



**Programa de Pós Graduação em Computação**

Mestrado em Engenharia de Computação

Universidade Federal do Rio Grande - FURG



# **Implementações do Filtro de Kalman para sincronização GPS-IMU**

**Oscar Arrúa**

[oarrua88@gmail.com](mailto:oarrua88@gmail.com)

Rio Grande, Maio 2020

\*AutoCERES



# Fusão Sensorial

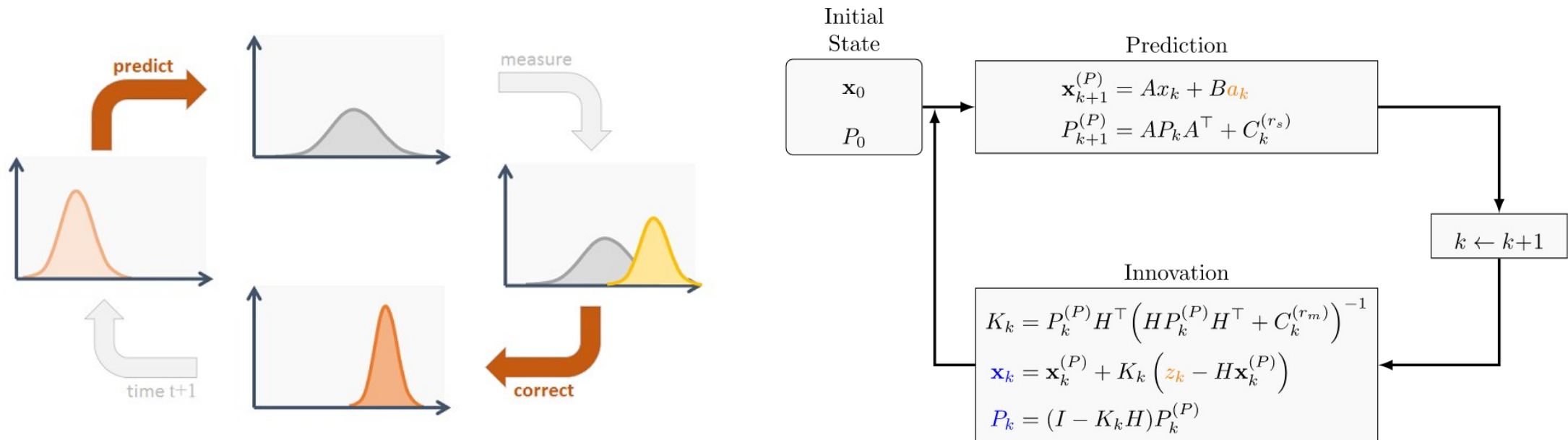
O trabalho consiste em melhorar a precisão dos dados de localização utilizando técnicas de filtragem. Para isso, são combinadas as informações de dois tipos de sensores através da implementação do Filtro de Kalman. Dispositivos utilizados:

- **IMU:** É um dispositivo eletrônico que mede a velocidade, orientação e forças gravitacionais de um aparelho, usando uma combinação de acelerômetros e giroscópios. IMUs de baixa qualidade (comum em smartphones) conseguem manter apenas por alguns poucos segundos a confiabilidade de sua navegação, necessitando desde o início de seu funcionamento de sensores externos, para que possa proporcionar uma performance aceitável.
- **GPS:** É um sistema de posicionamento via satélite desenvolvido pelos Estados Unidos da América que possui cobertura total do planeta, com exceção dos polos. Normalmente, a quantidade mínima de satélites para se conseguir calcular as coordenadas é 3. Porém, um número maior de satélites pode ser necessário para melhorar a qualidade do dado de localização.



# Fusão Sensorial

Desde a sua publicação, o filtro de Kalman vem sendo aplicado em diversas áreas, tais como: biomedicina, navegação e sensoriamento. Seu princípio de funcionamento consiste em uma estimação recursiva de estados de um sistema dinâmico sobre influência de um ruído branco. Para isso, utiliza valores médios das leituras dos sensores do sistema junto a variância também destes sensores, obtidas previamente, afim de estimar o melhor estado estimado  $\hat{\mathbf{x}}_t$  para o vetor  $\mathbf{x}_t$  de estados do sistema. O filtro de Kalman acontece em dois passo discretos: predição e correção.





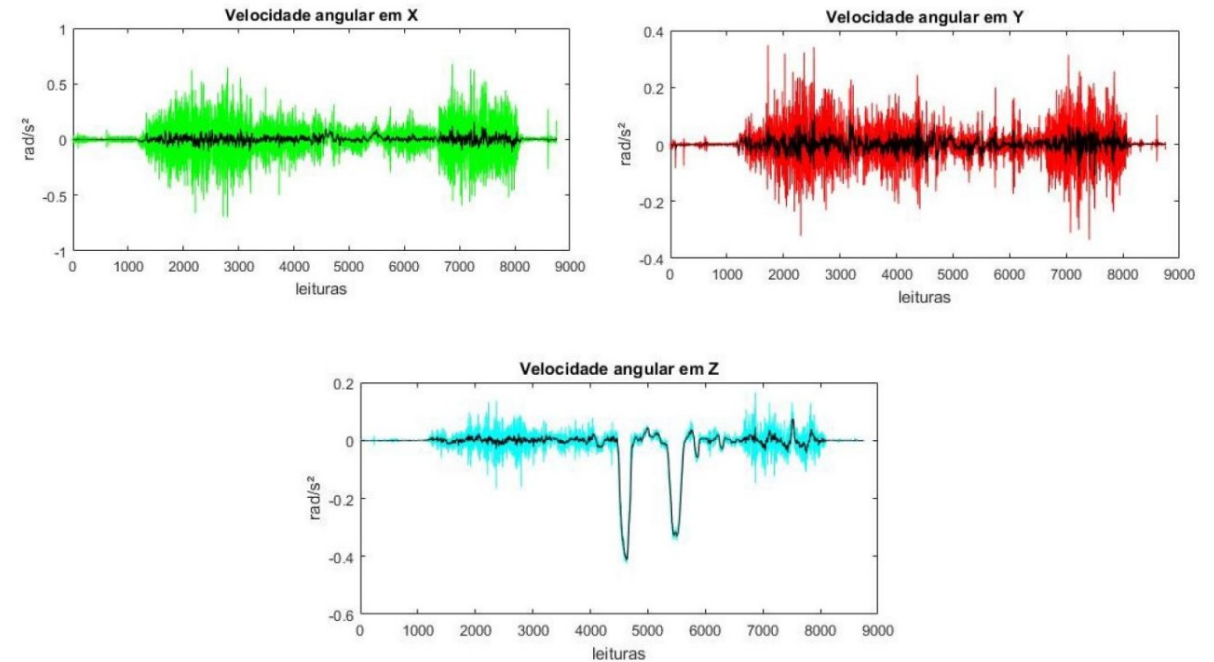
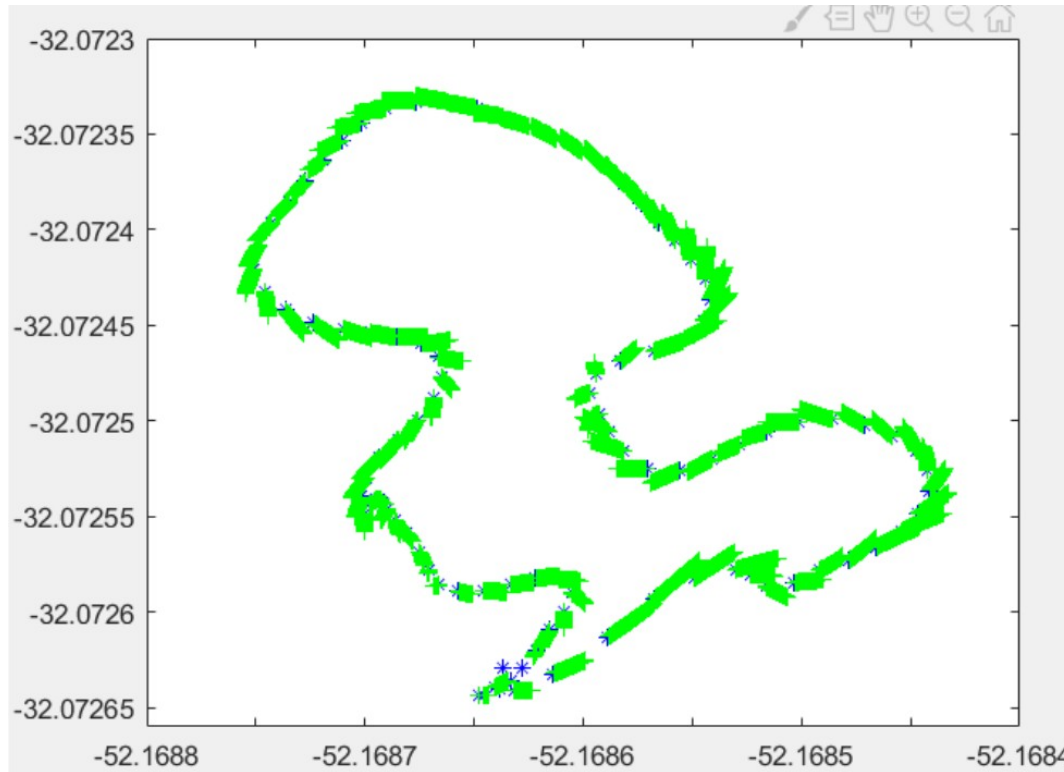
# Implementações

Sistema	Status
Matlab	Pronto
Java	Pronto
Android	Em desenvolvimento



# Matlab

Utiliza os dados coletados pelo smartphone para calcular as varianzas e as previsões.



- Também inclui correção da aceleração e da velocidade angular.



# Java

Utiliza os dados coletados pelo smartphone para calcular as varianzas e as previsões. Para realizar as operações de álgebra linear, utiliza a biblioteca **eigen** (<http://eigen.tuxfamily.org>)

```
oarrua@oarrua: ~/Desktop/Jar
File Edit View Search Terminal Help
(3_7) oarrua@oarrua:~/Desktop/Jar$ java -jar kalman-filter-1-0.jar -path /home/oarrua/
/Desktop/Input -output /home/oarrua/Desktop/Output
CALCULAR VARIÂNCIAS

Lat: 2.04487E-9 Lat: 1.49857E-9

AccX: 2.0448667205725195E-9      AccY: 1.4985666633650362E-9

Ang: 1.5941591562876796E-10
DONE
(3_7) oarrua@oarrua:~/Desktop/Jar$
```

```
oarrua@oarrua: ~/Desktop/Jar
File Edit View Search Terminal Help
ANGULOS:
0: -4.94833E0    1: 8.52804E1    2: 6.43497E-1    3: -5.02157E0    4: 6.64333E0    5: 2.
11352E0 6: -2.35166E0
7: 6.89782E0    8: 1.31471E1    9: -3.21265E1    10: 1.9973E0    11: 1.14001E1

n = 66

VALORES ESTIMADOS:
0: -5.21686E1    1: -3.20726E1    2: 6.43632E-7    3: 4.37799E-7    4: -1.78769E-7    5: 4.
70514E-7        6: -4.94074E0

ANGULOS:
0: -4.94074E0    1: 8.53728E1    2: 5.93128E-1    3: -5.01349E0    4: 6.58591E0    5: 1.
87888E0 6: -2.54915E0
7: 6.86663E0    8: 1.32559E1    9: -3.20204E1    10: 1.94446E0    11: 1.15857E1
```



# Android

Realiza as previsões em tempo real, coletando e combinando os dados dos dois sensores .



```
if (meuBluetoothAdapter.isEnabled())  
    meuBluetoothAdapter.disable();  
// Do nothing  
dialog.dismiss();  
}  
});  
AlertDialog alert = solicitaConexaoArduino.create();  
alert.show();  
  
temporizador = new Timer();  
rotinaIMU = new TaskIMU();  
rotinaGPS = new TaskGPS();  
rotinaKalman = new TaskKalman();  
  
temporizador.schedule(rotinaIMU, 20, 20);  
temporizador.schedule(rotinaGPS, 1000, 1000);  
temporizador.schedule(rotinaKalman, 20, 20);
```



# Inconveniência na implementação no Android

O principal problema para implementar o Filtro de Kalman no Android é encontrar uma biblioteca que permita calcular operações com matrizes. A biblioteca **jeigen** tem um inconveniente para ser implementado, ele cria arquivos temporários que reconhecem apenas os diretórios Windows, Linux e Mac. No caso de optar por continuar usando a biblioteca **jeigen**, seria conveniente atualizar seu código para os sistemas Android. Ele está disponível em:

**<https://github.com/hughperkins/jeigen>**

Outra alternativa seria implementar manualmente as operações de álgebra linear, que incluem adição, subtração, multiplicação, divisão, transposição de matrizes de diferentes ordens ou tamanhos.





# Efficient Java Matrix Library (EJML)

**Efficient Java Matrix Library (EJML)** é uma biblioteca de álgebra linear para manipular todo tipo de matrizes. Seus objetivos são: ter a maior eficiência computacional e de memória possível para matrizes pequenas e grandes; e ser acessível a iniciantes e especialistas.




Esta biblioteca foi adicionada recentemente ao projeto Android, possui todas as operações com matrizes necessárias e seu manual está disponível em:

**<http://ejml.org/wiki/index.php?title=Manual>**



# Gitlab do Projeto

<https://gitlab.com/oarrua881/autoceres>

Name	Last commit	Last update
 Kalman_Java	Lattest changes	5 hours ago
 Kalman_Matlab	First commit	10 months ago
 SmartMapping	Lattest changes	5 hours ago



**Obrigado!**