

**FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS
CURSO**

GIORGIOS PANAGIOTIS AMORIM BOKOS
Matrícula 21752093

**CONTROLADOR PID PARA PROJETO DE PORTA-COPO
RESPONSIVO DE CARRO**

BRASÍLIA
2022

GIORGIOS PANAGIOTIS AMORIM BOKOS

CONTROLADOR PID PARA PROJETO DE PORTA-COPO RESPONSIVO DE CARRO

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia de Computação do UniCEUB – Centro Universitário de Brasília

Orientador (a): Prof. MsC Francisco Javier De Obaldía Díaz

GIORGIOS PANAGIOTIS AMORIM BOKOS

**CONTROLADOR PID PARA PROJETO DE PORTA-COPO
RESPONSIVO DE CARRO**

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia de Computação do UniCEUB – Centro Universitário de Brasília

Orientador (a): Prof. MsC Francisco Javier De Obaldía Díaz

Brasília, 2022.

BANCA EXAMINADORA

Prof. Francisco Javier De Obaldía Díaz - Mestre.
Orientador (a)

Hudson Capanema Zaidan - titulação.
Examinador (a)

Ingrid Maria Dittert - titulação.
Examinador (a)

CONTROLADOR PID PARA PROJETO DE PORTA-COPO RESPONSIVO DE CARRO PID CONTROLER FOR A RESPONSIVE CAR CUP HOLDER

Giorgios Panagiotis Amorim Bokos¹, Prof. Francisco Javier De Obaldía Díaz², Hudson Capanema Zaidan³, Ingrid Maria Dittert⁴

Resumo

O objetivo deste artigo é apresentar um projeto de controle para um sistema de tempo real e mostrar a precisão de um algoritmo desse modelo para realizar um controle exato de um porta-copo responsivo de um automóvel de forma que o porta-copo se movimente para impedir que o líquido que está depositado no copo preso ao porta-copo caia, independentemente do movimento do carro. Isso mostra a capacidade do sistema de prever situações e controlá-las no menor espaço de tempo possível tendo a menor taxa de erro. Para este fim, foi utilizado como base projetos de controladores PID que realizam movimentos semelhantes, porém com estruturas e objetivos diferentes para chegar ao algoritmo que configura tal controlador. Com isso esse projeto pode apresentar resultados dentro do esperado no sentido de desempenho.

Palavras-chave: Controladores PID. Taxa de erro. Responsivo.

Abstract: This article objective is to present a control project for a real-time system and show the precision of this algorithm to accomplish an exact control of a responsive cup holder for a car that moves in a way that the liquid inside it won't fall off the cup, despite the movement. It shows the capability of the system to predict situations and control them as fast as possible having the smaller error rate. To that end, was used as inspiration, projects of PID controllers that have similar movements, but with different structures and purposes, to achieve the algorithm that configures such controller. Upon this, this project could present results as expected in performance.

keywords: PID Controller. Error Rate. Renpositive.

¹ UniCEUB, aluno

² UniCEUB, orientador.

³ UniCEUB, primeiro examinador.

⁴ UniCEUB, segundo examinador.

1 INTRODUÇÃO

O objeto do estudo, no qual está baseado o projeto, se deu por observância a uma situação do cotidiano e a possibilidade de implementar uma solução aplicando teorias da computação e eletrônica.

Todos já passaram por uma situação de desconforto ao estar no carro tomando alguma bebida, um refrigerante, por exemplo, e, devido a alguma ação repentina no trânsito, ver o líquido ser derramado no carro.

Essa ocasião alimentou a ideia para realização deste projeto, visto que, existem projetos similares de equilíbrio em tempo real para aplicações diferentes, mas com princípios tecnológicos semelhantes. Portanto, a ideia principal foi elaborar um processo para criar um porta-copo para veículos que compensasse os movimentos bruscos do carro inclinando o porta-copo em uma determinada direção e velocidade conseguindo impedir que o líquido seja derramado.

O sistema como um todo precisou ser elaborado utilizando teorias de circuitos de malha fechada e regras de sintonia para controladores PID, somado a componentes para realizar as leituras do sinal do ambiente, um microcontrolador e motores. Dessa forma, foi possível fechar o ciclo de realimentação do sistema usando o sinal captado para ajustar o sinal de saída para a saída desejada provocando, assim, a ação de inclinar o copo.

A possibilidade de aplicação desse sistema gera resultados significantes para implementações de projetos embarcados com base no modelo de controle PID.

2 REVISÃO BIBLIOGRÁFICA

Para iniciar, é importante destacar o funcionamento e a teoria por trás de cada elemento necessário para construção do projeto de automação em questão.

Existem algumas teorias primordiais para o desenvolvimento da pesquisa que precisam ser abrangidas no presente artigo.

Nesse sentido, será apresentada a revisão bibliográfica sobre os seguintes temas: malha fechada; PID; componentes utilizados.

2.1 Malha fechada⁵

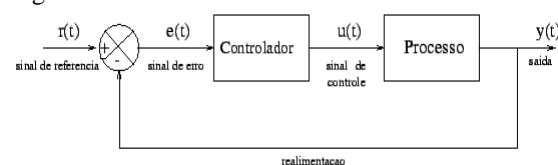
Um circuito de malha fechada consiste em um sistema em que a saída produzida é utilizada como entrada para determinar o sinal de controle do mesmo sistema em um instante específico.

O sistema de controle de malha fechada, que também é chamado de sistema de controle com realimentação, se baseia em um sinal de erro atuante $[e(t)]$, objetivando o estado desejado $[setpoint r(t)]$.

Isto é, o sistema de malha fechada busca a utilização do controle de realimentação para reduzir o erro do sistema.

O erro atuante é calculado pela diferença entre o sinal de entrada e o sinal de realimentação [sinal de saída $- y(t) -$ ou função do sinal de saída], conforme evidencia-se no processo mostrado na Figura 1.

Figura 1. Circuito de malha fechada.



Fonte: disponível em: <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node6.html>

⁵ Ogata, Katsuhiko. Engenharia de controle moderno / Katsuhiko Ogata; tradutora Heloísa Coimbra de Souza; revisor técnico Eduardo Aoun Tannuri. - 5. ed. - São Paulo: Pearson Prentice Hall, 2010.

Desse modo, o sinal de saída pode ser comparado com o *setpoint* do sistema e o erro medido para que a ação, se necessária, corrija o erro encontrado.

Com base nesse cálculo feito, o controlador do sistema emite sinais de controle [u(t)] visando alcançar o *setpoint*.

O *setpoint* é heurísticamente determinado conforme as particularidades de cada sistema. Ele se traduz no ponto ideal de operação do sistema.

Em breves palavras, o controle em malha fechada permite aumentar a precisão e estabilidade do sistema minimizando os ruídos externos e corrigindo os erros.

2.2 PID^{6,7,8}

O controlador PID é um método muito utilizado na indústria para sistemas de controle por possibilitar a sua aplicação em diversos projetos.

Se caracteriza por utilizar as ações proporcional, integral e derivativa para controlar um processo mecânico e reduzir o sinal de erro ao configurar corretamente essas variáveis.

Cada ação dessa age de uma forma sobre o sistema. Para tanto, detalha-se abaixo:

O controle proporcional ajusta a variável de controle de forma proporcional, multiplicando a magnitude do erro pelo o ganho proporcional estabelecido para o sistema. Conforme fórmula 1 abaixo:

$$P = e * kP \quad \text{Fórmula 1}$$

Em que:

e = erro medido

kP = coeficiente proporcional

Isso configura uma maior velocidade de resposta. Entretanto, cabe pontuar que, na hipótese de um ganho proporcional excessivo, o sistema tenderá a ficar instável, precisando ser calibrado corretamente para evitar essas oscilações.

O controle integral ajusta a variável de controle baseando-se na soma do termo de erro ao longo do tempo.

$$I = I + (e * kI) \quad \text{Fórmula 2}$$

Em que:

e = erro medido

I = Cálculo integral da última medição

kI = coeficiente integrativa

Em resumo, a ação integral conduz o erro de estado estacionário para zero, tendo em vista que a cada instante de tempo que o erro continua presente ele aumenta ainda mais a ação integral.

O controle derivativo ajusta a variável de controle tendo como base a taxa de variação do erro. Ou seja, conforme ocorre a variação da variável que ajusta o processo, a derivada acompanha proporcionalmente essa taxa de variação.

$$D = (\Delta S) * kD \quad \text{Fórmula 3}$$

Em que:

ΔS = Diferença entre as últimas duas medições do sensor

kD = coeficiente derivativa

Na prática, a diferença da última medição para a medição atual é multiplicada pelo ganho derivativo,

⁶ Graham C. Goodwin, Stefan F. Graebe, Mario E. Salgado. Control System Design, Prentice Hall PTR. Disponível em: https://www.academia.edu/23184065/CONTROL_SYSTEM_DESIGN

⁷ John W. Webb Ronald A. Reis. Programmable Logic Controllers, Fourth Edition, Prentice Hall PTR – 4.ed.

⁸ Ogata, Katsuhiko. Engenharia de controle moderno / Katsuhiko Ogata; tradutora Heloísa Coimbra de Souza; revisor técnico Eduardo Aoun Tannuri. - 5. ed.- São Paulo: Pearson Prentice Hall, 2010.

tornando esse componente diretamente proporcional à variação das últimas medições.

Ao final é feita a soma das 3 equações e chega-se ao ganho estabelecido.

$$PID = P + I + D \quad \text{Fórmula 4}$$

Em que:

P = Cálculo proporcional

I = Cálculo integral

D = Cálculo derivativo

Ao se aplicar os três tipos de controle juntos e configurá-los corretamente sobre a função de transferência do sistema pode-se obter um mecanismo de controle com uma taxa de erro dentro do aceitável.

2.3 Componentes

Na construção do projeto, fez-se o uso de alguns componentes essenciais para o funcionamento do sistema.

Observado o ciclo do processo, cada componente é responsável por uma ação. Primeiramente, o sensor é responsável por fazer a captação do sinal. Em seguida, o controlador executa sua rotina de cálculos para saber que ação deve ser tomada. Por fim, caso o controlador determine que alguma ação deva ser tomada o atuador é acionado para realizá-la.

Nesse sentido, abaixo serão mostrados e explicitados cada componente utilizado no projeto.

2.3.1 Acelerômetro

O acelerômetro é um tipo de sensor capaz de realizar medições de aceleração, sejam elas, estáticas ou dinâmicas.

Uma aceleração estática é uma força constante que age sobre um corpo, podendo ser uniforme e preditiva. Um exemplo para forças do tipo é a força da gravidade.

Já uma aceleração dinâmica, seria representada por um choque repentino, ou seja, uma força não uniforme aplicada sobre

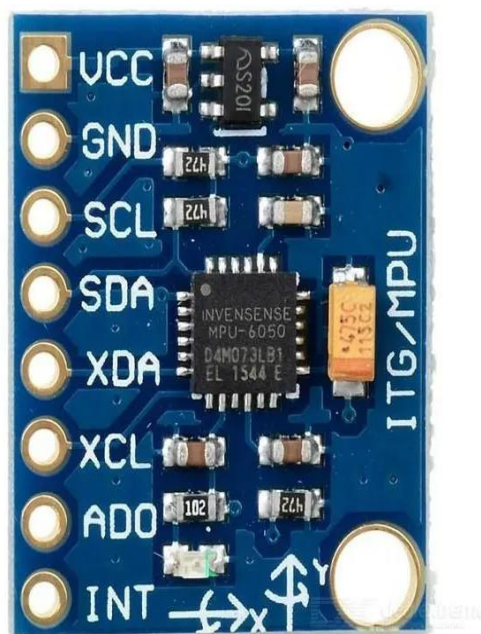
um corpo. Por exemplo: uma batida de carro.

Portanto, um acelerômetro se caracteriza por conseguir registrar acelerações de forma analógica e traduzir isso para sinais digitais.

Existem diversos tipos de acelerômetros que medem a aceleração de maneiras diferentes. Para o objeto de estudo utilizou-se um acelerômetro, comumente usado para projetos de arduino, chamado MPU-6050 que é do tipo MEMS (*Micro Electro-Mechanical Systems*). Este modelo de acelerômetro é utilizado para realizar medições de movimentos lineares sem um ponto de referência fixo.

A Figura 2 mostra um acelerômetro MPU-6050.

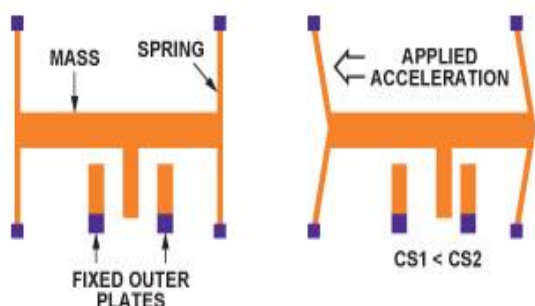
Figura 2. MPU-6050.



Fonte: disponível em: <https://www.microwat.com.br/arduino-prototipagem/modulos-e-componentes/acelerometro-e-giroscopio-3-eixos-mpu-6050-mpu6050>

O funcionamento de um acelerômetro MEMS se dá por uma estrutura que possui uma mola que se movimenta livremente dentro do corpo e placas fixas. Como mostrado na Figura 3.

Figura 3. Estrutura do acelerômetro MEMS.



Fonte: disponível em: <https://www.analog.com/en/analog-dialogue/articles/mems-accelerometers-as-acoustic-pickups.html>

A mola se encontra livre para movimentação em um dos eixos e quando o objeto é submetido a uma aceleração é possível medir a variação da capacitância entre as paredes dessa massa móvel e as paredes fixas. Essa diferença medida é proporcional à aceleração a que o objeto é submetido.

Dessa forma, para um acelerômetro que mede acelerações nos 3 eixos, basta que o componente tenha uma estrutura dessa orientada para cada eixo ou uma única estrutura com sensores capacitivos dispostos de forma a medir o movimento em cada eixo.

Sobre o MPU-6050, é possível destacar algumas importantes características:

- Possui 4 fundos de escala programados variando de $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$;
- Corrente de operação em torno de $500\mu A$;
- Opera com voltagem entre $2.375V - 3.46V$ e temperatura entre $-40^{\circ}C - 70^{\circ}C$;
- Possui sensores de giroscópio e de temperatura embutidos, além do sensor de aceleração;
- Comunicação I2C;
- Tamanho pequeno, contribuindo para utilização em sistemas embarcados.

2.3.2 Motor

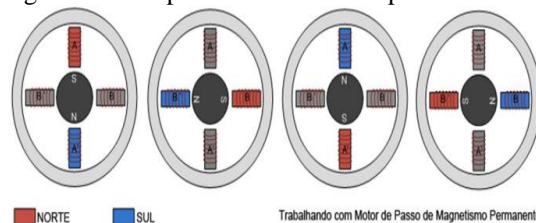
Um motor é um dos atuadores mais utilizados em sistemas de controle. Sua utilização garante mobilidade ao sistema podendo ser configurado de inúmeras maneiras.

Para sistemas de controle de menor porte são empregados dois tipos de motores, o servomotor e o motor de passo. O motor de passo possui uma precisão maior que o servomotor e, por esse motivo, foi aplicado no desenvolvimento deste projeto.

Dentre os motores de passo existem 3 tipos: motor de ímã permanente; motor de relutância variável; e motor híbrido.

Os motores de ímã permanente possuem um rotor magnetizado e bobinas em torno do rotor. Dessa forma, os acionamentos das bobinas geram um campo magnético que orienta o motor da forma desejada, conforme mostra a Figura 4.

Figura 4. Exemplo de motor de ímã permanente



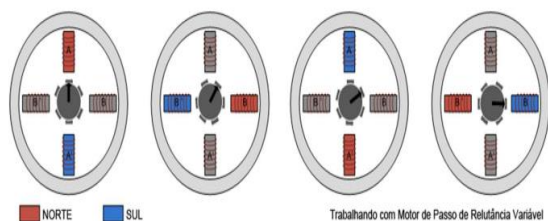
Fonte: disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>

Este tipo de motor pode ter sua precisão aumentada acrescentando o número de fases ou o número de polos no rotor.

O motor de relutância variável não possui um rotor como ímã permanente, nesse caso os dentes do rotor são alinhados e desalinhados com os dentes do estator conforme as bobinas são energizadas, provocando um torque no motor para dar cada passo e, assim, conseguindo girá-lo.

A imagem da Figura 5 ilustra um modelo de motor de relutância variável.

Figura 5. Exemplo de motor de relutância variável



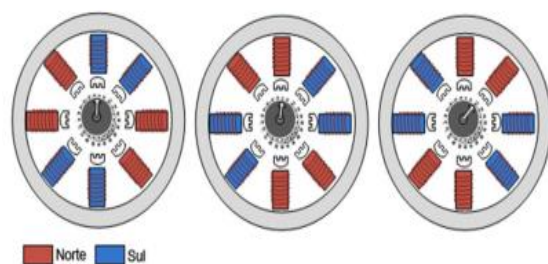
Fonte: disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>

Para aumentar a precisão do motor pode-se aumentar a quantidade de dentes no rotor ou aumentar o número de fases.

No caso do motor híbrido, como o próprio nome diz, este tipo de motor combina a ideia do motor de ímã permanente com a ideia do motor de relutância variável.

Nele o eixo do rotor é dividido em dois grupos de dentes, um com o polo sul e outro com o polo norte, assim os dentes de cada grupo ficam alternados. Dessa forma, para que o motor rotacione, assim como nos outros modelos, as bobinas são acionadas eletricamente em sequência como ilustrado na Figura 6.

Figura 6. Exemplo de motor híbrido



Fonte: disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>

Em resumo, o motor híbrido faz uso do alto torque fornecido por motores de ímã permanente e a alta precisão do motor de relutância variável.

Os motores de passo também podem ser caracterizados em unipolar ou multipolar.

No motor unipolar existe a ativação de uma bobina por vez para fazer o rotor girar. Já no motor bipolar, duas bobinas são ativadas por vez.

O motor bipolar costuma gastar mais energia para ativação, mas tem uma velocidade de resposta maior em comparação com o unipolar.

Uma vez explicados estes conceitos, é possível detalhar o motor utilizado neste projeto: o NEMA 17, modelo SY42STH47-1206A mostrado na Figura 7.

Figura 7. Motor NEMA 17.



Fonte: disponível em: <https://www.huinfinito.com.br/motores/696-motor-de-passo-unipolar-bipolar-nema-17.html>

Este motor é híbrido e possui as seguintes características que contribuíram para ser selecionado.

- Alta precisão com uma angulação de passo de 1.8°;
- 2 fases;
- Temperatura de operação de -20°C - +50°C;
- Voltagem de operação de 4 V e corrente de 1.2 A, por fase;
- Torque de 3.17 Kg.cm;
- Peso de 365g;

- Quantidade de passos para uma revolução: 200.

2.3.3 Driver do Motor

Os motores de passo normalmente precisam de um driver para que possam ser controlados corretamente e, para o caso do NEMA 17, um driver comumente utilizado para operar em conjunto é o A4988.

As principais características deste driver são:

- Voltagem de alimentação de no máximo 5.5 V;
- Controla motor com voltagem entre 8 V – 37 V;
- Opera em temperaturas entre -20°C – 85°C;
- Fornece corrente de até 2 A.

2.3.4 Controlador

O controlador é um elemento essencial para desenvolvimento de qualquer projeto de controle de malha fechada.

Devido ao seu baixo custo e facilidade de encontrar códigos *open-source*, o tipo de controlador mais escolhido são os arduinos.

Existem diversos tipos de arduinos que deverão ser selecionados com base no projeto no qual serão utilizados.

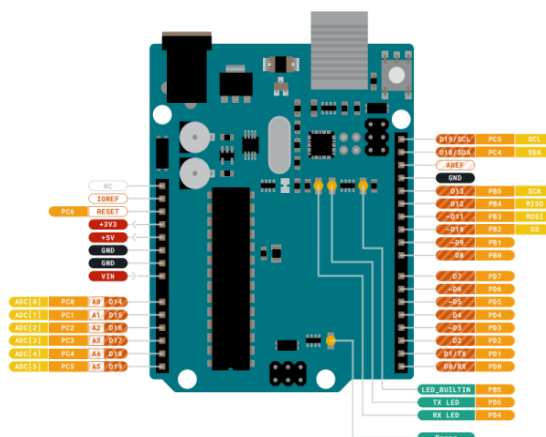
Para o objeto de estudo em questão foi possível aplicar o Arduino Uno R3. Este controlador faz uso dos processadores ATmega328P e ATmega16U2 e deve ser alimentado com voltagem entre 2.7V – 5.5V.

O arduino possui também uma IDE (*Integrated Drive Eletronics*) para desenvolvimento gratuita de fácil acesso e compreensão.

Dentre as várias possibilidades de pinagem no arduino pode-se destacar: os 6 pinos PWM; 1 pino de controle I2C; além de 6 pinos analógicos; e 14 digitais.

A Figura 8 demonstra a placa do arduino e todas suas pinagens.

Figura 8. Pinagem do arduino



Fonte: disponível em: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

É preciso destacar a importância dos pinos SDA e SCL, que funcionam de acordo com o protocolo de comunicação I2C.

Este protocolo faz uso destes pinos para estabelecer uma comunicação de mestre e escravo com os componentes. Neste caso, o arduino é o mestre e quaisquer dispositivos conectados por este pino serão tido como escravos.

A forma de endereçamento permite que cada escravo conectado tenha um endereço único, possibilitando a comunicação com o mestre.

Neste projeto, a comunicação feita com o módulo que mede aceleração (MPU-6050) é feita através do protocolo I2C utilizando os pinos SDA e SCL.

3 METODOLOGIA DE DESENVOLVIMENTO DO TRABALHO

O comportamento ideal do projeto se baseia em um defeito crítico ao funcionamento do projeto: o derramamento do líquido do copo.

Esta falha pode se originar por duas ocasiões: (i) o movimento responsivo programado no código não foi corretamente ajustado, provocando a queda do líquido por algum motivo (por exemplo, por passos além do ideal ou abaixo do ideal, ou até mesmo por velocidade e aceleração do

próprio movimento); ou (ii) uma queda por não movimentação do copo devido a algum erro dos componentes que atuam no projeto.

Dessa forma, para o correto desempenho, o código construído precisou passar por uma série de testes a fim de assegurar que os componentes escolhidos e detalhados acima se comportassem como o desejado.

Além disso, a estrutura construída precisou ser milimetricamente especificada para que o comando dado ao motor fosse traduzido em movimento compensatório para o copo.

3.1 Desenvolvimento do Protótipo

A construção da estrutura do projeto foi feita por impressora 3D e, para isso, foi preciso desenhar as peças utilizando *softwares* de impressão 3D.

Parte das peças foram desenhadas na plataforma do 'Fusion 360' e as outras peças foram desenhadas na plataforma *web* do 'Tinkercad'.

Como estrutura principal para o copo foi impressa uma plataforma com um furo no meio e dois encaixes laterais centralizados que serão usados para conectar as hastes que servirão de ligação e controle pelos motores. A Figura 9 mostra essa estrutura.

Figura 9. Suporte principal do porta-copo

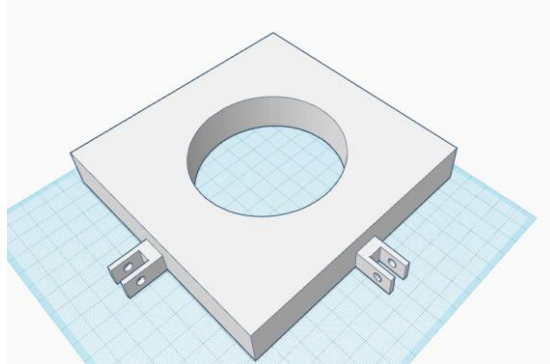


Imagem de 07.06.2022

O tempo de duração para impressão desta peça foi de 5 horas e 6 minutos e foi impressa com material de impressão ABS.

Uma segunda peça de estrutura foi impressa para realizar o suporte dos componentes. Nessa peça, os motores serão dispostos na parte superior. Como é mostrado na Figura 10.

Figura 10. Suporte inferior do porta-copo

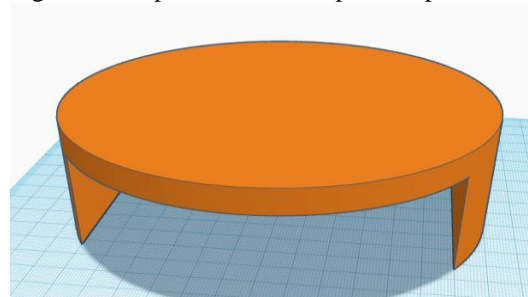


Imagem de 07.06.2022

O tempo de duração para impressão desta peça foi de 6 horas e 12 minutos e foi impressa com material de impressão PLA.

Além dessas peças de estrutura, algumas peças menores foram impressas para conectar as duas estruturas e realizar o movimento desejado.

Para acoplar ao motor foi desenhado um disco que traduz o movimento. Essa peça, por sua vez, foi acoplada a uma haste, também impressa, que é conectada a plataforma principal do copo.

Essa estrutura de movimento precisou ser feita dessa forma para transformar o movimento angular em movimento retilíneo, a fim de evitar problemas de travamento que poderiam ocorrer de acordo com o movimento das peças.

A Figura 11 ilustra essas peças citadas acima.

Figura 11. Peças de estrutura

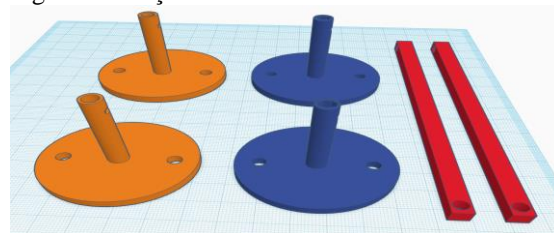


Imagem de 07.06.2022

O tempo de duração para impressão destas peças foi de 4 horas e 33 minutos e

foi impressa com material de impressão PLA.

Foram desenhados dois tipos de discos para testar o melhor encaixe no eixo do motor.

Com essas peças foi possível atingir o nível de movimentação e de suporte de peso necessários para o correto funcionamento do projeto.

3.2 Desenvolvimento do Circuito

Outra parte a ser desenvolvida trata das conexões dos componentes do circuito.

Foram utilizados 2 motores e, portanto, 2 *drivers*, além de 1 um sensor MPU-6050.

A Figura 12 mostra como foram feitas todas as conexões do sistema e a alimentação na *protoboard*.

Figura 12. Esquemático simplificado do sistema.

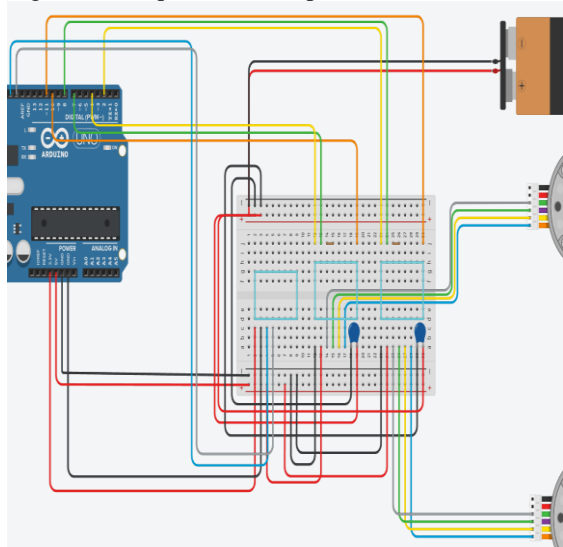


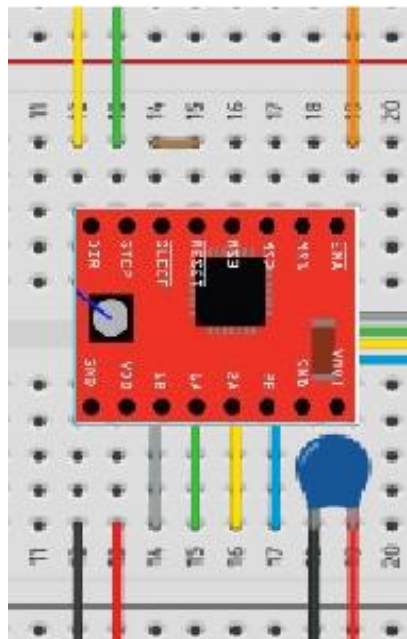
Imagem de 16.06.2022

De forma mais detalhada serão demonstradas cada uma dessas conexões.

Os *drivers* possuem duas fontes de alimentação, uma que vem do arduino com 5V e outra fonte externa de 12V, ambas alimentam a *protoboard* e, conseqüentemente, conseguem alimentar em paralelo cada driver do circuito.

A Figura 13 mostra cada conexão necessária para o funcionamento do *driver*.

Figura 13. Conexões do driver



Fonte: Giorgios Bokos (2022)

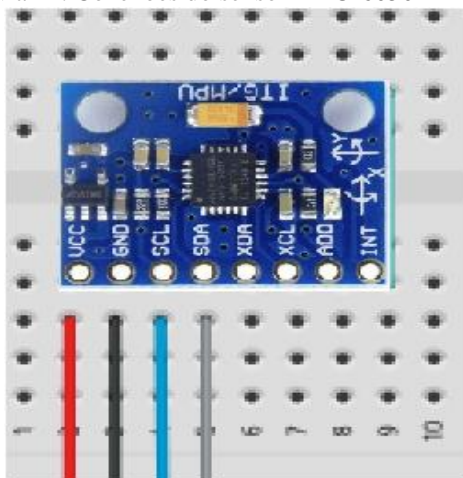
Como pode-se ver no detalhe da Figura 13, a alimentação feita nos pinos 12 e 13 da parte de baixo da *protoboard* advém do arduino e a alimentação feita nos pinos 18 e 19 vem da fonte externa. Um capacitor de 100 μ F é posicionado na fonte externa para evitar queima do componente.

Os 4 pinos, 14, 15, 16 e 17 da parte superior da *protoboard*, são conectados ao motor de passo para fazer o controle das bobinas. Já os pinos 12 e 13 são responsáveis por alimentar o *driver* com as informações de STEP (Quantidade de passos) e DIR (Direção do passo). Além disso, é preciso conectar os pinos 14 e 15 entre si para conectar o SLEEP ao RESET da placa e o pino 19 que passa informações de ENABLE ao *driver*.

Cada *driver* tem seus pinos conectados ao arduino, os cabos amarelos e verde da parte superior da *protoboard* são conectados a pinos digitais comuns do arduino. Já os pinos laranjas são conectados a pinos digitais PWM.

Além do *driver*, é preciso detalhar as conexões do sensor MPU-6050. A Figura 14 mostra as conexões deste dispositivo.

Figura 14. Conexões do sensor MPU-6050



Fonte: Giorgios Bokos (2022)

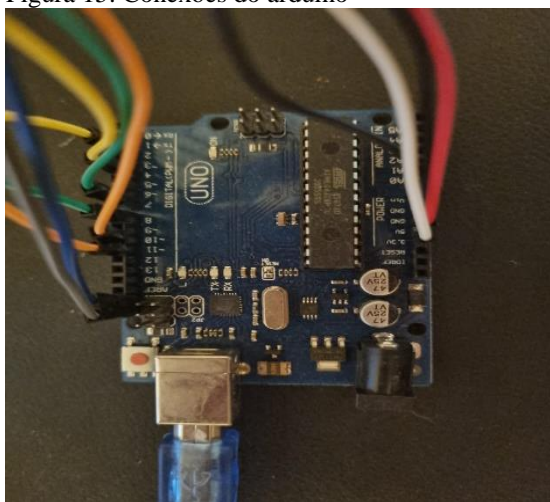
Como pode-se ver, a alimentação deste componente é feita através do arduino utilizando o pino de alimentação de 3.3V.

Para a comunicação do sensor foram utilizados os pinos I2C, SCL (*Clock line*) e SDA (*data line*), que facilitam a configuração do MPU-6050 com o arduino.

Todas as conexões foram exemplificadas pela plataforma do ‘Tinkercad’, no entanto, os dispositivos utilizados no projeto não estão disponíveis para uso nesta plataforma.

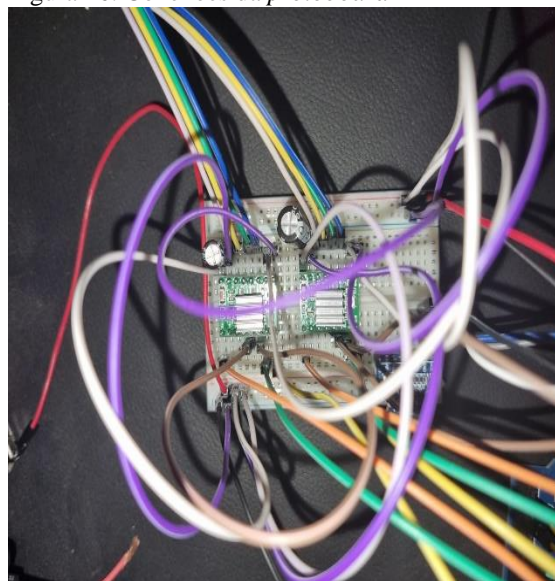
Para mostrar na prática como foram feitas as conexões foram selecionadas as Figuras 15, 16 e 17, que mostram as conexões na *proto-board* e no arduino.

Figura 15. Conexões do arduino



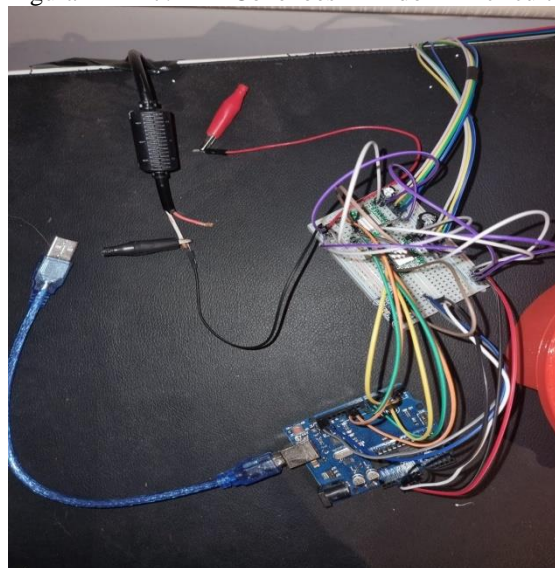
Fonte: Giorgios Bokos (2022)

Figura 16. Conexões da *proto-board*



Fonte: Giorgios Bokos (2022)

Figura 17. Conexões do circuito



Fonte: Giorgios Bokos (2022)

3.3 Desenvolvimento do Código

Como já citado anteriormente, o código desenvolvido foi baseado em dois conceitos de controle de componentes eletrônicos: circuito de malha fechada; e códigos PID.

De maneira geral, o código executado no arduino é baseado em um *loop* de execução tornando, de certa forma, trivial o processo de retroalimentação da malha de execução do ponto de vista da programação.

Uma vez que, basta que a saída obtida ao final do código seja comparada com a entrada medida em um dado momento da iteração de execução.

No entanto, o processo de construção do código com retroalimentação baseado na teoria de controladores PID torna o código mais complexo.

Cada um dos componentes (proporcional, integral e derivada) atuam sobre a saída agindo de alguma forma sobre o erro medido.

Como a movimentação da peça precisa ser feita nos eixos x e y para englobar os 360° do plano, foi preciso criar constantes PID para cada um dos eixos e configurá-las individualmente.

Cada eixo é controlado em conjunto pelos motores e possui uma função específica de controle PID. Dessa forma, ao somar a ação de cada motor pode-se obter qualquer ponto no espaço estabelecido de atuação.

Ainda se fez necessário, para obter um movimento mais suave na atuação dos motores, uma função com componentes PID para a velocidade e outra para determinar a aceleração com que o motor irá girar.

Para efeito de esclarecimento, os parâmetros iniciais estabelecidos para as constantes P, I e D foram definidos arbitrariamente em todos os casos e, posteriormente, foram ajustados conforme necessário para alcançar a estabilidade do processo.

No caso do objeto da presente pesquisa, o sistema é inicializado aproximando o valor percebido pelo acelerômetro no eixo x e y para 0 e iniciando o *setpoint*, para ambos os eixos, em 0.

Dessa forma, o processo é iniciado e espera até que uma medição de aceleração seja percebida. Quando uma medição é feita, este valor é comparado ao *setpoint* e é medido o erro. Em seguida, o erro passa por um filtro de média móvel das últimas 2 medições e é realizado os cálculos PID para

a posição, velocidade e aceleração do motor.

Caso a média dos últimos erros percebidos esteja acima dos valores determinados como limites, o controlador aciona o motor com base nos cálculos PID feitos para corrigir a aceleração medida. Se a média das últimas medições for abaixo dos limiares, o motor permanece parado.

As bibliotecas utilizadas foram a *'Wire.h'* – responsável pelas funções de comunicação I2C com o arduino; *'AccelStepper.h'* – responsável pelas funções de controle dos driver dos motores; e as funções *'LinkedList.h'*, *'Gaussian.h'* e *'GaussianAverage.h'* – que são as responsáveis por realizar os cálculos de média móvel do sistema.

As imagens das figuras 18, 19, 20 e 21 demonstram o código para as etapas citadas acima.

Figura 18. Comandos que recebem a aceleração medida pelo sensor.

```
Wire.beginTransmission(MPU);
Wire.write(0x3B);
Wire.endTransmission(false);
Wire.requestFrom(MPU, 6, true); // Solicita os dados

// Armazena o valor dos sensores nas variáveis corre
AccX = Wire.read() << 8 | Wire.read(); //0x3B (ACCEI
AccY = Wire.read() << 8 | Wire.read(); //0x3D (ACCEI

X = AccX/16384.0; // Fundo de escala -> -2g a 2g
Y = AccY/16384.0;
```

Fonte: Giorgios Bokos (2022)

Figura 19. Comandos que medem o erro e fazem o filtro de média móvel.

```
// implementando erros para os PID
error_X = setPoint - X;
Serial.print("Erro X: ");
Serial.println(error_X);

//Média móvel
mediaMovel_X += error_X;
mediaMovel_X.process();

Serial.print("Média em X: ");
Serial.println(mediaMovel_X.mean());
Serial.println();
```

Fonte: Giorgios Bokos (2022)

Para todos os cálculos PID foi utilizado o modelo abaixo, substituindo-se

somente os valores e as variáveis correspondentes quando necessário.

Figura 20. Cálculos PID que determinam a quantidade de passos do motor.

```
//PID para eixo X
// P
Px = error_X * kP_x;
// I
Ix += error_X * kI_x;
// D
Dx = (lastAccX - AccX) * kD_x;

// Soma tudo
PIDx = (Px + Ix + Dx)*100;

//converte para controle
controleX = (PIDx);
```

Fonte: Giorgios Bokos (2022)

A próxima etapa do código diz respeito a direção de rotação de um motor e qual motor deve ser acionado para estabilizar o sistema.

Como pode-se ver pela Figura 21, esta primeira condicional está atrelada ao movimento no eixo x e aciona um dos dois motores com base nos limiares estabelecidos para evitar ações de rotação desnecessárias.

Neste trecho do código o motor 2 é acionado caso seja percebido uma aceleração positiva no eixo x acima do limiar estabelecido. Caso a aceleração medida seja negativa e abaixo do limiar estabelecido, o motor acionado é o motor 1. Se a média móvel da aceleração não for suficiente para ultrapassar os limiares, o motor permanece parado.

No caso, o motor 2 giraria no sentido horário e o motor 1 giraria no sentido anti-horário em relação aos seus eixos. Em todos os casos, a velocidade e aceleração é ajustada de acordo com os cálculos feitos para evitar movimentos bruscos desnecessários.

Essa rotação inclina a superfície compensando a aceleração medida com base nos cálculos feitos previamente e, após a finalização do movimento, o motor retorna para sua posição de origem (a

posição 0 estabelecida ao inicializar o código).

O trecho de código da Figura 21 é replicado para acionar os motores em caso de aceleração medida no eixo y. Neste segundo caso, os motores, caso acionados, tem seu sentido de rotação alterado em relação ao mostrado no trecho de código da Figura 21.

Figura 21. Condicionais que determinam o movimento do motor.

```
motor1.enableOutputs();
motor2.enableOutputs();
if(motor1.run() == false && motor2.run() == false){
  if (mediaMovel_X.mean > 0.15){ //DESACELERAÇÃO NO EIXO X MEDE UMA FORÇA POSITIV
    Serial.println("Motor 1: Girando Sentido Horário");
    Serial.println();
    digitalWrite(pino_enable1, LOW);
    Serial.println(PID_vel_x);
    motor1.setSpeed(PID_vel_x);
    motor1.setAcceleration(PID_acc_x);
    motor1.runToNewPosition(controleX);
  } else if (mediaMovel_X.mean < -0.20){ //ACELERAÇÃO NO EIXO X MEDE UMA FORÇA N
    Serial.println("Motor 2: Girando Sentido horário");
    Serial.println();
    digitalWrite(pino_enable2, LOW);
    motor2.setSpeed(PID_vel_x);
    motor2.setAcceleration(PID_acc_x);
    motor2.runToNewPosition(-controleX);
  } else {
    Serial.println("Eixo X: Parando Motores");
    digitalWrite(pino_enable1, HIGH);
    digitalWrite(pino_enable2, HIGH);
  }
}
```

Fonte: Giorgios Bokos (2022)

O trecho do código da Figura 22 demonstra os comandos que fazem com que o motor retorne para a posição inicial ao final de um movimento.

Figura 22. Condicionais que determinam se o motor deve voltar para o ponto inicial.

```
if(motor1.currentPosition()>0 || motor1.currentPosition()<0){
  motor1.setSpeed(PID_vel_x);
  //motor1.setAcceleration(PID_acc_x);
  motor1.runToNewPosition(0);
}

if(motor2.currentPosition()>0 || motor2.currentPosition()<0){
  motor2.setSpeed(PID_vel_y);
  //motor2.setAcceleration(PID_acc_y);
  motor2.runToNewPosition(0);
}

motor1.disableOutputs();
motor2.disableOutputs();
}
```

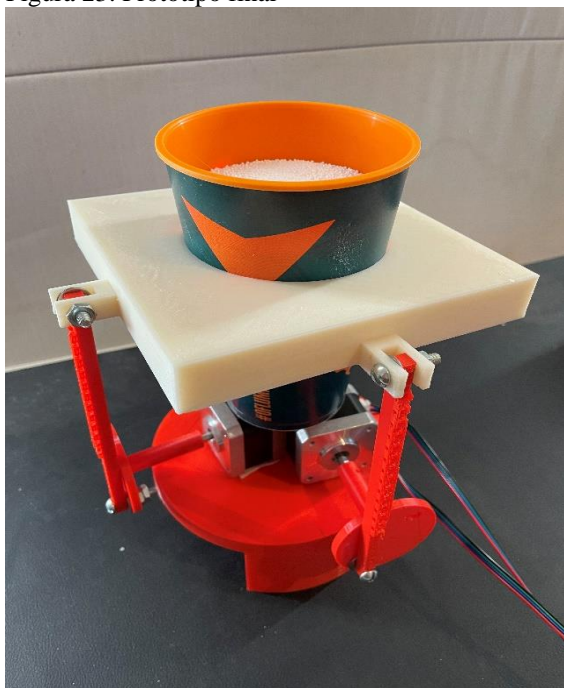
Fonte: Giorgios Bokos (2022)

Vale ressaltar que os recortes acima explicados representam apenas uma parte do código completo, este pode ser encontrado no apêndice do presente artigo.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Com o embasamento teórico e o desenvolvimento prático de todas as etapas de construção do protótipo, foi possível chegar ao modelo mostrado na Figura 23.

Figura 23. Protótipo final



Fonte: Giorgios Bokos (2022)

Dividindo por etapas de construção: (i) circuito eletrônico; (ii) desenho e impressão 3D; e (iii) código arduino.

A respeito do primeiro passo, a performance final do circuito correspondeu ao esquemático projetado em sua totalidade, não apresentando qualquer limitação ao processo.

No entanto, este estudo foi feito visando um protótipo funcional para demonstrar a aplicabilidade do sistema. Caso seja desenvolvido para uma aplicação prática desse modelo, a próxima etapa seria desenvolver um estudo sobre a necessidade de resfriamento do sistema e

encapsulamento, uma vez que, se trata de um sistema embarcado de tempo real que estaria sujeito a variações de temperatura e uso contínuo que poderiam acarretar no mau funcionamento ou até num desvio de especificação do projeto, demonstrando a grande utilidade da tecnologia de impressão 3D.

Em relação ao segundo passo, a estrutura projetada trouxe o grau de movimentação do sistema desejado e trouxe a sustentação necessária para manter a estabilidade proposta no projeto.

Entretanto, devido à fragilidade do material, à quantidade de pontos de sustentação e o desenho das peças articuladoras, um excesso de carga pode significar um mau funcionamento e possivelmente o rompimento da estrutura.

Por esse motivo e por precaução a eventuais derramamentos de líquidos que poderiam danificar permanentemente o sistema, foi utilizado, no lugar do líquido, pequenas bolas de isopor para simular o movimento da água.

Como alternativa de material poderia ser utilizado um metal leve, porém resistente, como o alumínio.

Conforme anteriormente mencionado, o fato da estrutura estar usando somente dois pontos de apoio, é outro agravante. Uma possível solução seria a inclusão de um terceiro ponto de apoio em conjunto com a ação de um terceiro motor, o que traria a necessidade de reajustes dos encaixes laterais centralizados, além de englobar mudanças no desenvolvimento do código e do esquemático do circuito com relação a ação dos motores.

Já quanto ao desenho das hastes, poderia ser projetado um novo posicionamento dos motores em conjunto com o modelo de haste utilizado. Esclarece-se que o projeto destas hastes levou em consideração a tentativa de evitar um problema de travamento das hastes na movimentação do sistema.

Embora essas melhorias estejam mapeadas, o projetado atende as

expectativas criadas para um primeiro protótipo.

Por fim, em análise ao último passo, a estrutura do código traduziu o movimento desejado, havendo a necessidade de realização de inúmeros testes a fim de alcançar o melhor desempenho possível dada as limitações no desenvolvimento do projeto.

O que ocorre na presente questão é que os testes realizados devem observar a união de todos os passos anteriores de forma harmônica e eficiente.

Nesse sentido, os testes tiveram sucesso em relação à movimentação prática proposta, tendo maior estabilidade quando melhor configurados os coeficientes PID.

Apesar disso, não foi viável chegar ao melhor desempenho possível quando levada em consideração, por exemplo, a inexistência de ambiente propício e seguro, além das instabilidades de percurso, tornando-se dificultoso precisar os coeficientes PID, fator esse que exponencialmente melhoraria o desempenho do sistema.

Para realizar estes testes de forma mais segura possível, por se tratar de variações bruscas de aceleração, o ideal seria alugar uma pista de automobilismo, ou algo do gênero, para garantir que as acelerações, frenagens e curvas pudessem estressar o sistema ao máximo sem violar nenhuma lei de trânsito e garantir a segurança de todos os envolvidos. Além disso, seria interessante realizar o encaixe do suporte de maneira fixa de forma que fique preso ao automóvel a todo instante.

5 CONSIDERAÇÕES FINAIS

Para o caso deste projeto, o conceito sobre este tipo de algoritmo foi o pilar principal, uma vez que, o objeto de estudo trata de um sistema de tempo real que precisa de rápida resposta à captação de sensores e tolerância baixíssima a erros.

Recapitulando, o desempenho, tanto do circuito quanto do código elaborado, foi

dentro do esperado conseguindo atingir o grau de movimentação proposto, porém para utilizar com uma carga maior de peso, deve-se repensar a estrutura construída por material de impressão 3D.

Dessa forma, aplicando todos os conceitos e métodos de desenvolvimento prova-se que é possível alcançar desempenhos ainda melhores tendo em vista os bons resultados já obtidos neste primeiro protótipo.

REFERÊNCIAS

Altana Tubes. Tipos de motores de passo e modos de operação. Disponível em: <https://www.altanatubes.com.br/webstore/?c=282&t=Tipos-de-motores-de-passo-e-modos-de-operacao>

AN907 Stepping Motors Fundamentals – Applications Note – Microchip, <http://ww1.microchip.com/downloads/en/AppNotes/00907a.pdf>.

Arduino® UNO R3. Product Reference Manual. SKU: A000066. <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

A4988 Datasheet(PDF) 2 Page - Allegro MicroSystems. <https://html.alldatasheet.com/html-pdf/455036/ALLEGRO/A4988/590/2/A4988.html>

Dadafshar, Majid. Accelerometer and Gyroscopes Sensors: Operation, Sensing, and Applications. Disponível em: <https://www.maximintegrated.com/jp/design/technical-documents/app-notes/5/5830.html>

Graham C. Goodwin, Stefan F. Graebe, Mario E. Salgado. Control System Design, Prentice Hall PTR. Disponível em: https://www.academia.edu/23184065/CONTROL_SYSTEM_DESIGN

High torque hybrid stepping motor specifications. Datasheet (PDF). Disponível

em:

https://components101.com/sites/default/files/component_datasheet/NEMA17.pdf

John W. Webb Ronald A. Reis. Programmable Logic Controllers, Fourth Edition, Prentice Hall PTR – 4.ed.

Nise, Norman S. Engenharia de sistemas de controle / Norman S. Nise; tradução e revisão técnica Jackson Paul Matsuura. - 7. ed. - Rio de Janeiro: LTC, 2017.

Locatelli, Caroline. Funcionamento de um Motor de Passo. Disponível em: <https://www.curtocircuito.com.br/blog/motor-de-passo/introducao-ao-motor-de-passo>

Madeira, Daniel. Protocolo I2C – Comunicação entre Arduinos. Disponível em: <https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinos/>

MPU-6050 Datasheet (PDF) - List of Unclassified Manufacturers. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/517744/ETC1/MPU-6050.html>

Ogata, Katsuhiko. Engenharia de controle moderno / Katsuhiko Ogata; tradutora Heloísa Coimbra de Souza; revisor técnico Eduardo Aoun Tannuri. - 5. ed.- São Paulo: Pearson Prentice Hall, 2010.

Silicon Sensing. MEMS Accelerometers. Disponível em: <https://www.siliconsensing.com/technology/mems-accelerometers/>

Stroski, Pedro. O que são MEMS? - Automação, Mecânica, Microeletrônica, Robótica, Sensores. Disponível em: <https://www.electricalibrary.com/2019/06/24/o-que-sao-mems/>

UNESP. Departamento de Engenharia Elétrica. Motor de passo. Disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>

FIGURAS

Figura 1. Circuito de malha fechada. Fonte: disponível em: <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node6.html>

Figura 2. Estrutura do acelerômetro MEMS. Fonte: disponível em: <https://www.analog.com/en/analog-dialogue/articles/mems-accelerometers-as-acoustic-pickups.html>

Figura 3. Exemplo de motor de ímã permanente. Fonte: disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>

Figura 4. Exemplo de motor de relutância variável. Fonte: disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>

Figura 5. Exemplo de motor híbrido. Fonte: disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>

Figura 6. Pinagem do arduino. Fonte: disponível em: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

Figura 7. Motor NEMA 17. Fonte: disponível em: <https://www.huinfinito.com.br/motores/696-motor-de-passo-unipolar-bipolar-nema-17.html>

Figura 8. Pinagem do arduino. Fonte: disponível em: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

