e de 70% para o tracking da bola. Além disso, o filtro fornece uma boa estimativa do valor das velocidades, linear e angular, o que não é possível alcançar só com a derivação numérica da posição com a informação da câmera, como percebe-se nos valores revelados nas tabelas 2 e 4. Observa-se também pelas figuras 2, 3 e 4 que o filtro mantém sempre o rastreo do objeto, e o erro quadrático não diverge em nenhum momento.

## 5 Conclusão

Através do estudo realizado, foi possível conhecer bem os principais filtros bayesianos empregados para problemas de tracking e localização global em robótica, assim como implementá-los em problemas de localização genéricos com linguagem de alto nível e entender suas aplicações e particularidades.

Para o desafio desse projeto, que consiste em rastrear e estimar as velocidades de robôs diferenciais e da bola na competição IEEE's Very Small Size Soccer (VSS), constatou-se a utilidade e aplicação dessas técnicas, em especial o filtro de Kalman estendido, uma extensão não linear do Filtro de Kalman. O resultado obtido através da simulação em MATLAB do problema com essa técnica de filtragem revelou um bom desempenho, mostrando sua eficácia em obter uma boa estimativa da posição do objeto e de suas velocidades linear e angular.

Tendo em vista esse resultado positivo, o próximo objetivo é aplicar a técnica explicada em robôs reais, utilizando um time de 3 robôs e câmera, além de integrar ao código principal da equipe ITAndrois. Além disso, espera-se também obter no futuro melhores estimativas para as covariâncias e parâmetros empregados, utilizando experimentos e técnicas de identificação de sistemas.

## 6 Plano de Trabalho e Cronograma Futuro

### março/abril

- Implementação do filtro no sistema embarcado do robô
- Criação de ferramentas de teste e análise do desempenho do código.

### maio/junho

- Testes no robô
- Otimização de parâmetros do filtro

### julho

- Integração com o código do time da ITAndroids
- Testes no robô
- Confecção do relatório final

## 7 Agradecimentos

Agradeço ao CNPQ, por prover ajuda financeira para realizar esse projeto. Agradeço à ITAndroids, por compartilhar conhecimento e ferramentas que tornam esse projeto possível.

## 8 Bibliografia

- [1] S. Thrun, W. Burgard e D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [2] D. Grisan e B. Rojovskii, *The Oxford Handbook of Nonlinear Filtering*. Oxford Univ. Press, 2011.
- [3] Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 1979.
- [4] B. D. O. Anderson e J. B. Moore, *Optimal Filtering*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 2013.
- [5] S. Julier e J. Uhlmann, *Unscented filtering and nonlinear estimation*. Proceedings of the IEEE 92(3), pp. 401–422.
- [6] E. A. Wan e R. V. D. Merwe, *The unscented kalman filter for nonlinear estimation*, pp. 153–158.
- [7] F. Dellaert, D. Fox, w. Burgard e S. Thrun, “Monte carlo localization for mobile robots.”, *ICRA*,
- [8] B. Ristic, S. Arulampulam e N. Gorlon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2014.