



III Encontro de Iniciação Científica e Tecnológica  
III EnICT  
ISSN: 2526-6772  
IFSP – Câmpus Araraquara  
19 e 20 de Setembro de 2018



## Filtro de Kalman aplicado a fusão sensorial com modelo tridimensional para aplicação em drone

Caio Beltani da Silva Fernandes<sup>1</sup>, Francisco Vargas Javier<sup>2</sup>, Fabiana Florian<sup>3</sup>

<sup>1</sup> Graduando em Engenharia Elétrica, Universidade de Araraquara-UNIARA, cbs.belt@hotmail.com.

<sup>2</sup> Docente do curso de Engenharia Elétrica, Universidade de Araraquara-UNIARA

<sup>3</sup> Docente do curso de Engenharia Elétrica, Universidade de Araraquara-UNIARA

Área de conhecimento (Tabela CNPq): Instrumentação Eletrônica – 3.04.02.04-2

**RESUMO:** Em um mundo onde se faz necessário a tradução de sinais analógicos para o ambiente virtual (digital), este artigo tem como objetivo demonstrar uma técnica, utilizando a plataforma *open-source* Arduino™, capaz de minimizar erros e instabilidade em sensores ou processos. Aplicado em diversas áreas de conhecimento este artigo tem como foco a área da aeronáutica, mais especificadamente a correção das instabilidades dos sensores giroscópio e acelerômetro em quadri-rotor ou popularmente drones. Pelo fato de ser um estudo preliminar incrementou-se um *software* 3D capaz de simular e demonstrar a melhora significativa na leitura dos dados. Conclui-se que a implementação do filtro de Kalman é uma ferramenta capaz de gerar dados confiáveis e estáveis tornando-se ideal para aplicação em drones.

**PALAVRAS-CHAVE:** arduino™; acelerômetro; aeronáutica; *drone*; giroscópio; processing™

## INTRODUÇÃO

O artigo apresenta o estudo do Filtro de Kalman aplicado à fusão sensorial, especificadamente fusão entre os sensores Acelerômetro e Giroscópio, a fim de minimizar problemas específicos de cada um dos sensores. Ressalta-se que o Acelerômetro gera instabilidade de leitura quando exposto a um ambiente com trepidação enquanto que o Giroscópio tem como principal distúrbio o acúmulo de erros de leitura ao passar do tempo.

O Filtro de Kalman é uma ferramenta matemática desenvolvida por Rudolf Kalman no ano de 1958 e conta com uma alta aplicabilidade, destacam-se; área aeronáutica, financeira e telecomunicações (WELCH; BISHOP, 2006). Uma das vantagens desse filtro é a sua volatilidade, em que normalmente, torna-se impossível fazer medições reais dos dados, por isso, se faz necessária a utilização de algoritmos capazes de estimar dados passados, presentes e até mesmo prever estados futuros.

Este artigo aborda um estudo preliminar voltado para a área aeronáutica, mais especificadamente, o desenvolvimento de um quadri-rotor, veículo aéreo com quatro motores *brushless* DC controlado remotamente. Apresenta-se neste artigo um algoritmo para a estabilização da leitura dos sensores Acelerômetro e Giroscópio utilizando a plataforma *open-source* Arduino™ como o dispositivo responsável por executar as lógicas de programação além de ser a plataforma responsável pela leitura dos dados provenientes da IMU GY-80, unidade essa em que estão os sensores.

Por se tratar de um estudo preliminar foi desenvolvido um algoritmo na plataforma *open-source* Processing™, capaz de traduzir os movimentos da placa física (sinais analógicos) ao ambiente virtual, a fim de demonstrar na prática a estabilização do sinal por meio da fusão sensorial.

Torna-se viável o estudo deste tema, primeiramente pela sua grande atuação em diversas áreas da engenharia e principalmente por ser parte inicial do desenvolvimento de um drone servindo deste modo com uma direção a novos projetos.

## FUNDAMENTAÇÃO TEÓRICA

### ALGORÍTIMO DE KALMAN

O Filtro de Kalman estima seus dados a partir da resposta do sistema, isso é, o filtro estima o estado do processo em algum momento do tempo e então obtém uma resposta, sendo essa resposta normalmente uma medição ruidosa. (WELCH; BISHOP, 2006)

Dentro do algoritmo de Kalman existem dois grupos de equações responsáveis pelo tratamento dos sinais. O primeiro grupo é responsável pelas equações de "atualização de tempo" enquanto que o segundo cuida da "atualização das medições". (WELCH; BISHOP, 2006)

Segundo WELCH; BISHOP (2006) as equações de atualização de tempo são responsáveis por projetar o estado atual um passo à frente e também fazer a estimativa do erro de covariância para obter a aproximação a priori para o próximo passo de tempo. Da mesma forma as equações de atualização das medições são responsáveis por gerar a resposta filtrada, de modo que são responsáveis por incorporar a nova medida a priori para obter uma melhor estimativa a posteriori.

As equações de atualização de tempo podem ser denominadas também como equações de predição, ou seja, ato de afirmar o que vai acontecer no futuro enquanto que as equações de medição podem ser representadas como equações de correção. O conjunto das equações de predição e correção formam o ciclo recursivo o que garante a confiabilidade do Filtro de Kalman nas mais variadas aplicações numéricas. (WELCH; BISHOP, 2006)

### DESENVOLVIMENTO MATEMÁTICO DO FILTRO DE KALMAN

O modelamento do Filtro de Kalman ocorre em duas etapas: etapas de predição e de correção. Na primeira etapa ocorre a predição sobre a dinâmica do sistema enquanto que a segunda atua na correção da covariância do sistema. Um ponto importante é que para ser obter uma resposta ótima do filtro é necessário que a situação problema seja um sistema linear com erros gaussianos brancos. Salienta-se que um erro gaussiano branco tem como característica um sinal aleatório que mesmo em diferentes frequências mantenha sua intensidade idêntica (OLIVEIRA, 2017)

O método utilizado por essa técnica constitui um processo cíclico de estimação, isso é, baseia-se em um sistema recursivo no qual através de uma leitura denominada "variável de observação" consegue-se estimar os dados passados, presentes e até mesmo prever estados futuros, sendo estes denominados "variável de estado." (WELCH; BISHOP, 2006)

O Filtro de Kalman em geral tenta estimar o estado  $x \in \mathbf{R}$  de um processo discreto no tempo definido por uma equação linear estocástica, tal como equações (1) e (2). (WELCH; BISHOP, 2006)

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (1)$$

$$z_k = Hx_k + v_k \quad (2)$$

onde,

$x$  – variável de estado do modelo matemático (temperatura, pressão, altitude..)

$z$  – variável de medida (valor real medido + ruído)

$v_k$  – ruído

$w_k$  – ruído

$A$  – matriz construtiva (datasheet)

$B$  – matriz construtiva (datasheet)

$H$  – matriz de observação

Vale ressaltar que essas equações estão representadas no sistema espaço de estados, tendo então que  $A$ ,  $B$  e  $H$  são matrizes. As matrizes " $A$ " e " $B$ " referem-se a forma construtiva do sensor, podendo ser obtidas

no *datasheet* do fabricante e "H" o modelo de observação do sistema, já que o modelo não pode ser observado diretamente.

De certo modo a matriz "A" refere-se o estado anterior de k-1 ao estado presente de k enquanto que "B" refere-se a entrada do controle ( $u_k$ ) opcional ao estado "x". Assume que a variável de entrada

$u_k$  também é dada em graus por segundos, onde  $u_k = \theta_k^*$ .

Para este artigo considera-se: A e B sendo a forma construtiva do acelerômetro (3) e (4).

$$A = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \quad (3)$$

$$B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \quad (4)$$

E define-se também a variável de estado. (5)

$$x_k = \begin{bmatrix} \theta \\ \theta_b \end{bmatrix} \quad (5)$$

onde,

$\theta$  – saída real do giroscópio

$\theta_b$  – escorregamento em graus por segundo

Conhecendo a variável  $z_k$  têm se o modelo de observação H:

$$H = [1 \quad 0] \quad (6)$$

Das equações 1 e 2 observa-se que os  $w_k$  e  $v_k$  são coeficientes variáveis não dependentes do sistema e aleatórias, esses coeficientes nada mais são do que os ruídos do sistema e da medição, respectivamente.

Na prática as matrizes "Q" e "R" podem variar ao decorrer do tempo, porém, é adotado um valor constante para cada matriz. O termo covariância refere-se ao nível que duas variáveis aleatórias têm de dependência uma da outra. No caso dessas duas variáveis analisadas serem totalmente independentes tem-se que o valor de sua covariância é zero (WELCH; BISHOP, 2006).

## AS EQUAÇÕES DE KALMAN E SEUS PARÂMETROS

De forma a deixar mais intuitiva a apresentação do conjunto matemático de Kalman toma-se a Figura 1 como base.

Já mencionada a característica recursiva do Filtro de Kalman, a Figura 1 apenas evidência e confirma o que foi apresentado. Fazendo a interpretação dessa mesma figura nota-se a presença das duas etapas já mencionadas, sendo elas a de predição e posteriormente a de correção.

Cada uma dessas etapas é descrita por um conjunto de equações; a etapa de atualização do tempo (predição), composta pela Equação 1 que projeta o estado a frente e também pela Equação 2, responsável pela projeção do erro de covariância a frente. Enquanto que a segunda etapa, atualização da medição (correção), é constituída pelo cálculo do ganho de Kalman (1), atualização do estado com a medida  $z_k$  (2) e por finalizar a atualização do erro de covariância.

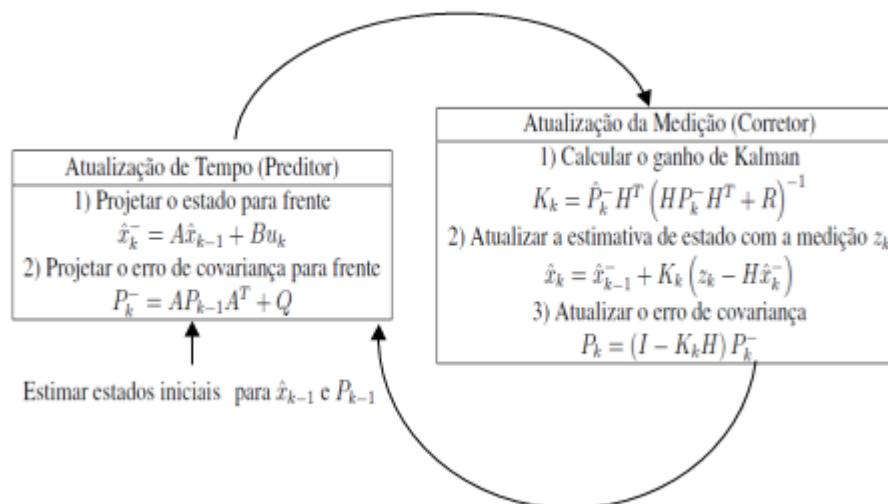


FIGURA 1. Representação do algoritmo de Kalman 1

Fonte: Adaptação - WELCH; BISHOP, 2006.

## ARDUINO UNO R3™

Arduino™ é uma placa de prototipagem eletrônica, desenvolvida na Itália meados de 2005. No início foi utilizada no meio acadêmico, para cunho educacional, porém, fez tanto sucesso que expandiu para o mundo todo. Vale ressaltar que o Arduino é uma plataforma *open source*, não há restrição para utilização. (MCROBERTS, 2015)

O termo Arduino™ estende-se tanto ao seu *hardware*, constituído simplificada por um microcontrolador, chip ATmega328, que conta com 20 pinos de entrada/saída, dentre os quais seis (6) para entradas analógicas e seis (6) para saídas PWM (*pulse with modulation*) quanto ao seu *software*, constituído por uma IDE (*Integrated Development Environment*), onde se desenvolve toda a programação em C/C++ (MCROBERTS, 2015)

## PROCESSING™

O Processing™ é uma linguagem de programação muito utilizada para desenvolvimento de artes visuais, foi desenvolvida por Casey Reas e Bem Fry em meados de 2001. Diferentemente do Arduino™ o Processing™ não possui Hardware, é apenas um software com uma IDE, ambiente de desenvolvimento integrado, onde toda sua programação é desenvolvida. (PROCESSING)

Utiliza a linguagem Java, C, C++ para programação, porém, seus desenvolvedores utilizam recursos personalizados para efeitos visuais entre as mais diversas aplicações. (PROCESSING)

Por ser uma poderosa ferramenta gráfica existem inúmeras aplicações, como; aplicação em jogos, desenvolvimento de artes conceituais, demonstração de gráficos e até a comunicação com periféricos eletrônicos. (PROCESSING)

## METODOLOGIA

A metodologia aplicada nesse artigo estende-se a montagem de um Arduino™ comunicando com os sensores (*hardware*) além da programação dos *softwares open-source* Arduino™ e Processing™, responsáveis por: leitura, processamento dos dados gerados pelos sensores giroscópio, acelerômetro e modelagem tridimensional para visualização dos resultados da aplicação do filtro de Kalman, respectivamente.

Para este artigo utilizou-se uma adaptação brasileira do Arduino UNO R3™, conhecida como Blackboard UNO R3™. Essa placa foi desenvolvida pela empresa brasileira Robocore™.

A Blackboard™ utiliza como microcontrolador o chip ATmega328, mesmo microcontrolador do Arduino UNO R3™ que conta com 20 pinos de entrada/saída, dentre os quais seis (6) são para entradas

analógicas e seis (6) para saídas PWM (pulse with modulation). Tem-se na Figura 2 a IDE do Arduino™, já programado para leitura dos sensores.

```

Acel_Giro | Arduino 1.6.13
Arquivo Editar Sketch Ferramentas Ajuda
Acel_Giro $
#include <Wire.h> // Biblioteca I2C

//----- CONFIGURAÇÃO DOS REGISTRADORES DO ACCELEROMETR

#define Register_ID 0
#define Register_Z0 0x2D // Acesso ao Eixo Z dado 0
#define Register_X0 0x32 // Acesso ao Eixo X dado 1
#define Register_Y0 0x34 // Acesso ao Eixo Y dado 0
#define Register_Y1 0x35 // Acesso ao Eixo Y dado 1
#define Register_Z0 0x36 // Acesso ao Eixo Z dado 0
#define Register_Z1 0x37 // Acesso ao Eixo Z dado 1

int I2CAddress = 0x53; // Endereço para comunicação via
int reading = 0;
int val=0;
int X0,X1,X_out;
int Y0,Y1,Y_out;
int Z1,Z0,Z_out;

float TangTeta, TetaRad, TetaGrau;
float TangFi, FiRad, FiGrau;

double Xg,Yg,Zg;

//----- CONFIGURAÇÃO DOS REGISTRADORES DO GIROSCOPIO

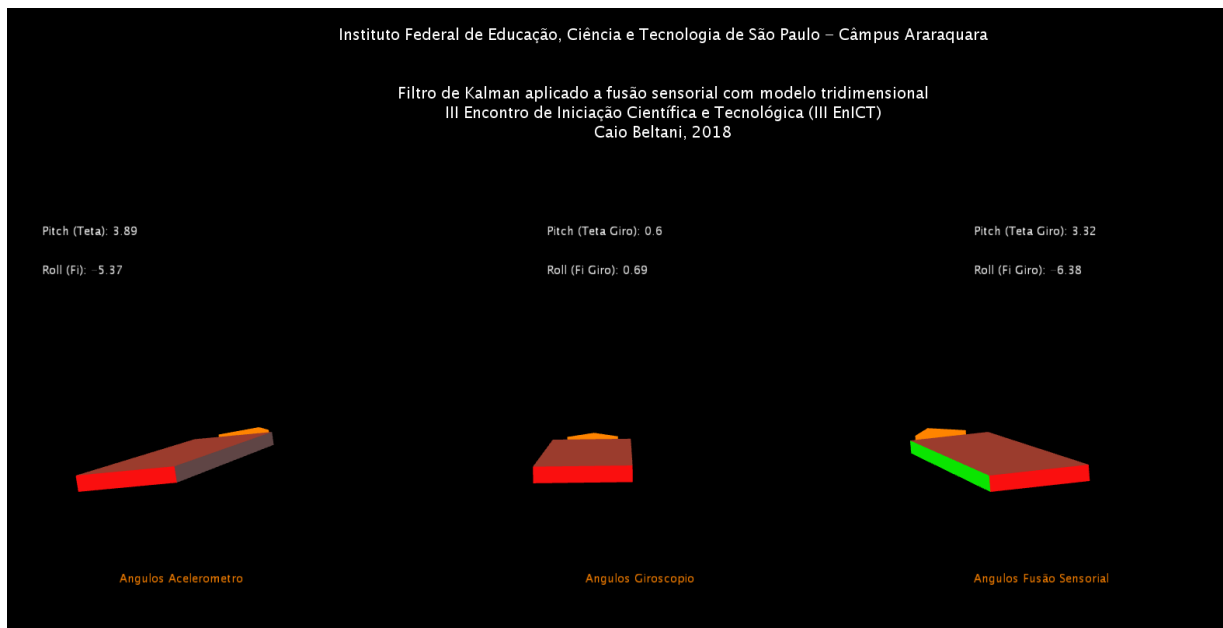
#define Gyro_g0 0x28
#define Gyro_gX1 0x29
#define Gyro_gY0 0x2a
#define Gyro_gY1 0x2b
#define Gyro_gZ0 0x2c

```

**FIGURA 2. IDE Arduino**

Fonte: Arquivo pessoal, 2018.

Utilizando o *software* Processing foi possível desenvolver um modelo tridimensional capaz de simular virtualmente os dados processados pela plataforma Blackboard™. Na Figura 3 mostra-se a aparência do *software* desenvolvido para análise em tempo real do funcionamento do filtro, em que é possível analisar, na prática os defeitos, já mencionados, dos sensores acelerômetro e giroscópio além do efeito de correção do filtro de Kalman.

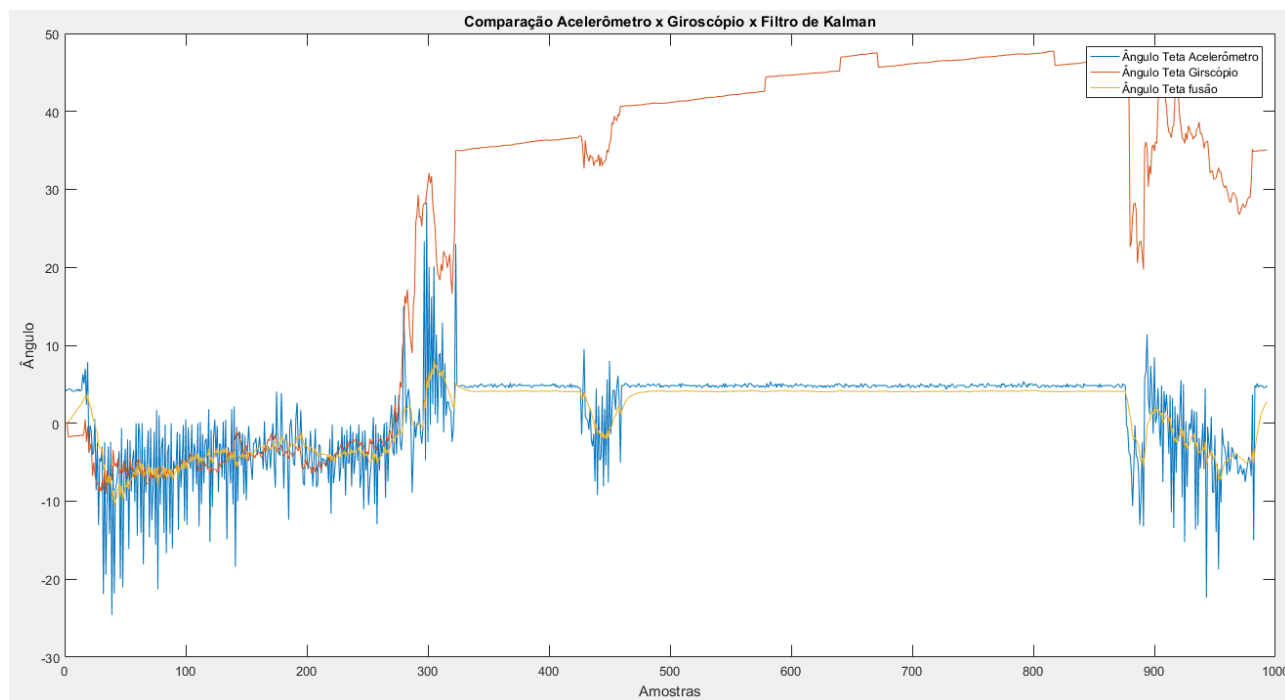


**FIGURA 3. Modelo Tridimensional**

Fonte: Arquivo pessoal, 2018.

## RESULTADOS E DISCUSSÃO

Para análise dos resultados foi gerado um gráfico no matlab™ para análise, conforme figura 4.



**FIGURA 4. Resultado da leitura dos sensores**

**Fonte: Arquivo Pessoal, 2018**

A figura a cima (4) demonstra a plotagem de 1000 amostras da leitura dos sensores acelerômetro, giroscópio e a fusão de ambos utilizando o filtro de Kalman. As cores azul, vermelho e amarelo correspondem, respectivamente à leitura do acelerômetro, giroscópio e a fusão de ambos. No período da amostragem houveram três intervenções manuais, sendo elas; no primeiro período simulou-se um ambiente de trepidação sobre os sensores, seguido por um período em que os sensores foram postos em repouso sobre uma bancada e o terceiro período em que os sensores foram postos novamente sobre uma trepidação. Com a análise do gráfico verifica-se que a fusão sensorial, realmente torna a leitura dos dados mais estáveis.

## CONCLUSÕES

Conclui-se que a aplicação do filtro realmente elimina os efeitos instáveis do acelerômetro quando exposto a um ambiente com trepidação e também descarta o acumulo de erro do giroscópio. Com isso prova-se que a implementação do filtro de Kalman gera dados confiáveis e viabiliza sua utilização para leitura de dados em drones.

## REFERÊNCIAS

MCCARRON, B. **Low-Cost IMU Implementation via Sensor Fusion Algorithms in the Arduino Environment**. Faculty of the Aerospace Engineering Department California Polytechnic State University. June, 2013.

MCROBERTS, M. **Arduino Básico**. 2. ed. São Paulo: Novatec Editora Ltda, 2015.

OLIVEIRA, W. S.; GONÇALVES, E. N. Implementação em c: filtro de Kalman, fusão de sensores para determinação de ângulos. **ForScience**: Revista Científica do IFMG, Formiga, v. 5, n. 3, e00287, jul./dez. 2017.

PROCESSING. **A short introduction to the Processing software and projects from the community**. Disponível em: < <https://processing.org/overview/>>. Acesso em: 17 fev. 2018.

WELCH, G.; BISHOP, G. An **Introduction to the Kalman Filter**. TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill. NC 27599-3175. July 24, 2006.