

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA
FATEC SANTO ANDRÉ

Tecnologia em Mecatrônica Industrial

HENRIQUE SATOSHI KISAKI
RENAN NUNES DE SIQUEIRA MANGANELLI

Santo André
2018

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA
FATEC SANTO ANDRÉ

Tecnologia em Mecatrônica Industrial

HENRIQUE SATOSHI KISAKI
RENAN NUNES DE SIQUEIRA MANGANELLI

ROBÔ COM SISTEMA GIROSCÓPIO E ACELERÔMETRO

Trabalho de Conclusão de Curso entregue à Fatec Santo André como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial.

Orientador: Prof. Me. Luiz Vasco Puglia

Santo André
2018

FICHA CATALOGRÁFICA

K614r

Kisaki, Henrique Satoshi

Robô com sistema giroscópio e acelerômetro / Henrique Satoshi
Kisaki, Renan Nunes de Siqueira Manganelli. - Santo André,
2018. - 83f. il.

Trabalho de Conclusão de Curso – FATEC Santo André.
Curso de Tecnologia em Mecatrônica Industrial, 2018.

Orientador: Prof. Luiz Vasco Puglia

1. Mecatrônica. 2. Robô. 3. Acelerômetro. 4. Giroscópio. 5.
Sensor. 6. MPU6050. 7. Sistema. I. Manganelli, Renan Nunes de
Siqueira II. Robô com sistema giroscópio e acelerômetro.

629.8

LISTA DE PRESENÇA

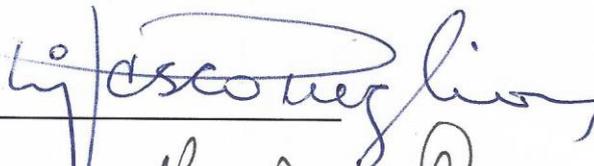
Santo André, 12 de Dezembro de 2018

**LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO
TRABALHO DE CONCLUSÃO DE CURSO COM O TEMA: "ROBÔ
COM SISTEMA GIROSCÓPIO E ACELERÔMETRO" DOS ALUNOS DO 6º
SEMESTRE DESTA U.E.**

BANCA

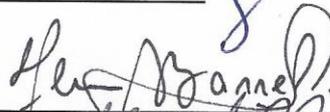
PRESIDENTE:

PROF. LUIZ VASCO PUGLIA

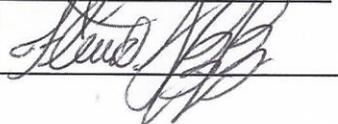


MEMBROS:

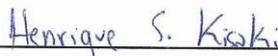
PROF. FLÁVIO AUGUSTO BARRELLA



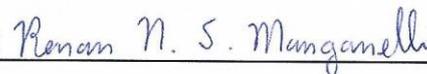
PROF. FLÁVIO JOSÉ BENTO BATISTA

**ALUNOS :**

HENRIQUE SATOSHI KISAKI



RENAN NUNES SIQUEIRA MANGANELLI



AGRADECIMENTOS

Primeiramente a Deus, por nos fortalecer e guiar em todas as decisões, não apenas acadêmicas, mas em todos os aspectos da vida.

Ao orientador, Professor Mestre Luiz Vasco Puglia, pelo auxílio, orientação, paciência e principalmente por acreditar em nós.

À Faculdade de Tecnologia de Santo André por nos proporcionar todas as ferramentas, necessárias ao crescimento profissional, em respeito tanto ao ambiente de estudo quanto ao corpo docente.

Aos professores que ministram as aulas do curso de Tecnologia em Mecatrônica Industrial, por nos proporcionar um ensino de excelência.

Aos familiares, por acreditar e nos apoiar em todos os momentos, incentivando e por vezes colaborando fisicamente e financeiramente com a conclusão do projeto.

Aos amigos e colegas da faculdade, por todo apoio que nos foi concedido.

Por fim, à dupla que sempre se manteve firme e unida, em todos os momentos, sendo eles bons ou ruins, enfrentando dificuldades e cumprindo desafios.

RESUMO

Sistemas naturais apresentam respostas características típicas. Um corpo esférico rolando sobre uma superfície côncava, sempre busca o centro desta mesma superfície, resultando, ao final, na estabilidade da sua posição. O desafio da ciência está na transição do plano, de côncavo para convexo, com a intenção de manter essa esfera parada sobre ele. Ao longo dos experimentos, os cientistas encontraram impossibilidades na atribuição de estabilidade a esse corpo, pois sua tendência é apresentar um sistema naturalmente instável. O desafio que se propõe, é o de promover um sistema de controle que permita, a esfera em questão, se encontrar estável nesta superfície. O exemplo da esfera, pode também ser aplicada também a outros conjuntos, como um foguete, uma bicicleta ou a um robô bípede, todos naturalmente instáveis. Neste estudo de caso, propõe-se a construção de um robô sobre duas rodas que permaneça constantemente na posição vertical. Promove-se este controle através da ação dos motores, que modificam o sentido e velocidade de rotação das rodas. Também, é necessário conhecer a inclinação deste robô para determinar a ação de resposta. Para que os objetivos sejam atingidos, é essencial modelar este sistema mecanicamente, gerando uma base física fundamental, que envolve conceitos como: momento de inércia, torque, momento angular e precessão. A partir destes elementos, houve uma busca pela mecatrônica necessária à estabilização do robô, procurando pelos sensores corretos e os programas mais eficazes. As respostas vieram na forma do módulo MPU6050 que é a união de um acelerômetro e um giroscópio. Foi decidido que o uso das tecnologias de impressão 3D aprimorou as qualidades mecânicas da estrutura do robô, além de introduzirem um novo espectro na gama de habilidades envolvidas nesse projeto.

Palavras chave: Acelerômetro, Giroscópio, MPU6050, Robô, Sensor.

ABSTRACT

Natural systems have typical characteristics. A spherical body rolling on a concave surface, always seeks the center of this surface, resulting in the end, the stability of its position. The challenge of science is in the transition of surfaces, from concave to convex, with the attempt to keep the sphere standing on this plane. Throughout the experiments, scientists have found impossibilities on promoting stability to the sphere, because it tends to a naturally unstable system. The challenge is then to promote a control system that allows the sphere in question to remain stable on this surface. The example of the sphere can also be applied to other systems, such as a rocket, a bicycle or a self-balancing robot, all naturally unstable. In this case study, it is proposed to construct a robot on two wheels that remains stationary in the vertical position. This control is promoted with action on the motors that modify the direction and rotational speed of the wheels. It is also necessary to know the inclination of this robot to determine the response. For achieving the goals, it's essential to mechanically model this system, creating a fundamental physical base, which involves concepts such as: moment of inertia, torque, angular momentum and precession. According with these elements, there was a search for the mechatronics needed to stabilize this robot, searching for the correct sensors and the most effective programs. The responses came in the form of the MPU6050 module which is the union of an accelerometer and a gyroscope. It was decided that the use of 3D printing technologies improved the mechanical qualities of the robot structure and introduced a new spectrum in the range of skills involved in this project.

Key words: Accelerometer, Gyroscope, MPU6050, Robot, Sensor.

LISTA DE EQUAÇÕES

EQUAÇÃO 1.....	22
EQUAÇÃO 2.....	23
EQUAÇÃO 3.....	25
EQUAÇÃO 4.....	25
EQUAÇÃO 5.....	27
EQUAÇÃO 6.....	38
EQUAÇÃO 7.....	38
EQUAÇÃO 8.....	54
EQUAÇÃO 9.....	56
EQUAÇÃO 10.....	56
EQUAÇÃO 11.....	57
EQUAÇÃO 12.....	57
EQUAÇÃO 13.....	57
EQUAÇÃO 14.....	60
EQUAÇÃO 15.....	60
EQUAÇÃO 16.....	60
EQUAÇÃO 17.....	60

LISTA DE FIGURAS

FIGURA 1. ORGANIZAÇÃO DO TRABALHO.....	19
FIGURA 2. EXEMPLO COM BARRAS	22
FIGURA 3. A ALAVANCA SERÁ ROTACIONADA EM VIRTUDE DA APLICAÇÃO DA FORÇA	23
FIGURA 4. TIPOS DE ESTABILIDADE	25
FIGURA 5. DIAGRAMA DE FORÇAS DE UM PLANO INCLINADO	25
FIGURA 6. VELOCIDADE ANGULAR	26
FIGURA 7. PRECESSÃO ATUANDO EM UMA RODA	27
FIGURA 8. GIROSCÓPIO NO PERFIL DE ILUSTRAÇÃO	28
FIGURA 9. MPU6050	30
FIGURA 10. EIXOS DE ORIENTAÇÃO DO SENSOR.....	31
FIGURA 11. ARDUINO UNO R3	32
FIGURA 12. ARDUINO IDE.....	32
FIGURA 13. PÁGINA DE INICIALIZAÇÃO DO FUSION 360.....	33
FIGURA 14. EXPORTANDO G-CODE DO FUSION 360.....	34
FIGURA 15. EXEMPLO DE FATIAMENTO	35
FIGURA 16. SISTEMA EM MALHA FECHADA	36
FIGURA 17. RESPOSTA AO DEGRAU DE PRIMEIRA ORDEM DE UM MOTOR DC	37
FIGURA 18. DIAGRAMA DE BLOCOS DE UM CONTROLADOR PROPORCIONAL + INTEGRAL + DERIVATIVO (PID).....	38
FIGURA 19. MODELO FÍSICO DO PÊNDULO INVERTIDO	39
FIGURA 20. PÊNDULO INVERTIDO CONTROLADO.....	40
FIGURA 21. IMAGEM EXPLODIDA DA ESTRUTURA.....	41
FIGURA 22. COMPONENTES E ENCAIXES	42

FIGURA 23. TESTE MOTOR DC	43
FIGURA 24. PRIMEIRA ESTRUTURA	44
FIGURA 25. SEGUNDA ESTRUTURA	45
FIGURA 26. PRIMEIRA PLACA	47
FIGURA 27. SEGUNDA PLACA.....	48
FIGURA 28. PLACA FINAL	49
FIGURA 29. SEGUNDO ROBÔ.....	50
FIGURA 30. ESTRUTURA FINAL	51
FIGURA 31. PROGRAMA DE ANÁLISE DA ZONA MORTA	52
FIGURA 32. VALOR DE PWM ANALISADO PELO MONITOR SERIAL.....	53
FIGURA 33. DRIVER PONTE H L298N.....	54
FIGURA 34. RESPOSTA ADQUIRIDA (AZUL) X RESPOSTA DESEJADA (LARANJA)	55
FIGURA 35. MÉTODO ADOTADO PARA ANÁLISE DE FREQUÊNCIA	56
FIGURA 36. GRÁFICO P.I.D.	59
FIGURA 37. DIAGRAMA FUNCIONAL.....	61

LISTA DE QUADROS

QUADRO 1. CONSEQUÊNCIA DA VARIAÇÃO DO ÂNGULO θ	24
---	----

LISTA DE SIGLAS E ABREVIATURAS

3D	Três Dimensões
CAD	<i>Computer Aided Design</i>
CAM	<i>Computer Aided Manufacturing</i>
CNC	Comando Numérico Computadorizado
DAC	Desenho Assistido por Computador
DMP	<i>Digital Motion Processor</i>
IDE	<i>Integrated Development Environment</i>
IMU	<i>Inertial Measurement Unit</i>
MEMS	<i>Micro-Electro-Mechanical System</i>
NiMh	Níquel-Hidreto Metálico
PCB	<i>Printed Circuit Board</i>
PCI	Placa de Circuito Impresso

LISTA DE SÍMBOLOS

μ	Coeficiente
θ	Ângulo Teta
ω_n	Velocidade Angular
\cos	Cosseno
d	Distância
F	Força
F_{at}	Força de Atrito
$F_{(s)}$	Função de Transferência em Malha Fechada
g	Aceleração da Gravidade
$G_{(s)}$	Função de Transferência em Malha Aberta
I	Momento de Inércia
L	Momento Angular
M	Massa
m_i	Massa Pontual
N	Força Normal
P	Força Peso/ Força da Gravidade
r_i	Distância em relação ao eixo de relação
T	Torque
V	Volts

SUMÁRIO

FICHA CATALOGRÁFICA.....	3
RESUMO.....	6
ABSTRACT	7
LISTA DE EQUAÇÕES	8
LISTA DE FIGURAS.....	9
LISTA DE QUADROS.....	11
LISTA DE SIGLAS E ABREVIATURAS	12
LISTA DE SÍMBOLOS	13
1 INTRODUÇÃO	17
1.1. MOTIVAÇÃO	17
1.2. OBJETIVOS.....	18
1.3. CONTRIBUIÇÕES ESPERADAS.....	18
1.4. ORGANIZAÇÃO DO TRABALHO	19
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 OBJETIVOS DO CAPÍTULO.....	21
2.2 INTRODUÇÃO AOS CONCEITOS FÍSICOS	21
2.2.1 MOMENTO DE INÉRCIA.....	22
2.2.2 TORQUE	23
2.2.3 CONCEITOS DE ESTABILIDADE	24
2.2.4 MOMENTO ANGULAR.....	26
2.2.5 PRECESSÃO.....	27
2.3 GIROSCÓPIO.....	27
2.3.1. EFEITO GIROSCÓPICO.....	28
2.4 ACELERÔMETRO	28
2.4.1 FUNCIONAMENTO DOS VARIADOS TIPOS DE ACELERÔMETROS.....	29
2.4.1.1 PIEZOELÉTRICOS:.....	29
2.4.1.2 PIEZORESISTIVOS:.....	29
2.4.1.3 CAPACITIVOS:	29
2.4.1.4 MEMS:.....	30

2.5 MPU6050.....	30
2.5.1 CARACTERÍSTICAS DO MPU6050.....	31
2.6 ARDUINO.....	31
2.7 IMPRESSÃO 3D.....	33
2.7.1 FUSION 360.....	33
2.7.2 ETAPA FINAL DESTE CAPÍTULO: MANUFATURA DA ESTRUTURA.....	34
2.7.2.1 CÓDIGO G.....	34
2.7.2.2 IMPRESSÃO 3D.....	34
2.8 TEORIA DE CONTROLE.....	35
2.9. PÊNULO INVERTIDO.....	39
3. METODOLOGIA.....	41
3.1 ESTRUTURA.....	41
3.2 CIRCUITO ELÉTRICO.....	43
3.3 TESTES DE VALIDAÇÃO.....	43
4. DESENVOLVIMENTO.....	44
4.1. CONFECÇÃO DA ESTRUTURA.....	44
4.2. ESQUEMÁTICO.....	46
4.3. PROTOTIPAGEM DA P.C.I.....	46
4.4. MELHORES RESULTADOS OBTIDOS.....	49
4.5. APLICAÇÃO DO PÊNULO INVERTIDO.....	51
4.6. ZONA MORTA.....	51
4.6.1. EXPLICAÇÃO.....	51
4.6.2. TESTES REALIZADOS.....	52
4.7. PONTE H L298N.....	53
4.8. CÁLCULOS PRIMORDIAIS.....	54
4.8.1. COMPENSADOR DE AVANÇO.....	57
4.8.1.1. COMPENSADOR DE AVANÇO VIA SOFTWARE.....	58
4.8.2. CÁLCULO DO P.I.D.....	59
4.8.3. APLICANDO VIA SOFTWARE.....	61
4.8.3.1. UTILIZAÇÃO DOS RECURSOS BÁSICOS DO MPU6050.....	62
4.8.3.2. DMP E UTILIZAÇÃO DOS RECURSOS AVANÇADOS DO MPU6050.....	64
5. CONSIDERAÇÕES FINAIS.....	66
5.1. CONCLUSÃO.....	66

5.2. SUGESTÃO PARA PROJETOS FUTUROS	67
6. REFERÊNCIAS BIBLIOGRÁFICAS	68
APÊNDICE A – ESTUDO MATEMÁTICO DO COMPENSADOR DE AVANÇO.....	72
APÊNDICE B – DOCUMENTAÇÃO DA ESTRUTURA PRINCIPAL.....	74
APÊNDICE C – ESQUEMA ELÉTRICO COM L293D.....	78
APÊNDICE D – ESQUEMA ELÉTRICO COM PONTE H L298N.....	79
ANEXO A – PROGRAMA P.I.D. COM MPU6050, REGINALD WATSON	80

1 Introdução

A partir dos primeiros estudos dos corpos celestes, a humanidade se depara com a natureza contra intuitiva dos movimentos de rotação, e desde então, a curiosidade tem carregado os pesquisadores em testes cada vez mais empíricos. Com os estudos de Galileo Galilei (1564-1642) diversos modelos matemáticos foram criados para representar a cinemática e a dinâmica dos corpos, desenvolvendo os conceitos fundamentais de inércia e propulsionando os fenômenos oscilatórios dos pêndulos. Foi a partir das profundas pesquisas de Johann Gottlieb Bohnenberger (1765-1831) que a dinâmica dos corpos em rotação recebeu seus primeiros modelos reais, na forma de um equipamento chamado Giroscópio.

Nas palavras de Ervin S. Ferry *“A girodinâmica é o fruto da genialidade matemática, invocado pelos físicos na produção de milagres maiores do que os feitos do mitológico Dédalo [...]”* (Ferry,1933), o que pode ser observado claramente na cada vez maior, influência dos equipamentos giroscópicos na engenharia atual. Graças ao advento dos sistemas posicionais, foi possível desenvolver veículos automóveis com uma grande estabilidade, aumentando a eficiência e reduzindo riscos, no entanto, o maior ganho se encontra na capacidade de produzir máquinas mais inteligentes.

1.1. Motivação

Sabe-se que junto à globalização, houveram diversos acontecimentos, tais como a integração social, política e econômica entre a maioria dos países, porém, junto a este evento, surgiu a tão conhecida Era do Conhecimento, ou seja, isso fez com que milhares de pessoas passassem a ter acesso às informações; Em consequência a isso, houve um aumento desproporcional no desenvolvimento, principalmente em relação às tecnologias, que em tão pouco tempo conseguiram esse avanço em diversas coisas que possivelmente o ser humano jamais esperava ver de fato (Lastres and Albagli, 1999).

Logo, isso não apenas atraiu as pessoas para seguirem o caminho de um “mundo mais tecnológico” ou com maior acesso a essas informações por assim dizer. Essa tecnologia se tornou cada vez mais parte do cotidiano de cada um, resultando em uma área de atuação muito disputada, onde apenas os mais qualificados para tal conseguem ingressar em um ramo profissional.

Portanto, uma das grandes motivações para este trabalho, foi baseada na ideia de adentrar a essa Era do Conhecimento, através do estudo com sistemas cujo qual utilizam-se de sensores giroscópicos e acelerométricos; e conseguir maior experiência para competir com o mercado de trabalho, provando assim, a qualificação esperada para isso.

1.2. Objetivos

O presente trabalho busca a integração dos conceitos acadêmicos envolvidos na área de controle para estruturas naturalmente instáveis. Espera-se que, como resultado do estudo aprofundado, de todo o conteúdo físico-matemático, tal como os conhecimentos advindos da mecatrônica relacionada, sejam adquiridos com propriedade e domínio. Ao final, pretende-se que a devida aplicação se efetue na forma de um mecanismo de auto balanço.

1.3. Contribuições Esperadas

Para este tópico aplica-se a ideia esperada no item anterior (1.2.), cujo qual encontram-se os objetivos de forma bastante sucinta, portanto, as contribuições esperadas são:

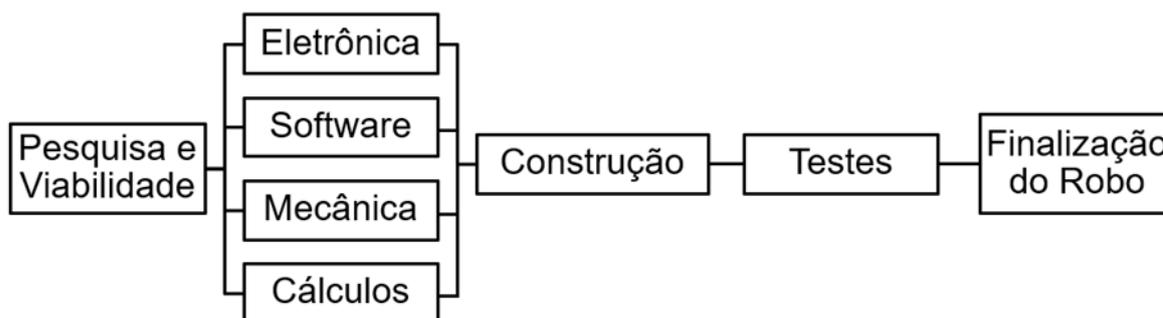
- a) Revisão atualizada não somente referente ao sensor em si, mas também, quanto aos conceitos teóricos que serão descritos ao longo da monografia;
- b) Utilização das tecnologias de *softwares CAD* e a impressão 3D;
- c) Utilização de *softwares*, cujos não foram definidos até o momento, porém sabe-se que estes servirão de auxílio para a implementação de controle na estabilidade de auto balanço citada anteriormente, por meio da comunicação de um Arduino com as linhas de código que serão geradas pelo uso do *software*;

- d) Reconhecimento do controle necessário, para aplicar ao tipo de estrutura a ser estudada.

1.4. Organização do Trabalho

O trabalho proposto, almeja a construção de um robô de auto balanço, e para tal, divide-se em algumas etapas, como representado na figura 1 a seguir:

Figura 1. Organização do Trabalho



Fonte: Própria.

O trabalho será apresentado segundo o avanço das pesquisas, iniciando-se com um amplo embasamento teórico, que precede o real escopo do projeto, e, portanto, comporta uma vasta gama conhecimentos na intenção de melhor transparecer a execução em si.

Durante a fundamentação teórica, buscou-se abranger todos os conceitos físico-matemáticos envolvidos na aplicação, além de introduzir temas referentes à composição, sendo eles: hardwares, detalhando os componentes eletrônicos utilizados, e softwares incluídos para controle.

O capítulo de metodologia tem como função levantar as premeditações referentes à: estrutura mecânica, conceitos fundamentais do circuito e proposição de testes. Ele demonstra os cuidados tomados na etapa de planejamento, assim como as preocupações e resultados iniciais de testes isolados.

O desenvolvimento, momento no qual todos os componentes foram reunidos e efetivamente colocados à prova. Este capítulo contém dados técnicos e todos os testes e resultados obtidos, relatando problemas e dificuldades, juntamente com suas soluções e lições aprendidas. Ao longo desta fase, percebeu-se a necessidade de expandir a conceituação teórica a respeito de tópicos pertinentes, dos quais jamais se teorizou haver relações.

Por fim, o capítulo de conclusão, apresenta um momento de reflexão sobre os principais pontos, além de levantar discussões sobre melhorias futuras. Esta parte possui a intenção de agregar crescimento e providenciar um fechamento formal para o trabalho.

2 Fundamentação Teórica

2.1 Objetivos do Capítulo

Para o futuro desenvolvimento do robô, é de suma importância a construção de um forte embasamento teórico, relacionado à: princípios físicos, matemáticos, eletrônicos e computacionais. A partir destes conceitos, será possível dimensionar os materiais, componentes e programação necessária a produção da estrutura, tal como os escopos de tempo e investimentos financeiros fundamentais à estabilidade dela.

O intuito deste capítulo, é apresentar todo o conteúdo teórico inerente aos sensores giroscópico e acelerométrico, que serão responsáveis por captar perturbações externas e corrigi-las, buscando a máxima estabilidade possível. Esse objetivo será alcançado através de um sinal que será enviado, interpretado e atuará nos motores, realizando a devida correção de controle, em resposta ao desbalanceamento causado por forças externamente aplicadas.

Quanto a hardware, serão aprofundadas as características do sensor MPU6050 e a sua interação com o microcontrolador Arduino. E por fim, serão levantadas as vantagens de se utilizar a técnica da impressão 3D para a estrutura do robô.

2.2 Introdução aos conceitos físicos

A estabilização de um corpo tanto verticalmente quanto horizontalmente, exige conceitos que englobam áreas da cinemática e da dinâmica que são respectivamente: O estudo que determina a posição, velocidade e aceleração de um corpo, com base no seu movimento; E o estudo que define as causas desses movimentos e todas suas possíveis repercussões.

Serão apresentados os princípios físicos mais relevantes para a definição de sistema estável, sendo eles: Momento de inércia, torque, força gravitacional, momento angular e precessão.

2.2.1 Momento de Inércia

Inércia é a propriedade de um corpo que exige a aplicação de uma força para que ocorra uma aceleração linear no momento deste corpo. Tudo que possui inércia é chamado matéria e a quantidade total de inércia em um corpo é a somatória da inércia em todas as suas partículas (FERRY, 1933).

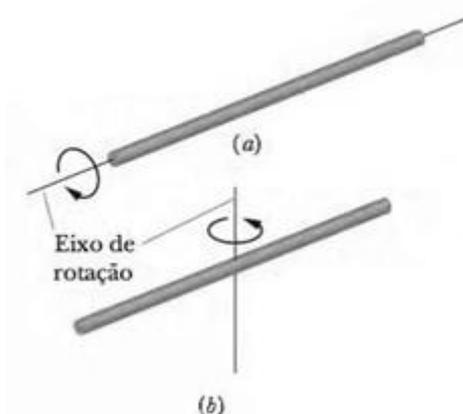
Momento de inércia (I) é descrito como a resistência existente em um corpo à rotação, exigindo a aplicação de um torque para que ocorra uma mudança em sua velocidade angular (equação 1).

Sendo matematicamente definida como o produto da somatória de todas as massas pontuais em um eixo, pela distância ao quadrado deste mesmo eixo, percebe-se que quanto maior for essa distância, mais força será necessária para realizar o movimento de rotação (RESNICK, HALLIDAY and WALKER et al.).

$$I = \sum m_i \cdot r_i^2 \quad \text{Equação 1}$$

“É mais fácil fazer girar uma barra comprida em torno (a) do eixo central (longitudinal) do que (b) de um eixo passando pelo centro e perpendicular à maior dimensão da barra. A razão para esta diferença é que a distribuição de massa está mais próxima do eixo de rotação em (a) do que em (b).”.

Figura 2. Exemplo com barras



Fonte: Livro – Fundamentos da Física – Halliday – 9ª Edição.

2.2.2 Torque

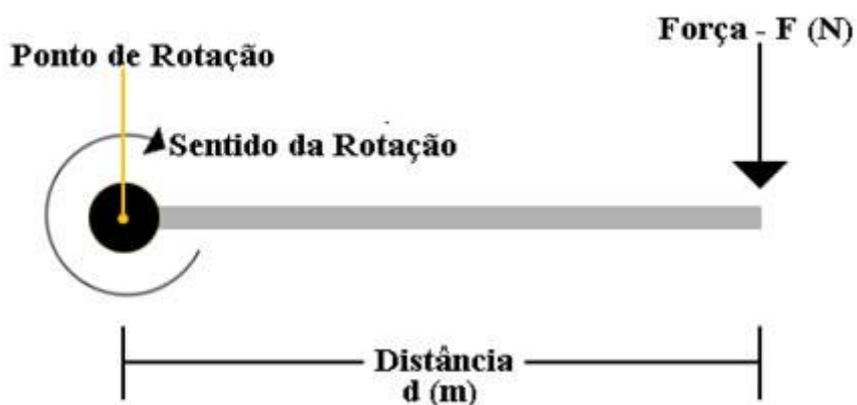
A qualquer efeito que cause, ou tenda a causar uma variação no momento angular de um corpo, é chamado torque. O torque pode ser definido como duas forças opostas e paralelas de igual magnitude separadas por uma distância (d) perpendicular às duas forças (FERRY, 1933).

$$T = F \cdot d \cdot \text{sen}\theta, \quad \text{Equação 2}$$

Onde: F = Força
 d = Distância
 $\text{sen}\theta$ = Seno do ângulo

Sendo assim, torna-se perceptível que quanto mais próximo do ponto de rotação, maior será o torque necessário para vencer o momento de inércia (figura 3).

Figura 3. A alavanca será rotacionada em virtude da aplicação da força



Fonte: mundoeducacao.bol.uol.com.br

O quadro 1, apresenta a variação gerada pelo ângulo θ conforme o movimento da alavanca em torno do ponto de rotação:

Quadro 1. Consequência da variação do ângulo θ

Ângulo	Torque horário	Torque Anti-horário
0°	0	0
90°	+	-
180°	0	0
270°	-	+
360°	0	0

Fonte: Própria.

A partir dessa tabela, torna-se possível analisar que, quando o ângulo θ for igual a 90°, resultará no torque máximo. Além disso, observa-se que, quando invertido o sentido de rotação, ocorrerá também uma alteração no sinal do torque.

2.2.3 Conceitos de estabilidade

É possível definir o nível de estabilidade de um corpo a partir da quantidade de trabalho necessário para retorná-lo ao ponto de referência. Segundo Ferry, a estabilidade de um corpo pode ser dividida em: estável, caso após perturbação, ele recupere sua posição original; instável, caso após perturbação, ele se afaste da posição original; neutro, caso não exista nenhuma tendência (FERRY, 1933).

Para que haja estabilidade na estrutura é necessário que o corpo de teste possa produzir trabalho suficiente para neutralizar as forças desestabilizadoras que agem sobre ele. Supondo uma superfície plana sobre o efeito do campo gravitacional da terra, será necessário que:

- 1° O projeto detecte a perturbação.
- 2° O projeto detecte o ângulo da perturbação.
- 3° O projeto calcule o trabalho necessário para retornar ao estado estacionário de equilíbrio.

Logo, conforme descrito pela segunda lei de Newton, o projeto deve compensar o desequilíbrio causado pela força resultante, que será descrita como o produto de sua massa pela aceleração, no cenário descrito, a gravidade, como representado pela figura 4.

$$P = M \cdot g$$

Equação 3

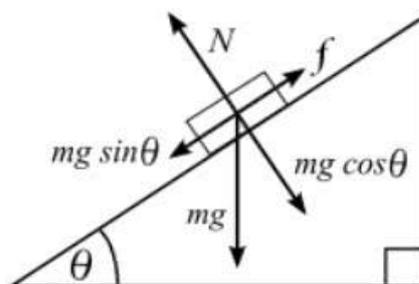
Onde: P = Peso
M = Massa
g = Aceleração da gravidade

Figura 4. Tipos de estabilidade



Em ambientes que contenham superfícies inclinadas, ocorre a decomposição das forças em 2 componentes perpendiculares, que correspondem respectivamente ao produto do $\sin\theta$, e $\cos\theta$ pela força peso P , conforme descrito na imagem (Figura 5), onde θ representa o ângulo de inclinação da superfície.

Figura 5. Diagrama de forças de um plano inclinado



Fonte: todamateria.com.br

Caso o objeto deslize sobre o plano, isso significa que, a força Normal, não anula completamente a força peso distribuída, deixando a componente $mg \cdot \sin\theta$ em oposição à uma terceira força, chamada de Atrito, definida matematicamente como:

$$F_{at} = \mu \cdot N$$

Equação 4

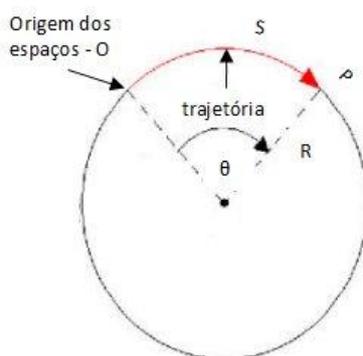
Onde: F_{at} = Força de atrito
 μ = Constante de atrito
 N = Força normal

Caso a superfície não possua atrito suficiente para a neutralização da força peso, então o sistema dinâmico se encontrará em aceleração constante a favor da componente $mg \cdot \sin\theta$, demonstrando um sistema instável que se afasta do estado inicial (RESNICK, HALLIDAY and WALKER et al.).

2.2.4 Momento Angular

Ao considerarmos um ponto (P) na superfície lateral de um corpo circular girante, podemos entender que, ao acompanhar o deslocamento espacial de P no tempo, a direção do vetor raio R se altera, gerando uma variação angular. À variação deste ângulo no tempo nomeia-se velocidade angular (Figura 6).

Figura 6. Velocidade angular



Fonte: infoescola.com.br

Logo, pode-se entender que momento angular (L) é a quantidade de movimento rotacional associada a um corpo. Essa quantidade é medida pela massa do corpo, pela distância de seu raio e pela velocidade linear. No entanto, para corpos com eixos fixos, o momento angular pode ser compreendido como o produto da velocidade angular (ω) pelo momento de inércia (I), resultando na quantidade total de movimento necessário para rotacionar um corpo.

$$L = I \cdot \omega$$

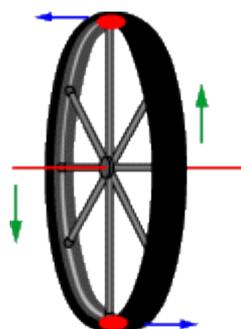
Equação 5

2.2.5 Precessão

A precessão é um movimento que ocorre em resposta à uma perturbação na direção do eixo de rotação de um corpo. É facilmente observada no movimento de uma bicicleta, onde a ação da gravidade aplica um segundo momento angular, que através de uma soma vetorial, resulta em um terceiro vetor em uma direção perpendicular as duas outras velocidades.

Este movimento pode ser calculado com base na força externa que causa a perturbação e a própria ação da gravidade, causando uma variação no momento angular. Desta forma, pode-se afirmar que, conforme a velocidade de rotação do objeto diminui, maior é sua velocidade de precessão. São grandezas inversamente proporcionais (figura 7).

Figura 7. Precessão atuando em uma roda



©2000 How Stuff Works

Fonte: science.howstuffworks.com

2.3 Giroscópio

O giroscópio (figura 8) é um sensor composto por uma roda livre que realiza um movimento giratório sobre um eixo de revolução suspenso. Todo movimento externo resultará em uma resposta imediata pelo momento de inércia, conforme descrito pela terceira e primeira lei de Newton respectivamente, que atribuirá toda a perturbação em cima dos eixos não-girantes. Sendo assim, a direção da roda girante não se alterará, se tornando uma referência para as medições de variáveis contínuas como

por exemplo a posição angular em que um corpo se encontra no espaço. Esta propriedade mantém-se quando há ausência de forças atuando diretamente no corpo central, porém tende a mudar de direção, quando esta se aplica sobre ele.

Figura 8. Giroscópio no perfil de ilustração



Fonte: pt.dreamstime.com

2.3.1. Efeito Giroscópico

O efeito de giroscópio, é perceptível quando ao aplicar uma perturbação angular sobre a roda livre em movimento giratório, como descrito anteriormente, oferece uma resistência. Segundo a primeira lei de Newton, ao incidir uma força sobre um corpo, será perceptível a presença de uma força oposta resistente à aceleração. Este fenômeno denomina-se **momento de Inércia**, que ao confrontar a aplicação de um torque externo, por somatória de vetores, gera um movimento de variação no eixo de rotação. Este fenômeno por sua vez, denomina-se **Precensão**.

2.4 Acelerômetro

Para que se mantenha a estabilidade de um sistema, mostra-se necessário a constante observação das variações posicionais da estrutura, que como analisado previamente, ocorrerão majoritariamente no momento angular.

Atentando-se a necessidade de uma reação rápida e coerente, não será medido apenas o deslocamento angular, mas também, o tempo em que esse deslocamento ocorre, desencadeando uma resposta mais forte para variações mais bruscas.

Com esse objetivo será utilizado um acelerômetro que pode atuar em até 3 eixos de orientação X, Y e Z, percebendo assim, mudanças na inclinação, rotação, colisão, aceleração e vibração da estrutura (Silva and Silva, 2011).

2.4.1 Funcionamento dos variados tipos de acelerômetros

Os acelerômetros podem ser encontrados de diversas formas, dentre eles, os mais comuns são os brevemente citados a seguir (Rodrigues et al., 2010):

2.4.1.1 Piezoelétricos:

O funcionamento desse tipo de acelerômetro, se baseia na variação de movimento, onde a massa gera uma pressão mecânica interna que reage produzindo uma carga elétrica proporcional ao seu deslocamento. Esse fenômeno se dá, graças a propriedade do material, que se denomina “efeito piezoelétrico”.

2.4.1.2 Piezoresistivos:

Semelhante ao funcionamento do acelerômetro piezoelétrico, esse gera uma variação na resistência de acordo com o movimento gerado, tornando-se assim, possível deduzir a aceleração. O maior problema em aplicar sensores desse tipo, se dá no seu grande consumo de energia, além de uma alta detecção de ruídos e não suportarem altas temperaturas.

2.4.1.3 Capacitivos:

Nos sensores capacitivos, a massa se encontra entre duas placas, assemelhando-se a um capacitor. Dessa forma, é possível detectar a capacitância de acordo com a variação de posição. Sua performance é alta ao se tratar de: tamanho da banda, baixo consumo de energia, sensibilidade, e baixa índice detecção de ruídos.

Esse tipo de sensor apresenta uma grande desvantagem quando em contato com campos magnéticos, por isso aplica-se uma blindagem, garantindo assim sua devida isolamento.

2.4.1.4 **MEMS:**

Os do tipo *MEMS* (*Microelectromechanical System*), incorporam às classes de acelerômetros, reduzindo seu custo, porém há uma curta banda de frequência, além de sofrer com a detecção de ruídos. Entretanto, mesmo com essas desvantagens, sua aplicação continua sendo a mais viável, devido a soluções desenvolvidas atualmente quanto à instabilidade e a baixa frequência em que trabalhava.

2.5 **MPU6050**

O sensor definido para este trabalho, foi o *MPU6050* (figura 9), em função das suas dimensões, baixo consumo de energia, alta precisão e repetibilidade, alta tolerância à choques, e grande desempenho quanto à programação em aplicações específicas (INVENSENSE, 2013).

Portanto para este trabalho, o acelerômetro juntamente com o giroscópio, enviarão ao microcontrolador, um sinal para que o atuador da estrutura corrija o erro quando detectado uma aceleração ou vibração, mantendo-o estável.

Os pontos que devem ser analisados ao escolher o modelo do sensor, é quanto aos intervalos de medidas, sensibilidade, sensibilidade transversal, intervalos de frequência, número de eixos em que atua, massa com que trabalha e condições ambientais.

Figura 9. MPU6050



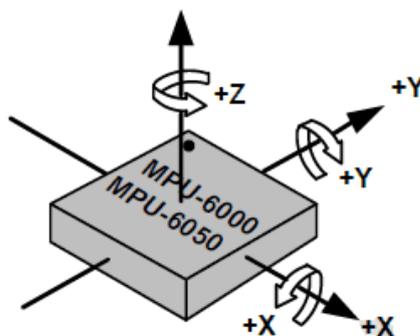
Fonte: martaduino.com.br

2.5.1 Características do MPU6050

O MPU6050 é um sensor do tipo MEMS que possui 6 eixos de rastreamento de perturbação, sendo 3 do giroscópio e 3 do acelerômetro (figura 10), possuindo também um *Digital Motion Processor*[™] (DMP) utilizado para cálculos complexos de tratamento dos valores de saída dos sensores. Aceita conexão com outros dispositivos, através de uma comunicação que se utiliza do barramento I²C.

Para rastreamento de precisão, tanto lento quanto rápido, o sensor detém um giroscópio de sensibilidade programável, com intervalos de escala ± 250 , ± 500 , ± 1000 e ± 2000 graus por segundo e um acelerômetro de $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$ (INVENSENSE, 2013).

Figura 10. Eixos de orientação do sensor



Fonte: INVENSENSE, 2013, p.21

A Invensense revolucionou a fundação do projeto ao reduzir as dimensões do sensor para 4x4x0.9mm, promovendo a mais alta performance, baixos ruídos e menor custo. Ele apresenta grande robustez, aguentando 10.000 g de choque (INVENSENSE, 2013).

2.6 Arduino

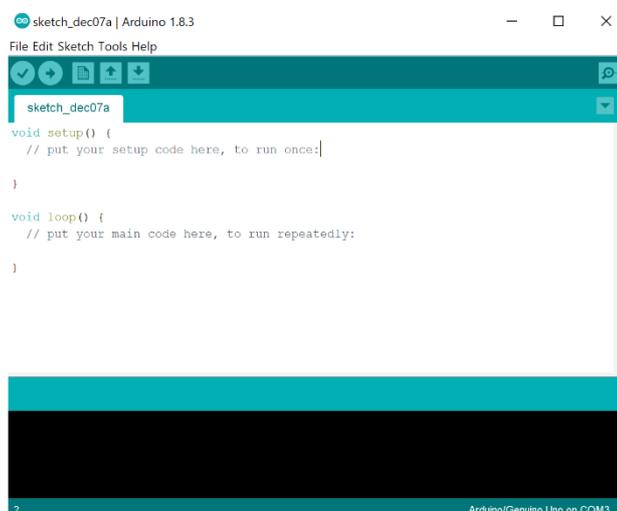
O Arduino (figura 11) é um hardware *open-source*, fácil de ser programado e adaptável de acordo com a necessidade. Para programá-lo é necessário a utilização do software Arduino (IDE) (figura 12) cuja plataforma é do tipo *Wiring* (BANZI, 2014).

Figura 11. Arduino UNO R3



Fonte: store.arduino.cc

Figura 12. Arduino IDE



Fonte: support.office.com

Segundo o próprio site da Arduino, essa ideia surgiu dos pesquisadores Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, e David Mellis no ano de 2000, no intuito de aprender a programar de uma forma mais fácil, utilizando a linguagem de programação C/C++ com algumas alterações. Sua primeira implementação no mundo acadêmico foi em 2005, para ajudar estudantes de design que não possuíam nenhum conhecimento prévio em programação, a criarem um projeto que conectasse o mundo digital ao mundo físico (Arduino, 2018).

Essa placa será responsável, por interpretar todas as mensagens enviadas pelo sensor MPU6050, e executar as devidas programações que serão previamente confi-

guradas para controlar o atuador da estrutura que será responsável por manter a estabilidade dela, ou seja, resumidamente, realizará as comunicações entre o sensor e o atuador.

2.7 Impressão 3D

O robô será constituído em grande parte da sua estrutura por peças modeladas virtualmente no *Fusion 360* e construídas concretamente no processo de impressão em 3D.

2.7.1 Fusion 360

O *Fusion 360* (figura 13), é um *software CAD (Computer-Aided Design)*, ou de Desenho Assistido por Computador (DAC), desenvolvido pela *Autodesk*, que é uma empresa que como propriamente dito:

“Criam softwares para pessoas que criam coisas. Nos setores de manufatura, arquitetura, construção, mídia e entretenimento, a Autodesk dá a você o poder de criar qualquer coisa.” – (Autodesk, 2018)

Este *software* servirá de auxílio para toda parte da modelagem digital, antes de enviar os comandos ao microcontrolador da impressora 3D.

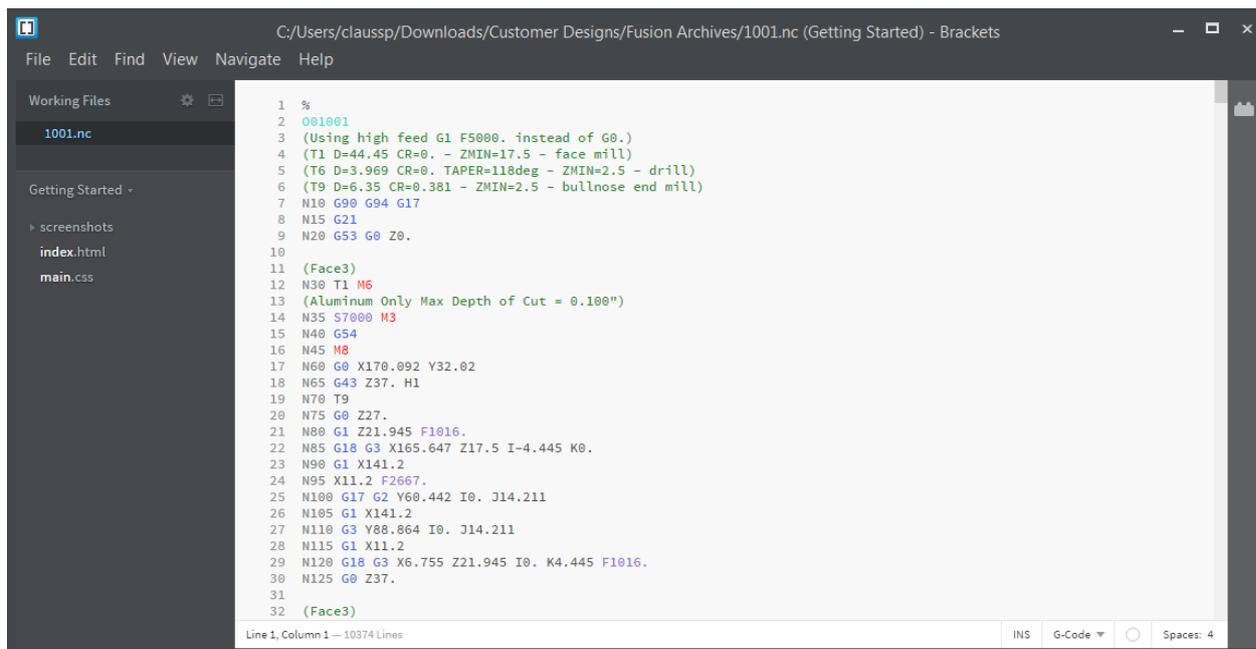
Figura 13. Página de inicialização do Fusion 360



Fonte: *Software Fusion 360*.

Os “comandos” citados anteriormente, são gerados pelo próprio *software* em código G (figura 14), sendo este o mesmo encontrado em máquinas CNC (Comando Numérico Computadorizado), tendo assim a vantagem de adquirir as informações de modelos visuais 3D criados em plataformas do tipo *CAD* (Miranda, 2009).

Figura 14. Exportando G-code do Fusion 360



```

1 %
2 O01001
3 (Using high feed G1 F5000. instead of G0.)
4 (T1 D=44.45 CR=0. - ZMIN=17.5 - face mill)
5 (T6 D=3.969 CR=0. TAPER=118deg - ZMIN=2.5 - drill)
6 (T9 D=6.35 CR=0.381 - ZMIN=2.5 - bullnose end mill)
7 N10 G90 G94 G17
8 N15 G21
9 N20 G53 G0 Z0.
10
11 (Face3)
12 N30 T1 M6
13 (Aluminum Only Max Depth of Cut = 0.100")
14 M35 S7000 M3
15 M40 G54
16 M45 M8
17 M60 G0 X170.092 Y32.02
18 M65 G43 Z37. H1
19 N70 T9
20 N75 G0 Z27.
21 N80 G1 Z21.945 F1016.
22 N85 G18 G3 X165.647 Z17.5 I=-4.445 K0.
23 N90 G1 X141.2
24 N95 X11.2 F2667.
25 N100 G17 G2 Y60.442 I0. J14.211
26 N105 G1 X141.2
27 N110 G3 Y88.864 I0. J14.211
28 N115 G1 X11.2
29 N120 G18 G3 X6.755 Z21.945 I0. K4.445 F1016.
30 N125 G0 Z37.
31
32 (Face3)

```

Fonte: knowledge.autodesk.com

2.7.2 Etapa final deste capítulo: Manufatura da Estrutura

2.7.2.1 Código G

Antes de qualquer aprofundamento sobre o processo em si, na impressora 3D, para este capítulo será necessário compreender o princípio básico do que é o código G, gerado pelo *software*, *Fusion 360*.

Então, interpreta-se que código G, é uma sequência de comandos gerados em um plano tridimensional (X, Y e Z), com inúmeras aplicações nas industriais. A abrangência de sua utilização, se dá graças à sua eficiência no processamento, precisão, e facilidade de uso (Azevedo, 2013).

O código G, é normalmente gerado para sistemas *CAM* (*Computer Aided Manufacturing*, ou seja, é o sistema aplicado para a modelagem física do produto, com o auxílio direto ou indireto de um o comando computacional) por um sistema *CAD*, que nesse caso se aplica ao *Fusion 360*.

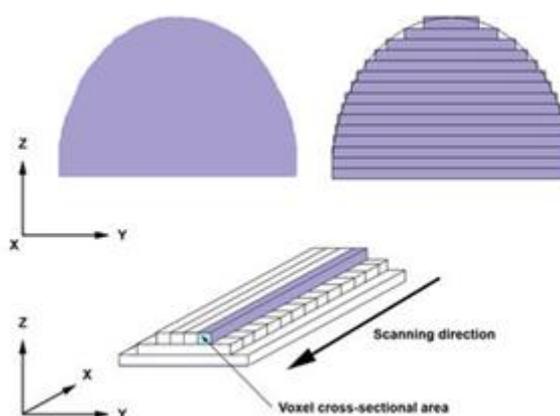
2.7.2.2 Impressão 3D

A impressão 3D, é um processo de sistema *CAM*, que gera um corpo sólido, na fusão e deposição de um filamento plástico.

A partir dos comandos gerados pelo *Fusion 360* em código G, adquire-se a orientação do trajeto que o filamento percorrerá para formar a peça final (figura 15). O processo de aquisição dessas coordenadas, denomina-se de fatiamento, devido à subdivisão em camadas, que ocorre a partir da peça modelada (Azevedo, 2013).

A altura da subdivisão, se dá entre 1 e 2 décimos, sendo que quanto menor a altura configurada para impressão, melhor será o acabamento da peça, e em consequência disso, maior será o tempo de execução.

Figura 15. Exemplo de fatiamento



Fonte: BH Group.

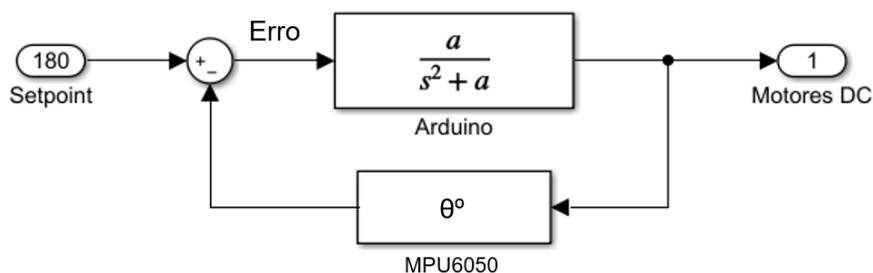
2.8 Teoria de Controle

A teoria de controle é um conceito muito antigo, mas o começo da análise formal da matéria é creditado ao físico James Clerk Maxwell pelas suas observações no Governador centrífugo, e desde então seus conceitos tem se expandido para todas as áreas. Segundo Mayr Otto, “Um sistema de controle realimentado, é um sistema que tende a manter uma relação pré-estabelecida entre uma variável à outra, por meio da comparação de funções e utilizando a diferença como método de controle”.

Controlar um sistema dinâmico, isto é, em constante mudança, exige uma comparação entre tudo que alimenta dado sistema (Input) e tudo que o deixa (Output), à essa comparação confere-se o nome de *Erro*, e seu valor é obtido através de uma realimentação por intermédio de um sensor competente. Quando isso acontece, todo

os integrantes do conjunto recebem o nome de Malha fechada, que pode ser representado como na figura 16:

Figura 16. Sistema em Malha Fechada



Fonte: Própria

Um controlador, busca aproximar a saída de um determinado valor, chamado *Setpoint (Referência)*, por meio de compensadores que alteram os valores de entrada considerando a mudança natural atribuída pelo sistema.

Essa característica natural pode ser escrita através de um modelo matemático, assumindo a forma de uma Função de Transferência. A forma mais comum de se lidar com as equações de controle, é por meio da transformada de Laplace, ferramenta que caminha entre os domínios do tempo e da frequência, alternando entre eles buscando a redução da complexidade convertendo equações diferenciais em equações algébricas.

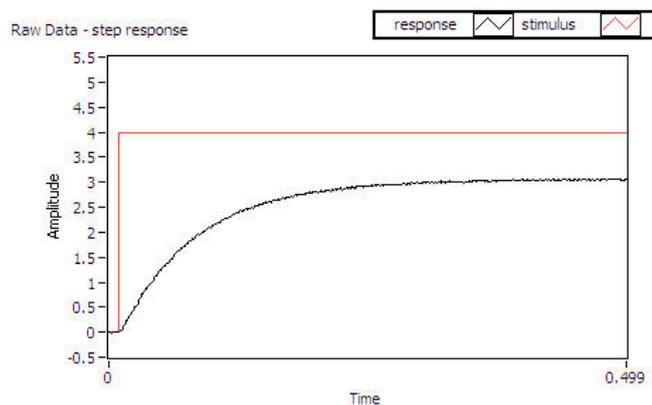
2.8.1 Função de Transferência

Muitos sistemas possuem um comportamento previsível, cuja entrada e a saída respondem proporcionalmente. Esses sistemas abrem uma brecha para uma representação matemática linear e invariante no tempo com resultados altamente compatíveis com a realidade.

Desta forma, ao estabelecer um sinal de entrada real ou discreto, mas invariante, pode-se traçar uma relação para com sua resposta, alterada pelas suas características físicas. Essas funções podem apresentar diferentes níveis de resposta de acordo com sua complexidade e ordem.

Essas funções podem ser analisadas mais a fundo através do estudo de sua resposta temporal, como mostrado na figura 17:

Figura 17. Resposta ao degrau de primeira ordem de um motor DC



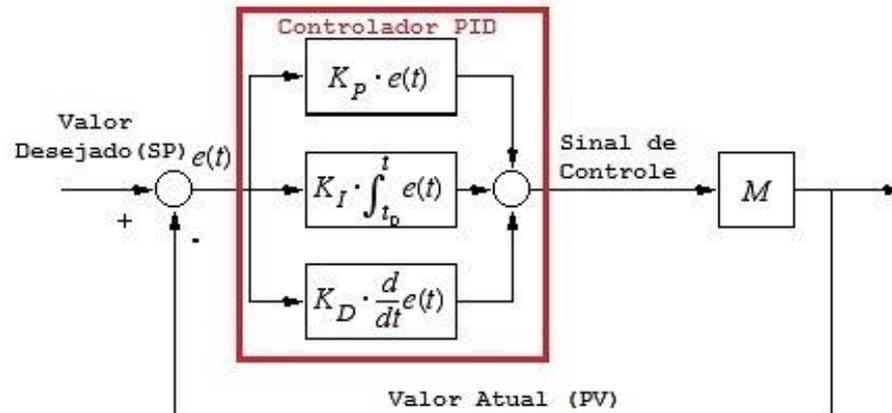
Fonte: ni.com

A teoria de controle se fundamenta da manipulação da resposta temporal de acordo com a necessidade apresentada pelo projeto, corrigindo aspectos indesejados, como sobressinal, alto tempo de acomodação ou erros de estado estacionário. Essa correção vem na forma de compensadores, que alocam alterações específicas dentro do sistema dinâmico.

2.8.2 Compensadores PID

Existem diversas técnicas para se compensar um sistema, as quais se baseiam na manipulação de ganhos. A mais utilizada devido ao seu amplo alcance é chamada de PID- Controle de ação Proporcional, Integral e Derivativo. A figura 18 apresenta um degrau do compensador PID em sua realização.

Figura 18. Diagrama de Blocos de um Controlador Proporcional + Integral + Derivativo (PID)



Fonte: embarcados.com.br

E pode ser matematicamente representado pela equação 6:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad \text{Equação 6}$$

Onde K_p representa a constante de ganho proporcional, K_i a constante de ganho integral, K_d a constante de ganho derivativo, e o erro, t o tempo e τ o tempo de integração.

Através da transformada de Laplace obtém-se uma visão holística mais compacta e abrangente, apresentada na equação 17:

$$L(s) = K_p + K_i/s + K_d s \quad \text{Equação 7}$$

A ação proporcional atua diretamente sobre o erro, aumentando o sinal de saída em diversas vezes. Apesar de muitas vezes ser suficiente para obter a resposta desejada, um grande valor de K_p pode gerar desequilíbrio na forma de sobressinal e erro de valor final.

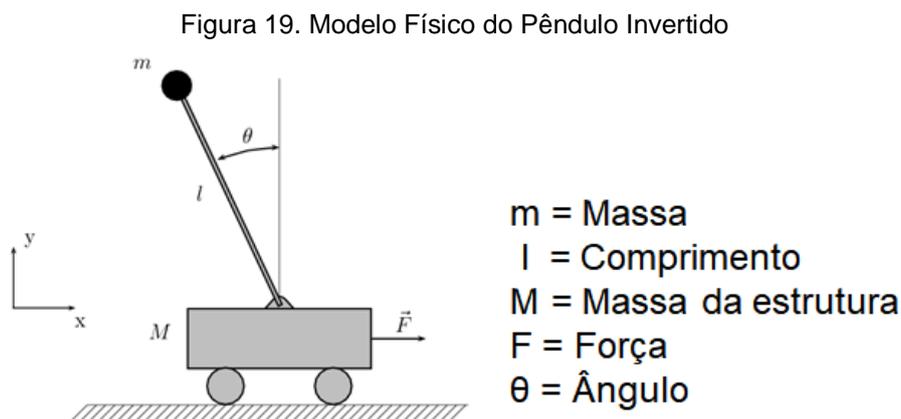
A ação integral, por sua vez, trabalha em relação a quantidade de erro 'acumulado', ou seja, acelerando a resposta caso o erro seja muito grande, e retardando caso o erro seja muito pequeno. Para sistemas discretos, essa correção é aplicada em intervalos constantes chamados de tempo integral. Caso este valor seja muito pequeno, a correção será lenta, no entanto, caso seja muito alta, também levará a instabilidade.

E por último, a ação derivativa que atua sobre a velocidade de variação do erro, antecipando a correção. Também trabalha com em intervalos regulares promovendo grande estabilidade, no entanto, não é recomendado para sistemas de grande oscilação.

2.9. Pêndulo Invertido

Uma das concepções iniciais era a necessidade de basear o projeto em um estudo de pêndulo invertido, onde a teoria de controle seria aplicada com maior propriedade, graças a instabilidade natural semelhante à do robô.

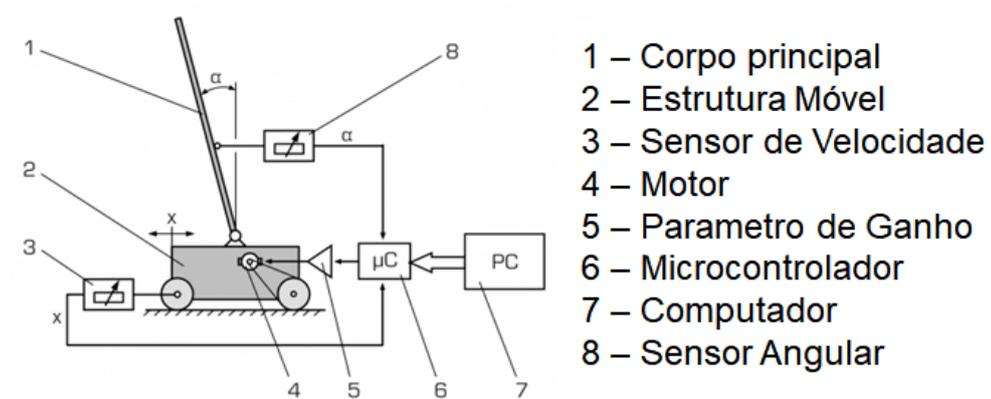
O pêndulo invertido consiste em uma barra, cujo centro de gravidade é localizado no topo da estrutura, através de uma concentração de massa, contribuindo para um atraso na velocidade angular, em outras palavras, atrasando a queda, e facilitando a medição da variação. A compensação ocorre por meio de atuadores horizontais em sua base (no oposto extremo da barra), como pode ser observado no clássico modelo de controle, mostrado a seguir (figura 19):



Fonte: Katsuhiko Ogata

Com essas informações, busca-se desenvolver um robô de estrutura naturalmente instável, cujo funcionamento baseia-se no modelo de pêndulo invertido com o sensor MPU6050 operando no topo e motores DC atuando em sua base, a partir dos valores processados pelo Arduino, de acordo com a figura 20.

Figura 20. Pêndulo invertido controlado



Fonte: Gunt.de

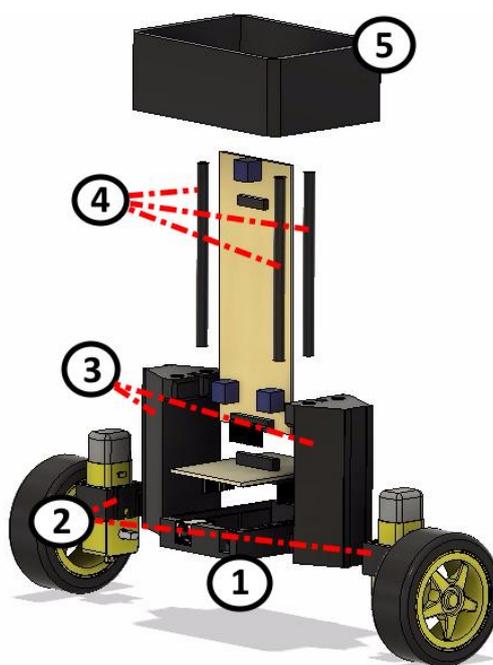
3. Metodologia

3.1 Estrutura

A modelagem da estrutura se iniciou com a premissa de sustentar a placa de circuito impresso, logo, absorvendo outras funções, como manter o apoio do sistema em um único eixo motorizado, e concentrar a maior quantidade de massa no topo.

Com o advento da impressão 3D, e por intermédio do software de modelagem *Fusion 360*, as primeiras peças começaram a ser desenvolvidas (figura 21).

Figura 21. Imagem explodida da Estrutura.

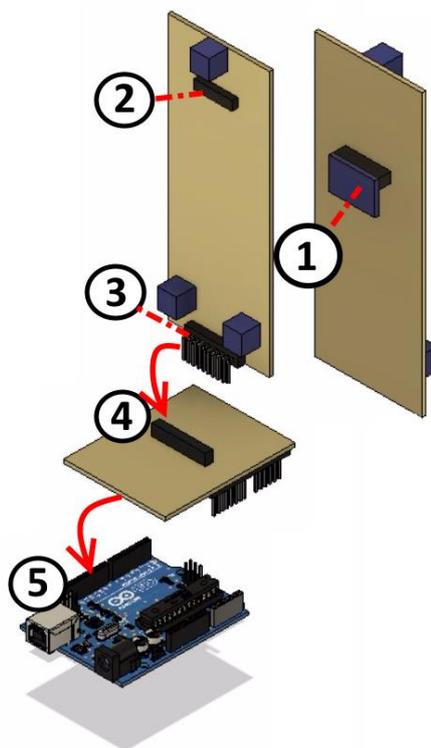


Fonte: Própria.

A peça central (figura 21. peça 1) tem o objetivo de acomodar o microcontrolador arduino, e foi desenvolvida para o modelo específico Blackboard Uno R3. Foram projetados em suas laterais, encaixes, que servem para acoplar os suportes laterais. Estes por sua vez, possuem duas unidades de apoio com o intuito de segurar a placa do circuito principal na posição vertical (figura 21. peça 3). A estrutura ainda conta com uma peça no topo (figura 21. peça 5), que carrega 2 baterias de 9V do tipo Ni-Mh, sustentada por 4 cilindros (figura 21. peça 4) cuja função é prolongar a estrutura,

gerando o momento de inércia necessário para otimização dos testes. Por fim, foram desenvolvidos acoplamentos especiais (figura 21. peça 2), dedicados aos 2 motores DC.

Figura 22. Componentes e Encaixes



Fonte: Própria.

A placa de circuito impresso, por sua vez, foi projetada com a intenção de manter a forma prolongada da estrutura, carregando o sensor giroscópico e acelerôométrico MPU6050 da Invensense (figura 22. peça 1). A opção de utilizar a ponte H L293D, gerando uma melhor integração do driver à estrutura pode ser observada na (figura 22. peça 2). A placa converge em um acoplamento generalizado em sua parte inferior (figura 22. peça 3), este é conectado a uma estrutura que se adere ao microcontrolador, na intenção de gerar um *Shield* (figura 22. peça 4).

3.2 Circuito Elétrico

As placas de circuito impresso apresentadas, foram elaboradas com base em diversos esquemas elétricos pesquisados para este tipo de aplicação, mais precisamente o elaborado, no site MakerPro, por Reginald Watson. A maior diferença realizada, foi na escolha de utilizar a própria ponte H L293D, ao invés de um módulo ponte H pronto. Este esquema elétrico se encontra claramente representado no apêndice C.

3.3 Testes de validação

Com o intuito de averiguar a funcionalidade do *hardware* desenvolvido, foram realizados alguns testes básicos (figura 23), sendo que o primeiro deles, é responsável por controlar os motores nos dois sentidos de rotação(horário e anti-horário), o segundo realiza a comunicação somente entre o MPU6050 e o monitor serial do Arduino, enquanto o terceiro, comunica o sensor (MPU6050) com seus atuadores (motores DC), conforme requisitado.

Figura 23. Teste Motor DC

```

1 //Primeiro teste - Motores DC
2 void setup() {
3   //Portas PWM do Arduino (0~255)
4   pinMode(3, OUTPUT);
5   pinMode(5, OUTPUT);
6   pinMode(6, OUTPUT);
7   pinMode(9, OUTPUT);
8 }
9
10 void loop() {
11   //Se for 0 é velocidade 0, caso seja 255 é velocidade máxima.
12   analogWrite(3, 255);
13   analogWrite(5, 0);
14   analogWrite(6, 0);
15   analogWrite(9, 255);
16   delay(300); //Tempo de inversão
17   analogWrite(3, 0);
18   analogWrite(5, 255);
19   analogWrite(6, 255);
20   analogWrite(9, 0);
21   delay(300); //Tempo de inversão
22 }

```

Fonte: Própria.

4. Desenvolvimento

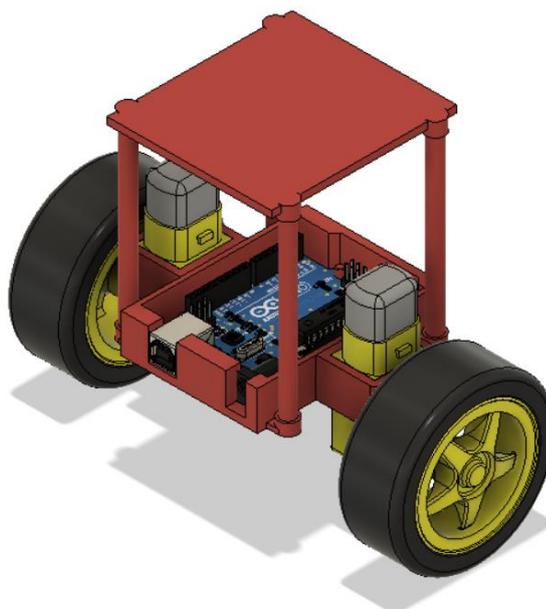
A princípio, buscou-se projetar a estrutura do robô baseando-se nos conceitos fundamentais de um pêndulo invertido, porém, devido a inexperiência com o dimensionamento das proporções mecânicas e a dificuldade na disponibilidade de recursos, o resultado esperado não foi obtido até as etapas finais do projeto.

A seguir, serão detalhados alguns fatores que dificultaram o desenvolvimento do projeto, mas que foram solucionados com um estudo mais aprofundado e com uma melhor utilização dos recursos.

4.1. Confecção da estrutura

O primeiro passo, após a aprovação do projeto, foi modelar a estrutura do robô, tomando como base, apenas que este deveria ter suas baterias alocadas ao topo, para a redução da velocidade angular e como tal, melhor compensação.

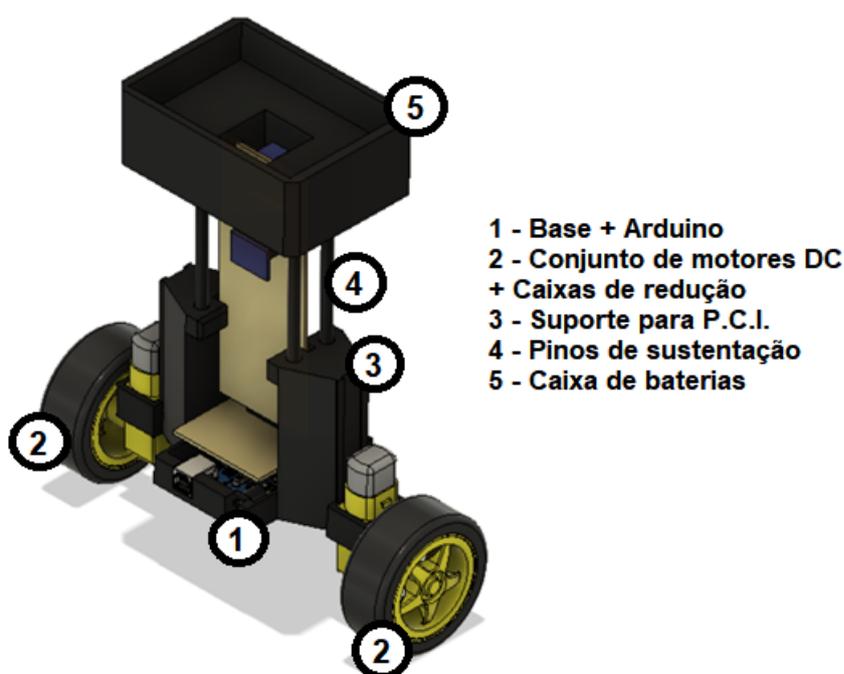
Figura 24. Primeira Estrutura



Fonte: Própria

Entretanto, como apresentado na estrutura da figura 24, a ideia inicial não foi levada adiante, pois possuía um topo relativamente baixo, o que acarretaria uma queda mais rápida, dificultando na hora de dimensionar seu devido compensador. Portanto, para solucionar este problema, buscou-se replanejar o projeto com uma estrutura de corpo alongado, cujo topo se encontraria em uma posição mais elevada. A placa de circuito impresso pensada para ela seguiria o mesmo princípio, garantindo maior facilidade na estabilização.

Figura 25. Segunda Estrutura



Fonte: Própria

Esta por sua vez, devido a indisponibilidade de recursos, não teve seu topo finalizado até as etapas finais do projeto, portanto diversos testes foram realizados, somente com a base da estrutura, parte na qual se apoiam as quatro barras, como representado na imagem (figura 25). Este fator desencadeou alguns problemas fundamentais que acarretaram diversas incongruências e testes sem grande valor significativo, mas que vieram a estimular pesquisas mais aprofundadas, com o intuito de compreender melhor sobre o conceito de estabilidade por trás do pêndulo invertido. Essa etapa providenciou informações que evidenciaram os erros cometidos na execução do projeto.

Foi assim que a ideia de conceber um novo robô, utilizando-se de materiais mais acessíveis e sem um maior compromisso com as formalidades, tal como baseando-se no replanejamento da estrutura, surgiu, e com a sua conclusão, adquiriu-se um segundo objeto de estudo, onde a resposta obtida, era próxima à esperada.

Em última instância houve modificações na estrutura principal, refletido pelos resultados adquiridos na execução do projeto auxiliar, essas alterações podem ser mais bem averiguadas no apêndice “B”.

4.2. Esquemático

Para a parte elétrica do projeto, os principais materiais utilizados foram:

1 x Arduino Uno

2 x Fontes de Alimentações

2 x Conjuntos de motores DC 6V_{DC} com caixa de redução

1 x Ponte H L298N

1 x Sensor MPU6050

Chicotes

O resultado obtido na integração de todos esses componentes, foi gerou-se o esquema elétrico representado no apêndice D.

Os testes que foram realizados com base nesse esquema elétrico, foram em sua maior parte bem-sucedidos, quando ocorria alguma falha, era devido à perda de comunicação, que mais tarde pode ser justificada pela perda de contato entre o sensor e o microcontrolador.

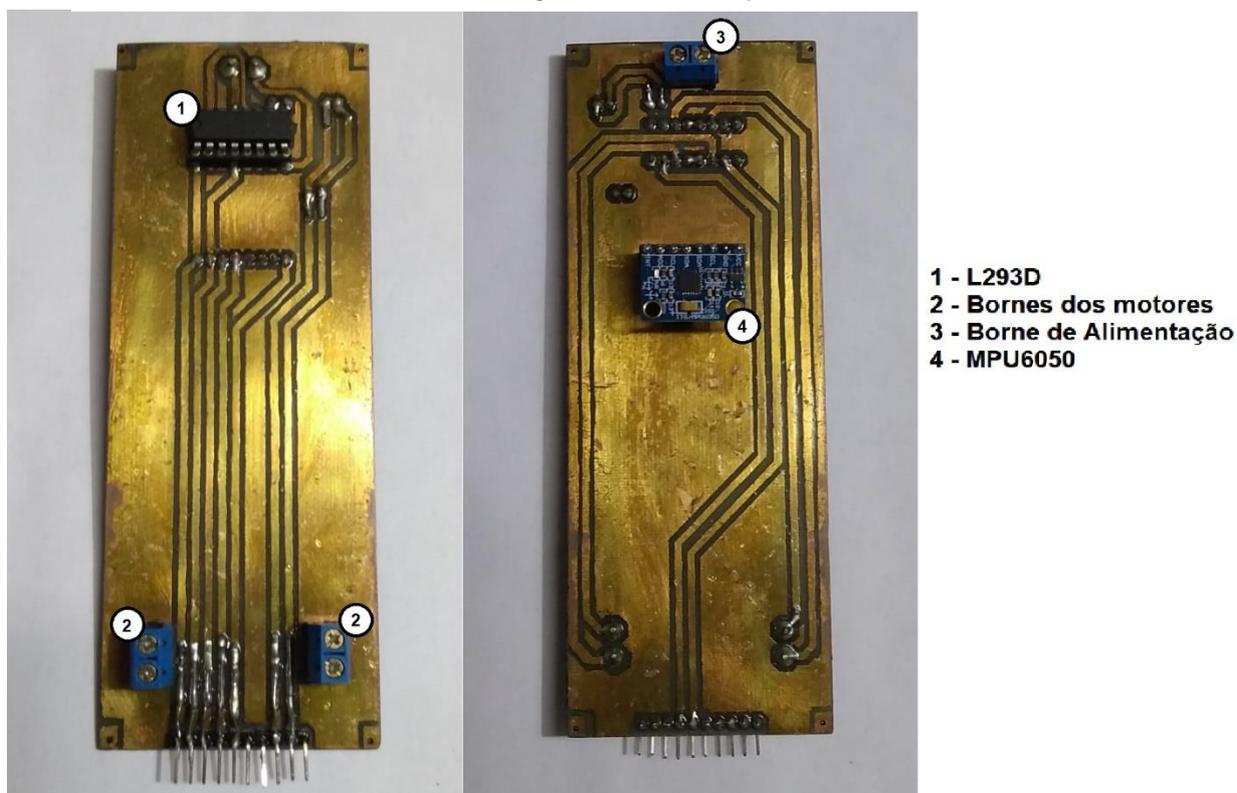
4.3. Prototipagem da P.C.I.

A Placa de Circuito Impresso (P.C.I.) foi concebida por meio de uma placa de fenolite e percloreto de ferro. Porém ao longo do projeto, mais de uma placa teve de

ser construída, pois em algumas instâncias, esta era peça fundamental para a resolução dos problemas, e, portanto, teve sua configuração alterada de acordo com as respostas encontradas.

A primeira placa (figura 26), teve que ser alterada, pois o modelo L293D de ponte H não era compatível com a corrente que os dois motores demandam, e ao sobreaquecer, ocasionava a interrupção da comunicação com o microcontrolador.

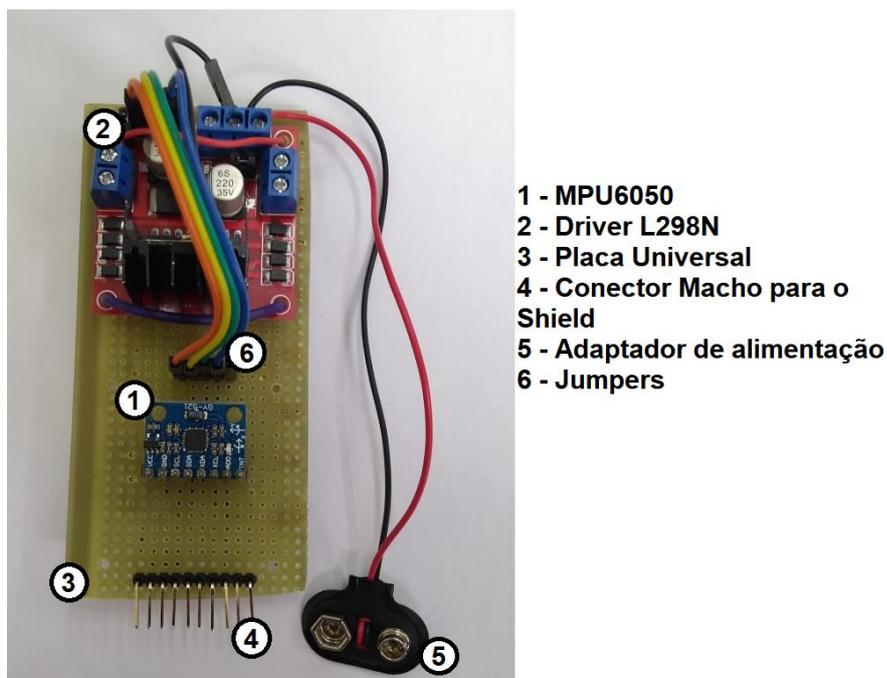
Figura 26. Primeira placa



Fonte: Própria.

Com a redução do tempo disponível, a segunda placa (figura 27), foi projetada sobre uma placa universal, permitindo uma maior velocidade de personalização e alteração durante os testes, e serviria como medida provisória antes da impressão de um novo circuito em placa de fenolite. Entretanto, a interrupção de comunicação com o MPU6050, continuou a ocorrer, exigindo uma segunda revisão.

Figura 27. Segunda Placa

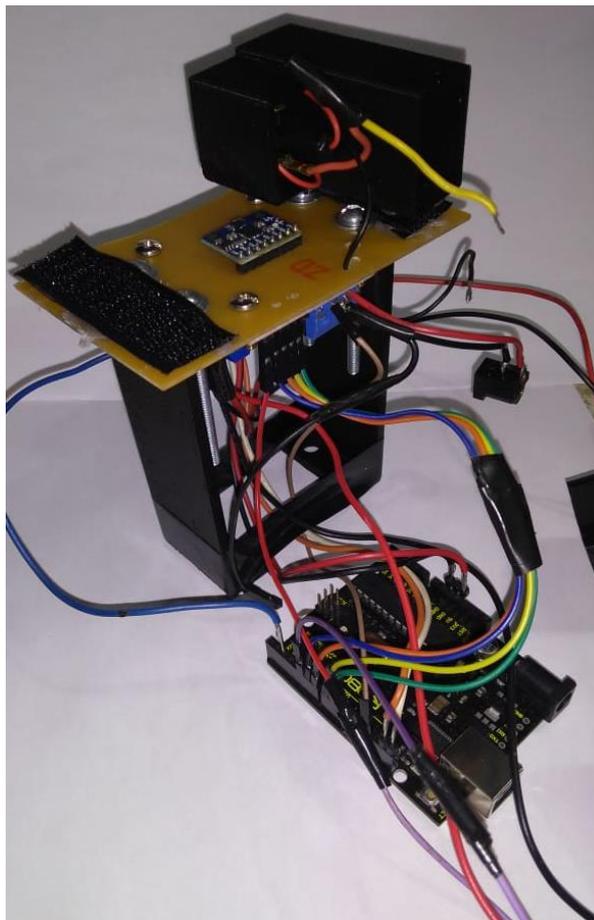


Fonte: Própria.

Durante a confecção da estrutura provisória, percebeu-se que grande parte dos erros até então confrontados eram provenientes da conexão entre os componentes do circuito, pois em função da dinâmica de movimento do projeto, diversos solavancos impediam o pleno funcionamento. Este problema foi solucionado com uma rígida fixação entre todos os integrantes através de soldas diretas, tal como a ampliação dos pontos de contato através duma melhor organização do cabeamento.

Ao final, o MPU6050 foi soldado diretamente na placa, para evitar esses mau-contatos, como apresentado na figura 28.

Figura 28. Placa final



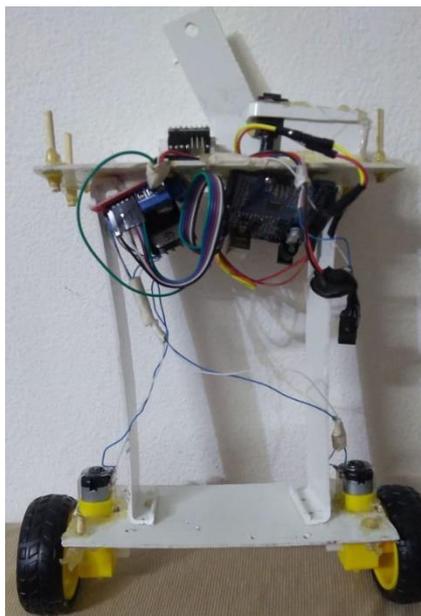
Fonte: Própria.

4.4. Melhores Resultados Obtidos

Como descrito anteriormente, com o decorrer do projeto, para que fosse possível adquirir uma maior confiabilidade e agilidade para trabalhar, foi realizada a construção de uma segunda estrutura (figura 29), que foi a obra responsável por apresentar os primeiros resultados sucedidos desta pesquisa.

O principal detalhe evidenciado pela segunda estrutura foi a importância de um centro de gravidade mais elevado, porém, que apresentasse carga proporcional à força dos motores disponíveis. Após uma sequência de testes com o objetivo de verificar a capacidade de “levantar” do robô, constatou-se que 28 centímetros providenciaram um atraso grande o suficiente para a recuperação dos motores em até 20° de queda. Esses resultados foram alcançados com a redução do peso total em aproximadamente 500 gramas, que mais tarde se revelaria um obstáculo.

Figura 29. Segundo Robô



Fonte: Própria.

Com o término da construção do segundo robô, iniciou-se o projeto do “compensador virtual”, chamado assim pois a malha completa de controle se conteria ao microcontrolador Arduino, através de equações e atribuição de testes lógicos para simular ganhos que de outra forma poderiam ser aplicados por meio de amplificadores operacionais.

Após a implementação deste conceito, obteve-se uma resposta instável, que exigiu uma apuração dos parâmetros dentro do código para que viesse a se tornar satisfatória, entretanto, devido à falta de peso no topo, que implicou em um centro de gravidade fora do desejado, grande parte dos testes realizados durante essa etapa apresentaram um comportamento inadequado e de caráter oscilatório, onde a estabilidade não se mantinha por um longo período de tempo.

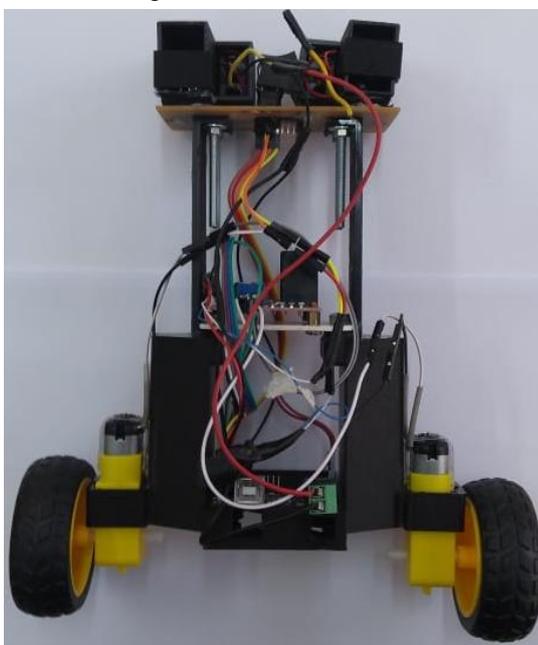
Após averiguados todos os pontos que impediam o funcionamento total do projeto, decidiu-se alterar a massa superior do robô, alcançando-se pleno equilíbrio por 20 minutos observados.

Durante uma das adaptações, percebeu-se que os motores utilizados no primeiro robô apresentavam uma ligeira defasagem em relação aos utilizados no segundo, o que levantou a possibilidade da existência de uma zona morta maior que o esperado, e que exigisse uma compensação individual, este tópico será mais bem abordado em capítulos posteriores.

4.5. Aplicação do Pêndulo Invertido

Então, para dar continuidade ao conceito do pêndulo invertido, seria necessário aplicar o centro de gravidade na parte superior da estrutura do robô. Isso foi feito, tomando como base o estudo desse efeito. O centro de gravidade é descrito pela empresa Tecnotextil, como sendo ponto de uma estrutura onde é aplicado a força peso. Portanto, a melhor maneira encontrada, para poder causar esse efeito no projeto estudado, foi alocando a maior concentração do peso, na forma de baterias responsáveis pela alimentação no topo da estrutura, tal como o sensor MPU6050 que demandava uma posição paralela ao topo, evitando que a dinâmica do sistema alterasse suas medições. Chegando a estrutura final apresentada na figura 30 a seguir:

Figura 30. Estrutura Final



Fonte: Própria.

4.6. Zona Morta

Zona morta é o nome dado a variação nos dados de entrada que não provocam resposta direta na saída.

4.6.1. Explicação

Ao aplicar-se tensão sobre os motores DC individualmente, percebeu-se que apresentavam uma necessidade maior de tensão para partir quando acoplados às suas devidas reduções, e que esta necessidade aumentava consideravelmente ao

conectá-los através do driver L298N, dificultando a compensação, pois para pequenos valores de erro, o motor não apresentava resposta.

4.6.2. Testes Realizados

Para solucionar esse problema, foram realizados alguns testes utilizando o Arduino como plataforma de comando, que seria o responsável por incrementar valores de PWM (Pulse Width Modulation) à saída através da integração com o driver L298N, e assim, foi possível capturar o valor mínimo necessário para partir o motor.

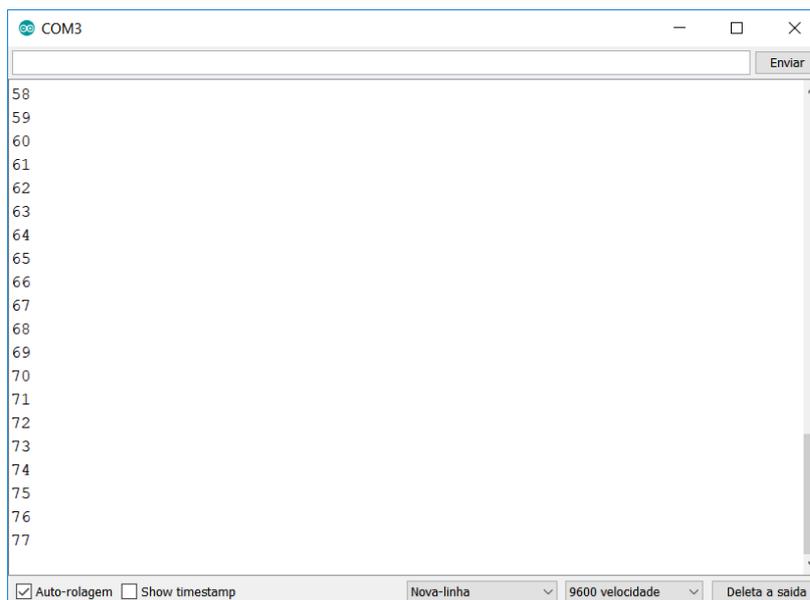
Abaixo (figura 31), é possível verificar o programa realizado, e os valores de saída (figura 32) que permitiram obter a análise descrita acima.

Figura 31. Programa de análise da Zona Morta

```
Zona_Morta
1 //Variável global
2 #define MOTOR 9 //Porta Digital de Saída PWM do Arduino,
3 //responsável por enviar o sinal PWM ao Driver L298N
4
5 int PWM; //Declarando a variável "PWM" como do tipo inteiro
6 void setup() {
7 //Inicializando a porta serial
8 Serial.begin(9600);
9 //Declarando que a porta MOTOR é de saída
10 pinMode(MOTOR, OUTPUT);
11 }
12
13 void loop() {
14 //Loop incremental de 0 a 255 (valores de PWM)
15 for (PWM = 0; PWM < 256; PWM++){
16 //Escrevendo no monitor Serial o valor atual de PWM
17 Serial.println(PWM);
18 //Aplicando a variável PWM no motor
19 analogWrite(MOTOR, PWM);
20 //Tempo de espera até incrementar novamente
21 delay(200);
22 }
23 }
```

Fonte: Própria.

Figura 32. Valor de PWM analisado pelo monitor serial



Fonte: Própria.

A partir dessa análise, ficou evidente que quando o PWM atingia o valor de 70 pulsos, o motor conseguia partir, e aplicando essa zona morta ao programa principal, percebeu-se uma grande diferença quanto ao tempo de resposta durante a inversão de sentido de giro dos motores.

4.7. Ponte H L298N

Enquanto era realizado o estudo, para compreender os problemas encontrados no projeto, tal como a diferença de rotação entre os motores, foram ponderadas algumas propostas levantadas pelo orientador.

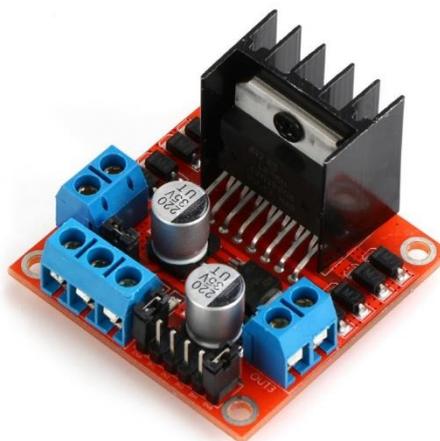
A primeira delas, foi tentar igualar a rotação entre ambos motores, e para isso, uniu-se a ligação deles em um mesmo borne do driver. Como resultado, os dois foram ativados ao mesmo tempo, em contrapartida, notou-se que a potência total havia sido reduzida. porém outro fato tornou-se perceptível, que foi a falta de potência aplicada no motor, não permitindo com que os dois girassem na mesma velocidade.

Para corrigir esse problema, alguns estudos foram feitos, tal como o da zona morta, apresentado no capítulo anterior (4.6.), até o momento na qual ficou perceptível uma avaria existente na utilização do L298N (figura 33). O driver apresenta uma diferença de potencial entre a tensão de entrada e de saída, pois em sua saída há uma

perda de 4V em relação à sua entrada. Portanto, para este projeto esperava-se uma tensão de saída aplicada nos motores de aproximadamente 6V, para isso, foi necessário aplicar ao valor de entrada cerca de 10V, obtendo assim a compensação dessa perda.

No caso, foi adotada uma tensão de 9V, que era a tensão fornecida pelas baterias que seriam utilizadas para apresentar o projeto.

Figura 33. Driver Ponte H L298N



Fonte: gearbest.com

4.8. Cálculos primordiais

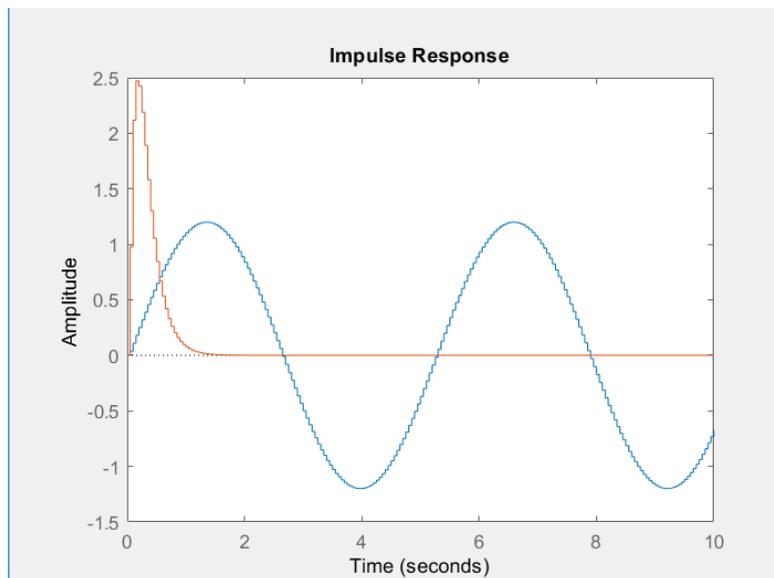
Para poder encontrar os compensadores necessários, e adquirir a maior estabilidade a este sistema, foi utilizado como ferramenta de auxílio a função de transferência genérica de um pêndulo invertido em malha aberta, mostrado a seguir:

$$G(s) = \frac{a}{s^2 + a} \quad \text{Equação 8}$$

É então possível analisar sobre o plano s , dois pontos opostos, equidistantes em relação à origem. Sendo assim, quando aplicado ao caminho pelo lugar das raízes (C.L.R.), analisa-se que há uma resposta instável até um determinado valor de ganho

K, e a partir disso, será marginalmente instável, como apresentado na imagem figura 34, gerada por meio de um software.

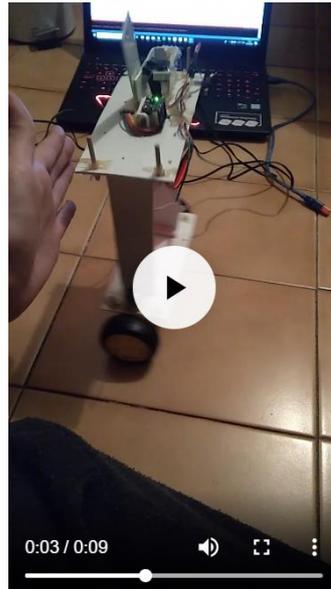
Figura 34. Resposta Adquirida (Azul) x Resposta Desejada (Laranja)



Fonte: Própria.

Para adquirir dados mais concretos, baseados no robô em si utilizado como ferramenta para essa pesquisa, foi gerado um vídeo, como representado na figura 35, para apresentar com maior precisão, a quantidade de oscilações que este sistema representava em um determinado tempo, quando tinha-se um valor de ganho K_p equivalente à 150 já inserido no sistema, tornando-se possível coletar a frequência aproximada do sistema.

Figura 35. Método adotado para análise de frequência



Fonte: Própria.

Como resultado dessa análise, obteve-se um resultado de 31 oscilações num período de 10 segundos, portanto, realizando os cálculos de frequência, é possível afirmar que este sistema é de 3.1Hz.

E foi assim que a partir dessa informação, foi possível gerar a velocidade angular (ω_n) do sistema, para tornar-se possível encontrar a função de transferência em malha fechada. O ω_n da função foi calculado conforme a equação 9.

$$\Omega_n = 2\pi f \quad \therefore \quad \omega_n = 2\pi 3.1 = 6.2\pi \cong 19.5 \text{ rad/s} \quad \text{Equação 9}$$

Obtendo-se assim a função de transferência em malha fechada da equação 10.

$$F_{(s)} = \frac{19.5^2}{s^2 + 19.5^2} = \frac{380.25}{s^2 + 380.25} \quad \text{Equação 10}$$

Após adquirir $F_{(s)}$ desse sistema, viu-se a necessidade de encontrar o valor de α , e isso foi feito a partir do tempo em que o robô demora para mover-se de 90° à 0° . Obtendo-se então um tempo de queda de 4 segundos.

A partir dessa resposta, tornou-se possível afirmar que os polos do robô se encontram nas posições de 1 e -1 no plano s, enquanto no plano z, foi necessário realizar o cálculo da posição da assíntota, determinando onde ela cruza o eixo x a 90°. Isso foi feito com base nos polos e zeros adicionados ao C.L.R., -10 e -5 respectivamente.

$$\sigma_A = \frac{\Sigma_{\text{polos_finitos}} - \Sigma_{\text{zeros_finitos}}}{n^{\circ}_{\text{polos_finitos}} - n^{\circ}_{\text{zeros_finitos}}} \quad \text{Equação 11}$$

$$\frac{(-10) + (-1) + (+1) - (-5)}{3 - 1} = -2.5$$

Portanto, colocando esses dados no software, tornou-se possível analisar um ponto dentro do ramo que tende ao infinito, possibilitando a procedência dos cálculos de compensadores. O valor encontrado foi de: $-1,5+7j$.

4.8.1. Compensador de Avanço

A alocação dos polos e zeros em -10 e -5, permitiram encontrar a equação do sistema, que é dado pela equação 12.

$$K = \frac{(s + p_1)(s + p_2) \dots (s + p_q)}{(s + z_1)(s + z_2) \dots (s + z_n)} \text{ ou } \frac{K(s + z_1)(s + z_2) \dots (s + z_n)}{(s + p_1)(s + p_2) \dots (s + p_q)} = 0 \quad \text{Equação 12}$$

Portanto, para esse sistema, obteve-se o resultado da equação 13.

$$\frac{K(s + 5)}{(s + 10)} \quad \text{Equação 13}$$

Então, sabe-se que o produto dessa equação do ganho K (equação 13), pela função de transferência em malha fechada (equação 10), tem-se que a resposta deve ser $1 \angle 180^\circ$, portanto:

$$\frac{380}{s^2 + 380} \times \frac{K(s + 5)}{(s + 1)} = 1 \angle 180^\circ$$

$$G_{(s)} = \frac{380K(s + 5)}{(s + 1)(s - 1)(s + 10)} /_{-1,5+7j} = 1 \angle 180^\circ$$

$$K = \frac{1L180^\circ(s+1)(s-1)(s+10)}{380L90^\circ(3,5+7j)} /_{-1,5+7j} = \frac{1L180^\circ(-0,5+7j)(-2,5+7j)(8,5+7j)}{380L0^\circ(3,5+7j)}$$

Resolvendo a equação resultante, obteve-se o resultado de que o ganho K é igual a 0,193. Portanto, aplicando este valor na equação de ganho (equação 13), adquire-se o compensador de avanço (CA) em s, como apresentado logo abaixo:

$$CA = \frac{0,193(s+5)}{(s+10)}$$

Porém, o sistema estudado está em z, logo, foi necessário realizar o cálculo desse compensador com a planta em z (CZ).

$$CZ = \frac{0,193z - 0,155}{z - 0,6065} \times z^{-1}$$

$$\frac{0,193 - 0,155z^{-1}}{1 - 0,6065z^{-1}} = \frac{Y}{X}$$

$$Y_{(n)} = 0,6065Y_{(n-1)} + 0,193X_{(n)} - 0,155X_{(n-1)}$$

A equação obtida, para o software que será aplicado no código do Arduino, representa o ganho necessário aplicado ao motor, portanto, $Y_{(n)}$ é o ganho que será aplicado aos motores do robô, $Y_{(n-1)}$ é o valor anterior, enquanto X representa os ângulos, sendo, $X_{(n)}$ o ângulo atual e $X_{(n-1)}$ o ângulo anterior.

4.8.1.1. Compensador de Avanço Via Software

Pelo software, é mais fácil e rápido adquirir os mesmos resultados obtidos nos cálculos realizados à mão, basta aplicar corretamente os parâmetros adquiridos nas devidas funções que já contém para programação. No apêndice “A”, estarão apresentadas as linhas de códigos desenvolvidas, comentadas linha a linha e os resultados gerados tais como as equações e gráficos.

4.8.2. Cálculo do P.I.D.

Na tentativa de estudar qual compensador apresentaria um resultado mais próximo ao desejado, além do compensador de avanço, estudou-se um método para aplicar um compensador P.I.D. esperando-se obter a mesma resposta sub amortecida da figura 34.

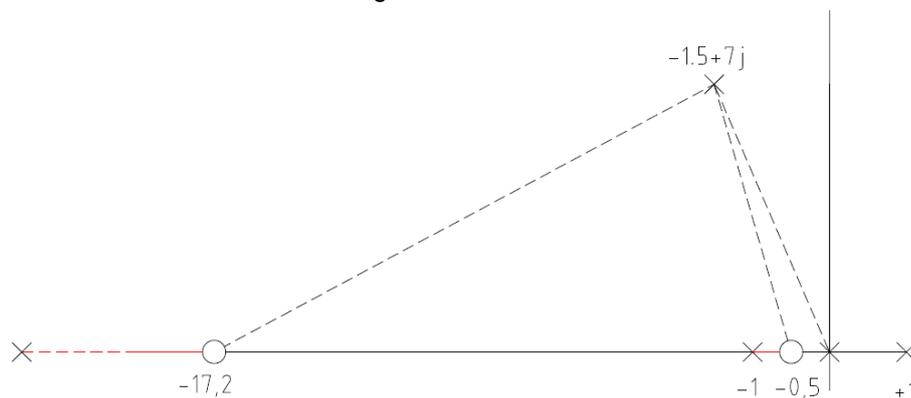
Para isso, foram reutilizados alguns dados adquiridos no capítulo de cálculos primordiais (4.8), sendo eles:

- Polos: 1 e -1
- Posição da Assíntota: 2,5
- Ponto utilizado: $-1,5+7j$

Com a intenção de adquirir os polos do lugar das raízes em s , e sabendo que um dos polos está na posição calculada como sendo a assíntota do sistema, os outros dois foram estipulados de acordo com as seguintes etapas:

- Primeiramente, sabia-se que um dos ângulos a serem formados, não deveriam ser maiores do que 5 graus, portanto, foi adotado um zero de valor $-0,5$ (figura 36).

Figura 36. Gráfico P.I.D.



Fonte: Própria.

- Feito isso, o próximo passo, era encontrar o terceiro ponto, portanto, sabendo-se que a somatória dos ângulos teria que ser equivalente à 180 graus, foram realizados os seguintes cálculos:

$$\alpha_1 = \arctg \frac{2,5}{7} + 90 = 109,6^\circ \quad \text{Equação 14}$$

$$\alpha_2 = \arctg \frac{0,5}{7} + 90 = 94^\circ \quad \text{Equação 15}$$

$$\alpha_3 = 180 - 109,6 - 94 = -24^\circ \quad \text{Equação 16}$$

- Com o resultado deste cálculo, percebeu-se que o ângulo que teria que ser formado, seria de 24 graus, e para conseguir obter o ponto em que este polo se encontra no lugar das raízes, foi realizado o cálculo a seguir:

$$\operatorname{tg} 24^\circ = \frac{7}{l} \Rightarrow l = \frac{7}{\operatorname{tg} 24^\circ} = 15,7$$

$$-15,7 - 1,5 = -17,2 \therefore (s + 17,2)$$

Aplicando estes valores à equação de ganho K (equação 13) apresentada anteriormente, foi possível substituir os valores adquiridos e assim, calcular o ganho P.I.D., utilizando a equação final encontrada.

$$\frac{K(s+0,5)(s+17,2)}{s}, \text{ para } K = 0,01$$

$$\frac{0,01(s + 0,5)(s + 17,2)}{s} = \frac{0,01(s^2 + 17,7s + 8,6)}{s}$$

$$\frac{0,01s^2 + 0,177s + 0,086}{s} \times 100$$

Observação: O valor de 100, foi utilizado na equação, pois para a aplicação na programação, foi o que apresentou o melhor resultado.

Sabendo-se que a equação 17 do cálculo de P.I.D. é dada por:

$$PID = Kp + \frac{Ki}{s} + Kds = \frac{Kds^2 + Kps + Ki}{s} \quad \text{Equação 17}$$

É possível realizar uma comparação entre as equações e extrair o valor de cada variável, para ser aplicada ao programa do robô.

Portanto, se:

$$\frac{Kds^2 + Kps + Ki}{s} = \frac{1ds^2 + 177s + 8,6i}{s}$$

Então, pode-se afirmar que:

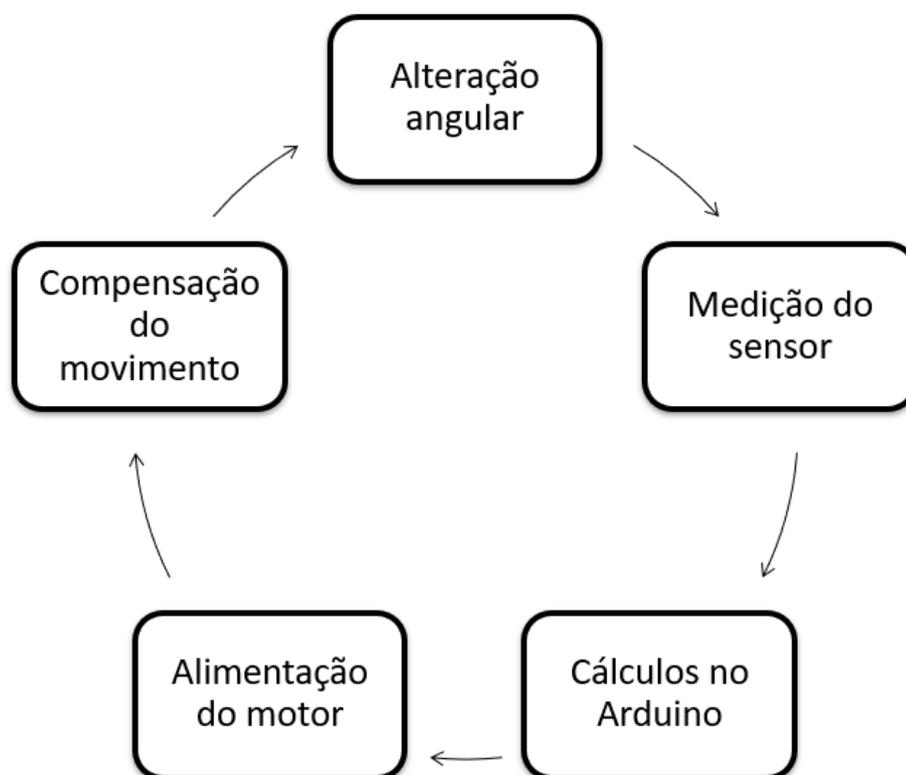
$$Kd = 1, Kp = 177 \text{ e } Ki = 8,6$$

4.8.3. Aplicando via Software

Todos os recursos utilizados, no que diz respeito à software, foram incorporados aos anexos. Agradecimentos especiais à Reginald Watson, por providenciar as bibliotecas essenciais, tal como o modelo padrão fundamental ao desenvolvimento do projeto, através da plataforma Maker Pro.

Como representação básica do software apresentado em anexo, tem-se o seguinte diagrama funcional da figura 37:

Figura 37. Diagrama Funcional



Fonte: Própria.

4.8.3.1. Utilização dos recursos básicos do MPU6050

Desde os projetos iniciais, a proposta de gerenciar todo o sistema de controle através de um “compensador virtual” foi levada em consideração. Essa ideia teve de transpassar as barreiras iniciais da captação dos dados de entrada e da velocidade de resposta sobre os dados de saída, na forma de sensor giroscópico e motores DC.

A princípio, observou-se que não havia maneira simples de requisitar os valores angulares do MPU6050, sem se apropriar da comunicação I²C que seria consumada pela biblioteca “WIRE.h”. Percebeu-se que a remoção dos dados deveria acontecer sobre cada um dos registradores diretamente, após a devida parametrização de cada um deles. Esses registros seriam armazenados em variáveis independentes para cada eixo, entretanto, apenas um deles seria foco do projeto final.

A primeira grande dificuldade se levantou no modo como esses dados eram concebidos, pois apresentavam valores indefinidos, fora de escala. Em função de diversas pesquisas, encontrou-se a relação estabelecida para a transformação em informação significativa.

O segundo problema surgiu na incapacidade do sensor de apresentar o valor angular atual, mostrando apenas a variação ocorrida. Este problema foi facilmente solucionado ao incrementar os valores de oscilação em uma única variável em relação ao tempo. Entretanto, logo percebeu-se à deriva numérica, decorrente pelo balançar do sensor, valor que se ampliava gradativamente e tardava em se recuperar. Este problema foi solucionado com extensa pesquisa, revelando a resposta nos ângulos de Euler, que aplicariam parte da aceleração no valor do giroscópio, proporcionando uma deriva extremamente reduzida.

Com os valores de defasagem angular confiáveis, pode-se partir para o desenvolvimento da malha de controle, que possuiria o compensador virtual. Essa etapa foi imensamente prejudicada pelos problemas elétricos e mecânicos, mencionados em capítulos anteriores. Com o incentivo do orientador, fundamentou-se o sistema em cima da diferença entre o ângulo atual, incrementado à variável, e o ângulo desejado, na forma de um valor fixo. A está diferença atribuiu-se o nome de: Erro.

Em função da discordância de escalas, não foi possível atribuir estabilidade ao protótipo apenas alimentando os motores com o valor de erro, e, portanto, nomeou-

se K_p (constante de ganho proporcional) a variável responsável por ampliar a ordem da alimentação dos motores.

Inicialmente, formulou-se que esta constante deveria levar o valor inteiro de 25, pois quando o protótipo atingisse a queda de 10° , a compensação se aproximaria do valor máximo. Essa hipótese logo foi descartada ao perceber, empiricamente, que a aceleração angular durante a queda se tornava irreparável, tendo em vista a magnitude dos motores disponíveis. Em segunda instância, conferiu-se o valor de 75 para K_p , observando resultados satisfatórios. Esses testes seriam mais bem explicados pela zona morta mais tarde. Outro detalhe importante a se acrescentar, foi a indiferença entre valores acima de 100, pois atingiam rapidamente o limite dos motores e não demonstravam melhorias quanto à estabilidade.

Com a intenção de implementar a constante Integradora, foi desenvolvida uma lógica de atuação apenas para valores pequenos de erro, e, portanto, caso a variação não estivesse entre a janela de $\pm 5^\circ$, não haveria ação integradora. Seu funcionamento se baseia na somatória de todos os erros vezes a constante de ganho integral K_i , dentro do intervalo selecionado. É importante salientar que esta somatória deveria encontrar valores negativos ao transitar entre ângulos maiores ou menores que a referência. Essa ação tem a função de prever a variação angular e, portanto, reduz a aceleração toda vez que houver uma alteração no sinal, resultando em uma resposta mais suave próxima do ângulo desejado e não interferindo na compensação de quedas mais bruscas.

O valor empírico encontrado para esta constante foi 100. Mas não houve melhora aparente em um primeiro momento, pois em pouco tempo o erro se acumulava e o robô se movimentava para fora da área de atuação do controle integrador.

Para acrescentar a constante de ganho derivativo, K_d , houve a necessidade de contabilizar a variação de erro em função do tempo decorrido. Essa ação agiria diretamente sobre a velocidade de queda, sendo que quanto mais rápida a queda (refletida na diferença entre erros), maior a compensação necessária para a recuperação do ângulo desejado. Na prática, essa constante gera o movimento de “guinada”, levantando bruscamente o topo da estrutura.

Durante a busca do valor empírico para K_d , descobriu-se que para valores muito altos, o protótipo se movimentaria em oscilação para os dois lados constantemente em alta frequência, levando a uma instabilidade indesejada. O valor encontrado foi de amplitude 2,8.

Por fim, associou-se todos os fatores de correção em uma única variável, de nome PID, ao qual aplicou-se sobre os motores, levando em consideração os limites de operação em cada um dos sentidos. Os resultados foram próximos ao esperado, entretanto, longe do ideal. Decidiu-se calcular os valores de compensação a partir das teorias matemáticas de controle, com a utilização de softwares.

4.8.3.2. DMP e utilização dos recursos avançados do MPU6050

Durante a primeira bateria de testes, observou-se um atraso relativo na resposta do microcontrolador e optou-se por procurar alternativas que pudessem solucionar o problema, levando em consideração a disponibilidade de recursos e o investimento necessário.

Descobriu-se que o sensor MPU6050 apresenta capacidade de processamento interno, limitada, porém suficiente tendo em vista a demanda. O acesso a este recurso se dá pelo DMP (Digital Motion Processor - processador de movimentação digital), recurso que apresenta alto desencorajamento por parte da empresa fabricante, que se recusa a fornecer qualquer informação a respeito. Portanto, toda informação utilizada provém de exemplos encontrados na internet, escritos por engenheiros dedicados a causa, como evidenciado no anexo A.

Este recurso se utiliza de um Buffer com registro FIFO (*First In First Out*- Primeiro a Entrar Primeiro a Sair) para armazenar os dados captados e processados pelo próprio sensor. Ele então se aproveita dos pinos de interrupções para indicar ao microcontrolador quando o pacote que contém as informações mais recentes estão prontos, e para isso precisa alterar a velocidade de comunicação e contabilização de *clock* serial inerentes aos *timers* do Arduino. Isso permite um processamento consideravelmente mais rápido, entretanto, mais propenso a erros, e, portanto, algumas redundâncias podem ser encontradas ao longo do programa base, utilizado como referência para a implementação do DMP, como reinicialização do pacote caso haja interrupções inesperadas e/ou retorne valores menores do que o calculado.

Pelos motivos mencionados anteriormente, somados à disponibilidade de tempo, avaliou-se que a implementação do recurso seria secundária, e portanto, não condizia com o escopo inicial do projeto. Logo, apesar de estudada a fundo, optou-se pela utilização do material referencial amplamente divulgado pela rede global.

5. Considerações finais

5.1. Conclusão

Os conceitos teóricos firmados desde a primeira documentação deste trabalho, se mantiveram entrelaçados com o funcionamento geral por todo o percurso das pesquisas. Forças como torque, momento angular e momento de inércia permitiram o entendimento das respostas físicas atreladas ao movimento da estrutura, e colaboraram para a confecção de corpos mais aptos ao equilíbrio pela redução de não-linearidades.

Quanto às aplicações, a IMU (Inertial Measurement Unit) já é incorporada na orientação de aeronaves, drones e embarcações, onde a precisão e controle se mostram de suma importância, e a capacidade de implementá-los, de alto valor. Há também, a chamada: robótica do entretenimento, onde cada vez mais a tecnologia avançada permeia o cotidiano. Nesta área, estima-se que a capacidade de medir a movimentação será enfoque e permitirá uma revolução na vida das pessoas, mais uma vez, experiência com essa tecnologia promoverá destaque.

Tendo em vista as crescentes discussões sobre os pilares da indústria 4.0, pode-se destacar a necessidade de robôs autônomos, cuja operação de auto balanço possui diversas aplicações, como: redução do consumo de espaço físico, possibilidade de operação em diversos eixos, além da gestão de movimento, permitindo a criação de robôs mais inteligentes, com deslocamento inteligente.

A proposta deste trabalho era o estudo e desenvolvimento de uma IMU, tal como a integração deste à um sistema naturalmente instável, onde atuaria como sensor de realimentação em uma malha de controle, colaborando para a compensação de dois motores responsáveis pela conservação de um setpoint pré-estabelecido, aumentando a estabilidade do sistema.

Este projeto permitiu a expansão matemática do estudo de controle, além de demonstrar, através de um meio observável, conceitos inerentes e inflexíveis do tema, como a relação entre ganhos, sobressinal e tempo de resposta.

Quanto ao desempenho do sensor MPU6050, destaca-se a qualidade e confiabilidade gerada na operação, tal como facilidade de implementação. Entretanto, deve-se ressaltar a complexidade de agir em camadas internas, onde a disponibilidade de informação é reduzida/escassa. No que diz respeito a custo benefício, tendo em vista

o escopo do projeto, este, em conjunto com a plataforma Arduino, representam um forte sistema, extremamente personalizável e eficiente.

Durante os testes, pode-se comprovar a aplicação de controle, além de verificar o funcionamento da realimentação negativa e principalmente os efeitos das constantes de ganho, onde construiu-se um impactante conhecimento empírico, graças a possibilidade de examinar cada resposta, em relação a cada alteração, individualmente e em conjunto, isso tudo em um ambiente onde o sucesso não era um objetivo maior do que a análise e estudo de cada caso.

5.2. Sugestão para projetos futuros

Uma das propostas previstas com o decorrer deste projeto, foi a implementação de um sistema que segue a mesma ideia do robô de auto balanço, porém desta vez, aplicado à uma bicicleta, que inicialmente permaneceria de pé sem o auxílio de qualquer componente físico sub-apoiado ao solo. A maneira idealizada para a realização desse trabalho, é aplicando um eixo, próximo ao assento da estrutura, com um peso que se desloca em um eixo secante à variação angular, conforme fosse necessário realizar a compensação de acordo com os dados coletados pelo sensor.

Quanto a melhorias propostas, sugere-se a substituição dos motores DC por motores de passo, os quais podem oferecer maior precisão e maior torque, além de apresentarem uma performance constante, mesmo com a depleção da bateria. Levantar as vantagens e desvantagens em comparação com uso de motores DC, pode se mostrar um excelente objeto de estudos.

6. Referências Bibliográficas

AUTODESK. Disponível em: < <https://www.autodesk.com.br/>>. Acesso em: 15 de maio de 2018.

ARDUINO. **“About us”**. Disponível em: <<https://www.arduino.cc/en/Main/AboutUs>>. Acesso em: 14 de maio de 2018.

AZEVEDO, F. M. D. **Estudo e projeto de melhoria em máquina de impressão 3D**. 2013. Universidade de São Paulo.

BANZI, M.; SHILOH, M. **Getting started with Arduino: the open source electronics prototyping platform**. Maker Media, Inc., 2014. ISBN 1449363296.

BH GROUP. **“Como Funciona a Impressão 3D, fantástico!”**. Disponível, em: <<http://bhgroup.com.br/noticias/como-funciona-a-impressao-3d-fantastico/>>. Acesso em: 20 de maio de 2018.

BROOKING. “Your Arduino Balancing Robot (YABR)”. Disponível em: <http://www.brokking.net/yabr_main.html>. Acesso em: 03 de dezembro de 2018.

CALACHE, D. C. **Caracterização de um Acelerômetro Baseado em Sistemas Microeletromecânicos (MEMS)**. Projeto de graduação, 2013.

CÁSSIO, H. **“O que é Arduino?”**. Disponível em: < <http://protovie.com/o-que-e-arduino/>>. Acesso em: 14 de maio de 2018.

DE MIRANDA, R. J. C. **Desenvolvimento de um programa didático computacional destinado à geração de códigos de Comando Numérico a partir de modelos 3D obtidos em plataforma CAD considerando a técnica prototipagem rápida**. 2009.

DOCPLAYER. **“Modelagem dos sistemas estruturais aula 03”**. Disponível em: <<http://docplayer.com.br/70822995-Modelagem-dos-sistemas-estruturais-aula-03-modelagem-de-cabos.html>>. Acesso em: 20 de maio de 2018.

DREAMSTIME. “**Giroscópio no perfil**”. Disponível em: <<https://pt.dreamstime.com/fotos-de-stock-royalty-free-girosc%C3%B3pio-no-perfil-image11315448>>.

Acesso em: 14 de maio de 2018.

EMBARCADOS. “**PID digital - Método de cálculo numérico**”. Disponível em: <<https://www.embarcados.com.br/pid-digital-metodo-de-calculo-numeric/>>. Acesso em: 16 de dezembro de 2018.

FERRY, E. S. (1933) Applied gyro dynamics: for students, engineers and users of gyroscopic apparatus. J. Wiley & Sons, inc.

HALLIDAY, D., RESNICK, R. e WALKER, J. – **Fundamentos de Física** – Volume 1- Mecânica – Livros Técnicos e Científicos Editora – 8a Edição.

INFOESCOLA. “**Cinemática**”. Disponível em: <<https://www.infoescola.com/mecanica/cinematica/>>. Acesso em: 30 de abril de 2018.

INFOESCOLA. “**Dinâmica**”. Disponível em: <<https://www.infoescola.com/fisica/dinamica/>>. Acesso em: 30 de abril de 2018.

INFOESCOLA. “**Velocidade Angular**”. Disponível em: < <https://www.infoescola.com/fisica/velocidade-angular/>>. Acesso em: 20 de maio de 2018.

KNOWLEDGE AUTODESK. “**Como exportar g-code do Fusion 360**”. Disponível em: <<https://knowledge.autodesk.com/pt-br/support/fusion-360/learn-explore/caas/sfdarticles/sfdarticles/PTB/Does-Fusion-360-Write-G-code.html>>.

Acesso em: 15 de maio de 2018.

LASTRES, H. M. and ALBAGLI, S. (1999) '**Informação e globalização na era do conhecimento**', Rio de Janeiro: Campus, 163.

MAKER PRO. “**Build arduino self-balancing robot**”. Disponível em: <<https://maker.pro/arduino/projects/build-arduino-self-balancing-robot>>. Acesso em: 14 de maio de 2018.

MARTADUINO. “**IMU 6 EIXOS MPU6050, ACELEROMETRO E GIRO**”. Disponível em: < <https://www.martaduino.com.br/imu-6-eixos-mpu-6050-acelerometro-e-giro>>.

Acesso em: 15 de maio de 2018.

MARTINS, R. A. Como distorcer a física: considerações sobre um exemplo de divulgação científica 1-Física clássica. **Caderno Brasileiro de Ensino de Física**, v. 15, n. 3, p. 243-264, 1998. ISSN 2175-7941.

MAYR, Otto. The origins of feedback control. **Scientific American**, v. 223, n. 4, p. 110-119, 1970.

MICROSOFT. “**Carregando código de quadro e Arduino IDE**”. Disponível em: <<https://support.office.com/pt-br/article/carregando-c%C3%B3digo-de-quadro-e-arduino-ide-a9723765-1314-49e0-a69b-bb5c3e1f628d>>. Acesso em: 15 de maio de 2018.

MPU6050 Datasheet. InvenSense Inc. 2013. url: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>

MUNDO EDUCAÇÃO. “**Momento ou torque de uma força**”. Disponível em: <<http://mundoeducacao.bol.uol.com.br/fisica/momento-ou-torque-uma-forca.htm>>. Acesso em: 13 de maio de 2018.

TODA MATERIA. “**Plano Inclinado**”. Disponível em: <<https://www.todamateria.com.br/plano-inclinado/>>. Acesso em: 20 de maio de 2018.

RAMALHO, G. L. B. **Monitoramento de condição de máquina utilizando acelerômetro MEMS**. Revista Tecnologia, v. 33, n. 1, p. 63-72, 2012. ISSN 2318-0730.

RODRIGUES, J. R. et al. Identificação das principais componentes de incertezas de calibração de acelerômetro piezoelétrico pelo método da comparação: Uma revisão atual. **VI SBEIN**, Simpósio Brasileiro de Engenharia Inercial, Rio de Janeiro, 2010.

SILVA, R. V.; SILVA, F. A. **Utilização de dispositivos móveis com acelerômetro para controle de aplicações**. Colloquium Exactarum, 2011. 12-20 p.

SOFISICA. “**Gravitação Universal**”. Disponível em: < <https://www.sofisica.com.br/conteudos/Mecanica/GravitacaoUniversal/gu.php>>. Acesso em: 13 de maio de 2018.

SOFISICA. “**Leis de Newton**”. Disponível em: <<http://www.sofisica.com.br/conteudos/Mecanica/Dinamica/leisdenewton.php>>. Acesso em: 30 de abril de 2018.

STORE ARDUINO. “**Arduino Uno Rev3**”. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 14 de maio de 2018.

TECNOTEXTIL. “**Centro de Gravidade Deslocado**”. Disponível em: < <https://www.tecnotextil.com.br/glossario/centro-de-gravidade-deslocado/>>. Acesso em: 18 de novembro de 2018.

Apêndice A – Estudo Matemático do Compensador de Avanço

Código:

```

syms s w z t           %Definição de símbolos
t=.05;                %Tempo de amostragem
T=0:t:10;             %Tempo total de 0s à 10s

%planta do robo em malha fechada com ganho de 380
npf=[380]              %Determinando numerador
dpf=[1 0 380]          %Determinando denominador
p=tf(npf,dpf)          %Func. de Trasn. malha fechada
pz=c2d(p,t,'ZOH')      %Transformando de contínuo para discreto com c2d

%compensador de avanço
ncs=0.193*[1 5]        %Numerador de Avanço
dcs=[1 10]             %Denominador de Avanço
cs=tf(ncs,dcs)          %Func. de Transf. do Compensador de Avanço
cz=c2d(cs,t,'ZOH')     %Transformando a resposta do Avanço com c2d

```

Resultado:

n = 1.0000 22.2000 86.0000

d = 1 0 -1 0

npf = 380

dpf = 1 0 380

p = 380

$s^2 + 380$

Continuous-time transfer function.

pz = 0.4386 z + 0.4386

$z^2 - 1.123 z + 1$

Sample time: 0.05 seconds

Discrete-time transfer function.

Current plot held

$$ncs = 0.1930 \quad 0.9650$$

$$dcs = 1 \quad 10$$

$$cs = 0.193 s + 0.965$$

$$s + 10$$

Continuous-time transfer function.

$$cz = 0.193 z - 0.155$$

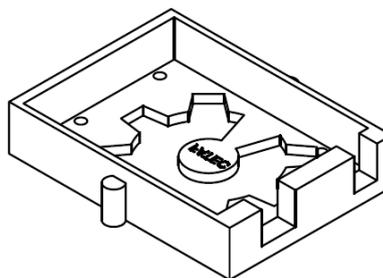
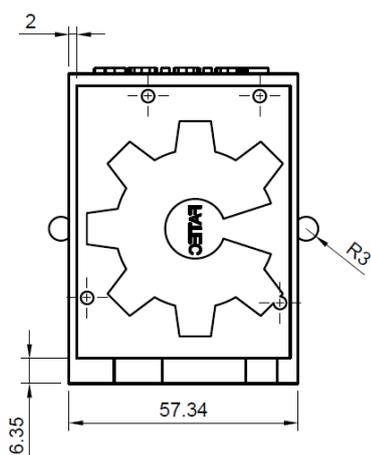
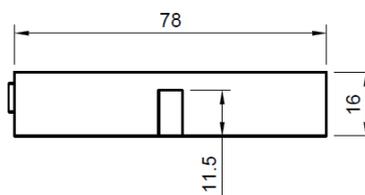
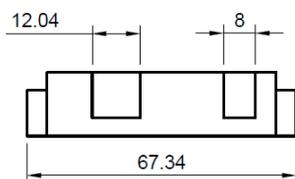
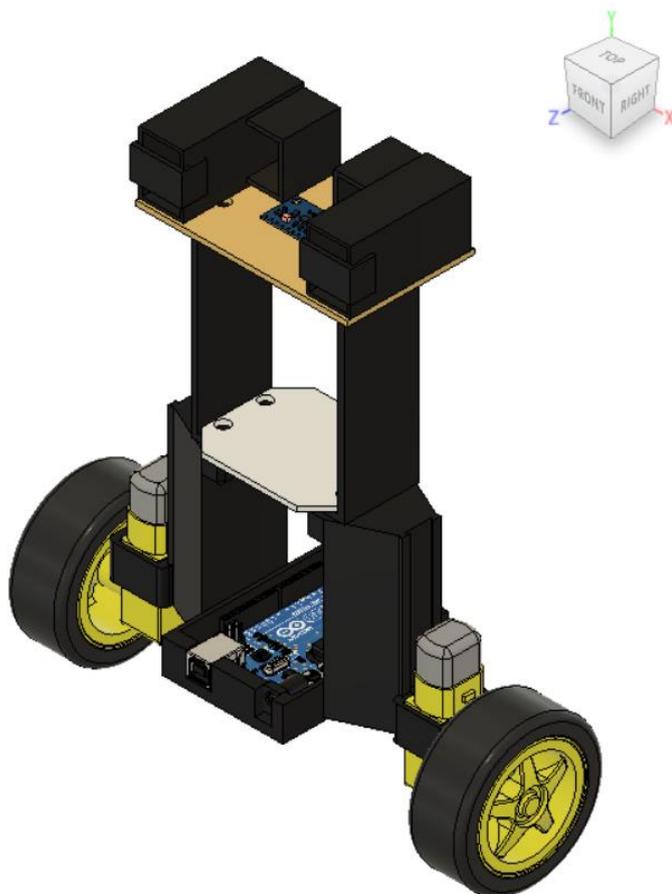
$$z - 0.6065$$

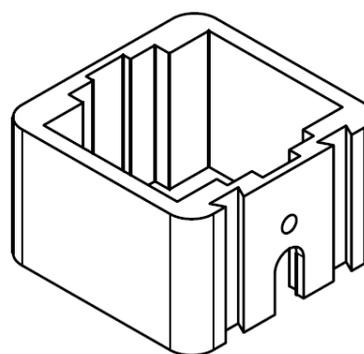
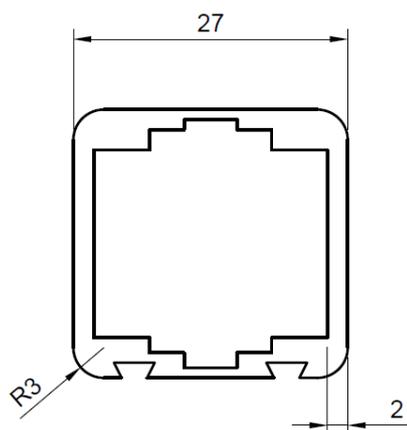
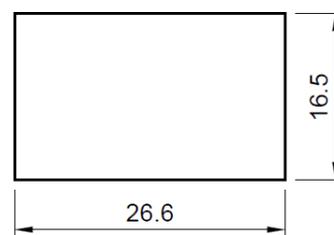
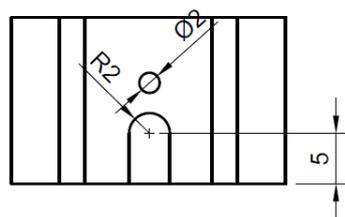
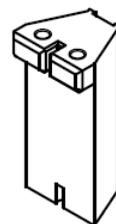
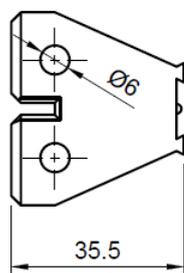
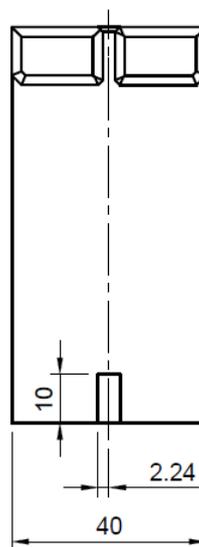
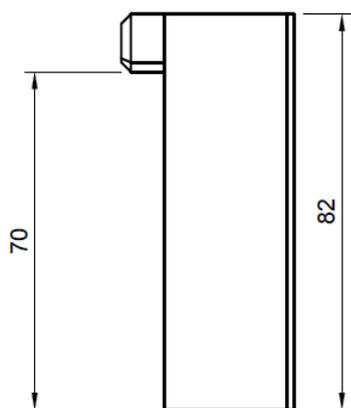
Sample time: 0.05 seconds

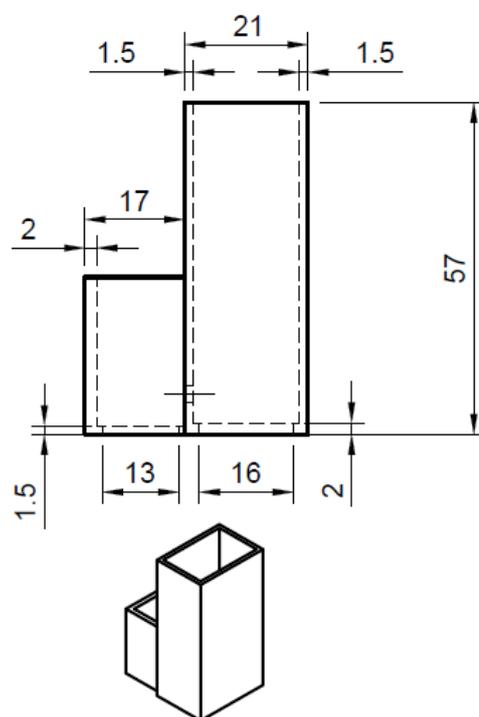
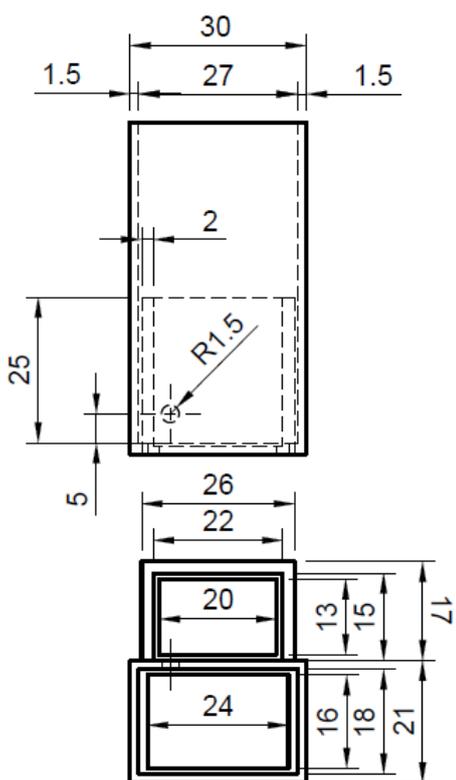
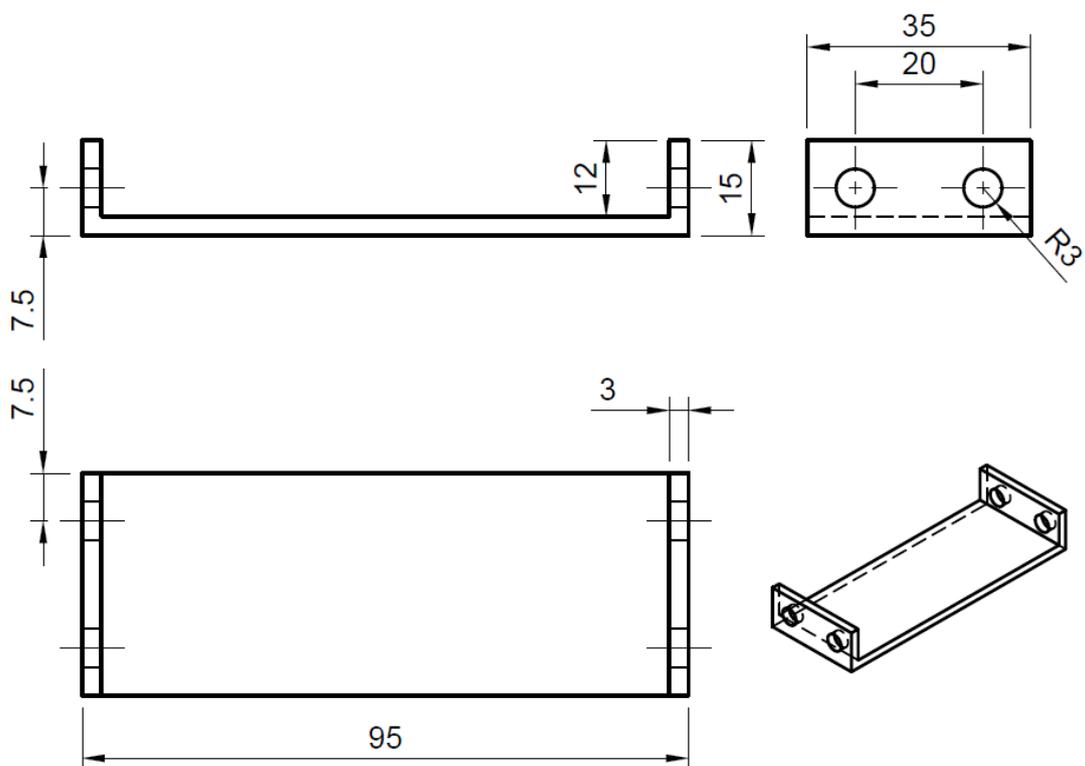
Discrete-time transfer function.

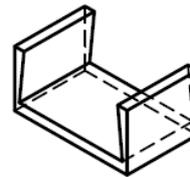
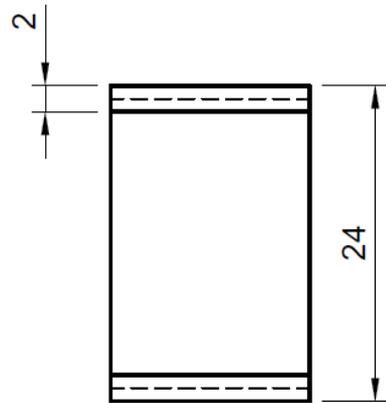
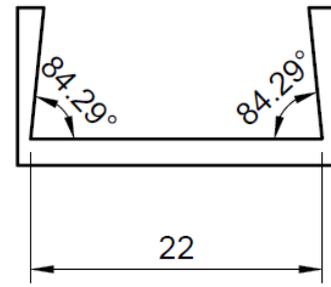
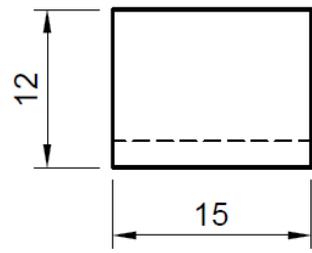
Current plot released

Apêndice B – Documentação da Estrutura Principal

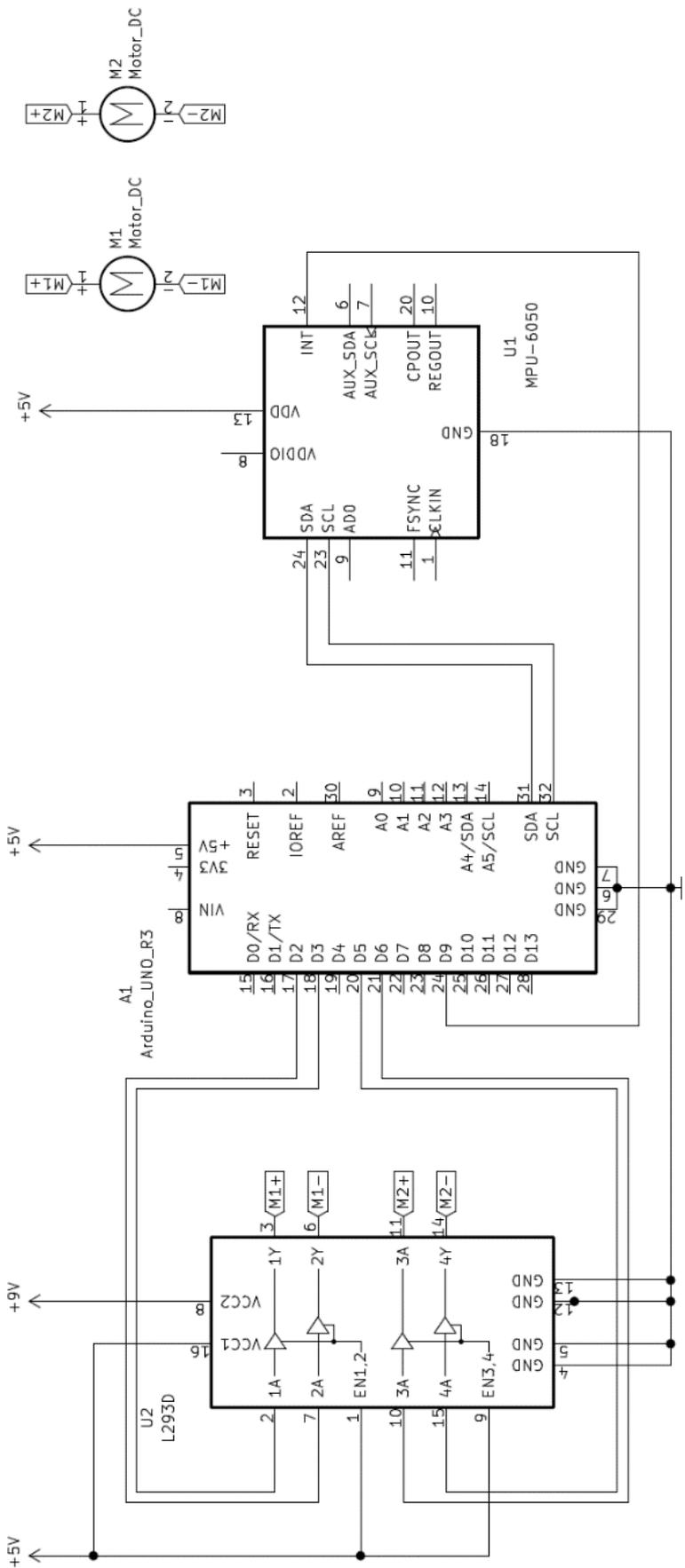




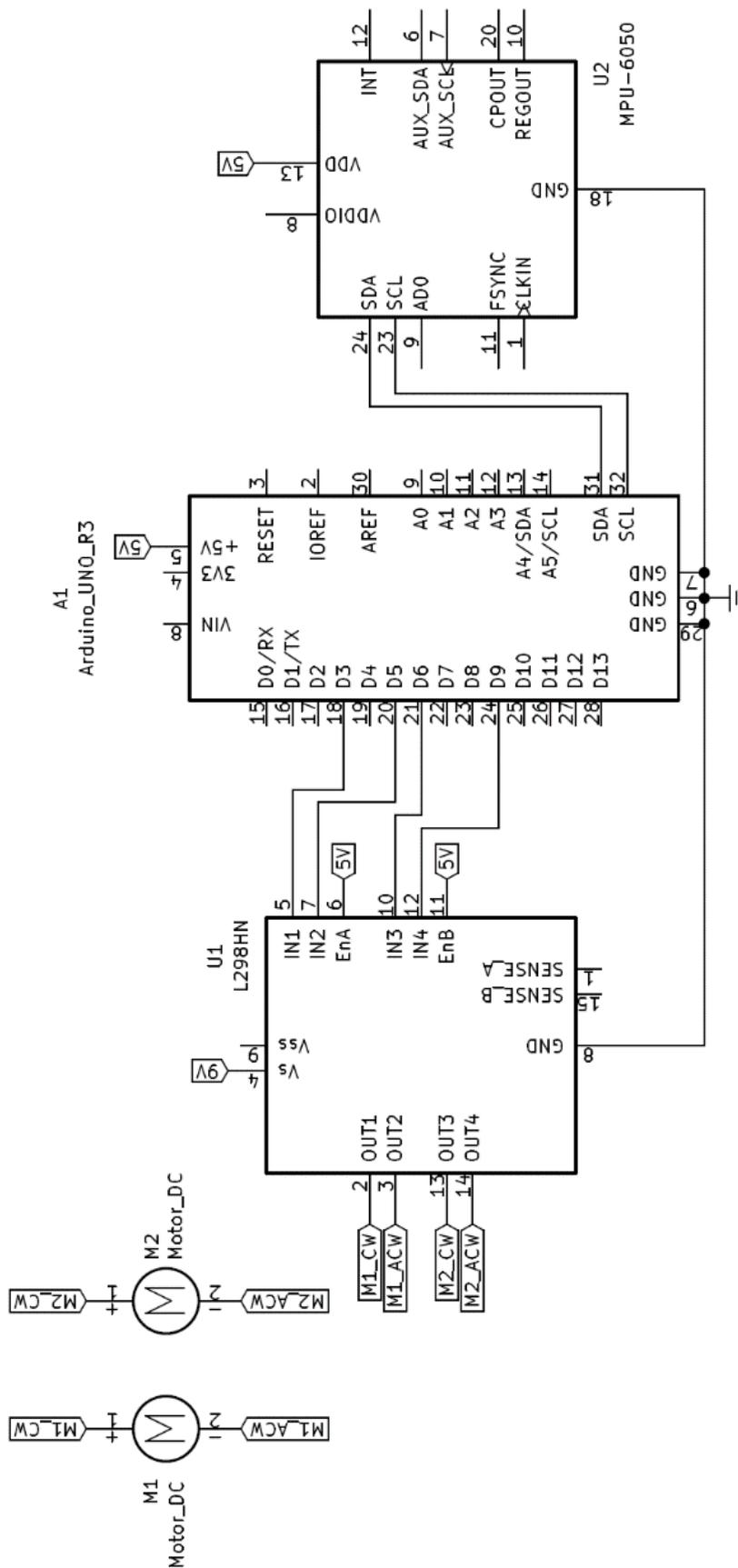




Apêndice C – Esquema Elétrico com L293D



Apêndice D – Esquema elétrico com ponte H L298N



Anexo A – Programa P.I.D. com MPU6050, Reginald Watson

//Conjunto de bibliotecas utilizado

```
#include <PID_v1.h>
#include <LMotorController.h>
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
```

```
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif
```

//Velocidade mínima de operação dos motores, relacionada à zona morta.
#define MIN_ABS_SPEED 60

MPU6050 mpu;//definindo a variável mpu como integrante da classe MPU6050

//Estado atual do MPU

```
bool dmpReady = false; // verifica se a função retornou um valor válido
uint8_t mpuintStatus; // carrega o estado da interrupção do MPU
uint8_t devStatus; // retorna o estado após cada operação (0 = sucesso, !0 = falha)
uint16_t packetSize; //tamanho esperado para o pacote do DMP (padrão de 42 bytes)
uint16_t fifoCount; // conta todos os bytes atuais dentro do FIFO
uint8_t fifoBuffer[64]; // vetor onde será armazenado o FIFO
```

// variáveis angulares extraídas do MPU

```
Quaternion q; // [w, x, y, z] contém os Quatérnios de espaço
VectorFloat gravity; // [x, y, z] vetor gravidade
float ypr[3]; // [yaw, pitch, roll] contém os valores de guinada/arfagem/rolamento, assim como o valor de gravidade
```

//variáveis de PID

```
double originalSetpoint = 175.8; //ângulo desejado
double setpoint = originalSetpoint; //variável de uso interno da biblioteca
double movingAngleOffset = 0.1; //correção de deriva
double input, output; //entrada e saída do PID
int moveState=0; //tipo de compensação : 0 = equilibrado; 1 = para trás; 2 = para frente
double Kp = 75; //constante de ganho proporcional
double Kd = 3.5; //constante de ganho diferencial
double Ki = 90; //constante de ganho integral
PID pid(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT); //chamada da função PID que realiza todas as contas explicadas na sessão de software
```

//fator de correção para possíveis diferenças entre os motores

```
double motorSpeedFactorLeft = 0.6;
double motorSpeedFactorRight = 0.5;
```

//Entradas do Driver

```
int ENA = 5;
int IN1 = 6;
int IN2 = 7;
int IN3 = 8;
int IN4 = 9;
int ENB = 10;
```

```

//chamada da função de controle do Driver
LMotorController motorController(ENA, IN1, IN2, ENB, IN3, IN4, motorSpeedFactorLeft, mo-
torSpeedFactorRight);

//inicialização dos Timers
long time1Hz = 0;
long time5Hz = 0;

volatile bool mpuInterrupt = false; // indicador de interrupção no MPU

void dmpDataReady()
{
  mpuInterrupt = true; //responde que o MPU preparou os dados
}

void setup()
{
  // iniciando comunicação I2C
  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();
    TWBR = 24; // 400kHz I2C clock (200kHz se a CPU for de 8MHz)
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
  #endif

  // iniciando comunicação Serial
  Serial.begin(115200); // taxa de baudrate
  while (!Serial); // Caso específico do arduino Leonardo

  // inicialização do MPU
  Serial.println(F("Initializing I2C devices..."));
  mpu.initialize(); //Chamada da função da biblioteca do MPU

  // verificando comunicação
  Serial.println(F("Testing device connections..."));
  Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050
connection failed"));

  // Carregando e configurando o DMP
  Serial.println(F("Initializing DMP..."));
  devStatus = mpu.dmpInitialize();

  // informações de offset do giroscópio
  mpu.setXGyroOffset(220);
  mpu.setYGyroOffset(76);
  mpu.setZGyroOffset(-85);
  mpu.setZAccelOffset(1788); // utilizando padrão de fábrica 1688

  // verificando se a comunicação ocorreu corretamente
  if (devStatus == 0)
  {
    // ligando o DMP
    Serial.println(F("Enabling DMP..."));
  }
}

```

```

mpu.setDMPEEnabled(true); // chamada da função de inicialização do DMP

// habilitando a interrupção do arduino
Serial.println(F("Enabling interrupt detection (Arduino external interrupt 0)..."));
attachInterrupt(0, dmpDataReady, RISING);
mpuIntStatus = mpu.getIntStatus(); //chamada da função que recebe o estado do pino
INT

// afirmando que o DMP está pronto para uso na void loop
Serial.println(F("DMP ready! Waiting for first interrupt..."));
dmpReady = true;

// medindo tamanho do pacote fornecido pelo DMP
packetSize = mpu.dmpGetFIFOpacketSize(); // chamada da função do Pacote

//configurando PID

pid.SetMode(AUTOMATIC); //chamada da função de modo de operação
pid.SetSampleTime(10); //nomeando tempo de amostragem
pid.SetOutputLimits(-255, 255); //limites para o teste de acordo com a utilização
}
else
{
  // Caso ocorra algum erro na comunicação
  // 1 = falha no carregamento da memória inicial
  // 2 = falha na atualização das configurações de DM
  Serial.print(F("DMP Initialization failed (code ");
  Serial.print(devStatus);
  Serial.println(F(")"));
}
}

void loop()
{
  // caso a configuração inicial tenha falhado, o programa será interrompido.
  if (!dmpReady) return;

  // O programa deve esperar até que o pacote do DMP esteja pronto ou uma interrupção
  ocorra
  while (!mpuInterrupt && fifoCount < packetSize)
  {
    //sem dados novos no MPU, os cálculos de PID serão realizados

    pid.Compute(); //chamada da função de cálculo
    motorController.move(output, MIN_ABS_SPEED); //aplicação do controle nos motores
  }

  // reinicia a variável de interrupção e recebe os bytes do pino INT
  mpuInterrupt = false;
  mpuIntStatus = mpu.getIntStatus(); //chamada da função que busca o estado do pino INT

  // medindo o FIFO atual

```

```

fifoCount = mpu.getFIFOCount();

// verificando por Overflow no pacote
if ((mpuIntStatus & 0x10) || fifoCount == 1024)
{
    // em caso de overflow, reinicia-se o FIFO
    mpu.resetFIFO(); //chamada da função de reset no FIFO
    Serial.println(F("FIFO overflow!"));

// em procedimento normal, verificar por novos pacotes de dados no MPU
}
else if (mpuIntStatus & 0x02)
{
    // esperando o tamanho correto do buffer
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

//leitura do pacote
    mpu.getFIFOBytes(fifoBuffer, packetSize); //chamada da função de leitura do DMP

// Verificando caso mais de um pacote tenha sido recebido antes que uma nova interrupção ocorra, isso permitirá a fluência de leitura.
    fifoCount -= packetSize;

//resgatando informações do MPU e alocando no buffer do FIFO
    mpu.dmpGetQuaternion(&q, fifoBuffer);    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    #if LOG_INPUT
//visualização serial, caso seja necessário monitoração
    Serial.print("ypr\t");
    Serial.print(ypr[0] * 180/M_PI);
    Serial.print("\t");
    Serial.print(ypr[1] * 180/M_PI);
    Serial.print("\t");
    Serial.println(ypr[2] * 180/M_PI);
    #endif
    input = ypr[1] * 180/M_PI + 180;
}
}

```