

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Samuel Rocha da Silva

**Desenvolvimento de uma Unidade de Medição Inercial
Integrando GPS, Filtro de Kalman e Comunicação CAN
para Veículo de Fórmula Estudantil**

São Carlos

2025

Samuel Rocha da Silva

**Desenvolvimento de uma Unidade de Medição Inercial
Integrando GPS, Filtro de Kalman e Comunicação CAN
para Veículo de Fórmula Estudantil**

Monografia apresentada ao Curso de Curso de Engenharia Elétrica com Ênfase em Sistemas de Energia e Automação, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Fábio Romano Lofrano Dotto

**São Carlos
2025**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

RS586d Rocha da Silva, Samuel
Desenvolvimento de uma Unidade de Medição
Inercial Integrando GPS, Filtro de Kalman e Comunicação
CAN para Veículo de Fórmula Estudantil / Samuel Rocha
da Silva; orientador Fabio Romano Lofrano Dotto. São
Carlos, 2025.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Sistemas de Energia e Automação) -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2025.

1. IMU. 2. Filtro de Kalman. 3. GPS. 4. Telemetria.
5. STM32. 6. Fusão de Sensores. 7. Formula Estudantil.
I. Título.

FOLHA DE APROVAÇÃO

Nome: Samuel Rocha da Silva

Título: “Desenvolvimento de uma Unidade de Medição Inercial Integrando GPS, Filtro de Kalman e Comunicação CAN para Veículo de Fórmula Estudantil”

**Trabalho de Conclusão de Curso defendido e aprovado em
25/06/2025,**

com NOTA 8,5 (oito vírgula cinco), pela Comissão Julgadora:

**Prof. Dr. Fabio Romano Lofrano Dotto - Orientador
SEL/EESC/USP**

Prof. Dr. Pedro de Oliveira Conceição Junior - SEL/EESC/USP

Prof. Dr. Augusto Matheus dos Santos Alonso - SEL/EESC/USP

**Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Professor Associado José Carlos de Melo Vieira Júnior**

RESUMO

SILVA, S. **Desenvolvimento de uma Unidade de Medição Inercial Integrando GPS, Filtro de Kalman e Comunicação CAN para Veículo de Fórmula Estudantil.** 2025. 74p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2025.

As Unidades de Medição Inercial (IMUs) são amplamente empregadas em veículos para estimar posição, velocidade e orientação, sendo essenciais em aplicações como navegação autônoma, controle dinâmico e telemetria. Este trabalho tem como objetivo o desenvolvimento e a integração de um hardware IMU de baixo custo para aquisição de dados em um veículo de Fórmula Estudantil. O sistema conta com um microcontrolador STM32, um acelerômetro, um giroscópio de nove eixos, um magnetômetro e um sensor de localização por satélite GNSS. Para o tratamento e a fusão dos dados, utilizou-se o Filtro de Kalman, sendo esses transmitidos via protocolo CAN (Controller Area Network). Na validação, verificou-se que a precisão dos dados de posição foi limitada pela ausência da tecnologia RTK no sensor utilizado, enquanto os demais sensores apresentaram precisão satisfatória.

Palavras-chave: IMU. Fusão de Sensores. STM32. Filtro de Kalman. Sistema Embarcado. Telemetria. MPU9250. Veículo Autônomo.

ABSTRACT

SILVA, S. **Design of an Inertial Measurement Unit Integrating GPS, Kalman Filter, and CAN Communication for a Formula Student Vehicle.** 2025. 74p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2025.

Inertial Measurement Units (IMUs) are widely used in vehicles to estimate position, velocity, and orientation, playing a crucial role in applications such as autonomous navigation, dynamic control, and telemetry. This study aims to develop and integrate a low-cost IMU hardware for data acquisition in a Formula Student vehicle. The system includes an STM32 microcontroller, a nine-axis accelerometer and gyroscope, a magnetometer, and a GNSS satellite positioning sensor. Data processing and fusion were performed using a Kalman Filter, with data transmission via the Controller Area Network (CAN) protocol. During validation, it was observed that the accuracy of position data was limited by the absence of RTK technology in the selected sensor, while the other sensors demonstrated satisfactory precision.

Palavras-chave: IMU. Sensor Fusion. STM32. Kalman Filter. Embedded System. Telemetry. MPU9250. Autonomous Vehicle.

LISTA DE FIGURAS

Figura 1 – Eixos de movimento rotacional de um veículo	22
Figura 2 – Típico gráfico para variância de Allan	24
Figura 3 – Etapas do funcionamento do Filtro de Kalman	26
Figura 4 – Diagrama ilustrativo da arquitetura do circuito eletrônico	29
Figura 5 – Circuito do microcontrolador - STM32	31
Figura 6 – Circuito do sensor - MPU9250	33
Figura 7 – Circuito do GPS - U-Blox	35
Figura 8 – Circuito do transceptor CAN - L9616	36
Figura 9 – Circuito Regulador de Tensão	37
Figura 10 – <i>Layout</i> de trilhas da PCI (visualização em duas camadas)	38
Figura 11 – Visualização 3D da PCI gerada no Altium Designer	39
Figura 12 – Processo de solda SMD com forno de refusão	40
Figura 13 – Placa após solda dos componentes	40
Figura 14 – Configuração de pinagem do microcontrolador	41
Figura 15 – Fluxograma do código de aquisição - IMU	43
Figura 16 – Fluxograma do código de aquisição - GPS	45
Figura 17 – Fluxograma do código de calibração dos sensores inerciais	47
Figura 18 – Fluxograma do código para orientação - Filtro de Madgwick	51
Figura 19 – Fluxograma do código para posição - Filtro de Kalman	57
Figura 20 – Fluxograma do código transmissão - CAN	60
Figura 21 – Pipeline do código	61
Figura 22 – Componentes do Quatérnion ao longo do tempo	64
Figura 23 – Orientação: Ângulos de Yaw, Pitch e Roll ao longo do tempo	64
Figura 24 – Comparação entre yaw calculado e o true heading do magnetômetro	65
Figura 25 – Desvio de Allan para os dados brutos do giroscópio	66
Figura 26 – Comparação dos dados de GPS brutos com a saída do Filtro de Kalman para Latitude e Longitude	67
Figura 27 – Mapa com trajetória estimada e dados brutos do GPS	68
Figura 28 – Dispersão dos dados brutos de GPS em relação à trajetória estimada pelo Filtro de Kalman	68
Figura 29 – Comparação entre a trajetória estimada e a trajetória real	69

LISTA DE TABELAS

Tabela 1 – Características técnicas do microcontrolador STM32F103C8T6	30
Tabela 2 – Características técnicas do sensor inercial MPU9250	33
Tabela 3 – Características da placa de circuito impresso (PCI)	39
Tabela 4 – Principais arquivos gerados após configuração	42
Tabela 5 – Estruturas de dados integradas ao algoritmo	48
Tabela 6 – Estruturas de dados integradas ao filtro de Kalman para posição	52
Tabela 7 – Mensagens e dados transmitidos via protocolo CAN	59

LISTA DE ABREVIATURAS E SIGLAS

ADC	Analog-to-Digital Converter
CAN	Controller Area Network
EEPROM	Electrically Erasable Programmable Read-Only Memory
EKF	Extended Kalman Filter
FS	File System
GNSS	Global Navigation Satellite System
GPIO	General Purpose Input/Output
GPS	Global Positioning System
<i>I²C</i>	Inter-Integrated Circuit (ou I2C)
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LQFP	Low-profile Quad Flat Package
MCU	Microcontroller Unit
MARG	Magnetometer, Accelerometer and Gyroscope
NMEA	National Marine Electronics Association
PCB	Printed Circuit Board
QoS	Quality of Service
RTC	Real-Time Clock
RTK	Real-Time Kinematic
RMS	Root Mean Square
SCL	Serial Clock
SDA	Serial Data
SiP	System in Package
SPI	Serial Peripheral Interface

SWCLK	Serial Wire Clock
SWD	Serial Wire Debug
SWDIO	Serial Wire Debug I/O
TTFF	Time To First Fix
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Objetivos	18
1.2	Organização do trabalho	18
2	REVISÃO BIBLIOGRÁFICA	21
2.1	Sensores para Navegação e Estimação de Estado	21
2.1.1	Natureza dos Dados e Erros Inerentes	22
2.1.2	Variância de Allan	24
2.2	Fusão Sensorial para Estimativa da Posição	25
2.2.1	Filtro de Kalman	25
2.3	Fusão Sensorial para Estimativa da Orientação	27
2.3.1	Filtro de Madgwick	27
2.4	Protocolo de Comunicação CAN	28
3	HARDWARE	29
3.1	Projeto do Circuito Eletrônico	29
3.1.1	Microcontrolador	30
3.1.2	Sensoriamento	32
3.1.3	Geolocalização	34
3.1.4	Telemetria	36
3.1.5	Regulador de Tensão	37
3.2	Projeto da Placa de Circuito Impresso	37
3.2.1	Considerações de Roteamento e <i>Layout</i>	37
3.2.2	Manufatura da PCI	39
3.2.3	Soldagem dos componentes	40
4	SOFTWARE	41
4.1	Configuração da Plataforma de Hardware no STM32CubeIDE	41
4.2	Implementação dos Drivers de Aquisição de Dados	42
4.2.1	Aquisição dos dados da IMU (MPU9250)	43
4.2.2	Aquisição dos dados do GPS (NEO-6M)	44
4.3	Algoritmos de Fusão Sensorial	45
4.3.1	Estimativa de Atitude (Roll, Pitch, Yaw)	46
4.3.1.1	Pré-calibração do sensor IMU	46
4.3.1.2	Cálculo da Orientação - Filtro de Madgwick	48
4.3.2	Estimativa de Posição e Velocidade - Filtro de Kalman	52

4.4	Implementação da Comunicação CAN	58
4.5	Considerações Finais	60
5	ANÁLISE DE DADOS E RESULTADOS	63
5.1	Metodologia de Coleta de Dados	63
5.2	Avaliação dos Dados de Orientação do Veículo	63
5.2.1	Comparação do Yaw Estimado com o Heading Magnético	65
5.2.2	Caracterização do Giroscópio: Desvio (Bias) e Ruído	66
5.3	Avaliação da Estimativa de Posição do Veículo	66
5.3.1	Visualização Espacial da Trajetória Filtrada	67
5.3.2	Análise Detalhada da Dispersão e Limitações do GPS	68
5.3.3	Desvio da Trajetória em Relação a uma Referência	69
6	CONCLUSÃO	71
6.1	Propostas para Trabalhos Futuros	72
	REFERÊNCIAS	73

1 INTRODUÇÃO

A Fórmula SAE (*Society of Automotive Engineers*) Elétrico é uma competição estudantil que propõe às equipes participantes projetar, manufaturar e demonstrar um veículo elétrico de alto desempenho. Nessa competição, parte da avaliação do projeto se dá através do desempenho do veículo em provas dinâmicas, tais como o de aceleração, *skidpad* (o qual avalia a capacidade de realizar curvas) e o *autocross*, que contém um circuito a ser percorrido pelo veículo (LAVEISSIERE, 2019).

Nesse contexto, a telemetria, ao monitorar e analisar dados em tempo real, possibilita uma tomada de decisão mais informada e precisa, desempenhando um papel crucial na otimização do desempenho veicular. Dentre os diversos sensores utilizados por um veículo de Fórmula Estudantil, destacam-se aqueles responsáveis pela estimativa de posição, velocidade e orientação do carro em tempo real, tornando assim a Unidade de Medição Inercial (IMU) um componente essencial para esse fim (LAVEISSIERE, 2019; MADGWICK, 2010).

Os dados fornecidos pela IMU podem atender a vários sistemas. Um exemplo são os carros autônomos utilizados na competição *Formula Driveless*, nos quais o uso desses sensores é indispensável, uma vez que são responsáveis pelos dados de posição e orientação do veículo. Outra aplicação para a IMU é fornecer dados para modelos de controle veicular. Um diferencial eletrônico (*e-diff*), responsável por ajustar a distribuição do torque com base na dinâmica veicular, também se beneficia desses dados (ANTUNES, 2017). Nesse contexto, a IMU tem um papel primordial, pois fornece dados com os quais é possível definir parâmetros como aceleração lateral e longitudinal, taxa de rotação (*yaw rate*), bem como a inclinação e rolagem do veículo (INVENSENSE, 2016).

Este trabalho foca no desenvolvimento de um sistema de sensoriamento de baixo custo, integrando dados de IMU (acelerômetro, giroscópio e magnetômetro) e GPS, projetado para atender às necessidades específicas de telemetria em veículos de Fórmula Estudantil. A integração desses sensores permitirá a aquisição e o processamento de dados essenciais como aceleração, orientação e posição. Para a fusão dos dados provenientes do GPS, será implementado um Filtro de Kalman, com o objetivo de suavizar as estimativas de posição e reduzir os efeitos de ruídos e perdas momentâneas de sinal. (SRINIVASAN, 2018; WANG; LIANG, 2018). Os dados serão transmitidos via CAN, protocolo de comunicação amplamente utilizado por equipes de Fórmula (ANTUNES, 2017; INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2024).

1.1 Objetivos

Este projeto tem como objetivo desenvolver um sistema capaz de fornecer dados confiáveis de posição, aceleração e orientação, que poderão ser utilizados posteriormente no desenvolvimento e aprimoramento de estratégias de controle dinâmico e navegação do veículo.

O sistema deve ser capaz de coletar e processar dados de um conjunto de sensores embarcados, incluindo acelerômetro, giroscópio, magnetômetro e um receptor GNSS (Sistema Global de Navegação por Satélite). A fusão dos sensores será feita por meio do Filtro de Kalman, visando reduzir o ruído e melhorar a precisão das estimativas em relação aos dados brutos. A transmissão dos dados, por sua vez, será realizada por meio do protocolo CAN (*Controller Area Network*), permitindo a integração com outros módulos eletrônicos do veículo.

Embora o sistema não tenha como objetivo atingir a precisão oferecida por soluções comerciais de alto custo, ele busca explorar o potencial dos sensores embarcados de baixo custo, por meio da aplicação de algoritmos de fusão sensorial. Com isso, pretende-se obter estimativas significativamente mais consistentes e úteis do que aquelas que seriam possíveis com sensores isolados e não processados, oferecendo assim uma solução acessível, robusta e adequada à realidade das equipes de Fórmula Estudantil.

Além disso, o sistema deve apresentar uma interface de comunicação eficiente, facilitando a extração e análise dos dados adquiridos. A robustez do *hardware* e do *firmware* será um ponto importante para um desempenho confiável em condições adversas, como vibrações e variações bruscas de aceleração.

1.2 Organização do trabalho

Este trabalho está dividido em seis capítulos principais. A **Introdução** apresenta a finalidade da pesquisa, assim como os objetivos do projeto e sua estruturação.

Na **Revisão Bibliográfica** expõe-se uma fundamentação teórica sobre os temas abordados no projeto, como Unidades de Medição Inercial (IMUs), sensoriamento, sistemas embarcados, Filtragem de Kalman, comunicação CAN e geolocalização.

O **Capítulo de Hardware** detalha como se deu o desenvolvimento do hardware do sistema embarcado. São descritos o projeto do circuito eletrônico, incluindo a seleção e integração de componentes como o microcontrolador STM32, o sensor inercial MPU9250, o módulo GNSS e o transceptor CAN L9616. O capítulo também abrange o planejamento e a manufatura da Placa de Circuito Impresso (PCI), além do processo de soldagem dos componentes.

Em seguida, no **Capítulo de Software**, delineia-se o desenvolvimento do código

utilizando o ambiente STM32CubeIDE e a linguagem C. Inclui a configuração da plataforma de hardware, a implementação dos drivers para aquisição de dados da IMU e do GPS, os algoritmos de calibração de sensores inerciais e os algoritmos de fusão sensorial, como o filtro de Madgwick para estimativa de atitude (*roll, pitch, yaw*) e o Filtro de Kalman para estimativa de posição e velocidade horizontal. A comunicação via protocolo CAN também é tratada.

Em **Resultados** apresenta-se a validação e os testes realizados. Discute-se a precisão dos dados obtidos em testes, mencionando as limitações observadas, como a precisão dos dados de posição devido à ausência da tecnologia RTK (*Real Time Kinematic*) no sensor GNSS utilizado, e avaliando o desempenho dos demais sensores.

Por fim, as **Conclusões** sobre o desenvolvimento deste projeto são apresentadas, e **Propostas para Trabalhos Futuros** são sugeridas, com base nos resultados e nas funcionalidades exploradas.

2 REVISÃO BIBLIOGRÁFICA

Em ambientes dinâmicos e desafiadores como Fórmula Estudantil, a capacidade de monitorar e analisar continuamente o comportamento do veículo é fundamental para a otimização do desempenho veicular. A análise dos dados coletados, incluindo aqueles provenientes da IMU, permite que as equipes compreendam o comportamento dinâmico do carro na pista e identifiquem pontos de melhoria no projeto e nos ajustes. Além disso, os dados da unidade são indispensáveis em aplicações como carros autônomos da Fórmula Driverless, onde são usados para a localização e o mapeamento em tempo real. (LAVEISSIERE, 2019).

2.1 Sensores para Navegação e Estimação de Estado

A navegação precisa e a estimação de estado de um veículo dependem fundamentalmente da coleta e interpretação de dados provenientes de diversos sensores. Dentre os mais utilizados em aplicações automotivas, destacam-se as Unidades de Medição Inercial (IMUs) e os Sistemas Globais de Navegação por Satélite (GNSS), como o GPS.

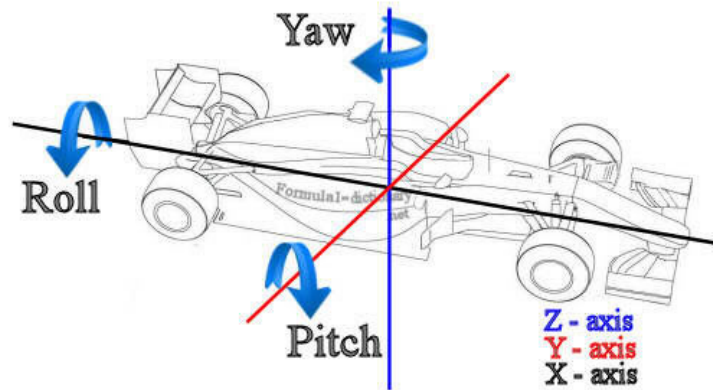
Unidades de Medição Inercial (IMU)

As Unidades de Medição Inercial (IMUs) são dispositivos responsáveis por fornecer dados precisos de aceleração e velocidade angular de um objeto. No contexto deste trabalho, esse tipo de sensoriamento desempenha um papel crucial na determinação do estado de um veículo, sendo empregado para estimar a orientação (atitude), representada pelos ângulos de rotação Pitch, Roll e Yaw, como mostra a Figura 1, além dos dados de posição e velocidade. Uma IMU tipicamente consiste em um conjunto de sensores que, juntos, fornecem dados sobre o movimento e a atitude do veículo (MADGWICK, 2010). Tipicamente, esses sensores são:

- **Acelerômetros:** Medem a aceleração linear própria, que inclui tanto a aceleração devido ao movimento do veículo quanto a aceleração gravitacional terrestre. Estes dados são fundamentais para determinar a inclinação do veículo. (MADGWICK, 2010)
- **Giroscópios:** Medem a velocidade angular, ou seja, a taxa de rotação em torno dos eixos do sensor (X, Y e Z). A integração desses dados ao longo do tempo permite calcular a orientação do veículo (MADGWICK, 2010).
- **Magnetômetros:** Medem a força e a direção de campos magnéticos, primariamente o campo magnético terrestre. Esta medição é utilizada para determinar a orientação

do veículo em relação ao norte magnético (MUN; SHIN; CHOI, 2018).

Figura 1 – Eixos de movimento rotacional de um veículo



Fonte: F1 Dictionary (2020)

Sistemas Globais de Navegação por Satélite (GNSS/GPS)

Os Sistemas Globais de Navegação por Satélite, sendo o GPS (*Global Positioning System*) um dos mais utilizados, são sistemas que utilizam sinais múltiplos satélites determinar a posição de um objeto. (SRINIVASAN, 2018). Os dados fornecidos tipicamente incluem:

- **Posição:** Fornecida em coordenadas geográficas (latitude, longitude e altitude) (ALABA, 2024).
- **Velocidade:** Calculada a partir da variação da posição ao longo do tempo ou pelo efeito Doppler dos sinais de satélite (AG, 2016).
- **Tempo:** Informação de tempo precisa, sincronizada com relógios atômicos a bordo dos satélites.

Os dados podem ser transmitidos por meio de diferentes protocolos, como o NMEA, que define um formato de mensagem padronizado para o envio de informações como latitude, longitude, altitude, velocidade, direção do movimento, hora, data, qualidade do sinal GPS e número de satélites visíveis. (AG, 2016).

2.1.1 Natureza dos Dados e Erros Inerentes

É crucial compreender que os dados de qualquer sensor, especialmente os de baixo custo frequentemente empregados em projetos acadêmicos e protótipos, estão sujeitos a erros e incertezas.

Erros da IMU

- Os **giroscópios** sofrem de *bias* (um desvio na leitura mesmo quando o sensor está parado) e ruído. A integração da velocidade angular para obter a orientação leva ao acúmulo desses erros, resultando em um desvio (*drift*) na estimativa da orientação ao longo do tempo (ALABA, 2024). O *bias* do giroscópio também pode variar com a temperatura e o movimento (MADGWICK, 2010).
- Os **acelerômetros** também possuem *bias* e ruído. Além disso, são sensíveis tanto à aceleração gravitacional quanto às acelerações lineares devido ao movimento do veículo. Distinguir entre estas duas componentes pode ser desafiador, e acelerações lineares podem corromper a medição da direção da gravidade, afetando as estimativas de inclinação (MADGWICK, 2010).
- Os **magnetômetros** são suscetíveis a distorções do campo magnético terrestre causadas por materiais ferromagnéticos ou campos eletromagnéticos gerados por componentes eletrônicos ou motores próximos ao sensor (distorções de *hard iron* e *soft iron*) (MUN; SHIN; CHOI, 2018).

Erros do GPS/GNSS

- A principal limitação do GPS é a necessidade de uma linha de visada direta com os satélites. Em ambientes como túneis, cânions urbanos densos ou espaços internos, os sinais de satélite podem ser obstruídos, atenuados ou refletidos (multicaminho), levando à degradação da precisão ou à perda total do sinal (ALABA, 2024; SRINIVASAN, 2018).
- A taxa de atualização de muitos receptores GPS de baixo custo pode ser relativamente baixa (por exemplo, 1 Hz ou 5 Hz), o que pode não ser ideal para aplicações que exigem alta dinâmica (SRINIVASAN, 2018).
- Mesmo em condições ideais, os dados de GPS possuem um erro de posicionamento inerente, tipicamente na ordem de alguns metros (WANG; LIANG, 2018). O modelo utilizado neste trabalho possui um erro aproximado de 2,5 metros (AG, 2016).

A compreensão dessas fontes de erro é fundamental para o desenvolvimento de algoritmos de fusão sensorial, como o Filtro de Kalman, que buscam combinar as informações de diferentes sensores para fornecer uma estimativa de estado mais precisa e robusta do que seria possível com qualquer sensor individualmente.

2.1.2 Variância de Allan

O Desvio de Allan (*Allan deviation*) é uma métrica estatística amplamente utilizada para caracterizar a estabilidade de sinais de sensores, em especial sinais de frequência e de velocidade angular provenientes de giroscópios (ALLAN, 1966). Foi proposto por David W. Allan na década de 1960 para avaliar a estabilidade de osciladores de frequência, mas tornou-se uma ferramenta fundamental na análise de ruído em sistemas inerciais.

Ao contrário da variância clássica, que não é adequada para sinais com componentes de ruído, o *desvio de Allan* permite a identificação e quantificação de diferentes tipos de ruído, como *white noise*, *bias*, *random walk*, entre outros. Isso é feito por meio da análise da variabilidade entre médias sucessivas de intervalos de tempo fixos (ALLAN, 1966).

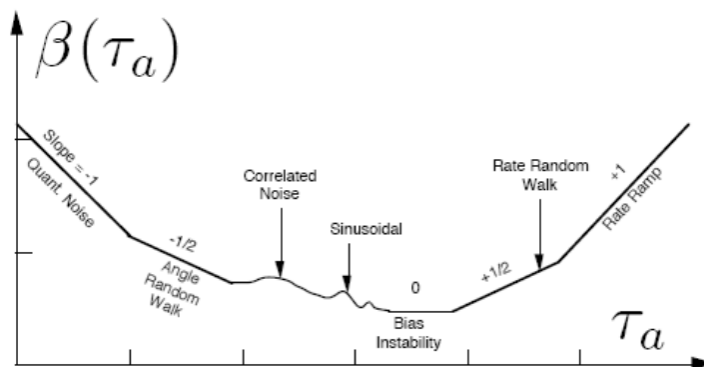
Matematicamente, para um sinal discreto x_i amostrado com intervalo τ_0 , o desvio de Allan para um tempo de integração $\tau = m \cdot \tau_0$ é definido como:

$$\beta_{\text{Allan}}^2(\tau) = \frac{1}{2(N-2m)} \sum_{k=1}^{N-2m} (\bar{x}_{k+2m} - 2\bar{x}_{k+m} + \bar{x}_k)^2, \quad (2.1)$$

onde \bar{x}_k representa a média das amostras no intervalo k de duração τ , N é o número total de amostras, e m é o número de amostras agrupadas.

Ao se representar $\beta_{\text{Allan}}(\tau)$ em escala log-log, é possível observar diferentes inclinações que correspondem a diferentes tipos de ruído. Por exemplo:

Figura 2 – Típico gráfico para variância de Allan



Fonte: (INOUE, 2012)

A inclinação de $-1/2$ está associada ao *white noise*, enquanto uma inclinação nula (0) indica a presença de *bias instability*. Já uma inclinação de $+1/2$ é característica do *random walk*.

Essa análise permite identificar os principais componentes de ruído presentes no sensor e é frequentemente usada para calibrar modelos de erro de sensores inerciais em

filtros de estimação, como o *Filtro de Kalman* (INOUE, 2012).

Na prática, o Desvio de Allan é uma etapa importante na validação de giroscópios e acelerômetros, principalmente em aplicações que exigem alta precisão e estabilidade a longo prazo, como navegação inercial e sistemas de posicionamento autônomo.

2.2 Fusão Sensorial para Estimativa da Posição

Para obter uma estimativa da posição geográfica que seja simultaneamente precisa e estável, é comum recorrer à combinação de dados provenientes de múltiplos sensores com características complementares. Nesse contexto, a fusão sensorial permite superar as limitações intrínsecas de sensores individuais, como as das Unidades de Medição Inercial e dos receptores GPS (ALABA, 2024).

A integração entre um sistema inercial (INS) e um receptor GPS é uma solução frequentemente utilizada na aquisição de dados de navegação. Enquanto a IMU fornece dados de movimento com alta taxa de atualização, mas sofre com o acúmulo de erros ao longo do tempo, o GPS, por sua vez, fornece posição absoluta relativamente precisa, porém é suscetível a falhas de sinal e possui baixa taxa de atualização (SRINIVASAN, 2018).

Nesse contexto, a estimativa de posição por fusão sensorial se dá na seguinte complementaridade: o INS fornece estimativas de posicionamento quando os dados do GPS estão indisponíveis ou são de baixa qualidade, e, reciprocamente, o GPS auxilia na correção do drift acumulado pelo INS, mantendo os erros de navegação dentro de uma faixa especificada (INOUE, 2012).

2.2.1 Filtro de Kalman

O Filtro de Kalman é amplamente utilizado na estimação de variáveis dinâmicas e fusão sensorial, sendo utilizado em diversos sistemas, incluindo na navegação inercial em aeronaves (INOUE, 2012). O filtro foi desenvolvido por Rudolf E. Kalman como um estimador recursivo eficiente que calcula o estado de um processo linear dinâmico a partir de uma série de medições incompletas e ruidosas (SRINIVASAN, 2018). Neste trabalho, o Filtro de Kalman é empregado para o tratamento e a fusão dos dados provenientes da IMU e do sensor GNSS, visando obter estimativas precisas de posição e velocidade.

Matematicamente, o filtro assume que o sistema pode ser descrito por um modelo de espaço de estados linear na forma:

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (2.2)$$

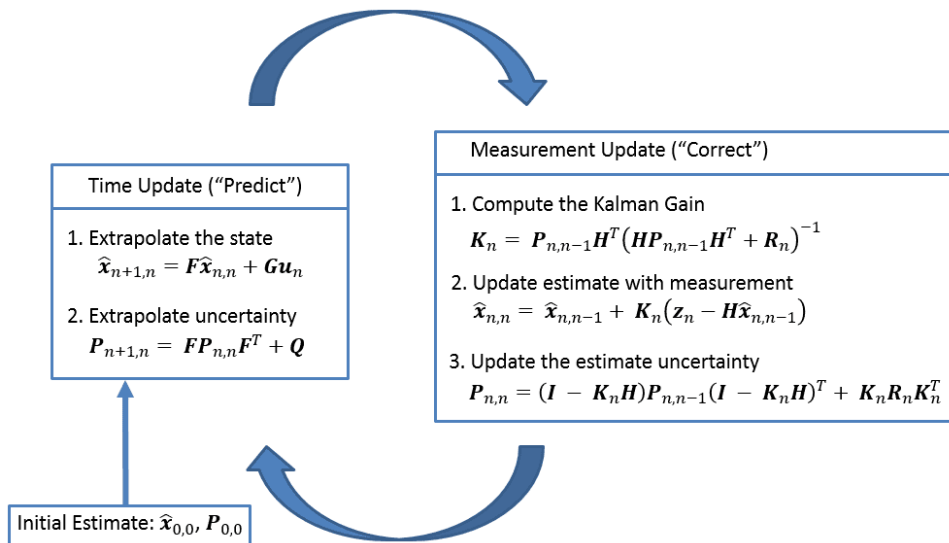
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \quad (2.3)$$

onde \mathbf{x}_k representa o vetor de estados no instante k , \mathbf{u}_k o vetor de entrada, \mathbf{z}_k o vetor de observações, \mathbf{w}_k o ruído de processo e \mathbf{v}_k o ruído de medição. Ambos os ruídos são assumidos como ruído branco gaussiano com covariância conhecida.

O princípio fundamental do Filtro de Kalman repousa em um ciclo de duas etapas: predição e atualização (correção) (SRINIVASAN, 2018).

- **Previsão (*predict*)**: Nesta etapa, o filtro projeta o estado atual e a sua incerteza (covariância do erro) para o próximo instante de tempo, utilizando um modelo matemático que descreve a dinâmica do sistema (SRINIVASAN, 2018).
- **Correção (*update*)**: Quando uma nova medição (z_k) se torna disponível, o filtro utiliza essa informação para refinar a predição. O *ganho de Kalman* (K_k) é calculado para ponderar a contribuição da medição em relação à predição, minimizando a covariância do erro da estimativa (SRINIVASAN, 2018).

Figura 3 – Etapas do funcionamento do Filtro de Kalman



Fonte: kalmanfilter.net

Observando a Figura 3, na etapa de previsão, o estado do sistema é estimado projetando o próximo estado a partir da estimativa anterior e de uma entrada de controle. Esse processo é representado pela equação $\hat{\mathbf{x}}_{n+1,n} = F \hat{\mathbf{x}}_{n,n} + G \mathbf{u}_n$, onde F é a matriz de transição de estados, G é a matriz de controle e \mathbf{u}_n é a entrada. Em seguida, a incerteza associada à estimativa é atualizada por meio da equação $P_{n+1,n} = F P_{n,n} F^T + Q$, considerando o modelo de ruído do processo Q .

Na etapa de correção, utiliza-se uma nova medição para ajustar a estimativa do estado. Primeiro, calcula-se o *ganho de Kalman* K_n , que determina o peso da medição na

correção, com base na incerteza da predição e na variância do ruído de medição R_n . A nova estimativa do estado é então corrigida com base no erro entre a medição observada z_n e a estimativa predita, conforme $\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n(z_n - H\hat{x}_{n,n-1})$. Por fim, a incerteza da nova estimativa é atualizada considerando a confiança na medição, de acordo com $P_{n,n} = (I - K_n H)P_{n,n-1}(I - K_n H)^T + K_n R_n K_n^T$.

Esse processo é repetido a cada novo ciclo de amostragem, permitindo ao filtro combinar eficientemente dados preditivos e observações ruidosas para obter uma estimativa precisa e estável do estado do sistema.

No contexto deste trabalho, o filtro é utilizado para fundir os dados provenientes da IMU e do GPS resultando em uma trajetória estimada mais suave e contínua, mesmo em situações de perda momentânea do sinal GNSS.

2.3 Fusão Sensorial para Estimativa da Orientação

A estimação da atitude de um veículo, ou seja, sua orientação no espaço tridimensional (Figura 1), é crucial para navegação e controle. Essa atitude é geralmente descrita pelos ângulos de Euler: rolagem (ϕ), arfagem (θ) e guinada (ψ), e sua determinação envolve a fusão dos sinais de sensores inerciais, como giroscópios, acelerômetros e magnetômetros (INOUE, 2012).

Um dos algoritmos comumente usados para estimação de atitude por meio da fusão de dados de uma IMU é o algoritmo de Madgwick (LAI, 2024).

2.3.1 Filtro de Madgwick

O filtro de Madgwick é um algoritmo de estimação de atitude desenvolvido por Sebastian Madgwick em 2010. Atualmente o filtro é amplamente utilizado em sistemas embarcados baseados em sensores inerciais por ser uma opção computacionalmente eficiente e de baixa latência. Diferentemente de métodos clássicos como o *Filtro de Kalman*, o filtro de Madgwick utiliza uma abordagem baseada em *quaternions* e em um *gradiente descendente* para minimizar o erro entre a orientação estimada e os vetores de aceleração e campo magnético medidos (MADGWICK, 2010).

Para essa solução os *quaternions* são preferidos em relação aos ângulos de Euler, pois são livres do problema de singularidades (como o *gimbal lock*) e oferecem maior robustez computacional (MADGWICK, 2010). Um *quaternion* de orientação $q = [q_w, q_x, q_y, q_z]$ pode descrever qualquer rotação no espaço 3D.

O filtro é projetado para ser aplicado em conjuntos de sensores do tipo MARG (Magnetômetro, Acelerômetro e Giroscópio). Sua eficiência está na capacidade de fazer a fusão dos dados do giroscópio — que oferecem resposta rápida, mas estão sujeitos a *drift*

— com os vetores de referência fornecidos pelo acelerômetro (gravidade) e magnetômetro (campo magnético da Terra), que são absolutos, mas mais ruidosos (LAI, 2024).

A atitude é representada por um quaternion unitário \mathbf{q} , o qual o algoritmo atualiza com base na taxa de rotação medida pelo giroscópio, corrigida por um termo proporcional ao erro entre os vetores medidos e os esperados com base na orientação atual (MADGWICK, 2010).

A equação de atualização do *quaternion* é expressa por:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \omega - \beta \nabla f(\mathbf{q}), \quad (2.4)$$

onde ω é o vetor de velocidade angular (convertido em *quaternion*), \otimes representa o produto de *quaternions*, $\nabla f(\mathbf{q})$ é o gradiente do erro e β é um ganho ajustável que controla a convergência do algoritmo. A derivada $\dot{\mathbf{q}}$ é integrada numericamente a cada ciclo, atualizando a estimativa de atitude.

Após cada atualização, o quatérnio é normalizado para garantir que represente uma rotação válida.

2.4 Protocolo de Comunicação CAN

O protocolo *Controller Area Network* (CAN) é um barramento de comunicação amplamente utilizado em sistemas embarcados, especialmente no setor automotivo, por sua robustez, confiabilidade e baixo custo. Ele permite a troca de mensagens entre diversos dispositivos (ou nós) sem a necessidade de um mestre central, utilizando um esquema de arbitragem por prioridade baseado em identificadores únicos (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2024).

As mensagens CAN são transmitidas em quadros padronizados que contêm um identificador, até 8 bytes de dados (ou até 64 bytes na versão *CAN FD*) e mecanismos de detecção de erro, como CRC. O protocolo opera com comunicação diferencial em dois fios (CAN High e CAN Low), oferecendo imunidade a ruídos eletromagnéticos e confiabilidade em ambientes hostis.

Neste trabalho, o protocolo CAN foi utilizado para transmitir os dados estimados pela IMU, como orientação e posição, garantindo comunicação eficiente e em tempo real com outros módulos do veículo.

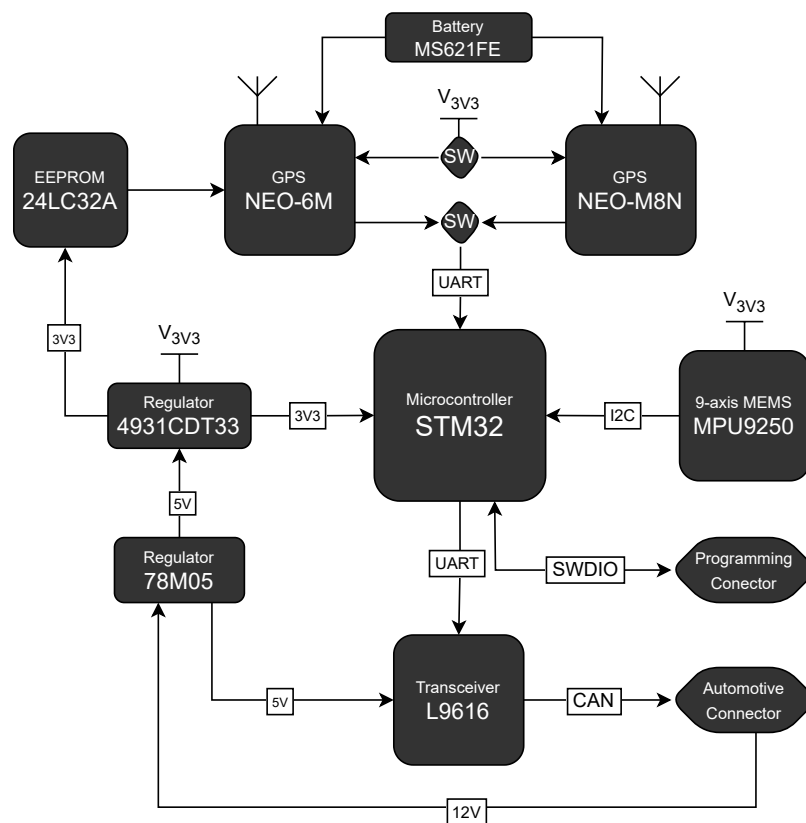
3 HARDWARE

Este capítulo descreve o desenvolvimento do sistema eletrônico embarcado, com foco na concepção, organização e implementação dos circuitos que integram os sensores, o microcontrolador e os módulos de comunicação. O objetivo é apresentar, de forma clara e segmentada, o processo de construção do sistema físico, desde o projeto do circuito eletrônico até a fabricação da placa de circuito impresso (PCI).

3.1 Projeto do Circuito Eletrônico

A etapa inicial do desenvolvimento de hardware consistiu na elaboração do circuito eletrônico, englobando a seleção e interconexão dos componentes do sistema. Esta fase foi realizada no *software* Altium Designer, que permitiu a elaboração dos digramas dos blocos principais, como o microcontrolador, os sensores, o receptor GNSS e o transceptor CAN.

Figura 4 – Diagrama ilustrativo da arquitetura do circuito eletrônico



Fonte: Elaborado pelo autor

A Figura 4 apresenta o diagrama de blocos do sistema embarcado desenvolvido, representando os módulos eletrônicos essenciais e a forma como se comunicam entre si por meio das interfaces implementadas.

As seções a seguir descrevem em detalhes os subsistemas representados no diagrama e apresentam as justificativas técnicas para as decisões adotadas ao longo do desenvolvimento do projeto.

3.1.1 Microcontrolador

Para a integração dos sensores embarcados, o processamento dos dados e a interface com o módulo transceptor CAN, optou-se pelo microcontrolador STM32F103C8T6. Este microcontrolador pertence à família STM32 da STMicroelectronics e foi escolhido por oferecer uma combinação adequada de desempenho, interfaces de comunicação integradas incluindo controlador CAN nativo além de ser, acima de tudo, um microcontrolador acessível.

Tabela 1 – Características técnicas do microcontrolador STM32F103C8T6

Parâmetro	Especificação
Núcleo (<i>core</i>)	ARM Cortex-M3, 32 bits
Frequência de operação	Até 72 MHz
Memória Flash	64 KB (até 128 KB em algumas versões)
SRAM	20 KB
Interfaces de comunicação	2x I ² C, 3x UART, 2x SPI, USB, CAN
Conversores A/D	2 ADCs de 12 bits com até 16 canais
Temporizadores	3 timers de uso geral, 1 timer avançado, <i>watchdog</i>
GPIOs disponíveis	Até 37 pinos de entrada/saída
Tensão de operação	2.0 V a 3.6 V (3.3 V típica)
Programação	Via SWD, ST-Link, <i>bootloader</i> USB/ <i>UART</i>
Encapsulamento / Dimensão	LQFP-48 / 7.0 (E 7.0 mm
Consumo típico	20 mA em operação
Faixa de temperatura	-40 °C a +85 °C

Fonte: Adaptado da STMicroelectronics e documentação técnica do STM32F103C8T6.

Para a implementação do circuito embarcado, utilizaram-se como base as soluções adaptadas do módulo *Blue Pill*, amplamente utilizado em projetos com o microcontrolador STM32F103 (STMICROELECTRONICS, 2020; STM32-BASE, 2016). Deste modo, a partir dos requisitos descritos no *datasheet* e nas *application notes* fornecidas pelo fabricante, foi possível projetar um circuito robusto e funcional, incorporando os elementos

Oscilador

Mesmo o STM32F103C8T6 possuindo um oscilador interno de 8 MHz, optou-se pela inclusão de cristais externos, sendo um de 8 MHz e outro de 32 kHz, os quais garantem maior precisão e confiabilidade em relação ao oscilador interno. Outra justificativa para a escolha é permitir maior versatilidade, garantindo uma temporização mais precisa e estável, essencial para aplicações que envolvem comunicação serial, filtros dependentes de tempo, como o filtro de Kalman, e o RTC (*Real-Time Clock*), que exige alta precisão para a manutenção de tempo real.

Conectores de *debug* e *boot*

Para possibilitar a programação e depuração do microcontrolador, foi adicionado um conector padrão SWD (*Serial Wire Debug*) com os sinais SWCLK, SWDIO, GND e 3.3 V. Dois jumpers conectados aos pinos BOOT0 e BOOT1, permitem a seleção do modo de *boot*. Essas adições são importantes, pois permitem realizar atualizações de *firmware* e depuração sem necessidade de intervenção no circuito principal.

3.1.2 Sensoriamento

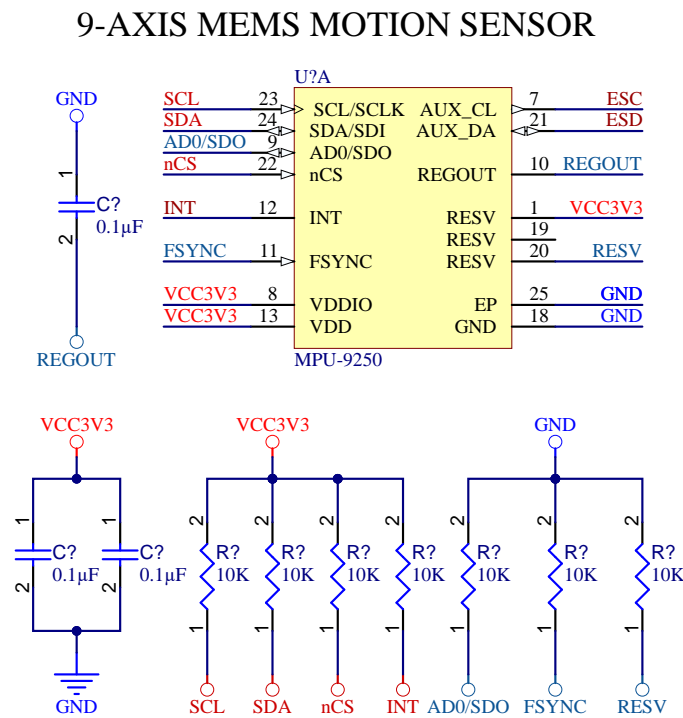
Para a aquisição dos dados inerciais e de orientação do sistema embarcado, foi selecionado o sensor MPU9250, fabricado pela InvenSense. O dispositivo foi escolhido por integrar, de forma compacta, múltiplos sensores em um único encapsulamento. Inicialmente, considerou-se a utilização de módulos de sensoriamento independentes para o giroscópio, magnetômetro e acelerômetro; no entanto, essa abordagem aumentaria desnecessariamente a complexidade do circuito.

O MPU9250 é um sensor inercial de nove graus de liberdade (9-DOF), implementado em um *System in Package (SiP)* que combina dois chips: o MPU-6500, que integra um acelerômetro triaxial e um giroscópio triaxial, e o AK8963, responsável pelas medições do magnetômetro triaxial.

- **Acelerômetro triaxial:** mede acelerações lineares nos três eixos. Permite a detecção de movimentos e inclinações, sendo essencial para o cálculo da posição relativa e análise de dinâmica veicular.
- **Giroscópio triaxial:** mede a velocidade angular nos três eixos. Permite estimar a rotação do sistema ao longo do tempo, fundamental para o rastreamento da orientação.
- **Magnetômetro triaxial (AK8963):** mede o campo magnético terrestre nos três eixos. Permite a obtenção do rumo (*heading*), atuando semelhante a uma bússula.

Dessa forma, o MPU9250 oferece uma solução eficiente e confiável para o sensoriamento inercial e a determinação da orientação, uma vez que a combinação dos sensores permite estimar com precisão o movimento e a atitude do sistema por meio de técnicas de fusão sensorial. As características técnicas do sensor podem ser observadas na Tabela 2, assim como o diagrama de ligação na Figura 6.

Figura 6 – Circuito do sensor - MPU9250



Fonte: Elaborado pelo autor

Tabela 2 – Características técnicas do sensor inercial MPU9250

Parâmetro	Especificação
Faixa do acelerômetro	$\pm 2 \text{ g}$, $\pm 4 \text{ g}$, $\pm 8 \text{ g}$ e $\pm 16 \text{ g}$
Faixa do giroscópio	$\pm 250^\circ \text{ s}^{-1}$, $\pm 500^\circ \text{ s}^{-1}$, $\pm 1000^\circ \text{ s}^{-1}$ e $\pm 2000^\circ \text{ s}^{-1}$
Faixa do magnetômetro	Até $4900 \mu\text{T}$
Comunicação	I ² C (até 400 kHz) ou SPI (até 1 MHz)
Tensão de operação	2.4 V a 3.6 V

Fonte: Adaptado da InvenSense MPU9250 Product Specification.

Interface de Comunicação I²C

Para o projeto, optou-se por utilizar a interface I²C para a comunicação entre o microcontrolador STM32F103C8T6 e o sensor MPU9250. Essa escolha se deu pela facilidade de integração e pela simplicidade do barramento, que é composto por duas linhas principais:

- SDA (Serial Data): responsável pela transmissão bidirecional de dados;
- SCL (Serial Clock): linha que fornece o sinal de clock gerado pelo mestre (neste caso, o STM32).

Para garantir a integridade do sinal e o funcionamento adequado do barramento I²C, foram adicionados resistores *pull-up* de 10 k Ω nas linhas SDA e SCL, conectando-as à tensão de alimentação de 3.3 V. Esses resistores são essenciais para manter o nível lógico alto nas linhas quando não estão sendo ativamente puxadas para o nível baixo pelos dispositivos conectados.

3.1.3 Geolocalização

Para aquisição dos dados de posição geográfica, foram escolhidos os módulos GNSS da família u-blox, sendo os modelos NEO-6M e NEO-M8N inicialmente selecionados para compor o projeto. Ambos os módulos são amplamente utilizados em aplicações embarcadas devido ao seu baixo custo e também devido à ampla documentação disponível.

O circuito da Figura 7 foi projetado para suportar múltiplas arquiteturas de módulos da família u-blox, com a seleção realizada por meio de jumpers na placa. Essa abordagem foi adotada com o objetivo de proporcionar maior flexibilidade ao sistema, permitindo a utilização tanto de módulos que requerem EEPROM externa, como o NEO-6M, quanto daqueles que possuem armazenamento interno, como o NEO-M8N. Dessa forma, foi possível implementar diferentes receptores GNSS sem a necessidade de alterações futuras no *layout* da placa. Além disso, essa solução facilita eventuais atualizações ou adaptações no projeto.

Memória EEPROM

O circuito de interface conta com uma EEPROM I²C externa. Para isso, foi utilizado o modelo 24LC32A-I/SN, com capacidade de 32 kbits, suficiente para atender às necessidades do módulo. Esse circuito integrado é essencial para permitir que o módulo armazene dados, como a última posição conhecida e parâmetros de configuração personalizados.

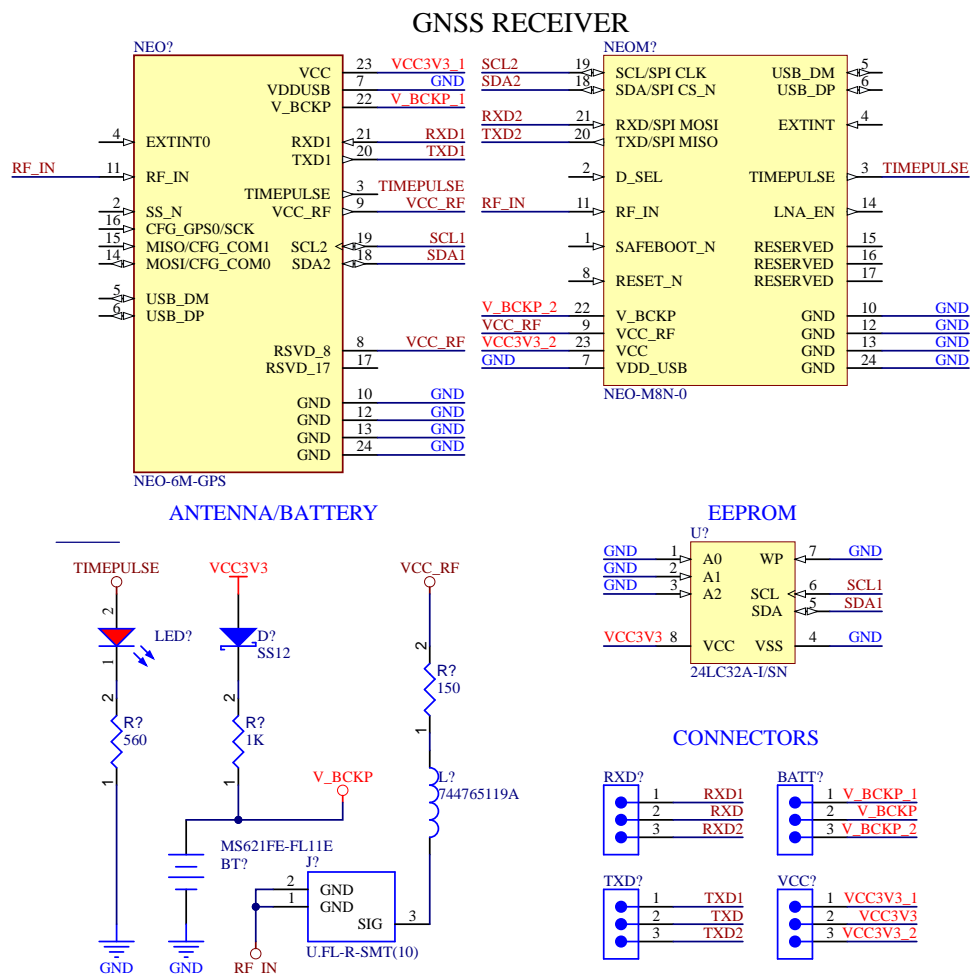
Bateria

Uma bateria do tipo botão, modelo MS621FE, foi adicionada ao circuito para fornecer energia ao RTC (*Real Time Clock*) interno e manter os dados da memória quando o módulo selecionado estiver desligado. Isso contribui para que o tempo de fixação inicial (*TTF Time To First Fix*) seja relativamente menor, pois permite que o módulo retome a navegação a partir do estado anterior.

Antena

Também foi incluído um conector para antena GPS ativa, conectada via cabo coaxial. A antena externa tem o papel de melhorar significativamente o ganho de sinal e trazer uma maior estabilidade de conexão com os satélites, especialmente em ambientes com interferência ou cobertura parcial.

Figura 7 – Circuito do GPS - U-Blox



Fonte: Elaborado pelo autor

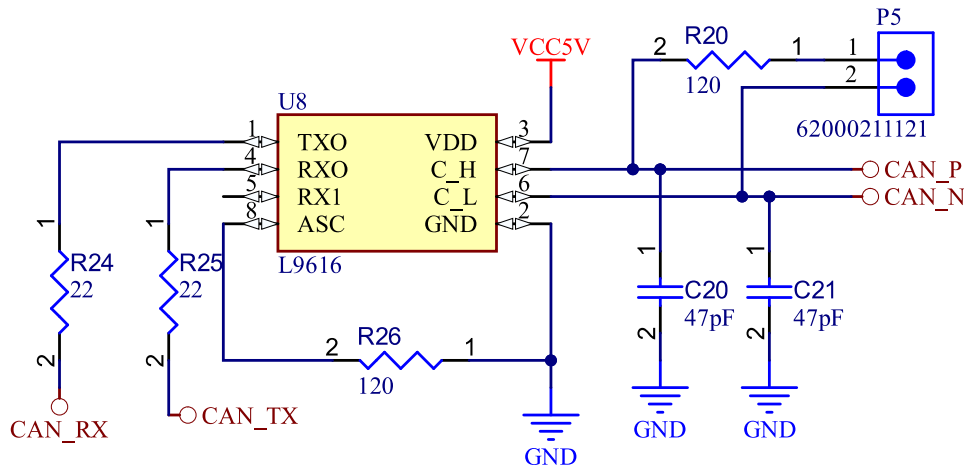
Comunicação com o Microcontrolador

Os dados são transmitidos ao microcontrolador via interface serial *UART*. O protocolo utilizado é o NMEA, que é o padrão na maioria dos receptores GNSS. O protocolo contém em sua estrutura informações de latitude, longitude, velocidade, número de satélites e horário UTC.

3.1.4 Telemetria

Para permitir a comunicação entre o sistema embarcado e outros módulos do veículo utilizando o protocolo *Controller Area Network* (CAN), foi necessário adicionar ao circuito um transceptor (Figura 8). O microcontrolador STM32F103C8T6 possui um controlador CAN integrado (periférico CAN), responsável pelo gerenciamento do protocolo no nível lógico. No entanto, para adaptar os níveis lógicos do microcontrolador aos níveis exigidos pelo barramento físico (CAN High / CAN Low), é necessário um transceptor CAN externo.

Figura 8 – Circuito do transceptor CAN - L9616



Fonte: Autor

O transceptor escolhido foi o L9616, fabricado pela STMicroelectronics. Este componente converte os sinais digitais do microcontrolador em sinais diferenciais compatíveis com a camada física do protocolo CAN e protege a linha de comunicação contra ruídos e transientes elétricos.

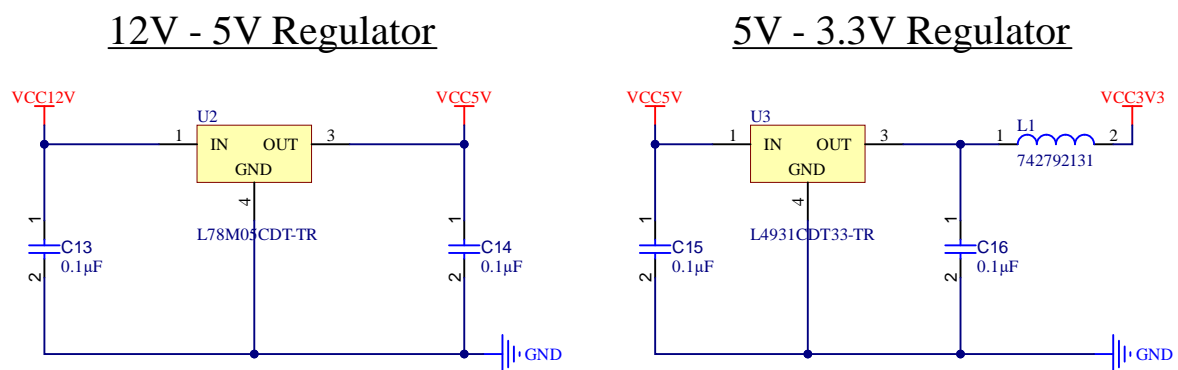
3.1.5 Regulador de Tensão

O circuito foi projetado considerando que a tensão de alimentação principal disponível é de 12V. No entanto, a maioria dos componentes eletrônicos da placa requer tensões menores e estáveis para operar corretamente. Por isso, foram utilizados reguladores lineares para converter essa tensão para 5V e 3,3V.

O regulador L78M05 converte a tensão de 12V para 5V, enquanto o regulador L4931CDT33 converte os 5V gerados anteriormente para 3,3V. Os capacitores de $0,1\ \mu\text{F}$ foram adicionados para filtrar ruídos de alta frequência e garantir maior estabilidade das tensões. Além disso, uma indutância foi utilizada para melhorar a rejeição de ruído nos circuitos sensíveis.

O circuito regulador de tensão pode ser visto na Figura 9.

Figura 9 – Circuito Regulador de Tensão



Fonte: Autor

3.2 Projeto da Placa de Circuito Impresso

Nesta etapa do desenvolvimento, o projeto da placa de circuito impresso (PCI) foi realizado utilizando o software Altium Designer, devido à sua capacidade de integração entre os esquemáticos e o *layout*. A placa integra, em uma única PCI, o microcontrolador STM32, sensores inerciais, módulo GPS, transceptor CAN, reguladores de tensão e os demais circuitos auxiliares descritos na seção anterior.

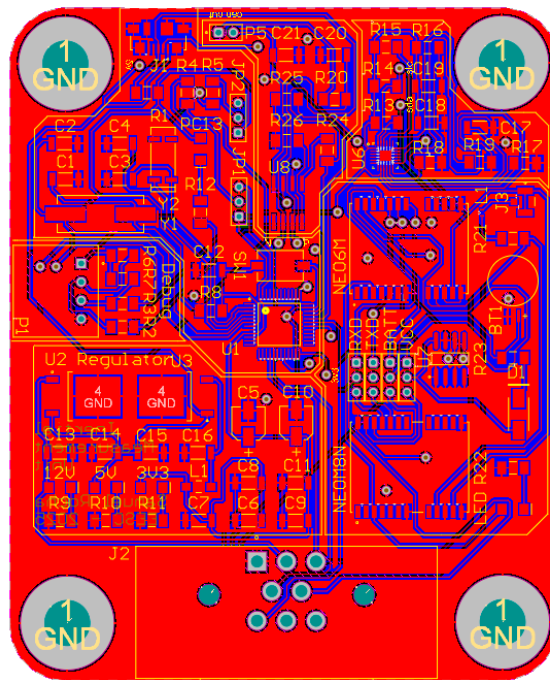
3.2.1 Considerações de Roteamento e *Layout*

Na modelagem, o microcontrolador foi posicionado de forma centralizada, com o objetivo de facilitar a conexão com os demais componentes do circuito. Idealmente, ao projetar uma PCI com um sensor inercial, recomenda-se que esse sensor seja posicionado o mais próximo possível do centro geométrico da placa, o que facilita a modelagem matemática. No entanto, essa escolha implicaria um roteamento mais complexo. Como a placa

é de pequenas dimensões, a posição do sensor em relação ao sistema como um todo tende a ser pouco relevante.

O processo de roteamento das trilhas da placa foi realizado manualmente, uma vez que a ferramenta de *autorouting* do software não apresentou boas soluções. Para permitir a interligação entre diferentes camadas e viabilizar algumas conexões, foi necessário o uso de furos de passagem (vias), especialmente em regiões com maior densidade de conexões. Além disso, foi implementado um plano de terra na placa, o que minimiza as interferências eletromagnéticas e elimina a necessidade de roteamento manual da trilha de GND. Também foram inseridos quatro furos aterrados, destinados à fixação mecânica à *housing*. A imagem com o *layout* das trilhas pode ser visualizada na figura 10.

Figura 10 – *Layout* de trilhas da PCI (visualização em duas camadas)



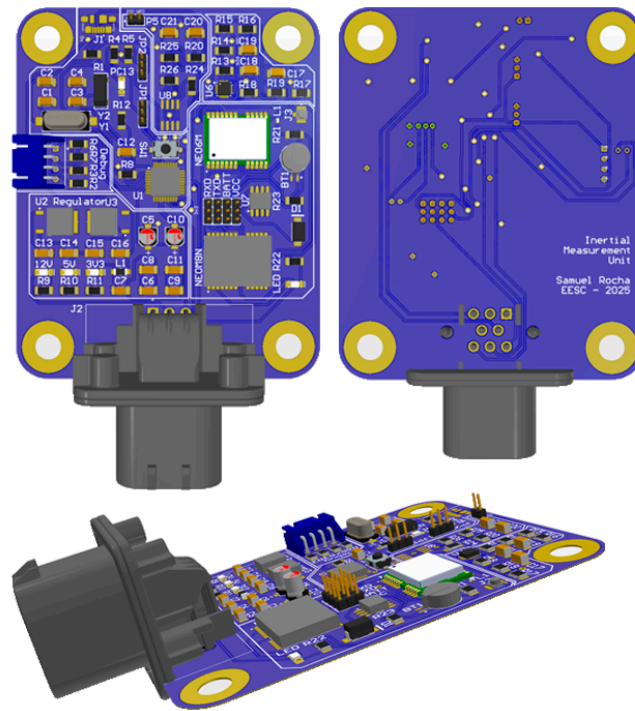
Fonte: Autor

Todos os componentes selecionados para a placa foram do tipo SMD. Para resistores e capacitores, adotou-se o encapsulamento 1206, por ser mais fácil de manusear e soldar manualmente, devido ao seu tamanho levemente maior em comparação com outros formatos.

Para atender ao padrão adotado pela equipe, o conector principal escolhido foi o modelo automotivo AMPSEAL de 8 vias. Além disso, para a depuração da placa, utilizou-se o conector JST da série XH, por ser um modelo simples e de fácil remoção.

A Figura 11, apresentada a seguir, mostra a visualização tridimensional completa da PCI.

Figura 11 – Visualização 3D da PCI gerada no Altium Designer



Fonte: Elaborado pelo autor

3.2.2 Manufatura da PCI

A fabricação da placa foi realizada pela empresa JLC PCB, utilizando um *stackup* de duas camadas. As características finais adotadas estão descritas na Tabela 3.

Tabela 3 – Características da placa de circuito impresso (PCI)

Parâmetro	Descrição
Dimensões aproximadas	90 mm (E) 75 mm
Número de camadas	Dupla face (2 camadas)
Espessura	1.6 mm
Espessura do cobre	1 oz (35 μ m)
Máscara de solda e serigrafia	Máscara azul, serigrafia branca
Furação	Metalização (vias)
Acabamento de superfície	HASL (com chumbo)

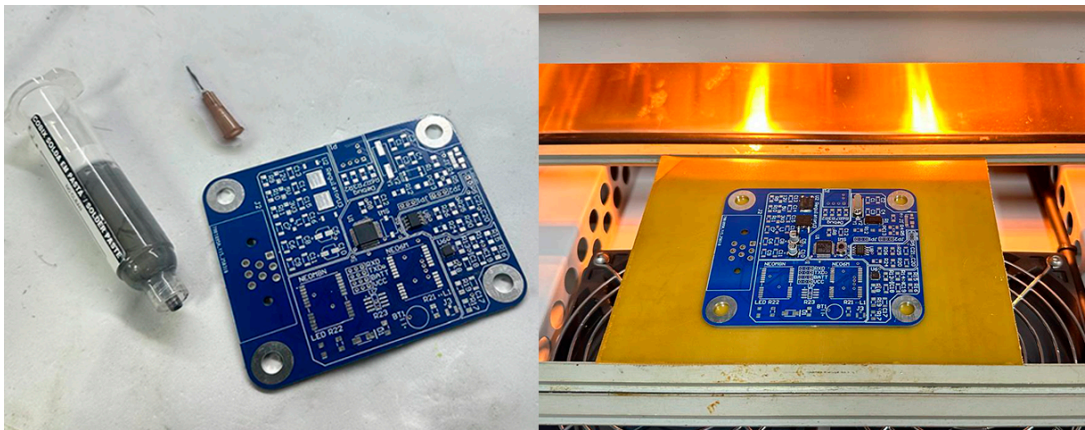
Fonte: Informações do fabricante da PCI.

3.2.3 Soldagem dos componentes

Após a fabricação da placa de circuito impresso o próximo passo no processo de manufatura é a soldagem dos componentes. Devido à complexidade imposta pelo reduzido tamanho da placa, a soldagem dos chips STM32F103C8T6 e MPU9250 foi realizada com o auxílio de pasta de solda e de um forno específico para a soldagem de componentes SMD. A Figura 12 ilustra o processo de soldagem desses componentes.

Todos os demais componentes da placa foram montados manualmente, com a utilização de uma estação de solda. Os subsistemas foram validados individualmente, assegurando inicialmente o correto funcionamento dos reguladores de tensão, seguido pela verificação do microcontrolador e, por fim, dos sensores em conjunto com o transceptor.

Figura 12 – Processo de solda SMD com forno de refusão



Fonte: Elaborado pelo autor

Figura 13 – Placa após solda dos componentes



Fonte: Elaborado pelo autor

4 SOFTWARE

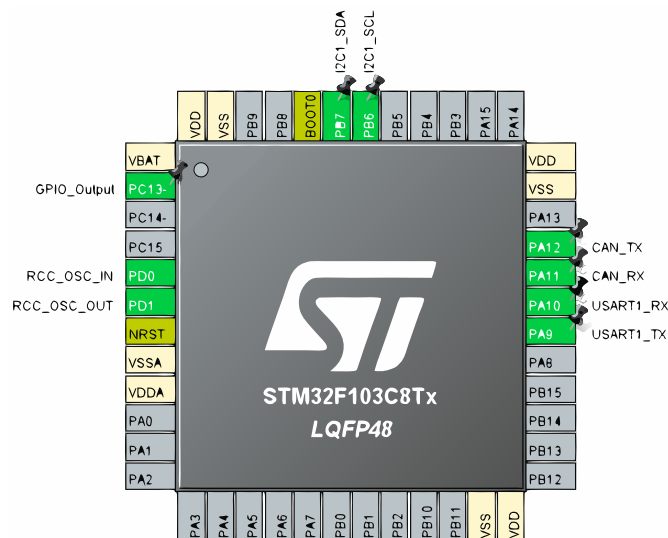
Este capítulo detalha o processo de desenvolvimento do software embarcado responsável pela aquisição, tratamento e transmissão dos dados dos sensores integrados à placa desenvolvida. A implementação foi realizada em linguagem C, utilizando o ambiente de desenvolvimento STM32CubeIDE.

O algoritmo de fusão sensorial foi baseado no projeto de código aberto de (UK, 2021), disponibilizado no seu GitHub, no qual se implementa um Filtro de Kalman em Python para realizar a previsão da posição com base em dados de GPS. No projeto em questão, utiliza-se um giroscópio MPU6050, um magnetômetro QMC5833L, um acelerômetro ADXL345 e um receptor GNSS da u-blox, modelo NEO-6M sendo este último o mesmo adotado neste trabalho.

4.1 Configuração da Plataforma de Hardware no STM32CubeIDE

O ambiente de desenvolvimento integrado fornecido pela STMicroelectronics possui ferramentas para a configuração de hardware, geração/edição de código e *debug* em um único ambiente. Uma das principais funcionalidades da ferramenta é o STM32CubeMX. Essa interface atua como uma camada de abstração de hardware, simplificando significativamente o processo de definição da pinagem da placa, inicialização e configuração de periféricos, eliminando a necessidade de manipulação direta de registradores. A Figura 14 mostra a configuração base definida para o STM32F103C8T6.

Figura 14 – Configuração de pinagem do microcontrolador



Fonte: Elaborado pelo autor

Para permitir a comunicação com a IMU foi configurada a interface I2C1 para os pinos PB6 (SCL) e PB7 (SDA). Para a aquisição dos dados do receptor NEO-6M, foi configurada a UART1, utilizando os pinos PA9 (TX) e PA10 (RX). Já a interface CAN1 do STM32 foi ativada para permitir a comunicação com o transceptor L9616, o qual realiza a interface elétrica entre o microcontrolador e o barramento CAN físico.

Um oscilador externo de 8 MHz (*HSE High Speed External*) foi habilitado, permitindo que, em conjunto com o PLL (textitPhase-Locked Loop), fosse gerado um clock de 72 MHz. Para garantir a execução periódica das tarefas de aquisição e processamento dos sensores, foi utilizado o *SysTick Timer*, configurado para gerar interrupções a cada 20 ms (frequência de 50 Hz). O *SysTick Timer* é usado como uma base temporal de referência do sistema, garantindo que os sensores, o algoritmo de Kalman e a transmissão CAN trabalhem de forma síncrona. Além disso, essa base de tempo é importante no cálculo do intervalo Δt , necessário para a integração da velocidade angular e para o modelo de predição.

Após a configuração de todos os periféricos, o software gerou os arquivos de abstração e inicialização do hardware com base na HAL da ST. Entre os arquivos criados, estão:

Tabela 4 – Principais arquivos gerados após configuração

Arquivo	Descrição
<code>main.c</code>	Entrada e inicialização do programa.
<code>stm32f1xx_hal_conf.c</code> , <code>stm32f1xx_it.c</code>	Configuração dos periféricos e tratadores de interrupção.
<code>stm32f1xx_hal_can.c</code> , <code>stm32f1xx_hal_gpio.c</code> , <code>stm32f1xx_hal_i2c.c</code> , <code>stm32f1xx_hal_usart.c</code>	Arquivos responsáveis por iniciar as interfaces CAN, GPIO, I2C e USART.

Fonte: Elaborado com base na estrutura gerada pelo STM32CubeIDE.

4.2 Implementação dos Drivers de Aquisição de Dados

No código de referência, desenvolvido em Python, o autor projeta a aquisição de dados de forma que os dados são inicialmente salvos em um arquivo .csv. O algoritmo é aplicado ao conjunto de dados em uma instância independente, posteriormente. Contudo, como uma das características cruciais do projeto é realizar o processamento e a transmissão em tempo real, foi necessário reformular o código em questão, retirando a etapa de armazenamento.

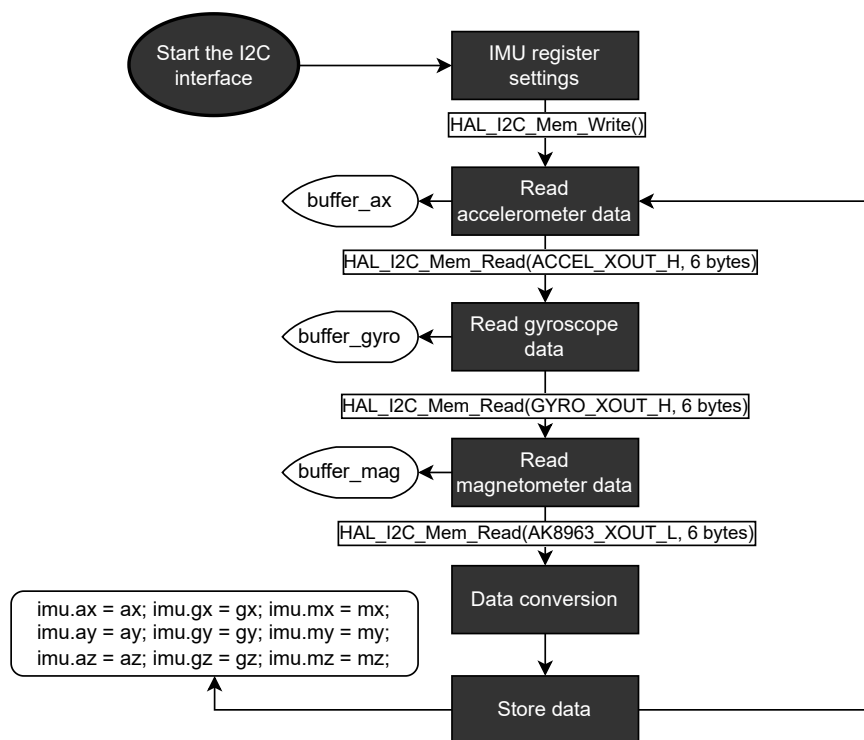
Assim, esta seção trata de uma parte importante do trabalho, que são os *drivers* de integração com os periféricos para aquisição dos dados dos sensores. O primeiro,

`imu_driver.c`, é responsável por coletar os dados do sensor de 9 graus de liberdade, e o segundo *driver*, `gps_driver.c`, é responsável pela aquisição e processamento dos dados recebidos via *UART* do receptor GPS do sistema embarcado. A imagem a baixo

4.2.1 Aquisição dos dados da IMU (MPU9250)

Como dito anteriormente, o sensor IMU utiliza uma interface I2C1 para comunicação com o microcontrolador. A interface foi configurada com clock de 100 kHz (Standard mode). O *driver* em questão tem o papel apenas de ler os dados dos nove eixos e armazená-los em variáveis globais correspondentes. Esses dados são brutos (*raw*) e, por esse motivo, não apresentam qualquer calibração ou compensação de bias para o acelerômetro, giroscópio ou magnetômetro. A Figura 15 apresenta um fluxograma organizado do funcionamento do *driver* de aquisição desses dados.

Figura 15 – Fluxograma do código de aquisição - IMU



Fonte: Elaborado pelo autor

A primeira função do *driver* é a de inicializar o sensor. Para isso, é necessário enviar um comando ao registrador `PWR_MGMT_1` para ativar o dispositivo. Em seguida, configuram-se as faixas de operação tanto do acelerômetro quanto do giroscópio, selecionando as escalas de $\pm 2g$ e $\pm 250^\circ/s$, as quais oferecem uma resolução adequada para a aplicação.

O acesso direto aos registradores internos é feito para obtenção dos dados do MPU9250. A função `HAL_I2C_Mem_Read()` permite acessar esses registradores específicos do sensor e, assim, obter os dados de leitura. O processo ocorre em ciclos de 20 ms (50 Hz), controlados pelo temporizador SysTick.

A cada ciclo, o microcontrolador realiza três etapas principais: leitura do acelerômetro, leitura do giroscópio e leitura do magnetômetro. Cada uma dessas etapas envolve a leitura de 6 bytes consecutivos (dois bytes por eixo, MSB + LSB), que são combinados para formar valores inteiros de 16 bits. Esses valores brutos são então convertidos em unidades físicas reais (m/s² para aceleração, rad/s para rotação e μ T para campo magnético), utilizando fatores de escala definidos com base na faixa de operação configurada durante a inicialização do sensor.

A leitura do magnetômetro AK8963 requer uma configuração especial: o MPU9250 deve estar com o modo de bypass ativado, permitindo que o STM32 comunique-se diretamente com o magnetômetro através do mesmo barramento I2C. Após a leitura dos valores dos três eixos magnéticos, os dados são organizados em uma estrutura única, contendo os nove valores referentes aos três sensores.

O código retorna uma estrutura `imu_data_t` com os campos `ax`, `ay`, `az`, `gx`, `gy`, `gz`, `mx`, `my` e `mz`. Essa estrutura é então passada diretamente ao algoritmo de calibração e aos algoritmos de fusão sensorial.

4.2.2 Aquisição dos dados do GPS (NEO-6M)

O módulo da *u-blox* transmite informações de posicionamento geográfico por meio de sentenças no padrão NMEA padrão utilizado pela maioria desses dispositivos. A comunicação com o microcontrolador STM32 é estabelecida utilizando a interface USART1, configurada para uma taxa de transmissão de 9600 bps, 8 bits de dados, sem paridade e 1 bit de parada, operando em modo de recepção assíncrona com interrupção.

O *driver* implementado tem como função capturar os dados diretamente do fluxo de caracteres da *UART*, identificar as sentenças válidas do tipo `GPRMC` e extrair as informações relevantes, como latitude, longitude e velocidade. A Figura 16 apresenta um fluxograma organizado do funcionamento do *driver* de aquisição desses dados.

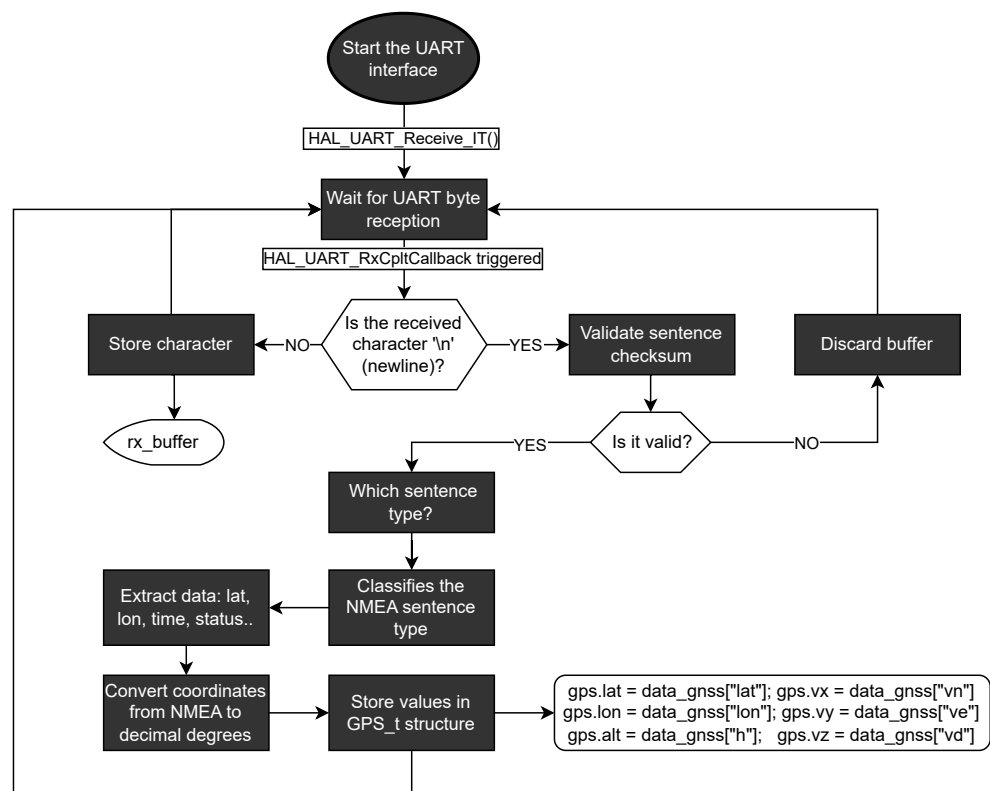
A aquisição inicia-se com a configuração da interrupção da *UART* para recepção de um byte por vez. A cada novo caractere recebido, o *driver* armazena os dados em um buffer até identificar o caractere de fim de linha (`\n`). Quando uma linha completa é recebida, o *driver* verifica se trata-se de uma sentença NMEA válida (como `GPGGA`, `GPRMC`, entre outras). Em caso afirmativo, os campos da sentença são extraídos utilizando funções de separação por delimitadores.

As coordenadas de latitude e longitude são inicialmente representadas no formato

NMEA (graus e minutos). Para serem utilizadas em cálculos internos e no filtro de Kalman, essas coordenadas são convertidas para o formato decimal em graus. A velocidade reportada em nós (knots) também é extraída da sentença e pode ser convertida para metros por segundo utilizando o fator de conversão $1 \text{ knots} = 0,5144 \text{ m/s}$.

As informações extraídas são armazenadas em uma estrutura `GPS_t`, contendo os dados de `latitude`, `longitude`, `altitude` e `velocidade`. A partir daí, essas variáveis globais ficam disponíveis para serem utilizadas nos algoritmos de predição e correção.

Figura 16 – Fluxograma do código de aquisição - GPS



Fonte: Elaborado pelo autor

4.3 Algoritmos de Fusão Sensorial

Nesta seção é apresentado o desenvolvimento dos algoritmos de processamento de dados utilizados no sistema embarcado para a estimativa da orientação (atitude) e da posição do veículo. Os modelos empregados baseiam-se na técnica de *fusão sensorial* e utilizam os dados brutos adquiridos por meio dos *drivers* previamente implementados, a fim de obter estimativas mais precisas, robustas e estáveis do estado do sistema.

A fusão sensorial é implementada de forma modular, considerando as limitações de processamento e memória do microcontrolador STM32F103C8T6, e estruturada para fa-

cilitar a manutenção e compreensão do código. Foram utilizados dois algoritmos distintos: o algoritmo de Madgwick, para a determinação da atitude (ângulos de *roll*, *pitch* e *yaw*), e o filtro de Kalman, para a fusão das informações de posição e velocidade horizontal fornecidas pelo GPS e pela IMU.

Dessa forma, o desenvolvimento é tratado em duas partes nesta seção. Na primeira, é apresentada a estimativa da atitude do sistema, a partir das leituras do MPU9250, com a aplicação do algoritmo de Madgwick para fusão sensorial. O algoritmo inclui uma etapa inicial de calibração dos sensores inerciais, responsável por eliminar desvios estáticos dos sinais.

Na segunda parte, é abordado o processo de estimativa da posição e velocidade horizontal do sistema, com base nos dados fornecidos pelo receptor GPS e na aplicação de um filtro de Kalman linear, inspirado na modelagem e estrutura do projeto de código aberto (UK, 2021), originalmente implementado em Python e adaptado neste trabalho para operação em tempo real na plataforma STM32.

Ambos os algoritmos foram implementados em linguagem C, com execução cíclica em tempo real e frequências de atualização adequadas a cada sensor (50 Hz para a IMU e 1 Hz para o GPS). Os fluxogramas apresentados ao longo da seção ilustram a arquitetura dos módulos implementados e o encadeamento lógico das etapas de processamento.

4.3.1 Estimativa de Atitude (Roll, Pitch, Yaw)

A atitude do veículo é dada pelos ângulos de *roll*, *pitch* e *yaw*, no qual o processo de estimativa é dividido em duas etapas principais. A primeira é composta pela calibração dos sensores inerciais para a correção de *bias* e offset estático advindos das leituras, garantindo, assim, uma maior precisão ao longo do tempo. Já na segunda etapa, as medições calibradas são processadas pelo algoritmo de Madgwick, que estima os ângulos de orientação durante o movimento em tempo real.

4.3.1.1 Pré-calibração do sensor IMU

O tratamento dos dados da IMU por meio da calibração visa compensar os erros sistemáticos dos sensores, especialmente os offsets (*bias*) presentes nas leituras do acelerômetro e do giroscópio. Desvios nas medições inerciais são esperados, pois estas estão sujeitas a incertezas e são corrompidas por diversas fontes de erro, incluindo ruídos, fatores de escala e variações de polarização (*bias*) com a temperatura, entre outros, afetando significativamente a acurácia dos valores estimados (INOUE, 2012).

O algoritmo realiza a calibração estática durante a inicialização do sistema do carro. Considera-se que o sistema está parado neste momento, e portanto, a aceleração medida deve ser próxima à gravidade no eixo vertical (geralmente $a_z \approx 9.81 \text{ m/s}^2$), e as taxas de rotação devem ser nulas.

Durante esse período, uma série de amostras é coletada e, em seguida, os valores médios de cada eixo são calculados, como mostra a Equação 4.1.

$$\text{bias}_{a,i} = \frac{1}{N} \sum_{k=1}^N a_{i,k}, \quad \text{bias}_{g,i} = \frac{1}{N} \sum_{k=1}^N g_{i,k} \quad (4.1)$$

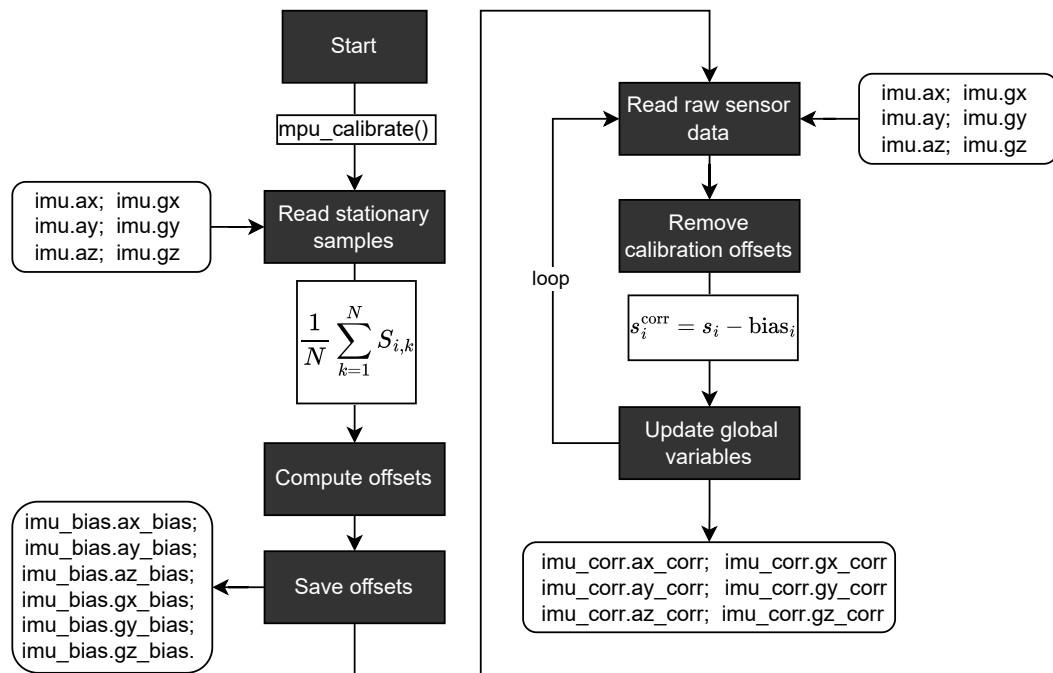
Na expressão, $a_{i,k}$ representa a leitura do acelerômetro no eixo i na amostra k , e $\omega_{i,k}$ representa a leitura do giroscópio no eixo i na amostra k . O termo N representa o número total de amostras coletadas durante o intervalo.

Após esse processo, os valores médios calculados são definidos como os offsets para cada eixo, sendo salvos em uma estrutura `imu_bias_t`. A partir de então, as próximas leituras são corrigidas, como demonstra a Equação 4.2, e então armazenadas em `imu_data_corr_t`.

$$a_i^{\text{corr}} = a_i - \text{bias}_{a,i}, \quad g_i^{\text{corr}} = g_i - \text{bias}_{g,i} \quad (4.2)$$

Esse procedimento permite reduzir significativamente o erro acumulado (*drift*) do giroscópio e melhorar a confiabilidade na estimativa dos ângulos de orientação. A Figura 17 apresenta o fluxograma que detalha o algoritmo de calibração.

Figura 17 – Fluxograma do código de calibração dos sensores inerciais



4.3.1.2 Cálculo da Orientação - Filtro de Madgwick

A partir das medições corrigidas (`imu_data_corr_t`), a estimação da atitude do veículo foi feita utilizando o método de fusão sensorial com filtro de Madgwick. Este algoritmo faz atualizações iterativas do *quaternion* $q = [q_0, q_1, q_2, q_3]$, o qual representa a orientação do sistema com base na integração das leituras de rotação (giroscópio) e na correção orientada pelos vetores de gravidade (acelerômetro) e campo magnético (magnetômetro)(MADGWICK, 2010).

Os ângulos de orientação são encapsulados em uma função dedicada do software embarcado e retorna a estrutura `attitude_data_t` como mostra a Tabela 5. Esta seção detalha o desenvolvimento e integração desse algoritmo ao projeto.

Tabela 5 – Estruturas de dados integradas ao algoritmo

Estrutura	Descrição
<code>imu_data_corr_t</code>	Contém os dados corrigidos de aceleração (a_x, a_y, a_z) e velocidade angular (g_x, g_y, g_z), livres de offsets.
<code>attitude_data_t</code>	Armazena os resultados finais dos ângulos de orientação em graus: <code>roll</code> , <code>pitch</code> e <code>yaw</code> .

Fonte: Elaboração própria, baseada na estrutura de dados do algoritmo.

Conversão das Unidades de Velocidade Angular

Como as velocidades angulares fornecidas pelo giroscópio são originalmente medidas em graus por segundo ($^\circ/s$), a primeira etapa do código está na conversão dessas medidas para radianos por segundo (rad/s) aplicada a cada eixo ($i = x, y, z$). Para essa conversão, utiliza-se a relação dada pela Equação 4.3.

$$\omega_i(\text{rad/s}) = g_i(^\circ/\text{s}) \cdot \frac{\pi}{180} \quad (4.3)$$

Cálculo da Derivada do *Quaternion*

Como introduzido anteriormente, a rotação tridimensional do sistema é representada por um *quaternion* $\mathbf{q} = [q_0, q_1, q_2, q_3]$. A atualização desse *quaternion* segue a taxa de variação ou derivada temporal, expressa pela seguinte equação:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega}_q \quad (4.4)$$

em que $\boldsymbol{\omega}_q = [0, \omega_x, \omega_y, \omega_z]$ representa as velocidades angulares corrigidas e convertidas. Estas são definidas como:

$$\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T \quad (4.5)$$

O operador \otimes denota o produto de *quaternions*, resultando na seguinte forma explícita para o cálculo de $\dot{\mathbf{q}}$:

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} -q_1\omega_x - q_2\omega_y - q_3\omega_z \\ q_0\omega_x + q_2\omega_z - q_3\omega_y \\ q_0\omega_y - q_1\omega_z + q_3\omega_x \\ q_0\omega_z + q_1\omega_y - q_2\omega_x \end{bmatrix} \quad (4.6)$$

É importante mencionar que a inicialização é feita com $\mathbf{q} = [1, 0, 0, 0]$, que representa a orientação neutra do sistema no instante inicial. Embora esse valor inicial não corresponda necessariamente à orientação real do corpo, o algoritmo de Madgwick conta com a etapa de correção por gradiente descendente, como será exposto a seguir, o que permite ajustar rapidamente o *quaternion* aos valores reais.

Normalização dos Vetores de Aceleração e Magnetômetro

Antes da correção e integração, e de modo a reduzir o impacto de ruídos e variações de ganho que podem acabar induzindo erros na estimativa da atitude, os vetores de aceleração \mathbf{a} e do magnetômetro \mathbf{m} são normalizados. Isso garante que apenas a direção das medições influencie a atualização do *quaternion*, removendo o efeito de variações de magnitude.

$$\hat{\mathbf{a}} = \frac{\mathbf{a}}{\|\mathbf{a}\|}, \quad \hat{\mathbf{m}} = \frac{\mathbf{m}}{\|\mathbf{m}\|} \quad (4.7)$$

Correção por Gradiente Descendente

Para alinhar a orientação estimada com as direções medidas a fusão sensorial de Madgwick inclui um termo de correção baseado em gradiente descendente. O termo atua para minimizar o erro entre as direções previstas (calculadas a partir do *quaternion* atual) e as direções reais de gravidade e campo magnético, garantindo estabilidade numérica e reduzindo o *drift* inerente ao giroscópio.

O cálculo de correção é baseado em um vetor de erro \mathbf{f} , que representa a discrepância entre o modelo interno (previsto) e as medições reais. O gradiente desse erro em

relação aos componentes do *quaternion*, $\nabla \mathbf{f}$, é então determinado e utilizado para ajustar a derivada temporal do *quaternion*.

Essa etapa de correção é expressa por:

$$\dot{\mathbf{q}}_{\text{corr}} = \dot{\mathbf{q}} - \beta \nabla \mathbf{f} \quad (4.8)$$

onde β é um parâmetro de ganho ajustável que controla a rapidez de convergência do filtro. Esse ajuste iterativo, incorporando a informação de gravidade e campo magnético, assegura que a estimativa final de atitude permaneça alinhada com as medições físicas e seja robusta frente a perturbações e ruídos.

Integração e Normalização do *Quaternion*

Concluídas as etapas de cálculo da derivada do *quaternion* e a aplicação do termo de correção, o algoritmo realiza a integração e normalização para atualização da orientação estimada. Essa fase, executada a cada ciclo de 20 ms (taxa de 50 Hz, definida pelo temporizador `SysTick`), consiste em integrar numericamente a derivada corrigida utilizando o método de Euler, resultando em um novo *quaternion* \mathbf{q}_{k+1} :

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \dot{\mathbf{q}}_{\text{corr}} \cdot \Delta t \quad (4.9)$$

Em seguida, para assegurar que o *quaternion* continue representando uma rotação unitária — uma exigência matemática para evitar distorções na estimativa de orientação — o *quaternion* resultante é normalizado:

$$\mathbf{q}_{k+1} = \frac{\mathbf{q}_{k+1}}{\|\mathbf{q}_{k+1}\|} \quad (4.10)$$

Esse processo iterativo garante a estabilidade numérica e a coerência física da orientação tridimensional estimada. Com isso, o sistema embarcado atualiza de forma contínua e eficiente os ângulos de Euler (*roll*, *pitch* e *yaw*) a cada ciclo, permitindo o acompanhamento em tempo real das variações de atitude do veículo, mesmo em ambientes dinâmicos e sujeitos a ruídos.

Conversão para Ângulos de Euler:

O *quaternion* final é convertido em ângulos de Euler (*roll*, *pitch*, *yaw*) usando as seguintes expressões:

$$\text{roll} = \arctan 2(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \quad (4.11)$$

$$\text{pitch} = \arcsin(2(q_0q_2 - q_3q_1)) \quad (4.12)$$

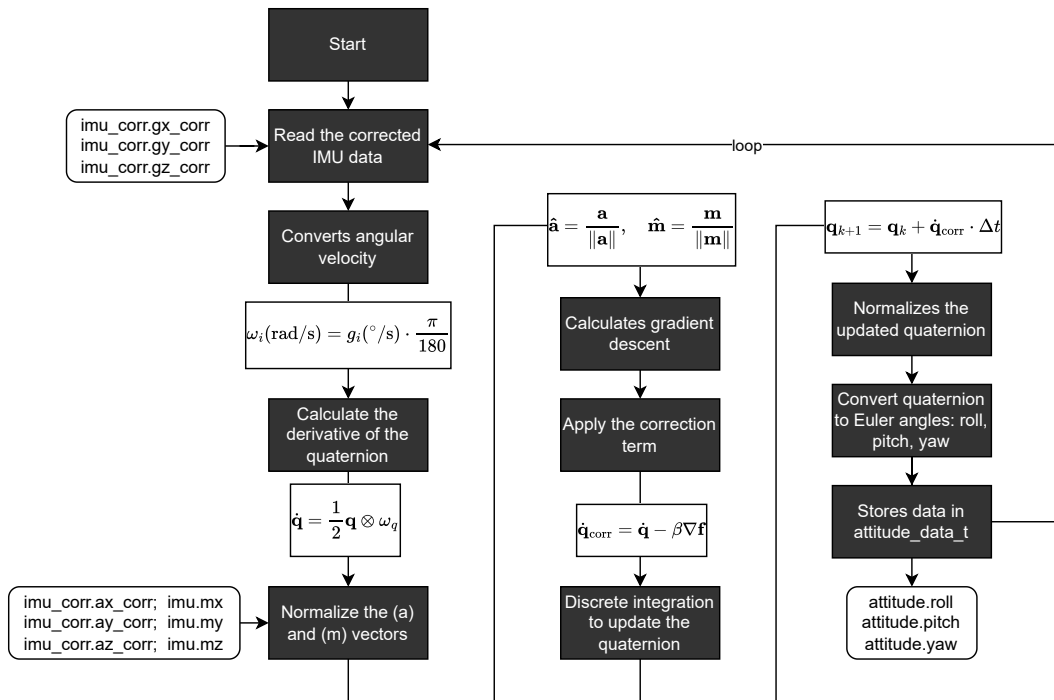
$$\text{yaw} = \arctan 2(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \quad (4.13)$$

Esses ângulos são então convertidos de radianos para graus e armazenados em `attitude_data_t`.

Fluxograma do Algoritmo

A Figura 18 apresenta o fluxograma detalhado de todas as etapas descritas, ilustrando a integração entre leituras de sensores, correção de drift e atualização final da orientação.

Figura 18 – Fluxograma do código para orientação - Filtro de Madgwick



Fonte: Elaborado pelo autor

Assim, com base em todas essas etapas, mesmo em ambientes sujeitos a ruído e perturbações, o filtro Madgwick implementado garante uma estimativa confiável dos ângulos de rolagem (ϕ), arfagem (θ) e guinada (ψ). Os valores de saída do filtro já são os valores finais que irão compor o pacote de dados a ser transmitido pelo protocolo de comunicação.

4.3.2 Estimativa de Posição e Velocidade - Filtro de Kalman

Uma das etapas cruciais desse projeto está na estimativa da posição do veículo, contudo, ao escolhermos sensores de baixo custo, estamos sujeitos a interferências. Suas medidas são transmitidas em baixas frequências, levando à perda de sinal com satélites, o qual é facilmente obstruído (INOUE, 2012).

Devido a essas limitações, é proposta a integração de um INS (Sistema de Navegação Inercial) com o receptor GPS para a estimativa da navegação. O filtro de Kalman linear discreto permite então calcular a posição e a velocidade horizontal do veículo, integrando as medições do GPS com um modelo dinâmico simplificado de movimento, resultando em estimativas mais confiáveis e estáveis. Desta forma, simplificada, o INS fornece o posicionamento quando os dados do GPS estão indisponíveis ou são de má qualidade, enquanto o GPS ajuda a manter os erros de navegação do INS dentro de uma faixa especificada. (ALABA, 2024)

Esta seção detalha o desenvolvimento e a integração desse algoritmo de aplicação do filtro implementado em linguagem C, adaptando a estrutura e as ideias centrais do código em Python (UK, 2021). Os dados de posição e velocidade são encapsulados e armazenados na estrutura `position_estimate_t`, como mostra a Tabela 6.

Tabela 6 – Estruturas de dados integradas ao filtro de Kalman para posição

Estrutura	Descrição
<code>gps_data_t</code>	Contém as medições brutas do receptor GPS, incluindo latitude (<code>gps.lat</code>), longitude (<code>gps.lon</code>), velocidades horizontais (<code>gps.vx</code> , <code>gps.vy</code>).
<code>imu_data_corr_t</code>	Contem os dados do acelerômetro e giroscópico corrigidos.
<code>attitude_data_t</code>	Contem estimativas de atitude (atribuídas pelo filtro Madgwick).
<code>position_estimate_t</code>	Armazena as estimativas finais de posição (<code>x</code> , <code>y</code>) [m] e velocidade horizontal (<code>vx</code> , <code>vy</code>) [m/s].

Fonte: Elaboração própria, baseada na estrutura de dados do algoritmo.

Conversão de Coordenadas para o Sistema Local

Como vimos, receptores *u-blox* fornecem coordenadas geográficas em latitude e longitude (`gps.lat` e `gps.lon`), expressas em graus decimais. Entretanto, para integrar essas medições no modelo de Kalman, tipicamente adota-se o sistema cartesiano local (em metros), sendo necessário adotar uma conversão. Para isso, a transformação é feita por

meio de uma *projeção equiretangular*, que aproxima localmente a superfície da Terra como um plano.

Matematicamente, essa conversão segue as seguintes expressões:

$$\begin{aligned}x &= R \cdot (\lambda - \lambda_0) \cdot \cos(\phi_0) \\y &= R \cdot (\phi - \phi_0)\end{aligned}\tag{4.14}$$

onde:

- λ, ϕ : longitude e latitude atuais do ponto (em radianos), respectivamente.
- λ_0, ϕ_0 : longitude e latitude do *ponto de referência fixo*, escolhido como origem do sistema local (em radianos).
- R : raio médio da Terra, assumido como $R \approx 6378137$ m.

A conversão de graus para radianos é feita pela relação:

$$\text{rad} = \text{graus} \cdot \frac{\pi}{180}\tag{4.15}$$

No contexto do projeto, a origem local (`lon_ref`, `lat_ref`) referência para as estimativas subsequentes é dada pela posição inicial do veículo ao iniciar o sistema.

Deste modo, a projeção equiretangular permite a aplicação do algoritmo ao simplificar o modelo de previsão (representado pela matriz \mathbf{A} em metros), essencial para integrar as posições em um sistema de referência coerente com as demais medições, como as velocidades lineares estimadas do GPS, já expressas em m/s.

Definição do Modelo de Estados e Dinâmica

Inicialmente, o filtro de Kalman adota um modelo de movimento simplificado, assumindo que o veículo se move com velocidade constante entre as amostras uma suposição razoável para sensores com baixa taxa de atualização, como o GPS operando a 1 Hz. O vetor de estado é definido como:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ v_{x_k} \\ v_{y_k} \end{bmatrix}$$

onde (x_k, y_k) representam as posições locais do veículo em metros, e (v_{x_k}, v_{y_k}) correspondem às velocidades projetadas nos eixos norte e leste, respectivamente.

A evolução desse estado é atualizada com base em um modelo de movimento com aceleração constante, no qual a dinâmica do sistema incorpora tanto a propagação do estado anterior por meio da matriz de transição \mathbf{A} , quanto a influência da aceleração medida pela IMU, por meio da matriz de controle \mathbf{B} . Assim, a predição do estado é descrita por:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A} \cdot \hat{\mathbf{x}}_{k-1} + \mathbf{B} \cdot \mathbf{u}_k \quad (4.16)$$

onde $\hat{\mathbf{x}}_{k-1}$ é o vetor de posição e velocidade estimado no instante anterior.

A matriz responsável pela transição de estado e a matriz de controle, que incorporam os efeitos da aceleração na predição da posição e da velocidade, são apresentadas a seguir:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \quad (4.17)$$

O vetor de entrada $\mathbf{u}_k = [a_{x_k}, a_{y_k}]^T$ da Equação 4.16 representa a aceleração linear no referencial do solo, obtida da IMU. Esse termo permite a predição da trajetória entre medições do GPS e durante eventuais perdas de sinal, com o sistema inercial (INS) atuando na predição e o GPS na correção.

O cálculo de \mathbf{u}_k é feito a partir da atitude e aceleração da IMU, transformadas pela matriz de cossenos direcionais (DCM) \mathbf{C}_{bn} , que relaciona o referencial do corpo ao de navegação (Equação 4.18).

$$\mathbf{u}_k = \mathbf{C}_{bn} \cdot \mathbf{a}_b \quad (4.18)$$

A matriz \mathbf{C}_{bn} é gerada a partir dos ângulos de Euler que representam a orientação do sistema, calculados pelo algoritmo de Madgwick, disponíveis em `attitude.roll`, `attitude.pitch` e `attitude.yaw`.

$$\mathbf{C}_{bn} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \quad (4.19)$$

Onde $c_\phi = \cos(\phi)$, $s_\phi = \sin(\phi)$, $c_\theta = \cos(\theta)$, $s_\theta = \sin(\theta)$, $c_\psi = \cos(\psi)$ e $s_\psi = \sin(\psi)$.

Tendo em mãos essa matriz, o algoritmo transforma o vetor de aceleração da IMU (medido no referencial do corpo) para o referencial de navegação. A entrada de controle \mathbf{u}_k

utilizada no filtro de Kalman é então representada pelas componentes horizontais dessa aceleração, para o modelo de estados, expressa no mesmo referencial das variáveis de posição e velocidade.

$$\mathbf{u}_k = \begin{bmatrix} a_{x_k} \\ a_{y_k} \end{bmatrix} \quad (4.20)$$

Por fim, é válido pontuar que a modelagem do filtro feita por (UK, 2021) considera que a velocidade permanece constante no intervalo Δt , o que simplifica os cálculos. Essa escolha é compatível com a dinâmica do veículo em percursos reduzidos, como ambientes urbanos, conforme discutido em (SRINIVASAN, 2018), ou, no nosso cenário, uma pista.

Modelo de Observação e Integração com GPS

As medições diretas fornecidas pelo GPS posição (`gps.lat`, `gps.lon`, convertidas para x , y) e velocidades horizontais (`gps.vx`, `gps.vy`) - são integradas diretamente ao filtro por meio do modelo de observação \mathbf{H} , o qual relaciona os estados observáveis com as medições disponíveis.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assim, sua função nada mais é do que informar o que está sendo medido e como isso se relaciona com o vetor de estado. Esse modelo permite que as medições do GPS sejam comparadas diretamente com as predições do filtro, possibilitando a correção eficiente dos estados estimados.

Incertezas associadas

Assim como qualquer modelo de predição, o filtro de Kalman também propaga as incertezas associadas à estimativa. Essa propagação é expressa por:

$$\mathbf{P}_{k|k-1} = \mathbf{A} \cdot \mathbf{P}_{k-1} \cdot \mathbf{A}^\top + \mathbf{Q} \quad (4.21)$$

Nessa equação, \mathbf{Q} representa a matriz de covariância do ruído de processo, ajustada para refletir a incerteza associada à hipótese de movimento constante; \mathbf{P}_{k-1} , a matriz de covariância do erro estimado no instante anterior; e \mathbf{A} , como já apresentado, representa a matriz de transição de estados do modelo.

A matriz \mathbf{Q} , nesse caso, é definida como uma matriz 4×4 , sendo não nula apenas nas componentes de posição x e y :

$$\mathbf{Q} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.22)$$

Os valores de σ_x e σ_y são calculados dinamicamente a partir da incerteza do GPS (desvios padrão de latitude e longitude). Portanto, a modelagem leva em conta apenas as incertezas associadas à posição enquanto os relacionados a velocidade é considerado como nulos.

Essa escolha reflete uma estratégia comum em filtros de Kalman para navegação com sensores de baixo custo, como discutido em (SRINIVASAN, 2018). Ao ignorar as incertezas associadas à velocidade, reduz-se o aumento variância dos estados, permitindo uma estimativa mais estável.

Correção com as Medições do GPS

A última etapa do código que implementa o filtro é correção o vetor de estados ajustando a predição feita com base na dinâmica do sistema e incorporando a informação de posição mais recente proveniente do módulo GPS.

O ganho de Kalman \mathbf{K}_k é o fator que faz a ponderação entre a estimativa feita pelo modelo e a informação do sensor. Seu cálculo leva em conta tanto a matriz de covariância predita $\mathbf{P}_{k|k-1}$ quanto a matriz de covariância das medições \mathbf{R} :

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \cdot \mathbf{H}^\top \cdot \left(\mathbf{H} \cdot \mathbf{P}_{k|k-1} \cdot \mathbf{H}^\top + \mathbf{R} \right)^{-1} \quad (4.23)$$

A matriz de covariância do ruído de medição, \mathbf{R} reflete a incerteza associada às medições fornecidas pelo GPS. No algoritmo implementado, essa matriz é definida como uma matriz diagonal com valores constantes assumindo empiricamente um desvio padrão de 100 unidades para as medições de posição e 0,5 m/s para as velocidades horizontais.

$$\mathbf{R} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_y}^2 \end{bmatrix} \quad (4.24)$$

Com o ganho definido, o vetor de estado é atualizado pela incorporação do erro de inovação, ou resíduo, que mede a diferença entre a medição real e a predita pelo modelo:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \cdot (\mathbf{z}_k - \mathbf{H} \cdot \hat{\mathbf{x}}_{k|k-1})$$

Além disso, a matriz de covariância é ajustada para refletir a nova incerteza após a atualização:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \cdot \mathbf{H}) \cdot \mathbf{P}_{k|k-1}$$

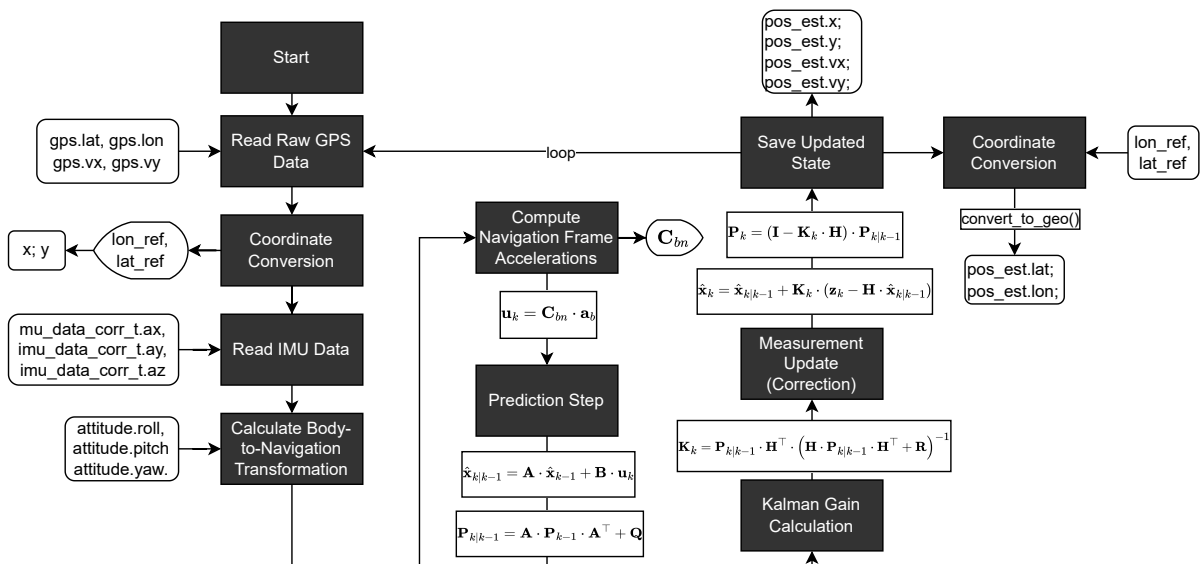
Ao término de cada ciclo, as estimativas de posição (x, y) e velocidades (v_x, v_y) corrigidos são salvas na estrutura `position_estimate_t`.

Fluxograma do Algoritmo

A Figura 19 apresenta o fluxograma detalhado do funcionamento do algoritmo de fusão de posição e velocidade, desde a leitura das medições do GPS até a atualização final dos estados estimados.

Com base em todas essas etapas de integração dos dados de GPS, projeção para o sistema local e atualização iterativa do filtro de Kalman — o sistema embarcado consegue obter uma estimativa de posição e da velocidade horizontal do veículo, mesmo em ambientes dinâmicos e sujeitos a perda de sinais de GPS.

Figura 19 – Fluxograma do código para posição - Filtro de Kalman



Fonte: Elaborado pelo autor

No diagrama, a função `convert_to_geo()` é quem faz a conversão reversa das coordenadas locais `position_estimate_t.x` e `position_estimate_t.y` para latitude e longi-

tude em graus decimais. Essa função aplica a projeção equiretangular inversa utilizando o ponto de referência inicial (`lon_ref`, `lat_ref`) e o raio médio da Terra. Os resultados dessa conversão são salvos nas variáveis `position_estimate_t.lat` e `position_estimate_t.lon`, respectivamente.

4.4 Implementação da Comunicação CAN

No pipeline do software embarcado podemos dizer que a transmissão dos dados é a última etapa do processo como um todo. O protocolo *Controller Area Network* (CAN) é responsável pelo envio das informações até aqui adquiridas e tratadas. Essa interface é o que permitirá a comunicação deste com os outros módulos eletrônicos embarcados do carro.

Todo o código compreendido no arquivo `can_comm.c` é responsável pela preparação, encapsulamento e envio dos dados aferidos e estimados. A transmissão de dados utiliza a biblioteca `STM32F1xx_HAL_CAN.c`, parte do HAL (Hardware Abstraction Layer) disponibilizado pela STMicroelectronics. As principais funções empregadas são:

- `HAL_CAN_AddTxMessage()`: responsável por adicionar a mensagem ao *mailbox* de transmissão do periférico CAN.
- `HAL_CAN_GetTxMailboxesFreeLevel()`: utilizada para verificar se há espaço disponível para transmissão antes de enviar a mensagem.

Estrutura Geral do Código

O primeiro ponto do código está na configuração do periférico CAN através do `CAN_HandleTypeDef` e na ativação do periférico com a biblioteca `stm32f1xx_hal_can.c`.

Após isso é feita a preparação dos dados tomando as informações relevantes, armazenadas em estruturas e organizando em variáveis locais. Esses dados são então convertidos e empacotados em uma *array* de 8 bytes.

Em seguida, cria-se a estrutura de transmissão. Para cada mensagem a ser enviada, é instanciada uma estrutura do tipo `CAN_TxHeaderTypeDef`, que define parâmetros como o identificador da mensagem, o tamanho dos dados e outras configurações necessárias para a transmissão.

Para enviar uma mensagem, o sistema realiza uma verificação dos *mailboxes* disponíveis no periférico CAN por meio da função `HAL_CAN_GetTxMailboxesFreeLevel()`, assegurando que há espaço livre para a nova mensagem ser transmitida.

Com os dados preparados e o canal de transmissão disponível, ocorre o envio da mensagem. A Função 4.25 é chamada, recebendo como parâmetros o identificador do barramento CAN (`hcan`), a estrutura de cabeçalho `CAN_TxHeaderTypeDef` e o array de dados empacotados.

```
HAL_CAN_AddTxMessage(&hcan, &TxHeader, data, &TxMailbox);      (4.25)
```

Esse processo é então repetido para cada tipo de mensagem que precisa ser transmitida. Por exemplo, mensagens com identificadores de `0x100` até `0x105` podem ser utilizadas para transmitir diferentes variáveis do sistema, garantindo que todas as informações relevantes sejam enviadas ciclicamente através do barramento CAN.

Estruturas de Dados Transmitidas

Os dados a serem transmitidos são organizados em estruturas (`structs`) que representam as informações sistema. Como o CAN permite enviar quadros de até 8 bytes, a ideia foi aproveitar ao máximo cada frame. Para isso, foi agrupado os dados relacionados ou com tamanho fixo.

A Tabela 7 define a estrutura adotada para o grupamento desses dados.

Tabela 7 – Mensagens e dados transmitidos via protocolo CAN

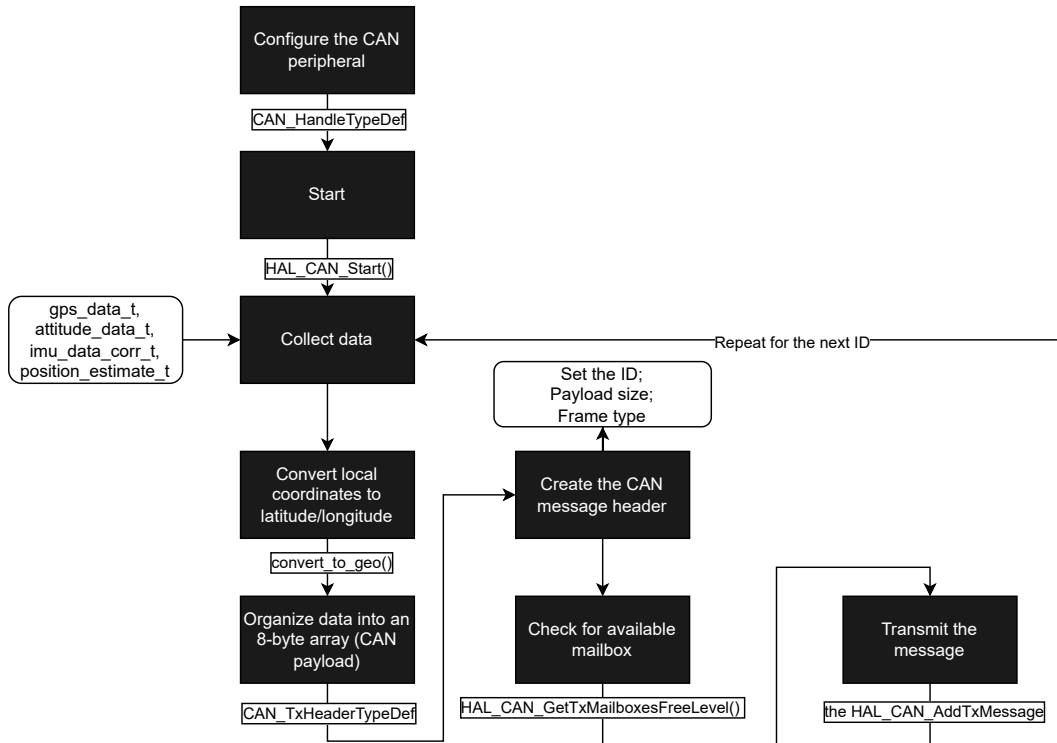
ID	Conteúdo	Bytes	Variáveis Usadas
0x100	Latitude (4B) e Longitude (4B). Posições em graus decimais.	8	<code>position_estimate_t.lat</code> <code>position_estimate_t.lon</code>
0x101	Altitude (4B) e <i>Heading</i> (4B). Altitude e ângulo de guinada.	8	<code>gps_data_t.alt</code> <code>attitude_data_t.yaw</code>
0x102	Roll (2B), Pitch (2B), Yaw (2B) e Reservado (2B). Ângulos de orientação.	8	<code>attitude_data_t.roll</code> <code>attitude_data_t.pitch</code> <code>attitude_data_t.yaw</code>
0x103	Acc X (2B), Acc Y (2B), Acc Z (2B) e Reservado (2B). Acelerações corrigidas.	8	<code>imu_data_corr_t.ax</code> <code>imu_data_corr_t.ay</code> <code>imu_data_corr_t.az</code>
0x104	Hora (1B), Min (1B), Seg (1B), Fix (1B), Dia (1B), Mês (1B), Ano (1B) e Reservado (1B). Tempo e qualidade do sinal.	8	<code>gps_data_t.time</code> <code>gps_data_t.validity</code>
0x105	Velocidade X (4B) e Velocidade Y (4B). Velocidades horizontais.	8	<code>position_estimate_t.vx</code> <code>position_estimate_t.vy</code>

Fonte: Elaboração própria

Fluxograma do Algoritmo

A Figura 20 apresenta o fluxograma detalhado do funcionamento do código de transmissão de dados via CAN, destacando as etapas de preparação, encapsulamento e envio das mensagens a partir das estruturas de dados geradas no sistema embarcado.

Figura 20 – Fluxograma do código transmissão - CAN



Fonte: Elaborado pelo autor

Com essa implementação, a comunicação entre o sistema embarcado e outros módulos (ou um módulo central de controle, por exemplo) é feita de forma eficiente, com atualização em tempo real e alta confiabilidade, mesmo em ambientes ruidosos. O protocolo CAN, aliado à estrutura modular do `can_comm.c`, garante a consistência e a segurança na transmissão e recepção dos dados críticos para a navegação e controle do veículo.

4.5 Considerações Finais

O software embarcado foi desenvolvido de forma que a arquitetura permite-se que cada camada de software seja testada, modificada e expandida de forma independente, garantindo robustez ao sistema como um todo. A Figura 21 ilustra essa estrutura com os diferentes módulos desde as rotinas principais de controle, passando pelos filtros de fusão sensorial (Madgwick e Kalman), até os *drivers* e bibliotecas de hardware.

Figura 21 – Pipeline do código

Main	main.c main.h	
Interface	position.c position.h	
	attitude.c attitude.h	
Filter	kalman_position.c kalman_position.h	
	madgwick_attitude.c madgwick_attitude.h	
Driver	imu_driver.c imu_driver.h	
	gps_driver.c gps_driver.h	
Protocol	can_comm.c can_comm.h	
SYS	stm32f1xx_hal_conf.c,	stm32f1xx_it.c
HAL	stm32f1xx_hal_can.c	stm32f1xx_hal_usart.c
	stm32f1xx_hal_gpio.c	stm32f1xx_hal_i2c.c

Fonte: Elaborado pelo autor

5 ANÁLISE DE DADOS E RESULTADOS

Este capítulo apresenta a análise dos dados coletados para a validação do desempenho da Unidade de Medição Inercial (IMU) desenvolvida. A análise avalia as estimativas de navegação e atitude obtidas por meio de fusão sensorial, com o objetivo de comprovar a robustez e a acurácia do sistema. Dessa forma, são discutidas as medições realizadas, as comparações com dados de referência, bem como as análises dos erros e das incertezas geradas pelo sistema.

5.1 Metodologia de Coleta de Dados

A validação foi realizada por meio da coleta de dados em condições que simulam a operação em um veículo de Fórmula SAE. Para isso, a placa desenvolvida foi instalada em uma moto, a qual percorreu um trajeto previamente definido. A saída CAN da placa foi conectada a um Arduino com o módulo MCP2515. Essa configuração permitiu que as informações transmitidas fossem salvas em um arquivo de log, possibilitando uma análise mais aprofundada.

Para fins de análise, dados como os quatérnios obtidos pelo filtro de Madgwick e os dados brutos do GPS foram adicionados à estrutura de dados transmitidos durante essa rodada de ensaios.

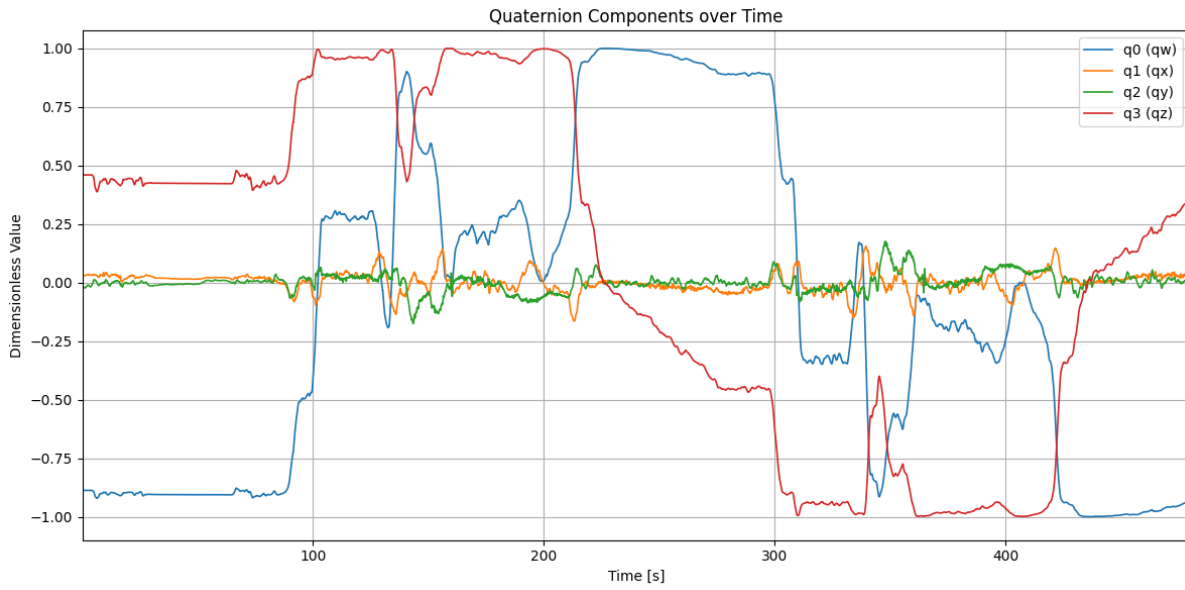
Com exceção dos dados destinados à avaliação do desvio de Allan, todos os demais dados foram coletados durante duas voltas em um percurso predefinido, com o objetivo de avaliar a repetibilidade dos resultados.

Os dados brutos e processados foram analisados e visualizados utilizando as bibliotecas `matplotlib` para os gráficos e `folium` para visualização dos mapas, além da biblioteca `allantools` para a análise do desvio de Allan. A rota de referência utilizada para comparação dos desvios de trajetória foi definida com o auxílio da ferramenta online GPSVisualizer.com.

5.2 Avaliação dos Dados de Orientação do Veículo

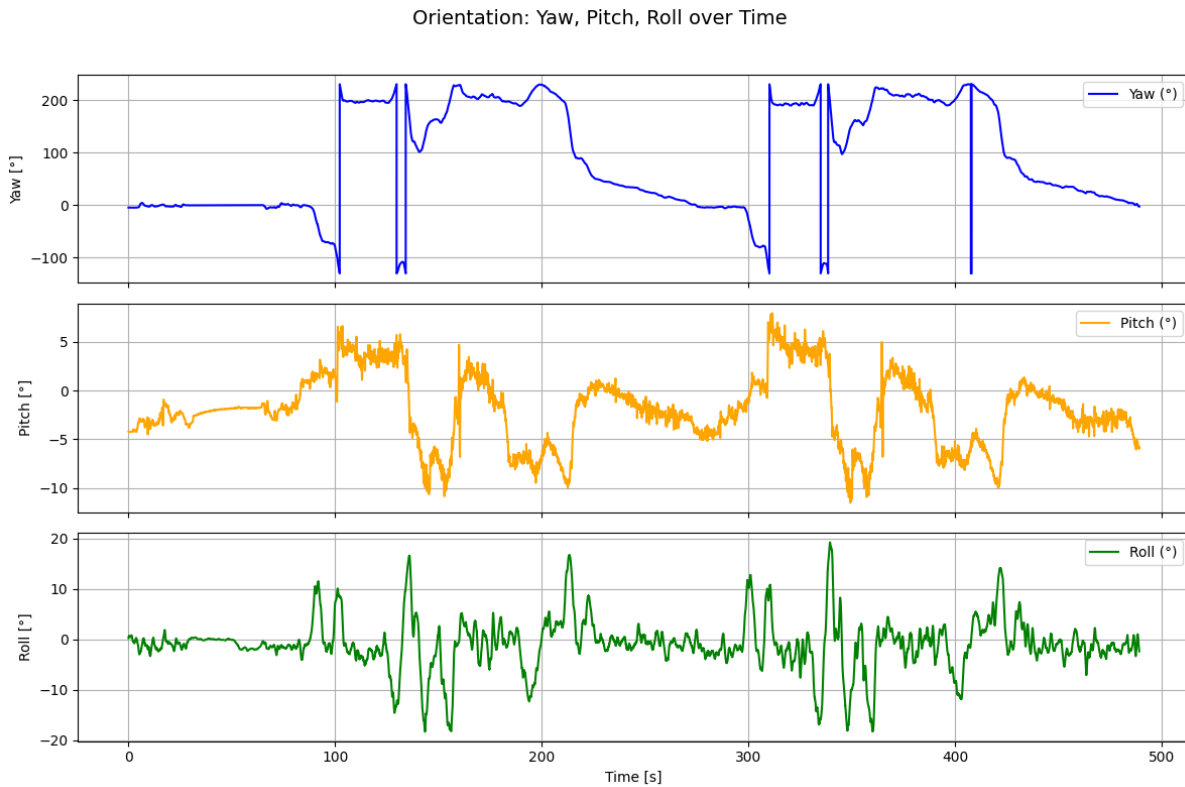
Como dito anteriormente, as estimativas de orientação obtidas através do Filtro de Madgwick baseia-se na rotação trimencional representada por um quaternium. A evolução dos quatérnios ao longo do trajeto é apresentada na Figura 22. A partir desses quatérnios o algoritmo obtém os ângulos de Pitch (arremesso), Roll (rolagem) e Yaw (guinada), mostrados na Figura 23.

Figura 22 – Componentes do Quatérnion ao longo do tempo



Fonte: Elaborado pelo autor.

Figura 23 – Orientação: Ângulos de Yaw, Pitch e Roll ao longo do tempo



Fonte: Elaborado pelo autor.

Ao observarmos o gráfico de orientação, levando em conta a amplitude dos sinais,

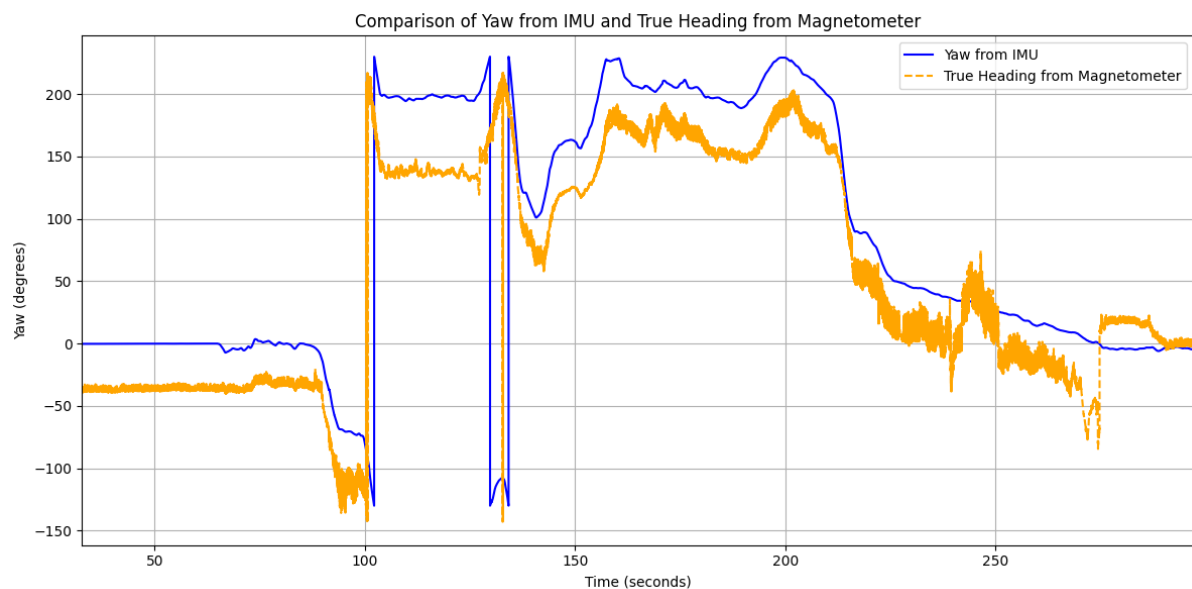
vemos que o *yaw* exibe uma movimentação mais acentuada, reflexo das curvas realizadas no percurso. Adicionalmente, o gráfico de atitude revela um perfil semelhante para as duas voltas do trajeto, o que evidencia a consistência das medições de orientação.

5.2.1 Comparação do Yaw Estimado com o Heading Magnético

Outro estudo realizado para certificar a coesão dos dados gerados pelo filtro de Madgwick foi compararmos o sinal estimado para um grau de orientação com um sinal de referência que não dependa do sistema interno da IMU. Para tal, o ângulo de *yaw* estimado pelo sistema foi comparado com o *true heading*, um dado de referência gerado a partir das leituras dos dados do magnetômetro, o qual indica a direção do sistema, atuando de forma semelhante a uma bússola.

No gráfico apresentado na Figura 24, os dados em azul representam o *yaw* estimado, que indica a direção do veículo em graus ao redor do eixo Z. Já os dados em laranja correspondem ao fluxo magnético medido pelo magnetômetro, referenciado ao norte magnético. Embora o magnetômetro não seja utilizado como saída final do sistema, ele serve como referência auxiliar para avaliarmos se a tendência dos dados estimados é compatível com a direção real do movimento.

Figura 24 – Comparação entre yaw calculado e o true heading do magnetômetro



Fonte: Elaborado pelo autor.

Em um primeiro momento, nota-se que o Yaw estimado inicia em 0° , enquanto o *True Heading* apresenta um claro descolamento. Este fato faz sentido ao lembrarmos que, durante a partida, a posição inicial é tomada como referencial para as medidas seguintes, enquanto o *True Heading* reflete, de fato, a orientação do sistema em relação ao Norte

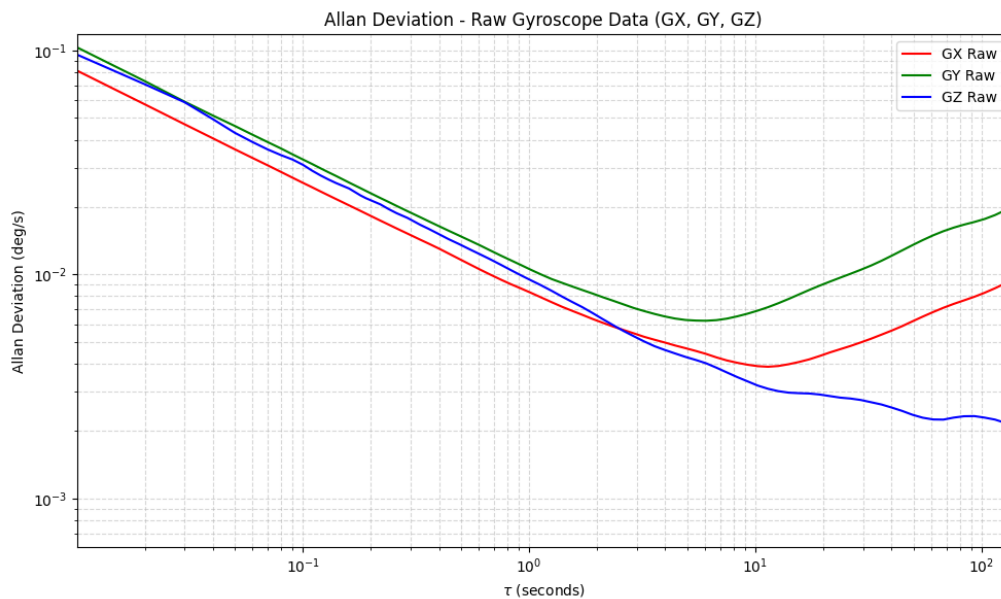
Magnético. Os dados ruidosos observados no sinal do magnetômetro são devidos à sua sensibilidade a interferências magnéticas, como, por exemplo, as provenientes do motor do veículo.

Apesar dessa diferença de referencial e dos dados ruidosos, a análise do gráfico demonstra uma correlação visual clara entre as duas medidas ao longo do percurso, confirmando que o filtro apresenta estimativas para a direção do veículo de forma coerente.

5.2.2 Caracterização do Giroscópio: Desvio (Bias) e Ruído

Para quantificar o ruído e o desvio angular (bias) do giroscópio, foi realizado um teste em bancada com a IMU mantida em repouso por um por cerca de 20 minutos. Os dados de taxa angular coletados foram então submetidos a uma análise de Desvio de Allan, cujos resultados são apresentados na Figura 25. Este gráfico permite identificar e quantificar as principais fontes de erro do giroscópio, como o ruído aleatório de ângulo e a instabilidade de bias, informações essenciais para a modelagem precisa do Filtro de Kalman.

Figura 25 – Desvio de Allan para os dados brutos do giroscópio



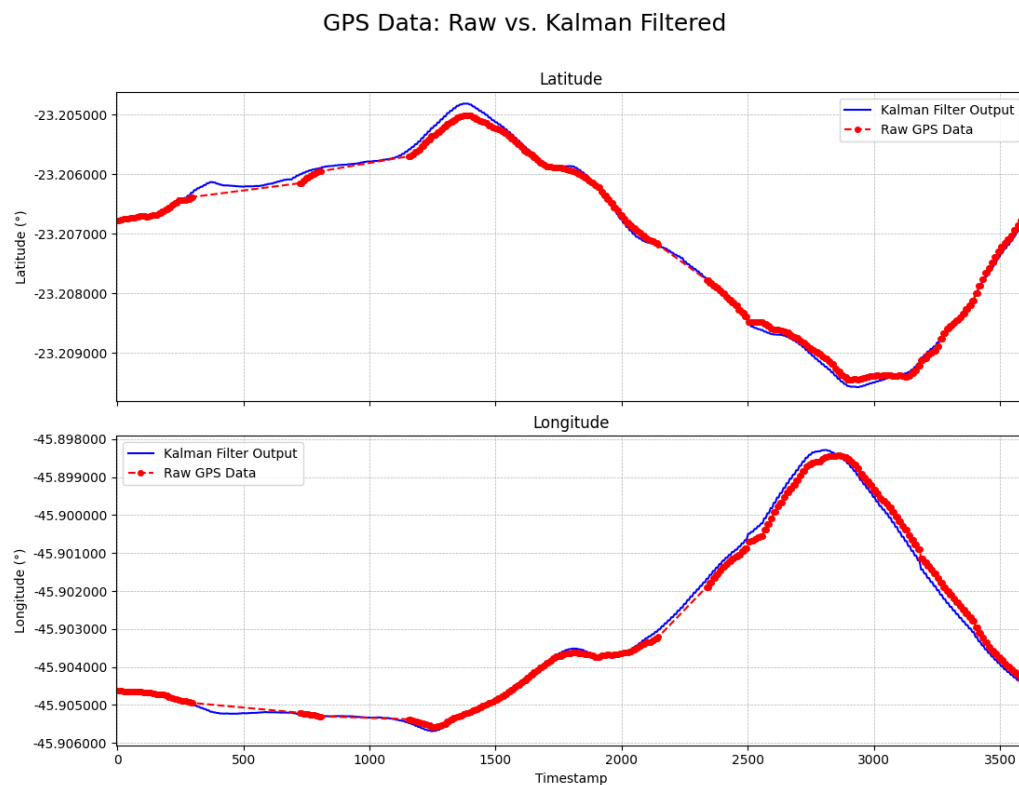
Fonte: Elaborado pelo autor.

5.3 Avaliação da Estimativa de Posição do Veículo

Para este estudo, buscou-se verificar a atuação do Filtro de Kalman ao mitigar as limitações intrínsecas do sensor de localização, especialmente a intermitência e o ruído do sinal. A Figura 26 apresenta uma comparação direta entre os dados brutos de Latitude e Longitude fornecidos pelo módulo GPS NEO-6M e a trajetória de posição estimada pelo algoritmo de Kalman.

Analisando, podemos notar que os dados de saída da fusão dos sensores são capazes de suavizar as medidas. Uma coisa que fica evidente ao observarmos o gráfico comparativo são os intervalos sem dados de GPS decorrentes da perda de sinal, um problema comum em ambientes urbanos e no nosso caso provavelmente devido à densidade de prédios no entorno do trajeto. O algoritmo de Kalman se mostra efetivo ao estimar dados de posição preenchendo essas lacunas de sinal.

Figura 26 – Comparação dos dados de GPS brutos com a saída do Filtro de Kalman para Latitude e Longitude



Fonte: Elaborado pelo autor.

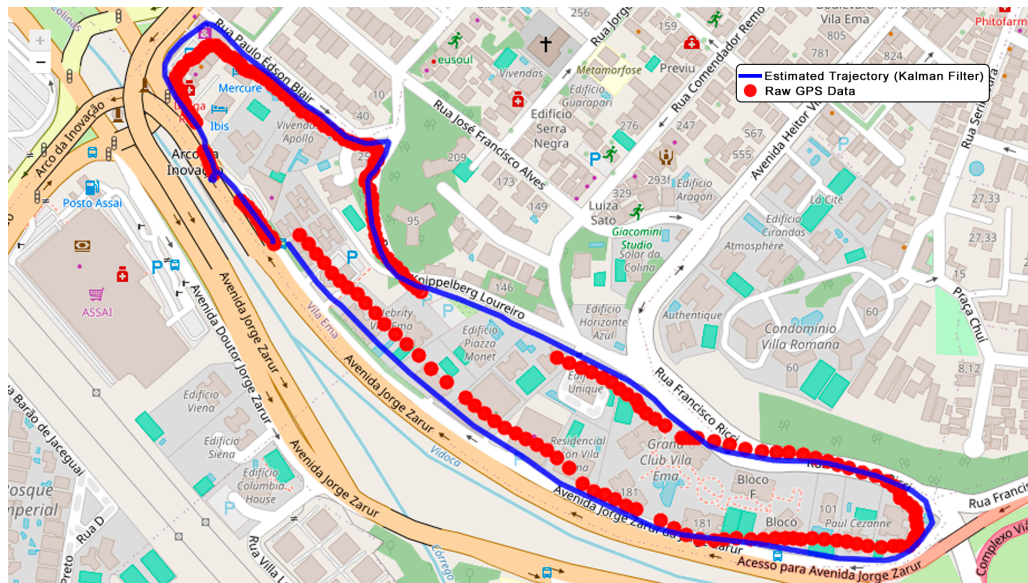
5.3.1 Visualização Espacial da Trajetória Filtrada

Para uma visualização espacial dos resultados, os mesmos dados de posição foram plotados em um mapa geográfico (27), mostrando a trajetória completa do percurso.

O mapa ilustra a diferença entre a trajetória bruta do GPS e a trajetória filtrada pelo Kalman. Os dados do módulo mostram-se dispersos e, em boa parte do tempo, distantes da rota esperada, refletindo o ruído e a baixa precisão do próprio GPS. A trajetória estimada pelo Filtro de Kalman forma uma linha contínua, visivelmente mais estável e representativa do movimento real do veículo, evidenciando o papel da IMU ao interpolar e corrigir a posição GPS. Podemos visualizar os momentos em que temos a atuação unicamente do sistema inercial durante a falta de dados do NEO-6M, o que

evidencia a eficácia do Kalman para a predição de trajetória.

Figura 27 – Mapa com trajetória estimada e dados brutos do GPS

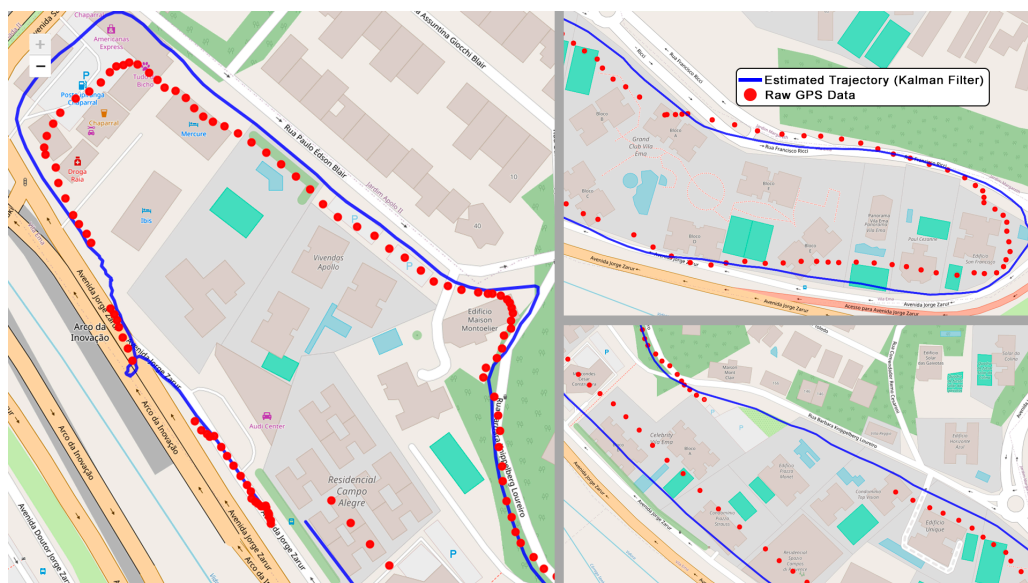


Fonte: Elaborado pelo autor.

5.3.2 Análise Detalhada da Dispersão e Limitações do GPS

Apesar dos ganhos trazidos pela fusão dos sensores, é fundamental analisar as limitações da precisão final, que são diretamente influenciadas pela qualidade dos dados brutos de entrada do GPS. A Figura 28 oferece uma visualização detalhada da dispersão do sinal GPS e da trajetória filtrada em trechos específicos do percurso.

Figura 28 – Dispersão dos dados brutos de GPS em relação à trajetória estimada pelo Filtro de Kalman



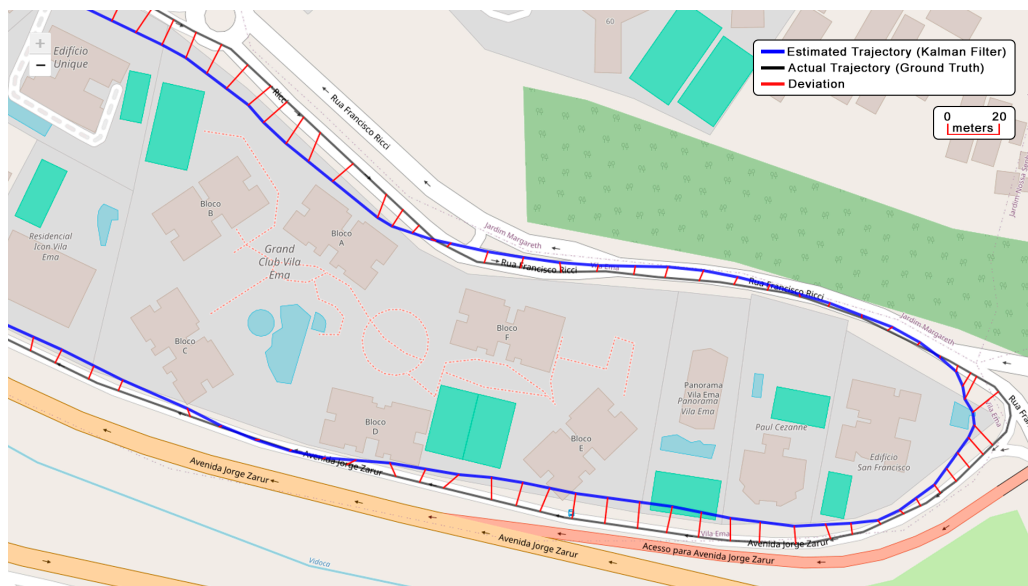
Fonte: Elaborado pelo autor.

Como pode ser observado no mapa, módulos GPS, particularmente de baixo custo, como o NEO-6M, apresentam uma dispersão considerável. Embora seja descrito que, para o módulo usado, o erro médio seja de 2,5 m, no mapa podemos ver pontos de localização registrados, sobretudo em áreas adjacentes à rodovia real, indicando um erro que extrapola essa medida (AG, 2016). Com isso, embora o Filtro de Kalman seja eficaz na suavização e estimativa de posição durante falhas, com base no movimento do veículo, sua acurácia final para um sistema de posicionamento é altamente influenciada pela precisão do GPS, uma vez que os dados são integrados a partir dessas medições.

5.3.3 Desvio da Trajetória em Relação a uma Referência

Para melhor quantificar esse erro na trajetória, foi realizada uma comparação desta com uma rota de referência esperada. A referência foi definida manualmente utilizando a ferramenta GPSVisualizer.com. A Figura 29 ilustra o desvio entre a trajetória calculada pela IMU (pós-Filtro de Kalman) e a rota esperada.

Figura 29 – Comparação entre a trajetória estimada e a trajetória real



Fonte: Elaborado pelo autor.

Na figura temos uma legenda que indica a escala das linhas vermelhas, as quais representam a magnitude do desvio em metros. A análise do gráfico revela que, em certas etapas do percurso, o erro entre a trajetória estimada e a referência pode atingir e até mesmo exceder 10 metros.

6 CONCLUSÃO

O desenvolvimento de um sistema embarcado de aquisição de dados para veículos de Fórmula Estudantil exige a integração de conhecimentos em diversas áreas — como hardware, software e análise de dados — para a criação de um sistema que atenda aos objetivos estabelecidos. Neste trabalho, entre os objetivos definidos, destacou-se o desenvolvimento e a validação de uma Unidade de Medição Inercial (IMU) voltada para veículos de Fórmula Estudantil, integrando dados de GPS e utilizando o Filtro de Kalman para aprimorar a estimação de posição, além de implementar a comunicação via barramento CAN. Ao longo do trabalho, foram abordados os desafios relacionados à fusão de sensores e às limitações impostas por um hardware de baixo custo.

Os resultados obtidos durante os ensaios demonstram que a placa desenvolvida atendeu aos propósitos iniciais, apresentando um desempenho satisfatório. A análise dos ângulos de atitude e dos quatérnios, estimados pelo Filtro de Madgwick, comprovou a capacidade do sistema em representar a dinâmica do veículo de forma consistente e precisa ao longo do percurso. A comparação do *Yaw* com uma referência de *True Heading* reforçou a coerência das estimativas de direção.

Contudo, a análise detalhada da dispersão do sinal GPS e do desvio da trajetória em relação a uma referência revelou as limitações da acurácia posicional final. Com base nos resultados, observou-se que a precisão do sistema é limitada pela qualidade intrínseca do módulo GPS de baixo custo, uma vez que seus dados são usados como base para integração. Este ponto ressalta a importância de um GPS de maior precisão para alcançar uma melhor acurácia em futuras iterações do projeto.

Ainda que os resultados para os dados de posição não sejam tão precisos quanto os dados de atitude fornecidos pelo sistema, as análises de posição em séries temporais e em mapas evidenciaram a capacidade do filtro de Kalman em suavizar os dados do GPS e, sobretudo, em compensar as perdas de sinal. A continuidade da trajetória, mesmo em trechos com ausência de dados GPS, demonstra a robustez do algoritmo de fusão de sensores e a relevância da IMU na interpolação e predição do deslocamento.

A comunicação via barramento CAN, utilizando a estrutura de dados definida, foi implementada com sucesso, permitindo a transmissão e aquisição dos dados processados de forma eficiente.

Em suma, o projeto da IMU atingiu os objetivos propostos, entregando um sistema funcional que integra sensores, aplica filtragem avançada em tempo real e se comunica via CAN. O sistema demonstrou ser capaz de fornecer dados de dinâmica veicular suficientemente precisos, podendo ser facilmente integrados a modelos de controle veicular.

A capacidade de manter uma estimativa contínua de posição e atitude, mesmo diante de falhas de sinal GPS, representa uma contribuição significativa para a autonomia e confiabilidade de sistemas de navegação e telemetria em veículos de Fórmula Estudantil.

6.1 Propostas para Trabalhos Futuros

Ao longo do projeto, algumas oportunidades de implementação para trabalhos futuros foram identificadas. A primeira delas diz respeito à precisão dos dados de posição. É importante destacar que o desenvolvimento da placa de circuito impresso foi realizado de forma a permitir a futura adição de módulos de maior precisão da *u-blox*. Essa decisão foi tomada prevendo uma possível necessidade de substituição do módulo GPS caso os resultados não fossem satisfatórios. Assim, o sistema está preparado para receber melhorias ao longo do tempo, sem a necessidade de reprojeter o circuito.

Dessa forma, uma possível implementação futura seria a adição de um módulo NEO-M8P, que conta com tecnologia RTK embarcada, permitindo uma alta precisão posicional. Ao integrar esse módulo ao sistema de navegação inercial, teríamos dados prévios de localização mais acurados para estimação, mesmo em condições de perda de sinal.

Outra proposta seria explorar a implementação de filtros de Kalman adaptativos ou estendidos (EKF/UKF), os quais podem otimizar ainda mais o desempenho do sistema.

Por fim, uma última proposta seria a de realizar os mesmos testes feitos até o momento, porém em um ambiente não urbano ou com poucas estruturas ao redor, simulando uma pista de competição. Dessa forma, seria possível comparar o desempenho em relação aos dados já adquiridos, evidenciando a influência do ambiente no desempenho do sistema.

REFERÊNCIAS

- AG u-blox. **NEO-6 Datasheet**. 2016. Acesso em 2 de junho de 2025. Disponível em: https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf.
- ALABA, S. Y. **GPS-IMU Sensor Fusion for Reliable Autonomous Vehicle Position Estimation**. 2024. ArXiv preprint arXiv:2405.08119. Disponível em: <https://arxiv.org/abs/2405.08119>.
- ALLAN, D. W. Statistics of atomic frequency standards. **Proceedings of the IEEE**, IEEE, v. 54, n. 2, p. 221–230, 1966.
- ANTUNES, A. M. P. **Sideslip Estimation of Formula Student Prototype (Extended Abstract)**. Lisboa, Portugal: [S.l.: s.n.], 2017. Resumo Estendido. Disponível em: https://fenix.tecnico.ulisboa.pt/downloadFile/1970719973966698/ExtendedAbstract_AndreAntunes.pdf.
- INOUE, R. S. **Controle robusto descentralizado de movimentos coordenados de robôs heterogêneos**. 2012. Tese (Doutorado em Engenharia Elétrica) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012. Orientador: Marco Henrique Terra. Disponível em: <https://doi.org/10.11606/T.18.2012.tde-14022012-095557>.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 11898-1:2024 Controller area network (CAN)**. 2024. Acesso em 2 de junho de 2025. Disponível em: <https://www.iso.org/standard/86384.html>.
- INVENSENSE, T. **MPU-9250 Product Specification Revision 1.1**. 2016. Acesso em 2 de junho de 2025. Disponível em: <https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>.
- LAI, T. **IMU Madgwick Filter Explanation**. 2024. Disponível em: <https://medium.com/@k66115704/imu-madgwick-filter-explanation-556fbe7f02e3>.
- LAVEISSIERE, J. **Formula Student: The Crucial Role of the IMU/GNSS**. 2019. Disponível em: <https://spectrum.ieee.org/formula-student-the-crucial-role-of-the-imu-gnss>.
- MADGWICK, S. O. H. **An efficient orientation filter for inertial and inertial/magnetic sensor arrays**. [S.l.], 2010. Disponível em: https://courses.cs.washington.edu/courses/cse466/14au/labs/14/madgwick_internal_report.pdf.
- MUN, J. H.; SHIN, H.-S.; CHOI, H.-L. Use of magnetic field for mitigating gyroscope errors for indoor pedestrian positioning. **Sensors**, v. 18, n. 8, p. 2592, 2018. Disponível em: <https://www.mdpi.com/1424-8220/18/8/2592>.
- SRINIVASAN, S. **Kalman Filter: An Algorithm for Making Sense from the Insights of Various Sensors Fused Together**. 2018. Disponível em: <https://medium.com/data-science/kalman-filter-an-algorithm-for-making-sense-from-the-insights-of-various-sensors-fused-together-ddf67597f35e>.

STM32-BASE. **Original Schematic STM32F103C8T6 Blue Pill**. 2016. Acesso em 29 de junho de 2025. Disponível em: https://stm32-base.org/assets/pdf/boards/original-schematic-STM32F103C8T6-Blue_Pill.pdf.

STMICROELECTRONICS. **STM32F103C8 Datasheet**. 2020. Acesso em 2 de junho de 2025. Disponível em: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>.

UK, S. **GPS position prediction and correction using Kalman Filter**. 2021. Repositório no GitHub. Disponível em: <https://github.com/spoorthiuk/GPS-position-prediction-and-correction-using-Kalman-Filter>.

WANG, X.; LIANG, M. Gps positioning method based on kalman filtering. *In: Proceedings of the 2018 IEEE International Conference on Robot Intelligence and Systems (ICRIS)*. [S.l.: s.n.], 2018. p. 130–133. Disponível em: <https://doi.org/10.1109/ICRIS.2018.00028>.