

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DAELN - DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ENGENHARIA ELETRÔNICA

DIOGO NOGUEIRA KNOP

**SISTEMA DE DETECÇÃO AUTOMÁTICO DE AVARIAS NA
PISTA DE ROLAMENTO**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2020

DIOGO NOGUEIRA KNOP

**SISTEMA DE DETECÇÃO AUTOMÁTICO DE AVARIAS NA
PISTA DE ROLAMENTO**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do grau de bacharel em Engenharia Eletrônica.

Orientador: Prof. Me. Luiz Fernando Copetti

CURITIBA
2020

DIOGO NOGUEIRA KNOP

SISTEMA DE DETECÇÃO AUTOMÁTICO DE AVARIAS NA PISTA DE ROLAMENTO

Este Trabalho de Conclusão de Curso de Graduação foi apresentado como requisito parcial para obtenção do título de Engenheiro Eletrônico, do curso de Engenharia Eletrônica do Departamento Acadêmico de Eletrônica (DAELN) outorgado pela Universidade Tecnológica Federal do Paraná (UTFPR). O aluno foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Curitiba, 30 de setembro de 2020.

Prof. Dr. Robinson Vida Noronha
Coordenador de Curso
Engenharia Eletrônica

Prof^a. Dr^a. Carmen Caroline Rasesa
Responsável pelos Trabalhos de Conclusão de Curso
de Engenharia Eletrônica do DAELN

BANCA EXAMINADORA

Prof. Me. Luiz Fernando Copetti
Universidade Tecnológica Federal do Paraná
Orientador

Prof. Dr. Guilherme de Santi Peron
Universidade Tecnológica Federal do Paraná

Prof. Dr. Hugo Vieira Neto
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia Eletrônica.

A toda minha família, por estarem sempre ao meu lado nos momentos mais difíceis e me incentivarem a continuar.

AGRADECIMENTOS

Gostaria de agradecer à minha esposa Aryadne pela paciência e ajuda para finalizar este projeto. Também agradeço ao Prof. Luiz Fernando Coppeti, orientador desse trabalho, e ao Prof. Hugo Vieira por sempre estar disponível quando eu necessitava. Agradeço também a toda comunidade da UTFPR, aos meus amigos e colegas.

Deixem que o futuro diga a verdade e avalie cada um de acordo com o seu trabalho e realizações. O presente pertence a eles, mas o futuro pelo qual eu sempre trabalhei pertence a mim. (TESLA, Nikola, 1927).

RESUMO

KNOP, Diogo Nogueira. SISTEMA DE DETECÇÃO AUTOMÁTICO DE AVARIAS NA PISTA DE ROLAMENTO. 2020. 46 f. Trabalho de Conclusão de Curso – Curso de Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2020.

A má conservação de estradas e rodovias é uma das principais causas de acidentes de trânsito. O Brasil, por ser um país de proporções continentais, enfrenta uma tarefa difícil para realizar as manutenções destas vias. O objetivo deste trabalho é o de desenvolver um sistema capaz de detectar quando um veículo passar sobre um buraco em uma pista de rolamento, informando a um servidor a localização e data de quando ocorreu a detecção. Foram utilizados sensores inerciais e uma plataforma de desenvolvimento que contém um SoC Xilinx Zynq-7000. Na FPGA, foram implementados filtros passa-baixas e a comunicação com os sensores inerciais. No microprocessador, foi implementando o Filtro Complementar, o algoritmo para detecção de buracos, e também feita a comunicação serial com sensor GPS e um servidor. A comunicação entre FPGA e microprocessador é feita através do protocolo AMBA AXI. Foi realizado um teste em campo, utilizando um veículo na cidade de Curitiba, o qual obteve resultado satisfatório para o sistema desenvolvido.

Palavras-chave: Fusão. Sensores. Filtros. Veículos.

ABSTRACT

KNOP, Diogo Nogueira. AUTOMATIC POTHOLE DETECTION SYSTEM ON ROAD SURFACE. 2020. 46 f. Trabalho de Conclusão de Curso – Curso de Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2020.

Poor conservation of roads and highways is a major cause of traffic accidents. Brazil, being a country of continental proportions, faces a difficult task to carry out the maintenance of these roads. The objective of this project is to develop a system capable of detecting when a vehicle passes over holes in a running track, informing a server its location and date of when the detection occurred. Inertial sensors and a development kit containing a Xilinx Zynq-7000 SoC were used. In the FPGA, low-pass filters and communication with inertial sensors were implemented. In the microprocessor, the Complementary Filter, the pothole detection algorithm, and the serial communication with a GPS sensor and a server were implemented. The communication between FPGA and microprocessor is done through the AMBA AXI protocol. A field test was performed using a vehicle in the city of Curitiba, which obtained a satisfactory result for the developed system.

Keywords: Sensors. Fusion. Filters. Vehicle.

LISTA DE FIGURAS

Figura 1 – Disposição das camadas de pavimento tipo flexível	16
Figura 2 – Ilustração da navegação estimada utilizando sensores inerciais	18
Figura 3 – Ângulos relativos ao sensor MPU-6050. Medição de aceleração nos eixos X, Y e Z. Medição da velocidade angular nos eixos ϕ , θ e ψ	19
Figura 4 – Arquitetura simplificada Xilinx Virtex 5	19
Figura 5 – Disposição dos satélites de GPS	20
Figura 6 – Filtro complementar. $G(s)$ é um filtro passa-baixas, então $1 - G(s)$ é o complemento, um filtro passa-altas	22
Figura 7 – Ilustração para cálculo de posição utilizando acelerômetro (força específica) e giroscópio (velocidade angular)	22
Figura 8 – Sensor InvenSense MPU-6050	24
Figura 9 – Exemplo de conexão I ² C com MPU-6050	25
Figura 10 – Exemplo de comunicação I ² C	25
Figura 11 – Módulo GPS NEO-6M	26
Figura 12 – Plataforma de desenvolvimento Zedboard	26
Figura 13 – Diagrama de blocos do SoC Xilinx Zynq-7000	27
Figura 14 – Arquitetura do sistema para detecção de buracos	28
Figura 15 – Diagrama de blocos do componente I ² C Mestre	29
Figura 16 – Máquinas de estados para mestre do barramento I ² C	30
Figura 17 – Criação de um novo IP de interface AXI4 tipo Lite.	33
Figura 18 – Projeto de blocos no software Vivado.	33
Figura 19 – Exemplo do arquivo de especificação da plataforma de hardware no software Xilinx SDK.	34
Figura 20 – Exemplo de leitura de um registrador da interface AXI4 em linguagem C.	34
Figura 21 – Máquina de estados utilizada no microprocessador	35
Figura 22 – Algoritmo utilizado para detecção de buracos	37
Figura 23 – Teste em bancada. Sinal do acelerômetro	38
Figura 24 – Teste em bancada. Sinal do giroscópio	39
Figura 25 – Sensores	40
Figura 26 – Ambiente de testes	41
Figura 27 – Resultados do teste em campo	42

LISTA DE TABELAS

Tabela 1 – Registradores da MPU-6050 utilizados	31
---	----

LISTA DE ABREVIATURAS E SIGLAS

AXI	Advanced Extensible Interface
CF	Complementary Filter
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
GPS	Global Positioning System
I ² C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
MEMS	Micro-Electro-Mechanical Systems
SDK	Software Development Kit
SoC	System-on-a-Chip
VHDL	Very High Speed Integrated Circuits HDL

LISTA DE SÍMBOLOS

θ	ângulo de arfada (aeronáutica)
ψ	ângulo de guinada (aeronáutica)
ϕ	ângulo de rolamento (aeronáutica)
Ta	taxa de amostragem do sinal

SUMÁRIO

1 – INTRODUÇÃO	14
1.1 JUSTIFICATIVA	14
1.2 OBJETIVOS	15
1.2.1 OBJETIVO GERAL	15
1.2.2 OBJETIVOS ESPECÍFICOS	15
2 – REVISÃO DE LITERATURA	16
2.1 DEFINIÇÃO DE BURACOS NA PISTA DE ROLAMENTO	16
2.2 DETECÇÃO DE BURACOS	16
2.3 UNIDADE DE MEDIÇÃO INERCIAL	18
2.4 ARRANJO DE PORTAS PROGRAMÁVEL EM CAMPO (FPGA)	19
2.5 SISTEMA DE POSICIONAMENTO GLOBAL	20
2.6 FILTROS	20
2.6.1 FILTRO PASSA-BAIXA	21
2.6.2 FILTRO COMPLEMENTAR	21
2.6.3 FILTRO DE KALMAN	23
3 – METODOLOGIA	24
3.1 DELIMITAÇÃO DA PESQUISA	24
3.2 RECURSOS UTILIZADOS	24
3.2.1 UNIDADE DE MEDIÇÃO INERCIAL	24
3.2.2 SISTEMA DE POSICIONAMENTO GLOBAL	25
3.2.3 SISTEMA EMBARCADO	26
3.2.4 SOFTWARE E APLICATIVOS	27
4 – PROJETO E DESENVOLVIMENTO	29
4.1 UNIDADE DE MEDIÇÃO INERCIAL	29
4.1.1 AQUISIÇÃO DE DADOS	29
4.1.2 FILTROS	31
4.2 COMUNICAÇÃO PL-PS	32
4.3 PROCESSAMENTO DOS SINAIS	35
4.3.1 CONVERSÃO DOS VALORES	35
4.3.2 FUSÃO DE SENSORES	36
4.3.3 DETECÇÃO DE BURACOS	36
5 – ANÁLISE E DISCUSSÃO DOS RESULTADOS	38
5.1 TESTES DE LABORATÓRIO	38

5.2	TESTE EM CAMPO	40
6	– CONCLUSÃO	44
	Referências	45

1 INTRODUÇÃO

Conforme a Confederação Nacional dos Transportes (2018), o Brasil possui apenas 12,4% da malha rodoviária asfaltada, e a maior parte (92,7%) é de pista simples. O levantamento, que avalia toda a malha federal e as principais rodovias estaduais, indica que dentro do total de rodovias pavimentadas, 61,8% são classificadas como regulares, ruins ou péssimas pois apresentam algum tipo de problema. A malha de rodovias asfaltadas no Brasil é de 211.468km, fazendo com que a identificação dos pontos que necessitam de reparo seja uma tarefa difícil de ser executada.

Uma vez identificada a necessidade de manutenção, é iniciada a análise de manutenção, que indicará o tipo de serviço que deverá ser executado. Estes serviços podem ser de reparo ou restauração. A restauração “é um conjunto de operações destinado a restabelecer o perfeito funcionamento de um bem deteriorado ou avariado, e restabelecer na íntegra, suas características técnicas e originais” (IPR, DNIT, 2017, p. 261). Reparo é “ação empreendida sobre um produto não-conforme, de modo que possa satisfazer às exigências de utilização previstas, embora não esteja necessariamente conforme as exigências especificadas originalmente como parte de uma operação de manutenção” (IPR, DNIT, 2017, p. 257).

Quanto maior o tempo para realizar o reparo do pavimento, maior será o custo, até atingir certo grau de deterioração em que as medidas de restauração se tornam ineficazes ou anti-econômicas DNIT (2006, p. 27), sendo necessária a reabilitação do pavimento. Sendo assim, os órgãos responsáveis pela manutenção das rodovias devem ser informados das avarias o mais rápido possível, e para isso contam com as manutenções preventivas e as informações dos usuários da rodovias através dos canais de ouvidoria. Nesse sentido, um sistema automático de detecção de buracos pode ser embarcado em veículos, e ao reconhecer a passagem por um buraco, enviar ao órgão responsável pelas manutenções as informações de onde está localizada esta avaria, diminuindo o tempo entre receber a informação e tomar a ação necessária.

Os sistemas automáticos para detecção de buracos podem ser do tipo passivo que, através de um conjunto de sensores, detecta a perturbação no sistema pela passagem do veículo por um buraco. O sistema ativo não necessita que o veículo passe por um buraco, pois utiliza recursos e sensores que fazem o reconhecimento de padrões. Os custos dos sensores para a detecção passiva são inferiores aos das câmeras para detecção ativa.

1.1 JUSTIFICATIVA

A manutenção do pavimento das rodovias tem impacto significativo na sociedade, pois além da segurança dos usuários, os custos para frete estão diretamente associados a condição do pavimento. Auxiliando os órgãos responsáveis pela manutenção a tomarem as decisões com menor tempo e de forma mais assertiva, haverá um retorno positivo a sociedade em forma de

segurança.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O objetivo deste trabalho é desenvolver um sistema capaz de mapear de forma passiva buracos na pista de rolamento.

1.2.2 OBJETIVOS ESPECÍFICOS

- utilizar arquitetura que contenha em um SoC um microcontrolador e um FPGA;
- utilizar sensor acelerômetro e giroscópio;
- utilizar sensor GPS para obter a posição e horário;
- projetar e implementar algoritmo de detecção de buracos;
- utilizar método de fusão de sensores;
- projetar e testar um sistema capaz de detectar buracos;
- utilizar comunicação serial com um computador;

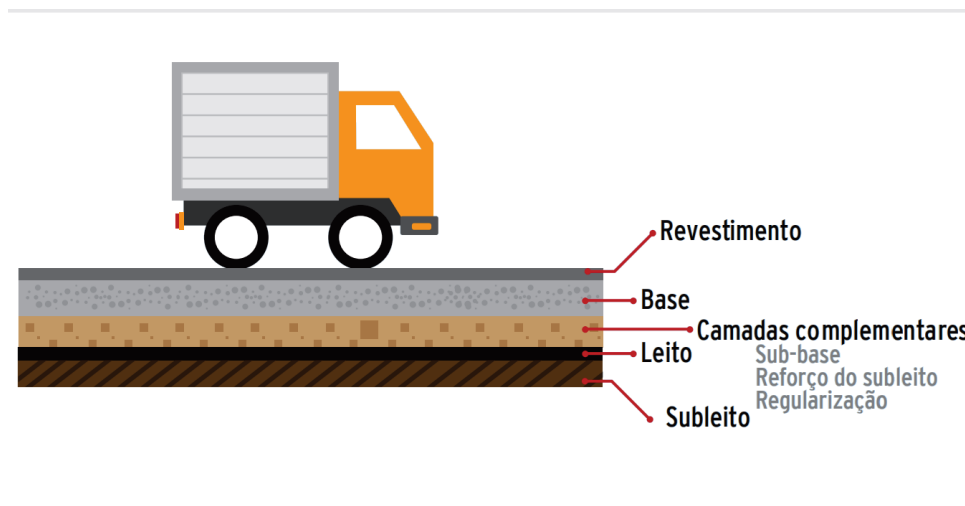
2 REVISÃO DE LITERATURA

Nesta seção é feita uma revisão do tema escolhido e das bases teóricas para realização deste trabalho. Serão apresentados dados sobre o tema e potencial ganho para a sociedade.

2.1 DEFINIÇÃO DE BURACOS NA PISTA DE ROLAMENTO

Pavimento de uma rodovia é a superestrutura constituída por um sistema de camadas de espessuras finitas, assentes sobre um semi-espaço considerado teoricamente como infinito - a infraestrutura ou terreno de fundação, a qual é designada de subleito. (IPR, DNIT, 2017, p. 226). A Figura 1 mostra a disposição das camadas de asfalto do tipo flexível.

Figura 1 – Disposição das camadas de pavimento tipo flexível



Fonte: CNT, SEST-SENAT (2018)

O modal rodoviário é o principal meio de transporte do Brasil, responsável por 61% da movimentação de mercadoria e de 95% da de passageiros (CNT, SEST-SENAT, 2018, p. 9). O buraco é um tipo de defeito no revestimento do pavimento, formando uma cavidade. Este defeito afeta estruturalmente o pavimento, pois permite acesso ao interior da estrutura de águas superficiais, além de afetar a segurança do tráfego e o custo do transporte (DNIT, 2006, p. 67).

São considerados buracos cavidades com extensão maiores ou iguais a 10cm, conforme (DNIT, 2006, p. 68).

2.2 DETECÇÃO DE BURACOS

Os métodos para detecção de buracos podem ser classificados em três categorias conforme Kim e Ryu (2014), de acordo com os métodos utilizados: Vibração, Reconstrução 3D

e Visão Computacional.

Os sistemas que utilizam vibração como base para detecção de buracos, geralmente fazem uso de sensores de aceleração. Estes possuem vantagem de possuírem a baixa necessidade de armazenamento de dados, bom custo benefício e fácil uso para processamento automático de dados em tempo real (KIM; RYU, 2014). Um problema encontrado neste tipo de sistema é a não identificação dos buracos que estão no meio das vias, em que as rodas do veículo não o acertam.

Nesse contexto, Burgart (2014) fez uso de sensores embarcados em dispositivos móveis. Através de um aplicativo executado em sistema *Android*, da Google, é feita a leitura do acelerômetro embarcado no dispositivo a cada 20 milissegundos. Um buraco é detectado quando a diferença na leitura da aceleração de um período ultrapassa um valor determinado, e quando detectado, é feita a leitura no sensor GPS embarcado no dispositivo móvel e é enviada uma mensagem a um servidor contendo: identificação do dispositivo, estampa de tempo, latitude, longitude e aceleração medida. O *Pothole Patrol*, (ERIKSSON et al., 2008), é uma iniciativa do Laboratório de Ciências da Computação e Inteligência Artificial (CSAIL) do Instituto de Tecnologia de Massachusetts (MIT), com objetivo de detectar e gerar relatório das superfícies das pistas de rolamento. Aquele trabalho desenvolveu um *hardware* específico para aquisição dos dados dos sensores de aceleração de GPS. Apesar de muito parecido com o trabalho de Burgart (2014), o tratamento dos dados não é feito localmente, e sim num servidor com um algoritmo de aprendizado de máquina. Foram instalados equipamentos em sete táxis que rodaram mais de 9000km para coletar dados de teste.

A reconstrução tridimensional também pode ser utilizada para detecção de buracos nos pavimentos, tendo como base tecnologias LiDAR (da sigla inglesa *Light Detection And Ranging*), e visão estéreo, que realiza a fusão de imagens bidimensionais para obter uma representação tridimensional do espaço. No trabalho desenvolvido por Kang e Choi (2017), a detecção de buracos utiliza LiDAR bidimensional e câmera. O ângulo e a profundidade das superfícies são obtidos utilizando LiDAR bidimensional, e a imagem obtida pela câmera melhora a precisão da detecção e fornece o contorno do buraco na superfície.

Wang (2004) fez um estudo de viabilidade para utilização de visão estéreo para levantamento das condições do asfalto. São necessárias duas câmeras digitais, que num primeiro momento são analisadas separadamente para detectar e classificar rachaduras. As imagens são então utilizadas para formar o modelo tridimensional da superfície do pavimento.

Utilizando a tecnologia de visão computacional, Jo e Ryu (2015) desenvolveram um sistema para detecção de buracos. O algoritmo consiste no pré-processamento da imagem, seleção da região de interesse e detecção do buraco conforme alguns parâmetros, como comprimento, área, variância e trajetória. Os experimentos foram realizados somente em dias ensolarados, precisando de mais testes para conclusão do trabalho.

2.3 UNIDADE DE MEDIÇÃO INERCIAL

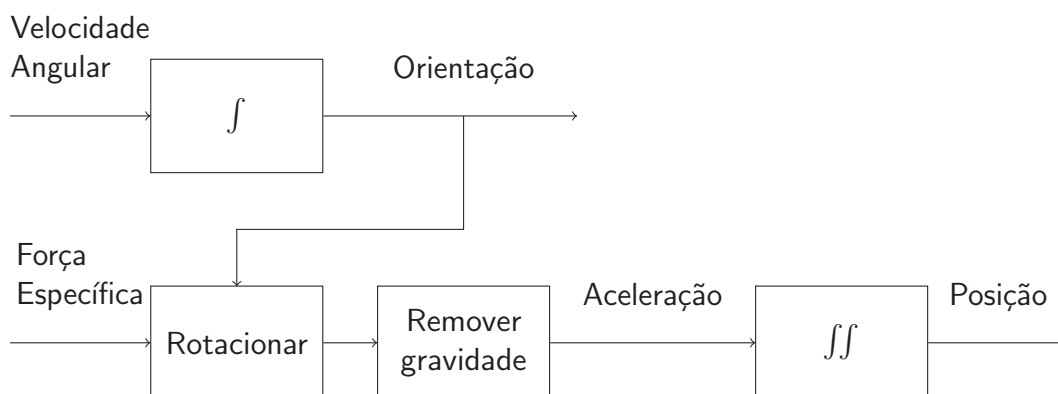
Uma unidade de medição inercial (IMU) é um dispositivo que agrupa sensores inerciais controlados por uma eletrônica embarcada, geralmente acelerômetros e giroscópios em arranjo ortogonal. As medidas fornecidas são utilizadas para navegação inercial, uma técnica para rastrear a posição e a orientação de um objeto. É amplamente utilizada na navegação de aeronaves, mísseis, submarinos, navios, robôs, veículos autônomos, entre outros (SANTANA, 2011).

Um Sistema de Navegação Inercial (SNI) de alto desempenho apresenta um alto custo (maior que U\$ 300.000,00), e possui uma regulamentação rígida, o que compromete sua utilização. Uma alternativa são os SNIs de baixo custo, que têm seu desenvolvimento impulsionado pelas IMUs que utilizam sistemas MicroEletroMecânicos (MEMS). As medidas das IMU-MEMS são corrompidas por diferentes fontes de erro, fazendo com que as medidas desses sensores inerciais sejam de baixa exatidão (DURÃO, 2009).

Um SNI utiliza as informações de sensores inerciais para obter a posição e a orientação angular de um objeto onde o sistema está instalado (DURÃO, 2009). Acelerômetros são sensores inerciais capazes de medir a força específica atuante sobre eles. Esta força é a resultante das ações da aceleração inercial e da aceleração da gravidade (DURÃO, 2009), e os giroscópios medem a velocidade angular do sensor (KOK; HOL; SCHÖN, 2017).

A Figura 2 representa a navegação estimada, processo este que utiliza as integrações das velocidades angulares e das acelerações para estimar a orientação e posição atuais (KOK; HOL; SCHÖN, 2017).

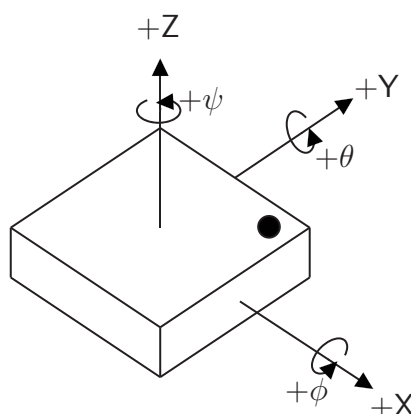
Figura 2 – Ilustração da navegação estimada utilizando sensores inerciais



Fonte: Kok, Hol e Schön (2017)

A Figura 3 apresenta a disposição dos eixos cartesianos com relação a leitura da aceleração e velocidade angular. O acelerômetro realiza a medição de aceleração nos eixos cartesianos X, Y e Z, enquanto o giroscópio realiza a medição da velocidade angular ao redor dos eixos X (Rolagem, ϕ), Y (Arfagem, θ) e Z (Guinada, ψ).

Figura 3 – Ângulos relativos ao sensor MPU-6050. Medição de aceleração nos eixos X, Y e Z. Medição da velocidade angular nos eixos ϕ , θ e ψ



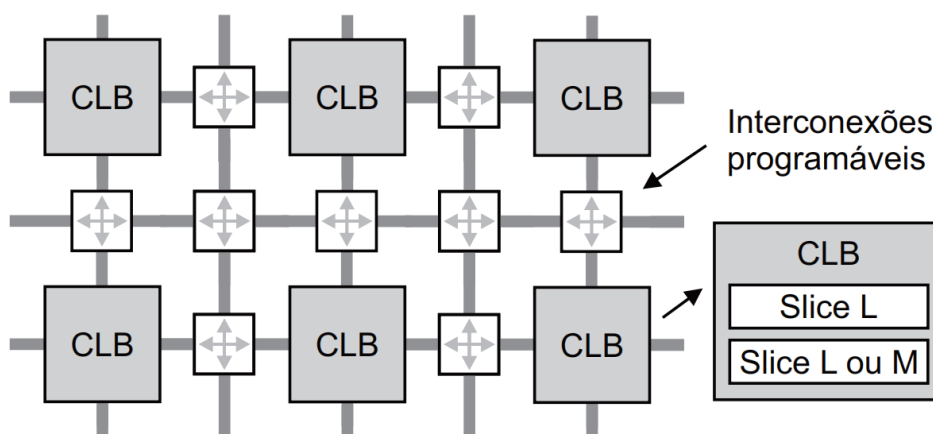
Fonte: Aatoria Própria

2.4 ARRANJO DE PORTAS PROGRAMÁVEL EM CAMPO (FPGA)

FPGA é uma plataforma que contém uma matriz de células lógicas genéricas e chaves programáveis, em que as células são programadas para desempenhar funções simples e as chaves interconectam as células conforme configuração (PÓVOA, 2010).

Uma arquitetura simplificada dos dispositivos Xilinx Virtex 5 é apresentada na Figura 4, em que os blocos lógicos são denominados *Control Logic Block* (CLB), e cada bloco é composto por duas *Slices* (fatias), uma do tipo L e outra que poder ser L ou M. A diferença entre as *Slices*, é que na *Slice M* a LUT (*look-up table*), pode ser utilizada pelo usuário como uma SRAM distribuída como memória de uso geral ou como registrador, quando a LUT não está sendo utilizada para implementar uma função. A LUT, ou tabela de busca, é um circuito combinacional com um flip-flop tipo D.

Figura 4 – Arquitetura simplificada Xilinx Virtex 5



Fonte: Pedroni (2010)

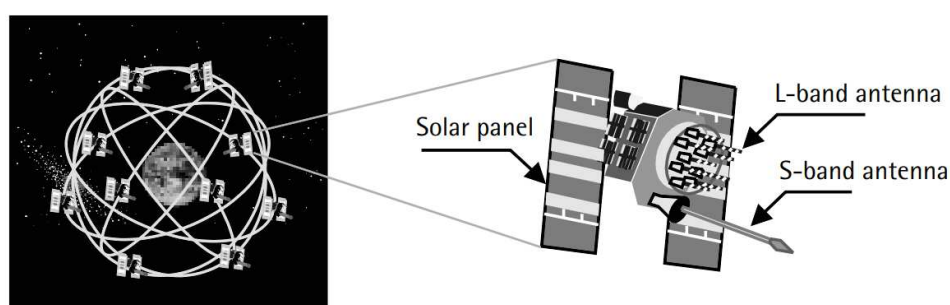
O projeto de uma FPGA é feito através de uma linguagem de descrição de hardware (HDL - *Hardware Description Language*). A linguagem VHDL (*Very-high-speed integrated circuit HDL*) foi escolhida para ser utilizada neste projeto.

A definição do comportamento dos blocos da FPGA e a configuração da conexão entre os blocos é feita utilizando arquivos HDL. Após definido o comportamento, é realizada a síntese e implementação, processos que em que o software de síntese transforma os arquivos HDL e gera o arquivo binário para configuração da FPGA. É feita a transformação para componentes a nível de porta lógica, o mapeamento de rotas e de portas lógicas, a conexão de sinais e análise de tempo estático (CHU, 2008).

2.5 SISTEMA DE POSICIONAMENTO GLOBAL

No começo dos anos 1970, o Departamento de Defesa (DoD) dos Estados Unidos da América criou um sistema de navegação por satélites, o Sistema de Posicionamento Global (GPS). O GPS fornece informações de posição e de tempo. O sistema é composto por 24 satélites de forma que existem quatro satélites em cada um dos seis planos orbitais, conforme a Figura 5 (EL-RABBANY, 2002).

Figura 5 – Disposição dos satélites de GPS



Fonte: El-Rabbany (2002)

Os satélites enviam ondas de rádio que incluem uma mensagem de navegação e dois códigos digitais. O dispositivo GPS receptor recebe as informações enviadas pelo satélite, armazena em sua memória interna e as processa utilizando um software embarcado. Parte deste processamento calcula a distância para os satélites através dos códigos digitais, e a posição dos satélites através das mensagens de navegação (EL-RABBANY, 2002).

2.6 FILTROS

Filtros digitais são muito importantes para o processamento digital de sinais, e utilizados para *separação* ou *restauração*. A separação de sinais é utilizada quando um sinal é contaminado com um outro sinal, ruído ou interferência. A restauração do sinal é usada quando o sinal é distorcido de alguma maneira (SMITH, 1999)

A performance de uma aplicação que utiliza IMUs é afetada pela precisão dos sensores utilizados, e conforme o preço do sensor é reduzido, sua precisão também é diminuída (ZHU; YU; XIAO, 2019). Os filtros são aplicados para filtrar os erros causados pela vibração interna do veículo, que pode, por exemplo, ser originada do motor, compressor ou de outras fontes como ar-condicionado ou alternador (SUWANDI; KITASUKA; ARITSUGI, 2019).

2.6.1 FILTRO PASSA-BAIXA

Filtros passa-baixas são filtros que atenuam a amplitude de sinais com frequência superior a uma determinada frequência de corte (f_0), permitindo a passagem de sinais com frequência inferior. Um filtro ideal não alteraria os sinais com frequência menor que a de corte, e atenuaria completamente os sinais com frequência maiores, sem região de transição.

A atenuação do filtro é dada pela sua ordem, número de polos e pela topologia escolhida. Quanto maior a atenuação, maior o atraso causado no sinal. Entre as topologias mais utilizadas estão Butterworth, Chebyshev I e II e Elípticos. A topologia Butterworth é uma das mais simples e foi projetada para ter uma resposta plana na banda de passagem. A Equação (1) mostra a função de transferência do filtro Butterworth de segunda ordem.

$$\frac{Y(s)}{X(s)} = \frac{1}{s^2 + \sqrt{2}s + 1} \quad (1)$$

Para implementação deste filtro em um sistema discreto, é necessária a transformação do plano S (contínuo) para o plano Z (discreto). Isto pode ser feito através da transformação bilinear, apresentada na Equação (2), onde T_a é a taxa de amostragem do sinal. A função de transferência do filtro no plano Z é da forma vista na Equação (3).

$$s = \frac{2}{T_a} \left(\frac{z - 1}{z + 1} \right) \quad (2)$$

$$H[z] = \frac{Y[z]}{X[z]} = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} \quad (3)$$

No plano Z, z^{-1} é um operador de atraso, então resolvendo a Equação (3) para $Y[z]$, e aplicando a transformada Z inversa, obtemos a equação a diferenças (4).

$$y[n] = b_0x[n] + b_1x[n - 1] + b_2x[n - 2] - a_1y[n - 1] - a_2y[n - 2] \quad (4)$$

2.6.2 FILTRO COMPLEMENTAR

O Filtro Complementar CF (Complementary Filter) é um técnica para estimar um sinal utilizando medições com diferentes fontes de ruído. Por exemplo, sejam x e y duas medições para um sinal z , e \hat{z} a estimativa do filtro para este sinal. Supondo que o ruído em x seja principalmente de baixa frequência, e o ruído em y principalmente de alta frequência, então

um filtro $G(s)$ do tipo passa-baixas pode ser aplicado para filtrar os ruídos de y , e o seu complementar $[1 - G(s)]$, um filtro passa-altas, irá filtrar o ruído de baixa frequência de x (HIGGINS, 1975). A Figura 6 mostra este exemplo.

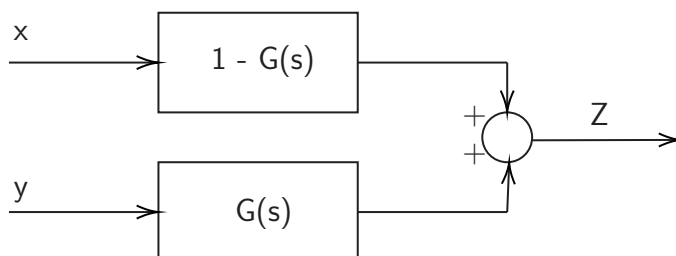


Figura 6 – Filtro complementar. $G(s)$ é um filtro passa-baixas, então $1 - G(s)$ é o complemento, um filtro passa-altas

Fonte: Higgins (1975)

Acelerômetro e giroscópio fornecem informações que podem ser utilizadas para estimar a orientação. O acelerômetro tem como característica possuir ruídos em suas medições, porém precisas em longos períodos de tempo, enquanto o giroscópio, devido a utilização de um integrador, fornece uma informação precisa em curtos períodos de tempo, mas tem um escorregamento (*drift*) ao longo do tempo (KOK; HOL; SCHÖN, 2017). Um filtro complementar, conforme Figura 7, pode ser aplicado nas medições do acelerômetro e do giroscópio para obter a orientação de um sensor, filtrando assim os ruídos característicos desses sensores. A Equação (5) mostra como o filtro é implementado, em que α é o coeficiente (peso) de complementação entre o ângulo calculado com o acelerômetro e o giroscópio.

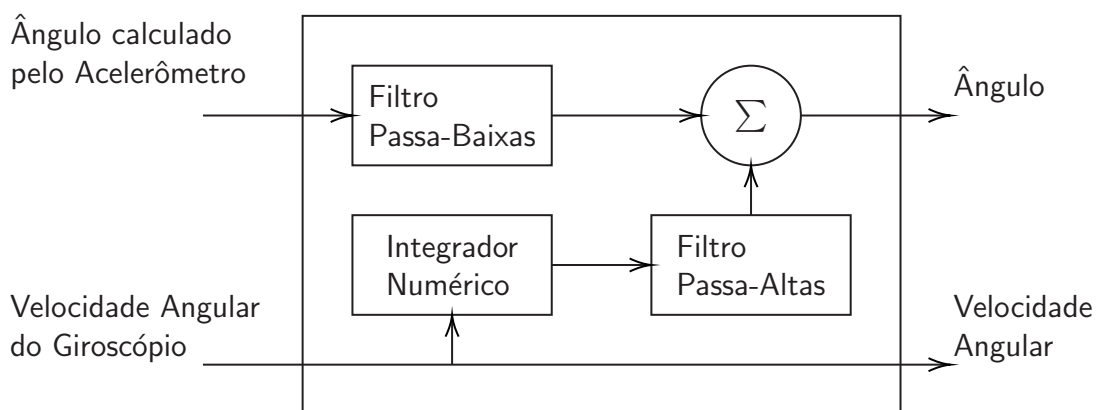


Figura 7 – Ilustração para cálculo de posição utilizando acelerômetro (força específica) e giroscópio (velocidade angular)

Fonte: Kok, Hol e Schön (2017)

$$\phi = (1 - \alpha)(\phi + \dot{\phi}_{gyr}\Delta t) + \alpha\phi_{acc} \tag{5}$$

2.6.3 FILTRO DE KALMAN

Publicado em 1960 por Rudolf E. Kalman, o Filtro de Kalman também é uma técnica para fusão de sensores que, através de sinais contaminados, utiliza as medições ao longo do tempo para estimar um valor próximo do sinal real (WELCH; BISHOP, 2006).

O Filtro de Kalman é uma forma de controle por realimentação e as equações são divididas em dois grupos: atualização temporal (predição) e atualização da medição (correção). As equações de atualização temporal tem como objetivo estimar o estado atual e a covariância do erro, enquanto as equações de atualização de medição fazem a realimentação (SANTANA, 2011).

O problema a ser resolvido é o de estimar o estado $x \in \mathfrak{R}^n$ a partir da medição $z \in \mathfrak{R}^m$, conforme a Equação (6) e a Equação (7) (WELCH; BISHOP, 2006).

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (6)$$

$$z_k = Hx_k + v_k \quad (7)$$

A matriz A é responsável por relacionar o estado anterior x_{k-1} ao estado atual x_k , a matriz B por sua vez relaciona o sinal de controle $u \in \mathfrak{R}^l$ do passo anterior ao estado x_k , enquanto a matriz H relaciona o estado x_k a medição z_k . As variáveis aleatórias w_k e v_k representam respectivamente os ruídos de processo e de medição, considerados como brancos, independentes entre si e com distribuições de probabilidade normal. A matriz Q representa a *covariância do ruído de processo* e a matriz R a *covariância do ruído de medição*.

$$p(w) \sim N(0, Q) \quad (8)$$

$$p(v) \sim N(0, R) \quad (9)$$

3 METODOLOGIA

Neste capítulo serão descritos os métodos utilizados para o desenvolvimento deste trabalho, além dos recursos necessários.

3.1 DELIMITAÇÃO DA PESQUISA

Este projeto pode ser executado utilizando diferentes abordagens e arquiteturas, e ser bem extensivo. A escolha da arquitetura foi como um desafio para o desenvolvedor, para consolidar os conhecimentos referentes a *FPGAs* e microcontroladores. A plataforma de desenvolvimento escolhida foi a Zedboard da AVNET, que possui o *SoC* (*System-on-a-chip*, em português *sistema-em-um-chip*) Xilinx Zynq[®]-7000.

3.2 RECURSOS UTILIZADOS

3.2.1 UNIDADE DE MEDIÇÃO INERCIAL

A IMU escolhida para este projeto é a MPU-6050 da InvenSense, que contém no mesmo chip um sensor acelerômetro e um sensor giroscópio do tipo MEMS, tendo cada sensor três eixos de medição, totalizando seis graus de liberdade (6DOF, em inglês *6 Degrees of Freedom*), e um sensor de temperatura. A decisão foi motivada pelo fato de o produto ser altamente disponível no mercado para desenvolvedores e o custo de aquisição ser baixo.

Este sensor possui a tecnologia Digital Motion Processing (DMP, em português Processador Digital de Movimento), que processa as medições dos seis eixos. Esses dados de acelerômetro e giroscópio são amostrados conforme a taxa definida pelo usuário, e a comunicação pode ser feita através do barramento I²C (Inter-Integrated Circuit, em português Inter-Circuito Integrado), desenvolvido pela Phillips. O barramento utiliza duas linhas bidirecionais, Dados Seriais (SDA) e Clock Serial (SCL). Um exemplo de configuração é mostrado na Figura 9. Utilizando 7 bits para endereçamento, é possível ter até 128 dispositivos escravos no barramento, com um dispositivo mestre.

Figura 8 – Sensor InvenSense MPU-6050

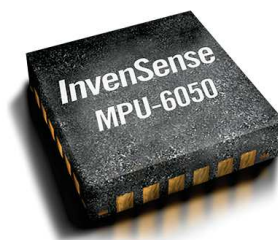
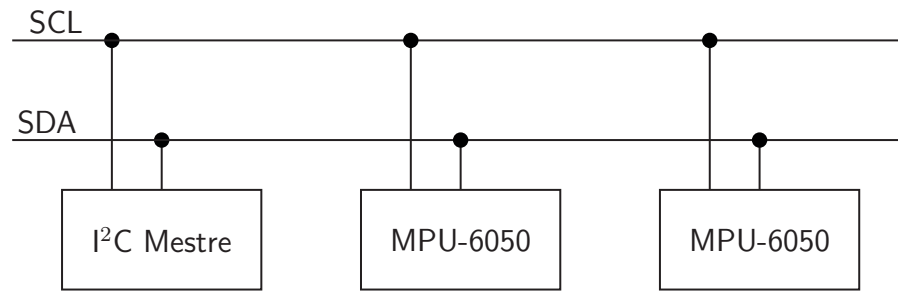
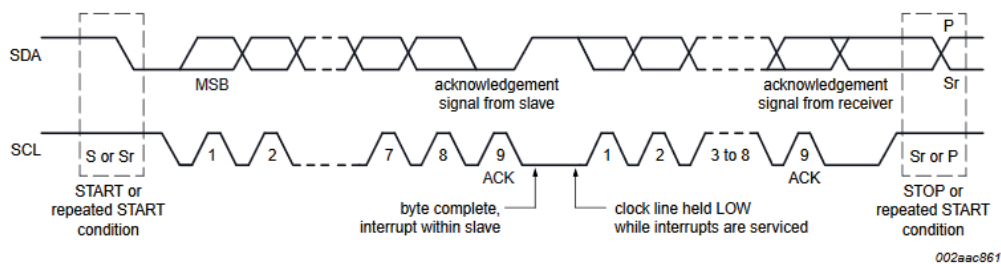


Figura 9 – Exemplo de conexão I²C com MPU-6050

Fonte: Kok, Hol e Schön (2017)

Para realizar a comunicação o mestre controla a linha de clock e inicia a comunicação com um bit de inicialização (Start), envia serialmente o endereço do dispositivo escravo, um bit para indicar se a operação é leitura ou escrita e aguarda pelo bit de confirmação do dispositivo escravo (ACK). É realizada a operação de leitura ou escrita, e então um comando para parar a comunicação é enviado pelo mestre (Stop). Esta operação é exemplificada na Figura 10.

Figura 10 – Exemplo de comunicação I²C

Fonte: NXP (2014)

Foram utilizadas duas IMUs, posicionadas acima das rodas dianteiras do veículo hospedeiro, pois geralmente as rodas dianteiras são mais propensas a acertarem um buraco na via. A opção de utilizar dois sensores, e não somente um, é para medir a aceleração em relação a gravidade em cada extremidade do veículo.

3.2.2 SISTEMA DE POSICIONAMENTO GLOBAL

Para obter a localização dos buracos, foi escolhido o módulo GPS Neo-6M, da u-blox, que possui interface serial. O módulo envia as informações no formato texto, que devem ser analisadas e manipuladas no receptor. Este módulo utiliza o conjunto de especificações de dados e elétricas para comunicação de dispositivos eletrônicos de navegação NMEA 0183.

Figura 11 – Módulo GPS NEO-6M

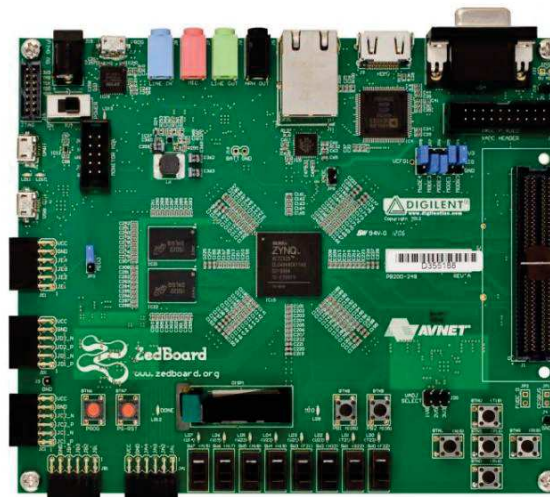


Fonte: <https://www.u-blox.com/en/product/neo-6-series>

3.2.3 SISTEMA EMBARCADO

Como já mencionado, foi escolhida a plataforma de desenvolvimento Zedboard, mostrada na Figura 12, pois integra em um SoC o sistema de processamento (PS, em inglês *Processing System*), um microprocessador dual-core ARM Cortex-A9, e o sistema de lógica programável (PL, do inglês *Programmable Logic*).

Figura 12 – Plataforma de desenvolvimento Zedboard

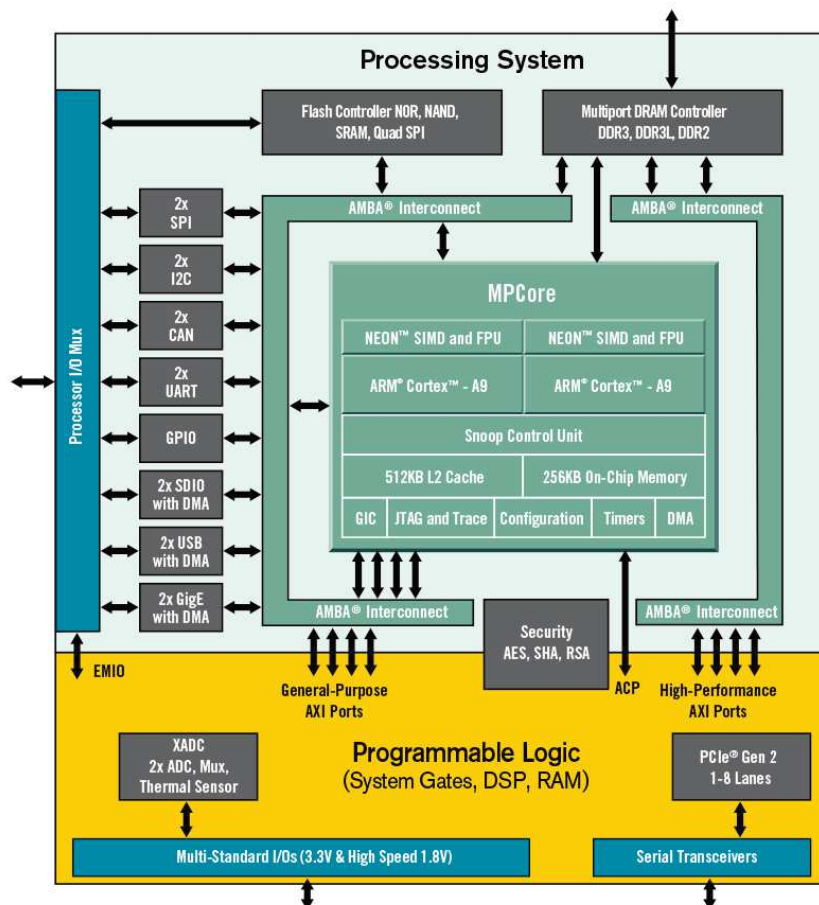


Fonte: <http://zedboard.org/product/zedboard>

O diagrama de blocos do SoC Zynq-7000 é mostrado na Figura 13. Dos recursos disponíveis neste SoC, foram utilizados as Entradas/Saídas da FPGA, o barramento AMBA e os periféricos de comunicação UART (Receptor/Transmissor Universal Assíncrono). A FPGA será responsável por fazer a interface e comunicação com as IMUs, além de aplicar um filtro de ruído nos sinais. Os dados filtrados são enviados através do barramento AMBA AXI4

(Advanced eXtensible Interface 4) para o microprocessador. O microprocessador aplicará o filtro complementar e executará o algoritmo para detecção de buracos, e também fará a comunicação com o módulo GPS e com um servidor responsável pelo armazenamento de informações. A arquitetura do sistema é mostrada na Figura 14.

Figura 13 – Diagrama de blocos do SoC Xilinx Zynq-7000



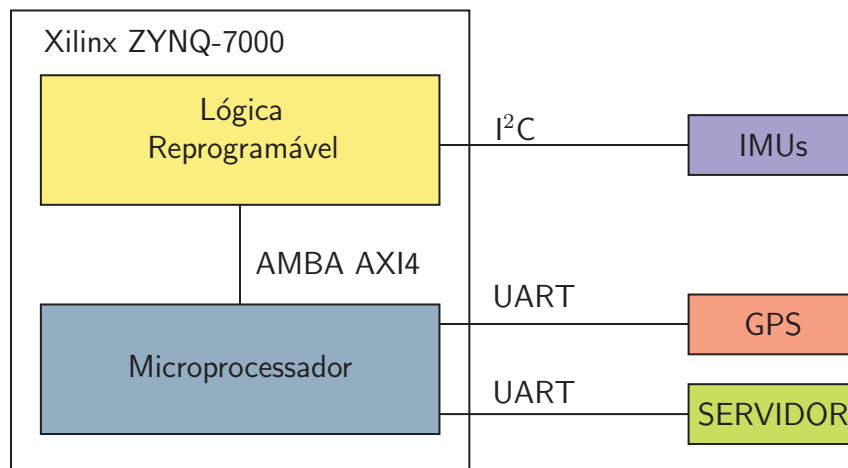
Fonte: <https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>

3.2.4 SOFTWARE E APLICATIVOS

A lista a seguir é referente aos recursos de *software* utilizados durante o projeto:

- Xilinx Vivado: Síntese e análise de projetos HDL.
- Xilinx Software Development Kit (XSDK): Síntese e análise de projetos para aplicações embarcadas.
- Matlab e Filter Design and Analysis Tool (FDATool): Análise de dados, projeto e simulação de filtros.
- draw.io: Construção de diagramas.

Figura 14 – Arquitetura do sistema para detecção de buracos



Fonte: Autoria Própria

4 PROJETO E DESENVOLVIMENTO

Neste capítulo será descrito o processo para o desenvolvimento do projeto. O processo seguiu as etapas abaixo:

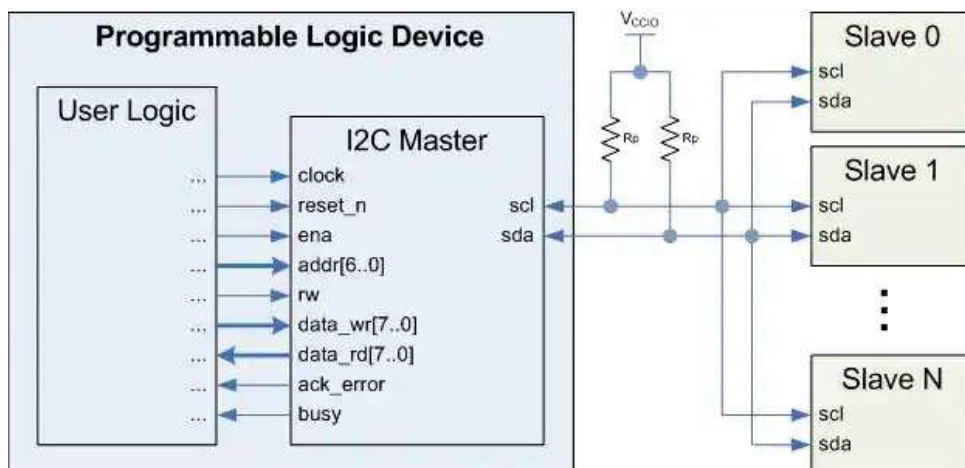
1. Desenvolver comunicação com IMU usando a *FPGA*
2. Implementar filtro digital para ruídos na *FPGA*
3. Realizar comunicação entre *FPGA* e microprocessador
4. Implementar método da fusão de sensores
5. Desenvolver algoritmo para detecção de buracos

4.1 UNIDADE DE MEDIÇÃO INERCIAL

4.1.1 AQUISIÇÃO DE DADOS

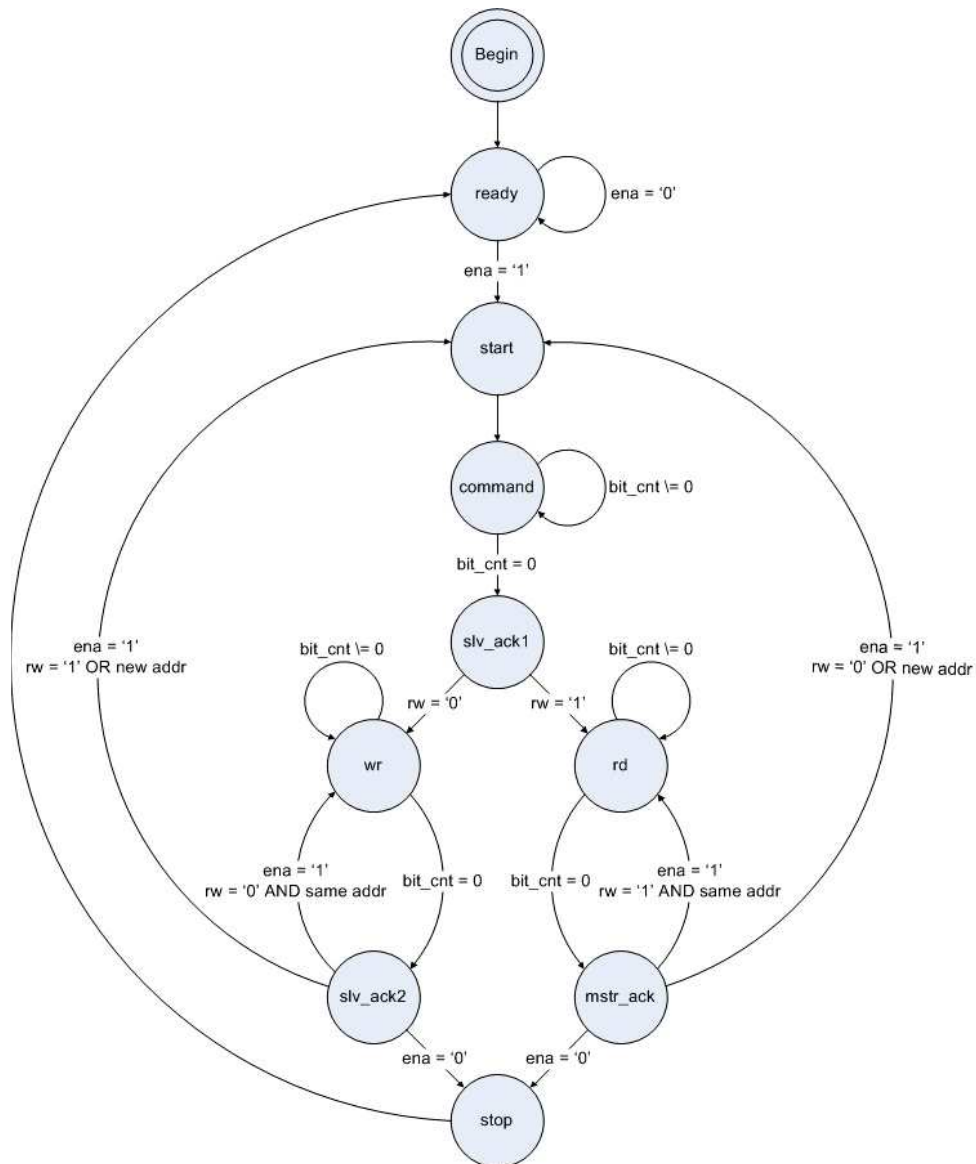
O primeiro passo para realizar a leitura dos dados da IMU, é implementar um componente I²C Mestre. Foi utilizado o componente mostrado na Figura 15, e os sinais são controlados conforme a máquina de estados apresentada na Figura 16. Foi escolhido o modo de operação padrão, que possui uma taxa de dados igual a 100 kbit/s. Considerando a operação descrita na Figura 10 e, contando com os comandos de inicialização e parada, são necessários 20 bits para realizar uma operação entre o dispositivo mestre e o dispositivo escravo, resultando em um total de até cinco mil operações por segundo.

Figura 15 – Diagrama de blocos do componente I²C Mestre



Fonte: Wiki Digi-Key¹.

¹<www.digikey.com/eewiki/pages/viewpage.action?pageId=10125324>

Figura 16 – Máquinas de estados para mestre do barramento I²C

Fonte: Wiki Digi-Key².

A comunicação com a IMU MPU-6050 também irá seguir uma máquina de estados. A sequência básica para realizar a leitura é:

1. Configurar gerenciamento de energia
2. Escrever o endereço do primeiro registrador a ser lido
3. Realizar leituras dos registradores desejados
4. Repetir passo 2

²Disponível em: <www.digikey.com/ee/wiki/pages/viewpage.action?pageId=10125324>

A leitura dos registradores do sensor é feita de maneira sequencial, e cada sinal de medição do sensor tem 16 bits. A cada leitura finalizada, o valor lido (8 bits) é armazenado em um registrador (16 bits) implementado na FPGA, conforme bit mais significativo. Os registradores dos sinais são apresentados na Tabela 1.

Tabela 1 – Registradores da MPU-6050 utilizados.

INFORMAÇÃO	REGISTRADORES	
	[MSB]	[LSB]
ACCEL_XOUT[15:0]	[0x3B]	[0x3C]
ACCEL_YOUT[15:0]	[0x3D]	[0x3E]
ACCEL_ZOUT[15:0]	[0x3F]	[0x40]
GYRO_XOUT[15:0]	[0x43]	[0x44]
GYRO_YOUT[15:0]	[0x45]	[0x46]
GYRO_ZOUT[15:0]	[0x47]	[0x48]

Fonte: (INVENSENSE, 2013)

Considerando que são necessárias uma operação de escrita e uma de leitura para cada registrador da Tabela 1, então a maior taxa de atualização dos registradores é de 417 Hz. Para facilitar a sincronia das frequências de operação do sistema, foi fixada a taxa de amostragem em 400 Hz.

4.1.2 FILTROS

O próximo passo é filtrar os sinais recebidos. Os filtros foram desenvolvidos utilizando a topologia Butterworth, conforme visto na Subseção 2.6.1. Os filtros foram projetados utilizando a ferramenta Matlab FDATool. A frequência de amostragem utilizada foi 400 Hz, conforme descrito na Subseção 4.1.1.

O acelerômetro é muito suscetível a ruídos, e para evitar variações da vibração do veículo ou da pista de rolamento, é aplicado um filtro passa-baixas ao sinal recebido. Foi escolhida uma frequência de corte de 5 Hz, e a equação a diferenças é apresentada na Equação (10):

$$y[n] = 0.00146x[n] + 0.00291x[n - 1] + 0.00146x[n - 2] - (-1.88904)y[n - 1] - 0.89489y[n - 2] \quad (10)$$

No giroscópio, as variações rápidas não podem ser desconsideradas, então uma frequência de corte de 50 Hz foi escolhida para calcular os coeficientes da Equação (11).

$$y[n] = 0.09762x[n] + 0.19526x[n - 1] + 0.09762x[n - 2] - (-0.94281)y[n - 1] - 0.33331y[n - 2] \quad (11)$$

4.2 COMUNICAÇÃO PL-PS

Como forma de utilizar os recursos disponíveis e também facilitar o cálculo com aritmética de ponto flutuante e de funções trigonométricas, os registradores implementados na FPGA (PL) são mapeados na memória do microprocessador (PS).

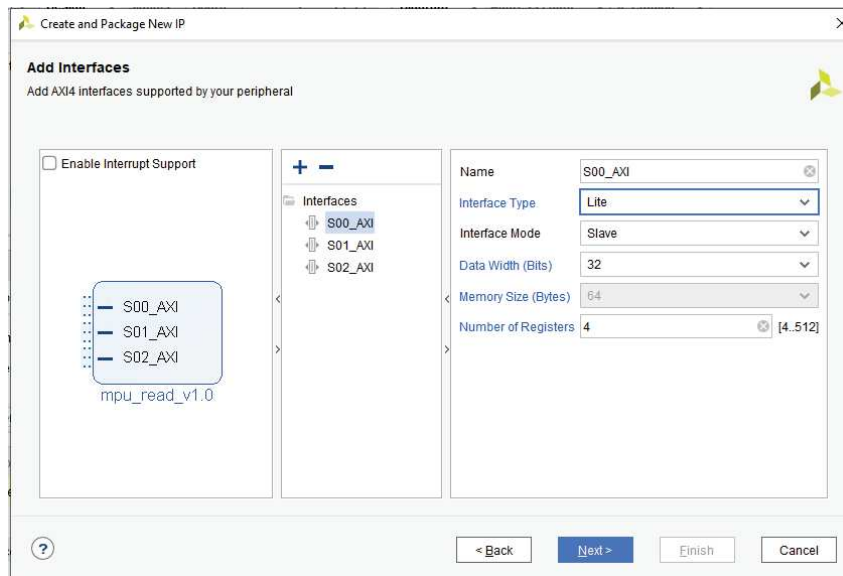
As comunicações disponíveis entre PL-PS são:

- Multiplas Entradas e Saídas Externas
- Portas AXI de Propósito Geral
- Portas AXI de Alta Performance
- Accelerator Coherency Port (ACP)
- Interrupções PS para PL
- Interrupções PL para PS

O protocolo AXI4 (Advanced eXtensible Interface, em português Interface Extensível Avançada), é parte da família de barramentos AMBA (Advanced Microcontroller Bus Architecture) para microcontroladores ARM. As especificações oferecem uma diversidade de definições, como dados variáveis, largura de barramento de endereçamento, entre outros, incluindo várias proteções de transações e permissões de acesso. O AXI4-Lite é um subconjunto do AXI4, que é utilizado em casos onde somente as transações de interconexão básicas são necessárias (XILINX, 2011).

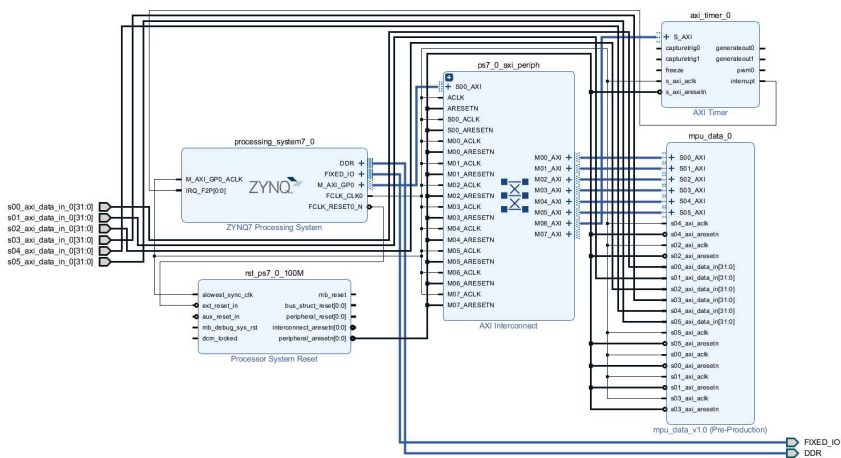
Foi utilizado um bloco IP (Intellectual Property, em português Propriedade Intelectual) disponível no software Vivado. Foi utilizado como base o tutorial apresentado por Johnson (2014). Foram configuradas 6 interfaces de 32 bits cada, exemplificadas na Figura 17. As medições da IMU possuem 16 bits cada, então são agrupados dois sinais em uma interface AXI4-Lite. São utilizadas três interfaces por IMU. O projeto final no software Vivado é apresentado na Figura 18.

Figura 17 – Criação de um novo IP de interface AXI4 tipo Lite.



Fonte: Autoria Própria

Figura 18 – Projeto de blocos no software Vivado.



Fonte: Autoria Própria

Após gerar o arquivo binário no software Vivado, é necessário desenvolver o programa para o microprocessador utilizando o software Xilinx SDK. Os sinais das interfaces AXI4-Lite são mapeados em endereços de memória no microprocessador e podem ser verificados no arquivo de especificação da plataforma de hardware, conforme exemplo mostrado na Figura 19. Um exemplo para realizar a leitura dos registradores utilizando a linguagem de programação C é apresentado na Figura 20.

Figura 19 – Exemplo do arquivo de especificação da plataforma de hardware no software Xilinx SDK.

system_wrapper_hw_platform_1 Hardware Platform Specification

Design Information

Target FPGA Device: 7z020
 Part: xc7z020clg484-1
 Created With: Vivado 2018.3
 Created On: Tue Oct 13 23:14:32 2020

Address Map for processor ps7_cortexa9_0[0-1]

Cell	Base Addr	High Addr	Slave I/f	Mem/Reg
axi_timer_0	0x42800000	0x4280ffff	S_AXI	REGISTER
mpu_data_0	0x43c20000	0x43c2ffff	S02_AXI	REGISTER
mpu_data_0	0x43c10000	0x43c1ffff	S01_AXI	REGISTER
mpu_data_0	0x43c40000	0x43c4ffff	S04_AXI	REGISTER
mpu_data_0	0x43c50000	0x43c5ffff	S05_AXI	REGISTER
mpu_data_0	0x43c30000	0x43c3ffff	S03_AXI	REGISTER
mpu_data_0	0x43c00000	0x43c0ffff	S00_AXI	REGISTER

Fonte: Autoria Própria

Figura 20 – Exemplo de leitura de um registrador da interface AXI4 em linguagem C.

```
#define XPAR_MPU_READ_0_S00_AXI_BASEADDR 0x43C00000
#define XPAR_MPU_READ_0_S01_AXI_BASEADDR 0x43C10000
#define XPAR_MPU_READ_0_S02_AXI_BASEADDR 0x43C20000
#define XPAR_MPU_READ_0_S03_AXI_BASEADDR 0x43C30000
#define XPAR_MPU_READ_0_S04_AXI_BASEADDR 0x43C40000
#define XPAR_MPU_READ_0_AXI_BASEADDR 0x43C50000

Xuint32 *baseaddr_p0 = (Xuint32 *)XPAR_MPU_READ_0_S00_AXI_BASEADDR;
Xuint32 *baseaddr_p1 = (Xuint32 *)XPAR_MPU_READ_0_S01_AXI_BASEADDR;
Xuint32 *baseaddr_p2 = (Xuint32 *)XPAR_MPU_READ_0_S02_AXI_BASEADDR;
Xuint32 *baseaddr_p3 = (Xuint32 *)XPAR_MPU_READ_0_S03_AXI_BASEADDR;
Xuint32 *baseaddr_p4 = (Xuint32 *)XPAR_MPU_READ_0_S04_AXI_BASEADDR;
Xuint32 *baseaddr_p5 = (Xuint32 *)XPAR_MPU_READ_0_AXI_BASEADDR;

short accX_1 = ((*baseaddr_p0 + 1) >> 16);
```

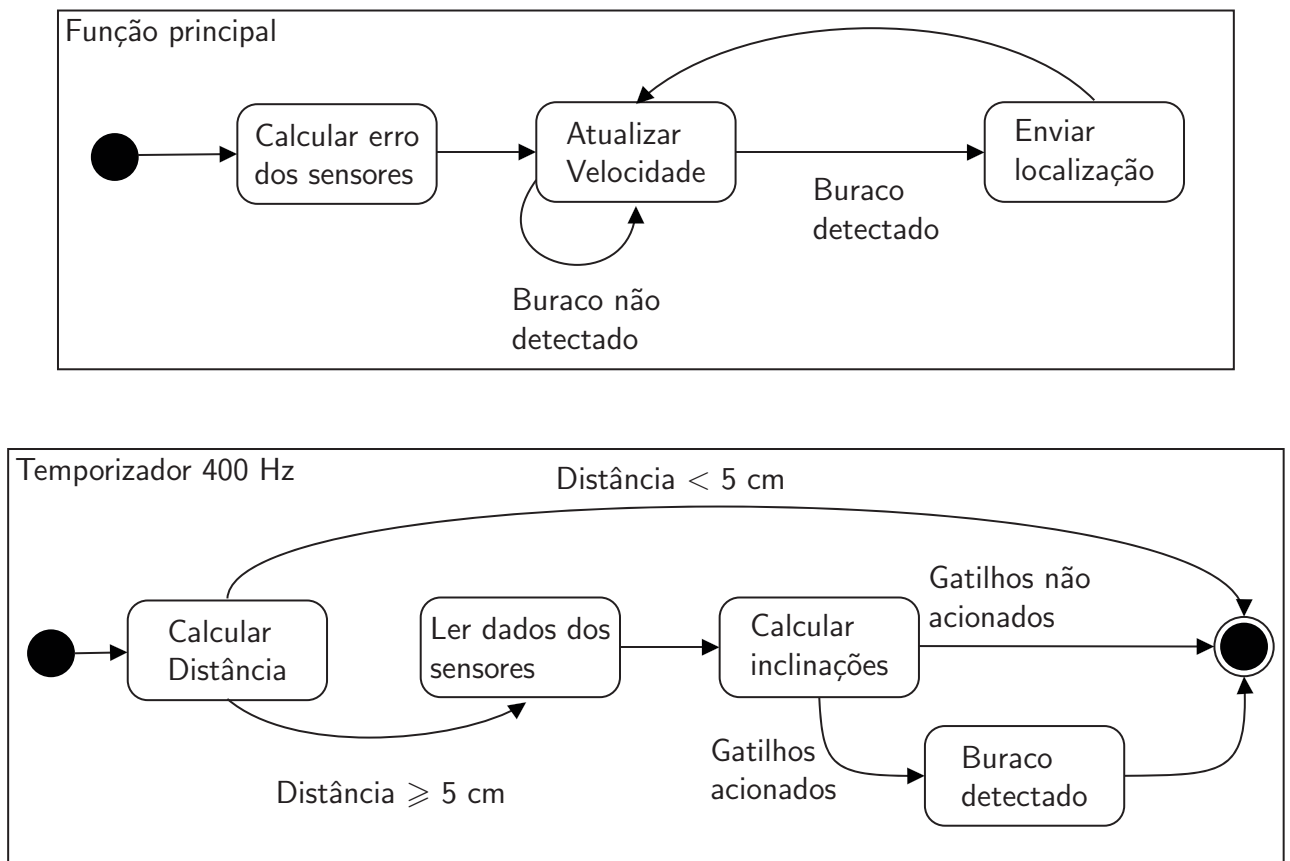
Fonte: Autoria Própria

4.3 PROCESSAMENTO DOS SINAIS

A Figura 21 mostra a máquina de estados do software do microprocessador. O processo principal faz a inicialização das variáveis e entra em um laço de repetição, onde é feita a leitura da velocidade do veículo através do GPS, e também verificado se o sinal de buraco detectado está ativo, para então enviar as coordenadas pela serial.

Um temporizador por hardware foi configurado para executar uma função numa frequência de 400 Hz. Nesta função, primeiro é calculada a distância percorrida pelo veículo, pois conforme visto no Capítulo 2, são considerados buracos cavidades com extensão maiores ou iguais a 10 cm. O algoritmo para calcular a inclinação e aceleração do veículo deve ser executado a cada 5 cm. Nesta mesma função, é calculada a diferença entre a leitura atual e a anterior.

Figura 21 – Máquina de estados utilizada no microprocessador



Fonte: Autoria Própria

4.3.1 CONVERSÃO DOS VALORES

Os valores da IMU precisam ser convertidos antes de serem utilizados. Na IMU estão sendo utilizadas as configurações de fábrica. Sendo assim, a escala do acelerômetro é de $\pm 2g$

e o giroscópio de $\pm 250^\circ/\text{s}$. A sensibilidade do acelerômetro é ± 16384 bits/g e do giroscópio ± 131 bits/($^\circ/\text{s}$) (INVENSENSE, 2013). Após a conversão, são calculadas as inclinações no eixo X (Rolagem, ϕ) e no eixo Y (Arfada, θ), conforme as equações (12) e (13).

$$\phi = \arctan\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \quad (12)$$

$$\theta = \arctan\left(\frac{x}{\sqrt{y^2 + z^2}}\right) \quad (13)$$

A velocidade angular medida no giroscópio é integrada em relação ao tempo para se obter a posição angular, conforme as equações (14) e (15).

$$\phi = \phi + \dot{\phi}\Delta t \quad (14)$$

$$\theta = \theta + \dot{\theta}\Delta t \quad (15)$$

4.3.2 FUSÃO DE SENSORES

Foi escolhido o Filtro Complementar devido a simplicidade para implementação quando comparado ao Filtro de Kalman. A implementação está descrita na Subseção 2.6.2. O valor escolhido para o coeficiente a foi 0,9, ou seja, 90% do valor utilizado para calcular os ângulos são do giroscópio, e 10% do acelerômetro, resultando na equação (16).

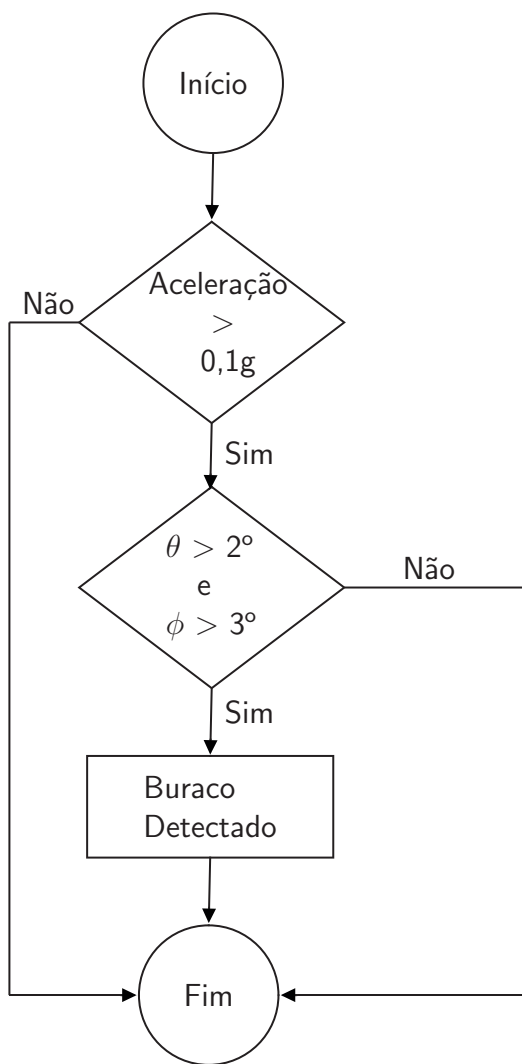
$$\phi = (0,9(\phi + \dot{\phi}_{gyr}\Delta t) + 0,1\phi_{acc}) \quad (16)$$

O filtro é aplicado para os ângulos Rolagem e Arfada.

4.3.3 DETECÇÃO DE BURACOS

Para realizar a detecção de buracos, foram utilizadas as medições de aceleração e inclinação do veículo. É calculada a diferença entre a leitura atual e a anterior, e caso a diferença seja superior aos valores estabelecidos, um sinal de buraco detectado é ativado. O fluxograma do algoritmo é apresentado na Figura 22. Os valores de decisão foram ajustados empiricamente para um melhor desempenho.

Figura 22 – Algoritmo utilizado para detecção de buracos



Fonte: Autoria Própria

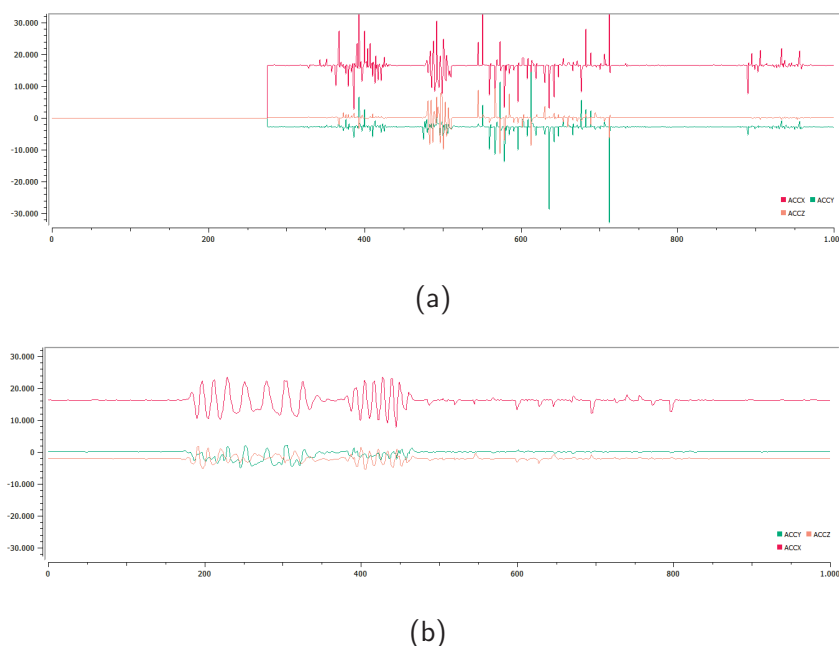
5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Neste capítulo são apresentados os resultados dos testes realizados após a etapa de desenvolvimento do sistema de detecção automático de buracos. Os primeiros testes foram realizados em laboratório, simulando aceleração e inclinação manualmente, para posteriormente serem realizados testes em campo.

5.1 TESTES DE LABORATÓRIO

Devido ao comportamento do acelerômetro, o primeiro passo realizado foi implementar e ajustar os coeficientes dos filtros. A Figura 23 mostra a diferença do sinal do acelerômetro, e a Figura 24 a do giroscópio. Como é possível observar, o filtro do acelerômetro rejeita as variações rápidas, ou possíveis amostras com ruídos. O sinal filtrado do giroscópio tem uma resposta mais rápida para voltar à condição de repouso, sendo assim os dados mais confiáveis.

Figura 23 – (a) Sinal proveniente do acelerômetro sem filtro, (b) Sinal proveniente do acelerômetro com filtro. Variações rápidas ou possíveis amostras com ruídos são rejeitadas

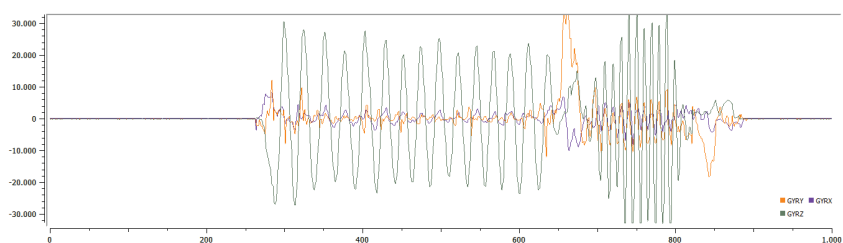


Fonte: Autoria Própria

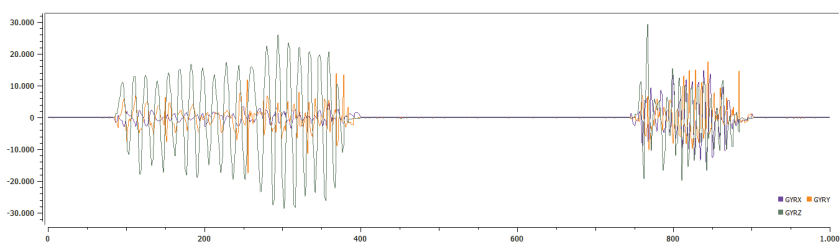
Antes de colocar o sistema em campo, primeiro foram ajustados os coeficientes para detecção dos buracos, como aceleração e o ângulo de inclinação. Os testes consistiam em, com uma das IMUs, acelerar em direção à gravidade e fazê-la sofrer uma inclinação, e verificar se a resposta do sistema era positiva para detecção de buracos.

Uma vez ajustados os coeficientes, foram realizados os testes em campo.

Figura 24 – (a) Sinal proveniente do giroscópio sem filtro, (b) Sinal proveniente do giroscópio com filtro. O sinal volta à condição de repouso em um intervalo de tempo menor.



(a)



(b)

Fonte: Autoria Própria

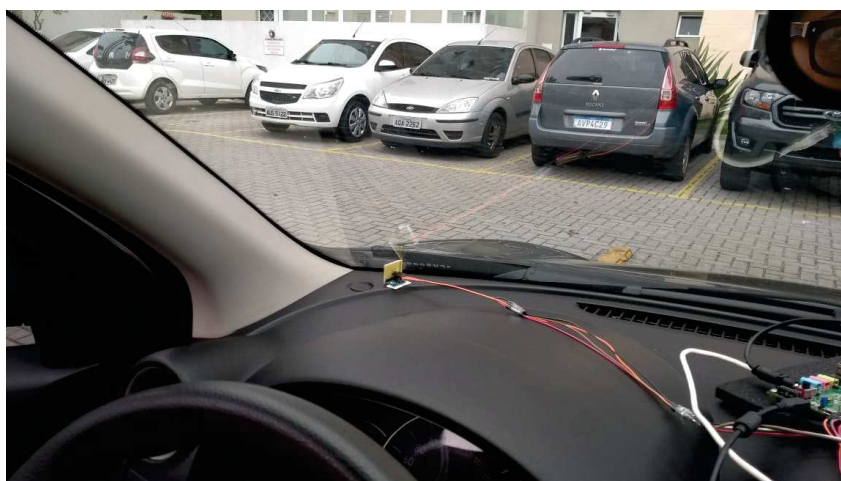
5.2 TESTE EM CAMPO

Os sensores e a placa de desenvolvimento foram instalados no interior de um veículo para ser realizado o teste em campo. A Figura 25 mostra a instalação dos sensores na extremidade interna do veículo, enquanto a Figura 26 mostra a placa de desenvolvimento e o computador utilizado para o teste. Foi percorrido um trecho previamente conhecido de 8,6km na cidade de Curitiba, Paraná.

Figura 25 – As figuras (a) Sensor UMI lado direito, (b) Sensor UMI lado esquerdo



(a)



(b)

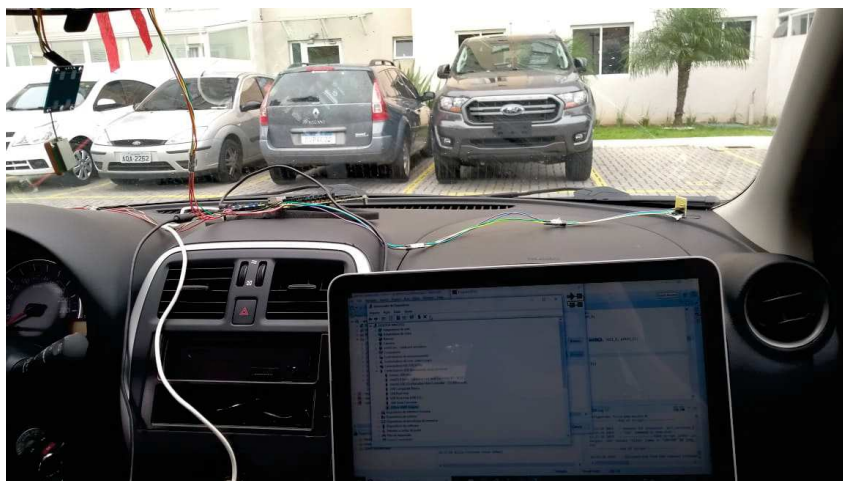
Fonte: Autoria Própria

Durante o teste, para ajustar os valores dos coeficientes para detecção de buraco, também foram coletadas as leituras da aceleração no eixo Z e da inclinação do veículo, que podem ser verificadas na Figura 27a e na Figura 27b. Estes dados foram analisados utilizando o algoritmo representado na Figura 22, e na Figura 27c pode ser visualizado o resultado do teste em campo, onde os pontos indicam os buracos detectados em cada sensor.

Figura 26 – (a) Placa de desenvolvimento Zedboard, (b) Computador com ambiente de desenvolvimento



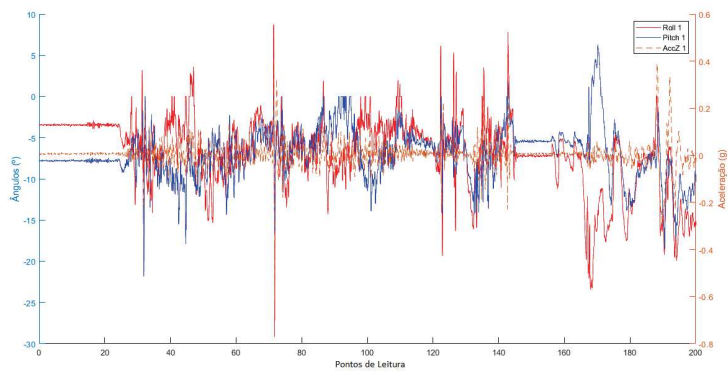
(a)



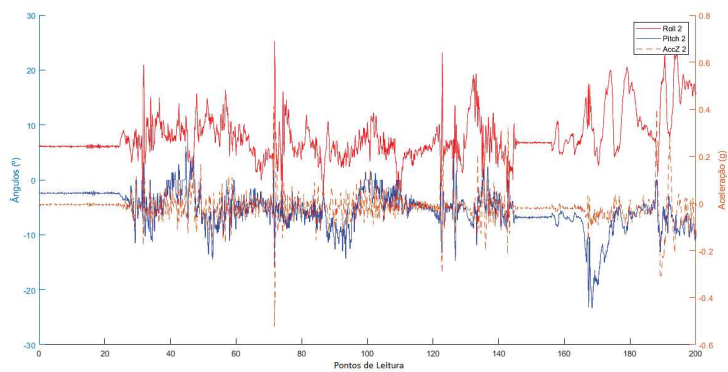
(b)

Fonte: Autoria Própria

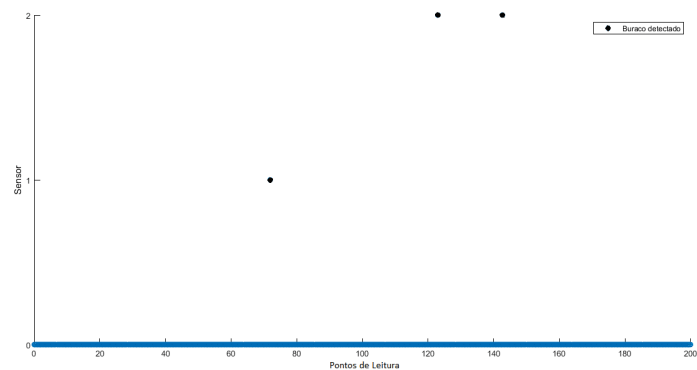
Figura 27 – Resultados do teste em campo realizado em um trajeto conhecido (a) Sinais do sensor 1, (b) Sinais do sensor 2, (c) Buracos detectados conforme algoritmo



(a)



(b)



(c)

Fonte: Autoria Própria

O sistema pode acusar buracos falsos, como tampas de bueiros abaixo do nível da pista de rolamento, e também quando o veículo trafega sobre tachões. Para evitar esta situação, podem ser utilizados recursos de visão computacional, como reconhecimento de padrões.

6 CONCLUSÃO

Este trabalho foi desenvolvido utilizando grande parte dos conhecimentos adquiridos no curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná. Foram utilizados conceitos de processamento de sinais, desenvolvimento de *hardware* e *software*. Grande parte do esforço utilizado foi em aprender uma nova arquitetura, pois ao utilizar uma plataforma de desenvolvimento que contém o SoC (System On Chip) Xilinx Zynq-7000, foi necessário decidir os prós e contras de realizar determinada tarefa em *hardware* ou em *software*.

Foram muitos conhecimentos adquiridos e consolidados. A decisão de realizar a leitura dos dados das IMUs com a FPGA trouxe um grande aprendizado de protocolos e barramentos, aplicação de máquinas de estado finito, sincronização de sinais, e também como a aplicação de filtros de digitais escritos em VHDL.

Durante o desenvolvimento do projeto, as principais dificuldades encontradas foram em fazer a comunicação I²C em VHDL, e o mapeamento dos sinais oriundos da programação lógica (FPGA) e o sistema de processamento (microprocessador). A utilização do microprocessador ARM foi de suma importância para o projeto, pois proporcionou o uso de bibliotecas para uso de funções trigonométricas com aritmética de ponto flutuante.

Para ser um projeto viável economicamente, todas as rotinas podem ser realizadas no microprocessador, diminuindo assim custos para o desenvolvimento. Porém, o objetivo deste trabalho era aprimorar o conhecimento para futuros projetos que façam uso de FPGA e microcontroladores, onde cada componente possa realizar uma tarefa com maior eficiência.

Referências

- BURGART, S. Gap trap: A pothole detection and reporting system utilizing mobile devices. 2014. Citado na página 17.
- CHU, P. **FPGA prototyping by VHDL examples**. 1. ed. New Jersey: Wiley-Interscience, 2008. Citado na página 20.
- CONFEDERAÇÃO NACIONAL DOS TRANSPORTE, SERVIÇO SOCIAL DO TRANSPORTE, SERVIÇO NACIONAL DE APRENDIZAGEM DO TRANSPORTE. **Pesquisa CNT de rodovias 2018**: relatório gerencial. [S.l.], 2018. 405 p. Disponível em: <<http://cms.cnt.org.br>>. Acesso em: 28 de setembro de 2019. Citado na página 16.
- CONFEDERAÇÃO NACIONAL DOS TRANSPORTES. **Anuário CNT do Transporte 2018**. 2018. Disponível em: <<http://anuariodotransporte.cnt.org.br/2018/>>. Acesso em: 22 de setembro de 2019. Citado na página 14.
- DEPARTAMENTO NACIONAL DE INFRAESTRUTURA TERRESTRE. **Manual de Restauração de Pavimentos Asfálticos**. [S.l.], 2006. 310 p. Disponível em: <<http://ipr.dnit.gov.br>>. Acesso em: 22 de setembro de 2019. Citado 2 vezes nas páginas 14 e 16.
- DURÃO, C. R. C. **Utilização de Unidades de Medidas Inerciais Baseadas em Sistemas Microeletromecânicos em Navegadores Integrados**. 2009. 183 f. Tese (Doutorado em Ciências em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2009. Citado na página 18.
- EL-RABBANY, A. **Introduction GPS: The global positioning system**. 1. ed. Boston: Artech House, 2002. Citado na página 20.
- ERIKSSON, J. et al. The pothole patrol: Using a mobile sensor network for road surface monitoring. **Proceedings of the 6th international conference on mobile systems, applications, and services**, p. 29–39, 2008. Citado na página 17.
- HIGGINS, J. W. T. A comparison of complementary and kalman filtering. **IEEE Transactions on Aerospace and Electronic Systems**, AES-11, n. 3, 1975. Citado na página 22.
- INSTITUTO DE PESQUISAS RODOVIÁRIAS, DEPARTAMENTO NACIONAL DE INFRAESTRUTURA TERRESTRE. **Glossário de termos técnicos Rodoviários**. [S.l.], 2017. 321 p. Disponível em: <<http://ipr.dnit.gov.br>>. Acesso em: 22 de setembro de 2019. Citado 2 vezes nas páginas 14 e 16.
- INVENSENSE. **MPU-6000 and MPU-6050 Product Specification**. [S.l.], 2013. 52 p. Disponível em: <<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>>. Acesso em: 17 de setembro de 2020. Citado 2 vezes nas páginas 31 e 36.
- JO, Y.; RYU, S. Pothole detection system using a black-box camera. **Sensors**, n. 15, p. 29316–29331, 2015. Citado na página 17.
- JOHNSON, J. Creating a custom ip block in vivado. 2014. Disponível em: <<http://www.fpgadeveloper.com/2014/08/creating-a-custom-ip-block-in-vivado.html>>. Acesso em: 5 de agosto de 2019. Citado na página 32.

- KANG, B. ho; CHOI, S. il. Pothole detection system using 2d lidar and camera. **2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)**, 2017. Citado na página 17.
- KIM, T.; RYU, S.-K. Review and analysis of pothole detection methods. **Journal of Emerging Trends in Computing and Information Sciences**, v. 5, n. 5, p. 603–608, 2014. Citado 2 vezes nas páginas 16 e 17.
- KOK, M.; HOL, J. D.; SCHÖN, T. B. Using inertial sensors for position and orientation estimation. **Foundations and Trends in Signal Processing**, Now Foundations and Trends, v. 11, n. 1-2, p. 1–153, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8187588>>. Acesso em: 10 de outubro de 2019. Citado 3 vezes nas páginas 18, 22 e 25.
- NXP SEMICONDUCTORS. **I²C-bus specification and user manual**. [S.l.], 2014. 64 p. Disponível em: <<https://www.nxp.com/docs/en/user-guide/UM10204.pdf>>. Acesso em: 17 de setembro de 2020. Citado na página 25.
- PEDRONI, V. A. **Circuit Design and Simulation with VHDL**. 2. ed. Cambridge, Massachusetts: The MIT Press, 2010. Citado na página 19.
- PÓVOA, R. C. B. L. **Síntese Evolucionária de Circuitos Digitais Empregando FPGA's**. 2010. Disponível em: <http://www.puc-rio.br/pibic/relatorio_resumo2010/relatorios/ctc/ele/ELE-Rogério_Povoa.pdf>. Acesso em: 07 de dezembro de 2019. Citado na página 19.
- SANTANA, D. D. S. **Navegação terrestre usando unidade de medição inercial de baixo desempenho e fusão sensorial com filtro de Kalman adaptativo suavizado**. 2011. 209 f. Tese (Doutorado em Engenharia) — Escola Politécnica, Universidade de São Paulo, São Paulo, 2011. Citado 2 vezes nas páginas 18 e 23.
- SMITH, S. W. **The Scientist and Engineer's Guide to Digital Signal Processing**. 2. ed. San Diego, California: California Technical Publishing, 1999. Citado na página 20.
- SUWANDI, B.; KITASUKA, T.; ARITSUGI, M. Vehicle vibration error compensation on imu-accelerometer sensor using adaptive filter and low-pass filter approaches. **Journal of Information Processing**, v. 27, p. 33–40, 2019. Disponível em: <https://www.jstage.jst.go.jp/article/ipsjip/27/0/27_33/_pdf>. Acesso em: 30 de setembro de 2020. Citado na página 21.
- WANG, K. C. P. Challenges and feasibility for comprehensive automated survey of pavement conditions. **In Proceedings of 8th International Conference on Applications of Advanced Technologies in Transportation Engineering**, p. 531–536, 2004. Citado na página 17.
- WELCH, G.; BISHOP, G. An introduction to the kalman filter. University of North Carolina at Chapel Hill, 2006. Citado na página 23.
- XILINX. **AXI Reference Guide**. [S.l.], 2011. 80 p. Disponível em: <<https://www.xilinx.com>>. Acesso em: 17 de novembro de 2019. Citado na página 32.
- ZHU, M.; YU, F.; XIAO, S. An unconventional multiple low-cost imu and gps-integrated kinematic positioning and navigation method based on singer model. **Sensors**, 2019. Disponível em: <<https://europepmc.org/article/PMC/6806061>>. Acesso em: 30 de setembro de 2020. Citado na página 21.