PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA

IMPLEMENTAÇÃO E CONTROLE DE UMA PLATAFORMA ESTABILIZADA ATRAVÉS DE SENSORES INERCIAIS

Porto Alegre, 28 de novembro de 2019.

Autor: Kelvin Soares da Silva
Pontifícia Universidade Católica do Rio Grande do Sul
Curso de Engenharia de Controle e Automação
Av. Ipiranga, 6681 - Prédio 30 - CEP: 90619-900 - Porto Alegre - RS - Brasil
Email: kelvin.silva@edu.pucrs.br

Orientador: Prof. Rafael da Silveira Castro
Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681 - Prédio 30 - Bloco A - Sala 226 - CEP: 90619-900 - Porto Alegre – RS - Brasil
Email: rafael.castro@pucrs.br

RESUMO

As plataformas inercialmente estabilizadas têm o objetivo de manter ou controlar a linha de visão de um objeto em relação ao espaço inercial e está presente em diversas aplicações, como câmeras, telescópios, antenas de comunicação e armamento (LIMA, 2013). Devido à necessidade de manter estabilizados produtos de alta sensibilidade, projetos de implementação de uma plataforma estabilizada tornam-se essenciais para que instrumentos de medições não se descalibre com a movimentação do transporte. O projeto apresenta uma solução a partir de sistemas embarcados de código livre, para a implementação de baixo custo. Considerando o objetivo em manter e controlar a linha de visão de um objeto em relação ao espaço inercial, os resultados alcançados foram satisfatórios e de grande aprendizado, e os mesmos são apresentados no fim do presente trabalho.

Palavras-chave: Plataforma estabilizada. Sistema de controle. Sensor inercial.

1 INTRODUÇÃO

O homem sempre buscou meios que facilitassem sua sobrevivência e busca por alimento, e essa constante busca por tecnologias e técnicas de trabalho proporcionaram um grande avanço para a humanidade. O grande salto tecnológico veio com a Revolução Industrial, trazendo novos processos e meios de produção em massa (Souza, 2019). Durante as grandes navegações, instrumentos de navegação como compassos e lunetas já estavam presentes no meio científico, no qual as estrelas auxiliavam na localização. Mais tarde, sistemas mais precisos, eficazes e tecnológicos surgiram, tais como o *Global Positioning System* (GPS), que funcionam basicamente através de sinais emitidos por satélites e a triangulação desses sinais. O presente trabalho demonstra uma plataforma inercial estabilizada e tem como objetivo manter e controlar a linha de visão de um objeto em relação ao espaço inercial.

1.1 Motivação

As plataformas estabilizadas são apontadas na literatura como estrutura de grande potencial para utilização em diversas aplicações, incluindo navios, robôs, lentes em câmeras portáteis, automóveis, sensores ou antenas e até mesmo satélites. Segundo Hilkert (2008), o telescópio espacial Hubble é uma plataforma giro-estabilizada, que foi projetada para manter-se constante com uma precisão de 10 milisegundos de arco. Desta maneira, permitem que os astros distantes sejam focalizados, possibilitando imagens com nitidez em alta resolução. Portanto, as plataformas estabilizadas são submetidas em aplicações de grandes variações durante as operações (Santos, 2007; Scheffe, 2002).

Acompanhando esse avanço tecnológico, os sistemas embarcados estão cada vez mais popularizados e acessíveis à sociedade, possibilitando realizar projetos com dispositivos para as mais variadas aplicações e com baixo custo, confiabilidade e facilidades, integrando com as diversas bibliotecas, códigos livres e softwares matemáticos como o Matlab.

Essas vantagens receberam uma atenção considerável no meio acadêmico e profissional, pois permitiram realizar projetos práticos de sistemas de controle mais acessíveis e simplificados.

1.2 Objetivo

Pretende-se com esse projeto aplicar os fundamentos da teoria de controle moderno e projetar uma plataforma capaz de estabilizar um objeto, utilizando dispositivos eletrônicos e eletromecânicos de baixo custo, a partir das informações angulares provindas dos sensores inerciais e corresponder de forma satisfatória às variações bruscas e/ou delicadas. O projeto

deverá ser simplificado e móvel para ser utilizado em demonstrações e reforço de estudo para fins educacionais ou em aplicações práticas de projeto.

1.3 Delimitações do Trabalho

Este trabalho utiliza a técnica da coleta das informações angulares propostas pelo autor Luis Lamas (2016) e a partir deste sinal, foram corrigidos os erros e ruídos dos sensores inerciais e projetado o sistema de controle para o ajuste de velocidade e sentido de rotação do servo motor.

Este projeto não inclui a compensação de um terceiro eixo, denominado YAW e não possui ajuste na compensação a partir da variação da massa do objeto a ser estabilizado. Também não foram avaliados os aspectos econômicos, visto que todos os componentes já apresentam um valor baixo agregado comparado a sua funcionalidade.

2 REVISÃO BIBLIOGRÁFICA

2.1 Gimbal como mecanismo de estabilização

Um gimbal está fortemente atrelado a um giroscópio e acelerômetro, conforme pode ser visto na Figura 1. A primeira descrição de um gimbal foi dada pelo engenheiro grego Filão de Bizâncio, cerca de 300 a.C., no qual ele cita que ao se girar um pote de tinta com uma abertura nas faces, a tinta permanece dentro do pote enquanto ele estiver girando. Embora as fontes desse experimento sejam escassas, seu design era aplicado à construção de balistas na Grécia antiga. Mais tarde surgiu a junta universal, baseada no design de um gimbal. Girolamo Cardano foi o primeiro a sugerir que o design da junta era capaz de transmitir potência motora, no ano de 1545. Na Europa a junta ficou conhecida como junta de Cardan. O mecanismo também é conhecido como junta de Hooke, pois em 1676 Robert Hooke construiu um protótipo funcional. Por fim, Henry Ford foi o pioneiro na utilização da junta em seus carros, sendo o mesmo quem apelidou de o mecanismo de junta universal.

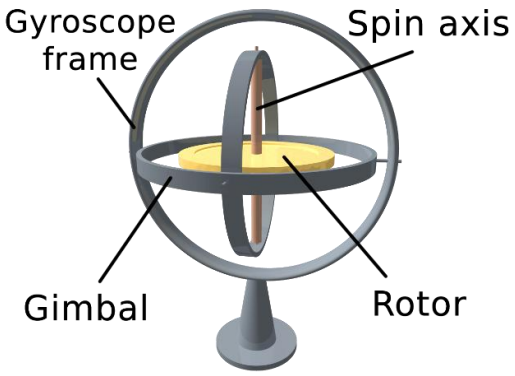

Figura 1 Construção do gimbal

Dado o conceito histórico do gimbal, pode-se ter a ideia da importância desse mecanismo. Atualmente é muito aplicado na indústria cinematográfica, no qual é utilizado na estabilização de câmeras, sistemas de filmagem por drones, que também utilizam este sistema de estabilização. Na Figura 2 é possível ver uma aplicação prática do mecanismo.

Figura 2 Aplicação do sistema gimbal para estabilização de câmeras.

2.2 Sistemas Embarcados

A união de software e hardware integrados em um sistema é chamado de sistema embarcado. Os sistemas embarcados são sistemas de computação com integração de hardware e software, e são projetados para executar uma função dedicada (Li; YAO, 2003). Com essas definições citadas acima, conclui-se que dispositivos como calculadora, máquina de lavar, telefone, caixa eletrônico são sistemas embarcados, pois possuem integração de hardware e software para um fim específico.

2.3 Arduino

Devido à necessidade de criar um produto com diversas funcionalidades integradas e de baixo custo, foi desenvolvido no ano de 2005 o controlador Arduino em Ivrea, na Itália para

ensinar programação aos estudantes de design. Como os estudantes não eram da área, o projeto foi elaborado com o intuito de ser um dispositivo de fácil programação e operação.

O Arduino baseia-se na plataforma *Wiring* para prototipagem eletrônica de software livre para o controle de interatividade entre dispositivos. É constituído de um ambiente de desenvolvimento integrado (IDE), linguagem de programação e o dispositivo microcontrolado de placa única. Os códigos de programação são comumente escritos a partir da linguagem Processing, essencialmente C/C++ (MEINECKE, JONAS. LOPES, PAULO).

O microcontrolador Arduino Mega 2560 dispõe de 54 pinos que configurados podem ser utilizados como entradas e saídas (E/S) digital e 16 entradas analógica de 10 bits, 4 portas de comunicação para a interface serial ou via USB, para a integração e programação e comunicação SPI e I2C (SOUZA, 2014).

Os dispositivos em questão possui o protocolo I2C (do inglês, Inter-Integrated Circuit) que transfere dados em 8 bits através de sua serial. O protocolo permite a conexão de diversos periféricos através do seu barramento tipo multi-master, possibilitando o endereçamento de cada dispositivo. Utilizam-se duas linhas bidirecionais, denominada SDA e SCL para o clock do barramento (NXP, 2014).

2.4 Unidade de medição inercial (IMU)

A unidade de medição inercial é o dispositivo capaz de precisar a posição e/ou orientação atual de um objeto através de sensores inerciais (SAHAWNEH; JARRAH, 2008). A IMU consiste de um sistema composto de acelerômetro, giroscópio e magnetômetro. A utilização conjunta de ambos os sistemas se faz necessária devido aos ruídos e a sensibilidade de translação produzidos pelo acelerômetro e a diferença entre o valor real e o da medição (drift), assim, se faz uma média aproximada do valor real diminuindo a sensibilidade a translações. (MOURA, 2013).

2.4.1 Acelerômetros

É um sensor que determina a aceleração ao longo de uma direção, sendo que a elaboração mais comum é a partir da variação da capacitância.

O funcionamento do sistema é a partir da movimentação de uma massa de prova, suspensa por molas. Assim, de acordo com a distância entre a parte fixa e a parte móvel, tem-se a variação da capacitância, sendo C1 a diferença entre a parte fixa e a móvel quando deslocada à esquerda e C2 quando deslocada à direita. Como demonstra a figura 3, a capacitância proporcional é a diferença das duas capacitâncias (C1 e C2).

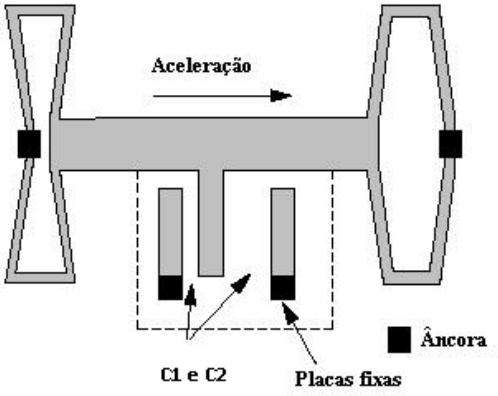

Figura 3 Construção do acelerômetro

O cálculo dos ângulos de arfagem (pitch, $\emptyset$) e rolamento (roll, θ), demonstrados nas Equações abaixo, é gerado pelo sensor através da aceleração da gravidade a partir dos valores de leitura em cada eixo (X, Y e Z) (REIS et al., 2014). Por determinação dos eixos de rotação definimos qual deverá ser o roll e o pitch (MOURA, 2013).

$$\emptyset = atan2(Y,(\sqrt{X2+Z2}))$$
$$\theta = atan2(X,(\sqrt{Y2+Z2}))$$

2.4.2 Giroscópio

Um dos equipamentos que mais revolucionaram os meios de navegação e localização foi o giroscópio, inventado por Jean Bernard Léon Foucault em 1850 [3]. O objetivo de Foucault era mostrar que a terra girava em torno do seu próprio eixo, e seu princípio de funcionamento pode ser comparado a uma roda girando em um eixo, que se opõe à mudança de direção. Esta oposição à mudança de direção pode ser melhor entendida com o giro de um peão, que ao girar sobre uma superfície plana a roda permanece com seu eixo vertical, porém, ao se inclinar, o plano do eixo tende a permanecer apontando para a mesma direção vertical de antes, conforme pode ser visualizado na Figura 4. Este funcionamento proporcionou um grande avanço aos sistemas de navegação e balanceamento, usado inclusive em satélites, sondas espaciais, navios, e outros meios de navegação.

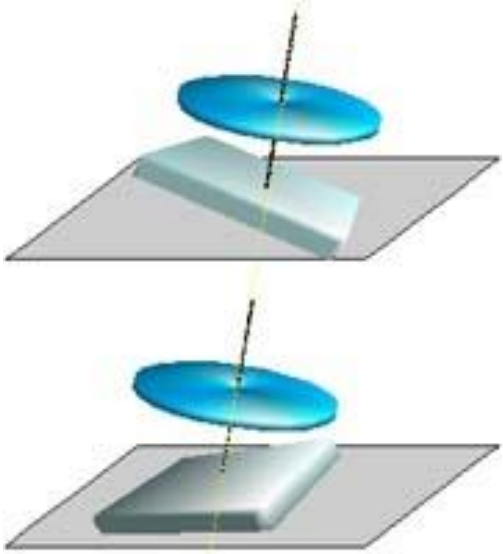

Figura 4 Princípio de funcionamento de um giroscópio.

São mecanismos que mede a rotação em relação ao seu próprio eixo com base no princípio de conservação angular para a orientação do objeto. O giroscópio mede a velocidade angular a partir da integração da posição angular. A Equação abaixo demonstra essa integral, composta por wi (velocidade angular), xi (posição angular) e t (intervalo de tempo) (REIS et al., 2014).

$$xi=xi-1+ wi \text{ t} \quad (3)$$

A construção interna mais utilizada é feita por microestruturas vibrantes que quando movimentada gera um sinal de saída devido ao distanciamento ou aproximação entre as aletas fixas (subtrato do CI) e o corpo de prova. A deflexão das hastes é proporcional a velocidade angular devido a inércia das vibrações a partir da rotação do sensor (MOURA, 2013).

2.5 Erros e ruídos dos sensores inerciais

2.5.1 Ruído Branco (White noise/Random Walk)

Este erro é gerado a partir da integração da parte do ruído presente nos sinais dos sensores inerciais devido a variação em um curto período no sinal da saída, que produz uma alta frequência quando os sensores estão em repouso. A saída é perturbada por um ruído termomecânico com flutuação presente, com uma taxa maior que a taxa amostrada pelo sensor. O white noise nos giroscópios é dado por “graus/√h” e “g/√hz” para os acelerômetros. (SANTANA, 2011).

2.5.2 Polarização constante (Bias)

O erro Bias é entendido como um sinal constante ou que possui uma variação lenta independente do sinal de entrada. Essa constância tem o seu valor médio alterado em algumas circunstâncias, como na religação do equipamento. Esse erro angular em (º/h) cresce linearmente com o tempo, através da equação $\emptyset(t) = Ɛbg.t$, sendo o tempo t o de integração. (SANTANA, 2011).

2.5.3 Deriva térmica (Drift)

Trata-se de um erro em virtude da temperatura (mudanças climáticas ou auto-aquecimento do próprio integrado) que altera a magnitude do Bias (Ɛbg) e que não está incluída na parcela determinada quando o sensor está em repouso, assim, deve-se compensar na lógica de programação no controle. A relação entre bias (Ɛbg) e a temperatura causa um erro de orientação não-linear. Atualmente as IMUs dispõe internamente de sensores de temperaturas para que possamos corrigir o sinal de saída a fim de compensar as variações do Bias (SANTANA, 2011).

2.5.4 Erro de quantização

Os dados gerados pela IMU são digitalizados e fornecidos em instantes discretos de tempo, devido a isso, a saída gera um ruído branco proporcional à magnitude quantizada. Este valor é diretamente proporcional a aproximação da ordem de grandeza do sinal elétrico devido à taxa amostrada referente ao sensor ser menor que a taxa osciladora do ruído (SANTANA, 2011).

2.5.5 Desalinhamento

Os acelerômetros e giroscópios realizam suas medições a partir de uma base ortogonal perfeitamente alinhada, porém, devido ao desalinhamento estrutural dos eixos do projeto, definimos constantes para cada erro de alinhamento relacionando aos próprios eixos da plataforma (SANTANA, 2011).

2.5.6 Erros de calibragem

Referente ao conjuntos de erros envolvendo a linearidade dos sensores angulares, fator de escala e desalinhamento entre eixos. O erro de calibragem também acrescenta uma parcela de Bias (Ɛbg), no giroscópio quando os sensores estão rotacionando esse acréscimo pode ser observado e no acelerômetro essa parcela de Bias (Ɛbg) pode ser observada no estado estacionário devido a aceleração gravitacional (g). O erro soma uma parcela na deriva angular

Ø(t) com uma relação direta entre o tempo em que o movimento permanece e a magnitude da velocidade angular (Wib(t)) (WOODMAN, 2007).

2.6 Filtro complementar

O Filtro Complementar CF (do inglês, Complementary Filter) combina os dados dos sensores e atua para diminuir os ruídos, e integra um filtro passa-baixas e um filtro passa-altas.

Com isso, o CF elimina os dois problemas na aquisição dos dados dos sensores, ruído e acuracidade. O acelerômetro insere muito ruído na leitura e para resolvermos utilizamos o filtro passa-baixas. O giroscópio, quando utilizado por um longo período, não retorna ao valor zero, então, deve ser utilizado um filtro passa-altas para melhorar a acuracidade (TAKAHASHI, 2016).

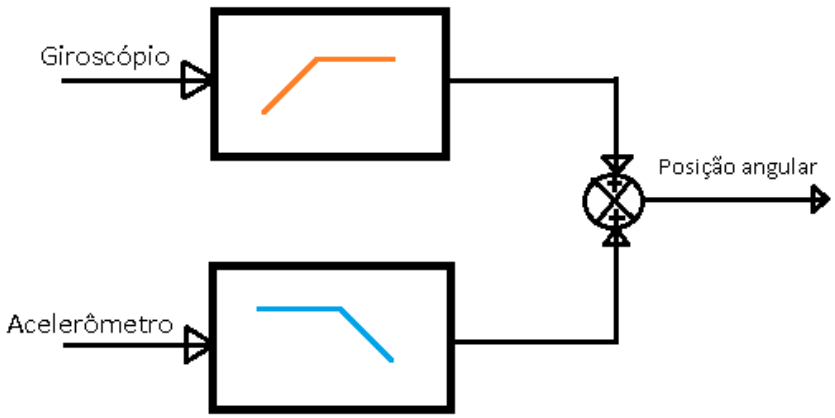

Figura 5 Filtro complementar alternativo

A Equação abaixo expressa o CF:

$\theta = A \cdot (\theta + \Omega \cdot dt) + (1 - A) \cdot Aa$

Onde P1 e P2 são os coeficientes do filtro, Ang é a variável do ângulo da fusão dos dados, Gv é a velocidade angular data pelo giroscópio, dt é a constante de tempo para a integração e Aa é o ângulo dado pelo acelerômetro (BRAGA, CAVALCANTE, SÁ, 2017).

2.7 Servo motor 360º por pulsos PWM

São dispositivos eletromecânicos para a movimentação de forma precisa e controlada, o mesmo deve manter o seu eixo estático mesmo sendo submetido a uma força externa quando está atuando ou em seu estado de repouso, mas para isso deve estar sendo energizado.

A utilização desse dispositivo nos permite um range maior de velocidade e que o torque de saída se mantenha constante independente de sua velocidade, além, de proporcionar um torque no eixo maior em relação a outros motores (MORAES, 2011).

O Servo motor é composto, demonstrado na figura 6, por carcaça, uma caixa para acomodar as diversas partes do dispositivo, o motor, que é o dispositivo elétrico para a movimentação do eixo principal, transmissão mecânica, composta por um sistema de engrenagens, ou por polias e correias, que nos permitem que as grandezas de torque, posição, velocidade, aceleração e desaceleração tornem-se compatíveis com a necessidade da carga, circuito de controle, responsável por a identificação dos pulsos Pulse-Width Modulation (PWM) e potenciômetro de ajuste, utilizado para definir a posição de repouso. Para permitir um sistema de rotação contínua, é retirado o potenciômetro do eixo, desta forma o mesmo fica livre para rotação em 360 graus, não mais limitando-o a 180 graus para posicionamento.

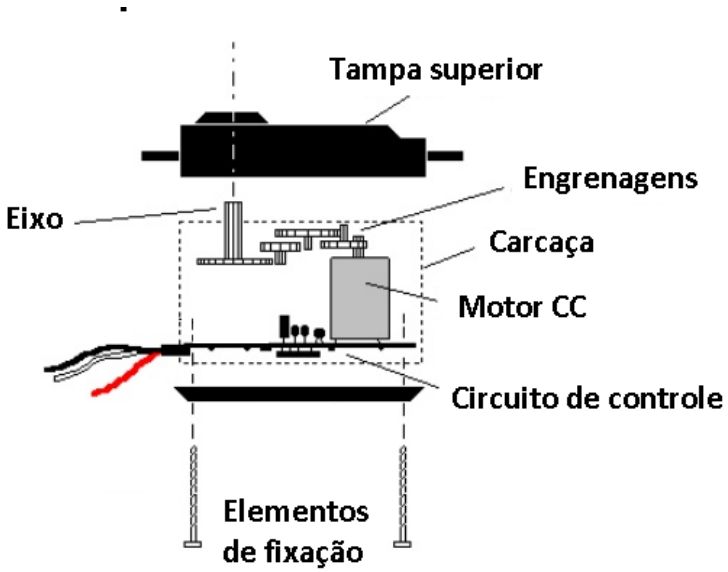

Figura 6 Estrutura do servo motor

Para o sinal de controle, conforme a figura 7, o sentido e a velocidade são realizados através do tempo em que o sinal permanece em nível lógico alto (variado de 1ms a 2ms) dentro de um intervalo de 20ms, com uma frequência de 50 Hz (MOURA, 2013). Uma largura de pulso de 1,5 ms fará o eixo parar de se mover, com 1 ms o eixo moverá em sua velocidade máxima para o sentido anti-horário (CCW) e um trem de pulsos de 2 ms fará o eixo se mover com velocidade máxima para o sentido horário (CW). Assim, o ajuste de velocidade é dado através do valor intermédio de 1,5 ms aos seus valores extremos, 1 ms e 2 ms (DRONEBOT, 2018).

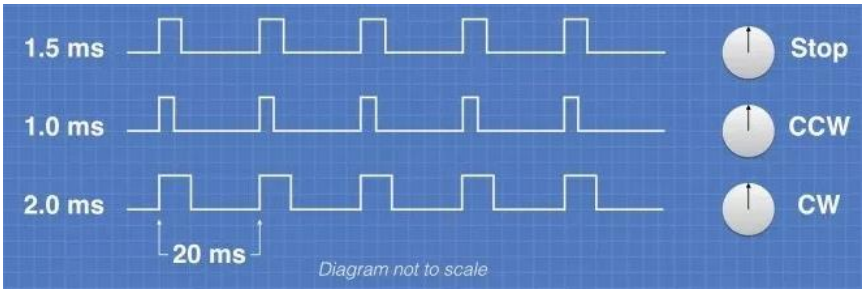

Figura 7 Movimentação por pulsos PWM

2.8 Sistema de controle

2.8.1 Controle através do espaço de estados

Para sistemas denominados MIMO (do inglês, múltiplas entradas e múltiplas saídas) os quais são mais complexos do que os habituais que relacionam uma entrada e uma saída ou uma função de transferência, necessitamos utilizar ferramentas computacionais para reduzir a complexidade das expresões matemática.

Para a representação de sistemas no espaço de estados, utilizamos matrizes. Desta maneira, através de matrizes canônicas conseguimos transformar as funções de transferências em espaço de estado (OGATA, 2010).

Define-se que um sistema é totalmente controlável no espaço de estados se o posto da matriz de controlabilidade Cx é igual a dimensão da matriz A (OGATA, 2010), conforme apresentado abaixo:

$$Cx = [B \vdots A.B \vdots \cdots \vdots A\,n{-}1\,.\,B]$$

A observabilidade permite estimar estados no qual não dispomos acesso. Um sistema é observável se o posto da matriz é igual a ordem n da Matriz A, onde Ox é mostrado abaixo:

$$Ox = [\ C\ C.A \vdots C.A\,n{-}1$$

2.8.2 Matriz de ganho K por alocação de pólos

Ao inserir uma matriz de realimentação, conforme e figura 8, obtemos o sistema de controle, caso o mesmo seja dito controlável, assim tornamos o sistema assintóticamente estável quando definimos um ganho K adequado. A partir disso, obtemos as características de estabilidade e de transição através de A-B.K.

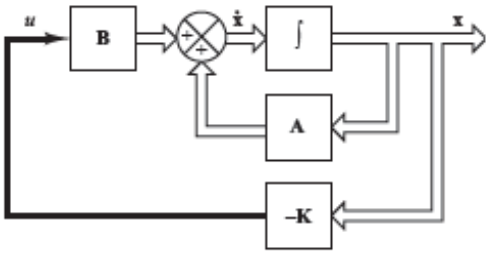

Figura 8 Sistema de realimentação de estados

3 METODOLOGIA

A seguir serão demonstrados os métodos utilizados para a construção do protótipo do gimbal com o enfoque no sistema de controle, visto que as ferramentas, como as bibliotecas wire.h e Servo.h, utilizadas no desenvolvimento do código de programação estão largamente disponibilizadas na internet e podem facilmente ser consultadas.

3.1 Estrutura do protótipo

Foi realizado um esboço do protótipo no software Sketchup da empresa At Last e desenhado o modelo no software AutoCad da empresa Autodesk, Inc. no formato .dxf e idealizado a partir dos cortes a laser em MDF no laboratório de fabricação LABFAB da PUCRS.

O protótipo é capaz de se mover livremente no eixo X e no eixo Y (ambos com 360º de liberdade) a partir de uma estrutura leve e rígida para que o sistema eletromecânico possa atuar de forma suave e precisa para compensar a diferença angular.

Foram definidas as trajetórias dos fios elétricos do controlador aos periféricos visando reduzir a interferência da soma das massas e da tração dos condutores, para evitar maiores esforços mecânicos na atuação dos servos motores. O compartimento do controlador, placa impressa, bateria e outros periféricos, foi definido para fácil manutenção, consulta e proteção dos dispositivos.

3.2 Filtro complementar

Existem vários parâmetros de sensibilidade que variam o valor da saída do sensor, alterando a validade do cálculo, mesmo quando o dispositivo encontra-se em repouso, causando incertezas da medida devido aos denominados erros de processo.

Para que possamos utilizar a fusão dos sensores, utiliza-se na lógica de programação um CF, a partir de um passa-baixas para eliminar os ruídos captados através do acelerômetro e um filtro passa-altas para melhorar a acuracidade provindo do giroscópio.

3.3 Calibrações

A placa MPU-6050 e os servos motores devem ser previamente calibrados para o melhor funcionamento do conjunto. Desta forma possibilita maior confiabilidade e um controle mais eficaz. Para isso, foram utilizados códigos de programação isolados para realizar as calibrações de cada dispositivo separadamente.

3.3.1 Calibração do acelerômetro da placa MPU-6050

Em virtude dos erros de alinhamento entre a plataforma estabilizadora e o posicionamento da placa MPU-6050 com a LOS, foram definidos valores nas variáveis de pitch0 e roll0 com toda a estrutura posicionada paralela ao chão, e retirado a média para definir os valores nas variáveis acima citada.

3.3.2 Calibração do giroscópio da placa MPU-6050

Como o acelerômetro, o giroscópio deve ser calibrado para compensar o erro de posicionamento estrutural. Assim o processo para definir as variáveis gyrox0 e gyroy0, novamente utiliza-se o valor médio do erro coletado a partir dos valores das variáveis Gy[0] e Gy[1] mantendo o sensor imóvel.

3.3.3 Calibração dos Servo motores

Para calibrar esse dispositivo, inserimos o valor de 1500 microsegundos na função denominada *writeMicroseconds* correspondente ao objeto servo1 ou servo2, referente ao tempo do pulso em nível lógico alto na porta, e ajustamos o potenciômetro para um sentido até parar o eixo; caso o mesmo movimente-se mais rápido, ajustamos o potenciômetro para o outro sentido.

3.4 Sistema de controle

Para melhor estabilizar do objeto, projeta-se o sistema de controle no intuito de atuar mais rapidamente quando a diferença angular (erro) for grande e mais lentamente quando a diferença angular for menor. Deste modo, conforme o erro diminui, a velocidade também diminuirá.

O controle será em malha aberta, visto que o erro de inclinação de 6 DOF do MPU-6050 e da LOS possibilita um ótimo controle utilizando uma plataforma quadrada de 0,2 m.

3.4.1 Identificação do Pitch

Para determinarmos o controle do Pitch inserimos um pulso com duração de 1,2 segundos e uma amplitude de 80 e realizamos a coleta das informações das variáveis Control[0] e Angle[0] a partir da serial do Arduino IDE. Com a ferramenta *Ident* do Matlab extraímos a função de transferência (FT), passamos para o espaço de estados e determinamos os autovalores para então projetar o sistema de controle de pitch.

3.4.2 Identificação do Roll

Da mesma forma que o controle do Pitch, para o controle do Roll foi projetado um pulso com a mesma duração e amplitude, que a partir da coleta e análise dos dados nas variáveis Control[1] e Angle[1], importamos para o Ident do Matlab e extraímos a FT. Posteriormente utilizamos o espaço de estados para encontrar os autovalores para determinar o sistema de controle para o Roll.

4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

4.1 Estrutura do protótipo

A figura 9 representa o esboço com vista superior do protótipo no software Sketchup. Desta maneira, a partir da movimentação operacional, verificou-se a eficiência no deslocamento dos eixos pitch e roll.

Figura 9 Modelo do Gimbal

A figura 10, projetada no AutoCad e enviada no formato.dxf para serem idealizados os cortes a laser em MDF no laboratório de fabricação.

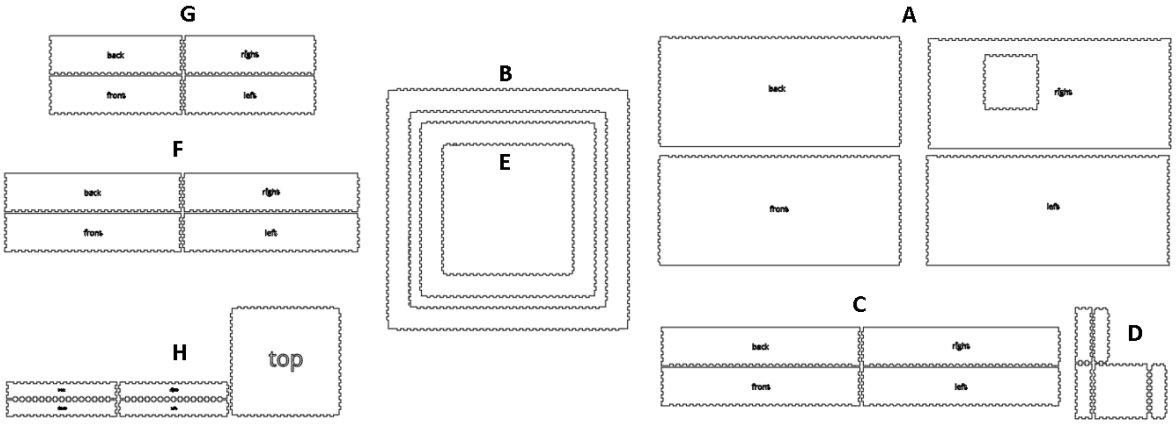

Figura 10 Cortes da construção do protótipo

Descrição das partes:

A - Paredes externas da estrutura do gimbal;

B - Tampa para a estrutura externa;

C - Paredes internas da estrutura do gimbal;

D - Gaveta para alocação de parte dos dispositivos;

E - Tampa para a estrutura do eixo de fora;

F - Parede externa da estrutura do eixo de dentro;

G - Parede interna da estrutura do eixo de dentro;

H - Plataforma de estabilização.

A figura 11 apresenta o protótipo montado e pronto para a operação, visto que as dimensões podem proporcionalmente serem ajustadas para demais fins, respeitando as limitações dos servo motores utilizados.

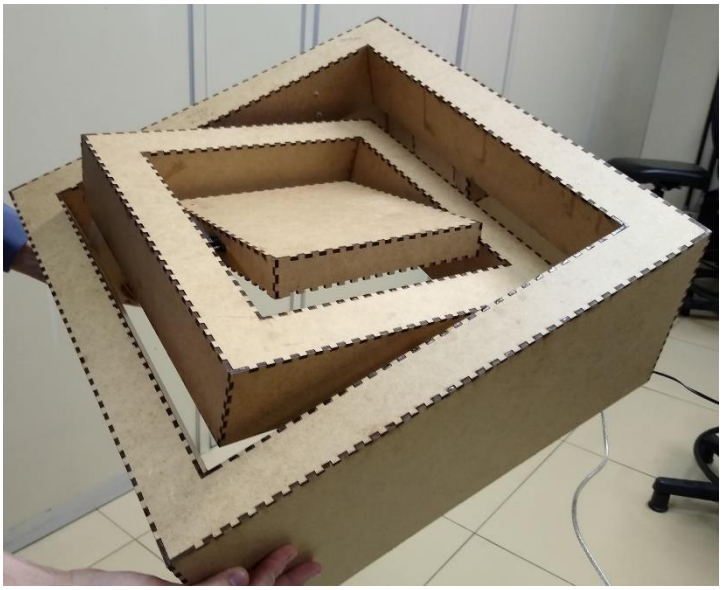

Figura 11 Protótipo montado

4.2 Parâmetros do filtro complementar

Conforme a figura 12, definimos a variável denominada "A" para inserir um valor para os coeficientes do filtro.

```
//Aplicacao do Filtro Complementar
Angle[0] = A * (Angle[0] + (Gy[0] - gyrox0) * dt) + (1 - A) * (Acc[0] - pitch0);
Angle[1] = A * (Angle[1] + (Gy[1] - gyroy0) * dt) + (1 - A) * (Acc[1] - roll0);
```

Figura 12 Aplicação do filtro complementar

O tempo de amostragem utilizado foi de dt = 0,04s e uma constante de tempo $\tau = 1$, obtém-se um valor de $\alpha \approx 0.96$. Esse valor será utilizado durante todo este trabalho.

Utilizando esse valor para α, chega-se a uma medição mais precisa e menos ruidosa. Outra abordagem mais robusta, porém mais complexa, é a utilização do Filtro de Kalman.

4.3 Calibrações

4.3.1 Calibração do acelerômetro da placa MPU-6050

Com os eixos em paralelo à linha do horizonte, realizou-se uma média das amostras dos ângulos no Matlab (utilizando a função mean) a partir dos valores das variáveis de pitch (Acc[0] e Acc[1]), através da comunicação serial do Arduino IDE com o sistema mecânico em repouso para definir o valor do erros nas variáveis pitch0 e roll0, conforme o código de programação no apêndice.

4.3.2 Calibração do giroscópio da placa MPU-6050

Da mesma forma que a calibração do acelerômetro, a calibração do giroscópio é realizada a partir da média, porémessa calibração é realizada a partir das variáveis Gy[0] e Gy[1], conforme o código de programação no apêndice.

4.3.3 Calibração dos Servo motores

O valor em microsegundos referente ao tempo do pulso em nível lógico alto nas saídas PWM no pino 10 e pino 11. O valor de 1500 us definido em ambas as portas é um processo de calibração e, caso o eixo se movimente, o parafuso do potenciômetro deve ser ajustado até parar o eixo do dispositivo.

Esse valor foi definido nas variáveis motPitch0 e motRoll0 no programa principal, de forma a serem reajustadas sempre que houver uma movimentação inesperada de algum dos servo motores quando o conjunto mecânico estiver em repouso, de forma que o eixo pare, pois torna-se inconveniente realizar esse ajuste através dos potenciômetros de cada servo motor.

4.4 Sistema de controle

As leis de controles para a atuação em Pitch e Roll mostradas na figura 13 foram determinadas a partir dos autovalores de cada subsistema com o sinal invertido, no intuito de realizar a compensação da diferença angular e não a soma delas. Desta forma, a lei de controle denominada Control[0] realiza a compensação de pitch e a lei de controle Control[1] atua sobre o erro de roll.

```
// Leis de Controle
  Control[0] = -5.8226*Angle[0]-0.1344*Gy[0];
  Control[1] = -10.7688*Angle[1]-0.1496*Gy[1];
```

Figura 13 Lei de controle

4.4.1 Identificação do Pitch

A identificação do controle do Pitch foi realizada a partir da variável denominada Control[0] e da variável Angle[0]. A partir do tempo de amostragem e tamanho da amostra, determinamos o intervalo entre as amostras, assim, conseguimos plotar o degrau de entrada na variável u0 e a rampa de saída na variável y0, comparando a atuação do servo motor e a respectiva diferença angular a partir do degrau, para então determinar a dinâmica desse subsistema, conforme a figura 14.

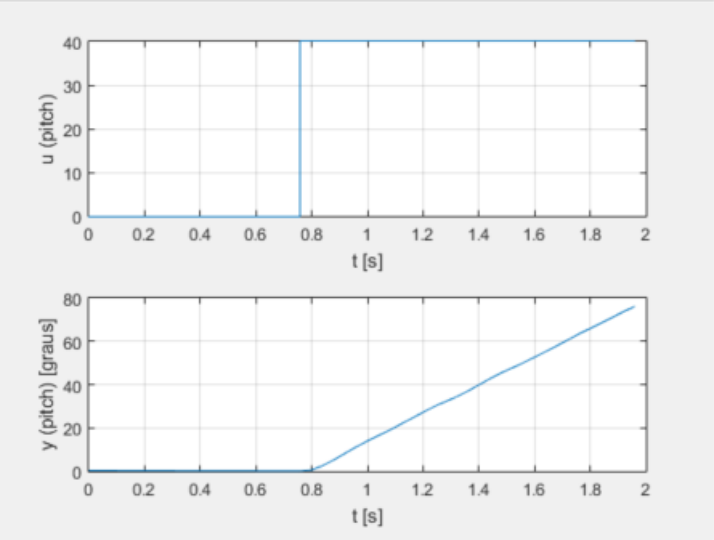

Figura 14 Identificação do pitch

A ferramenta ident do Matlab dispõe de um sistema de identificação e que a partir da relação das variáveis u0 e y0 foi gerado o modelo de saída correspondente a dinâmica do pitch com um determinado valor da estimativa em porcentagem, conforme demonstrado na figura 15, e a função de transferência mostrada na figura 16.

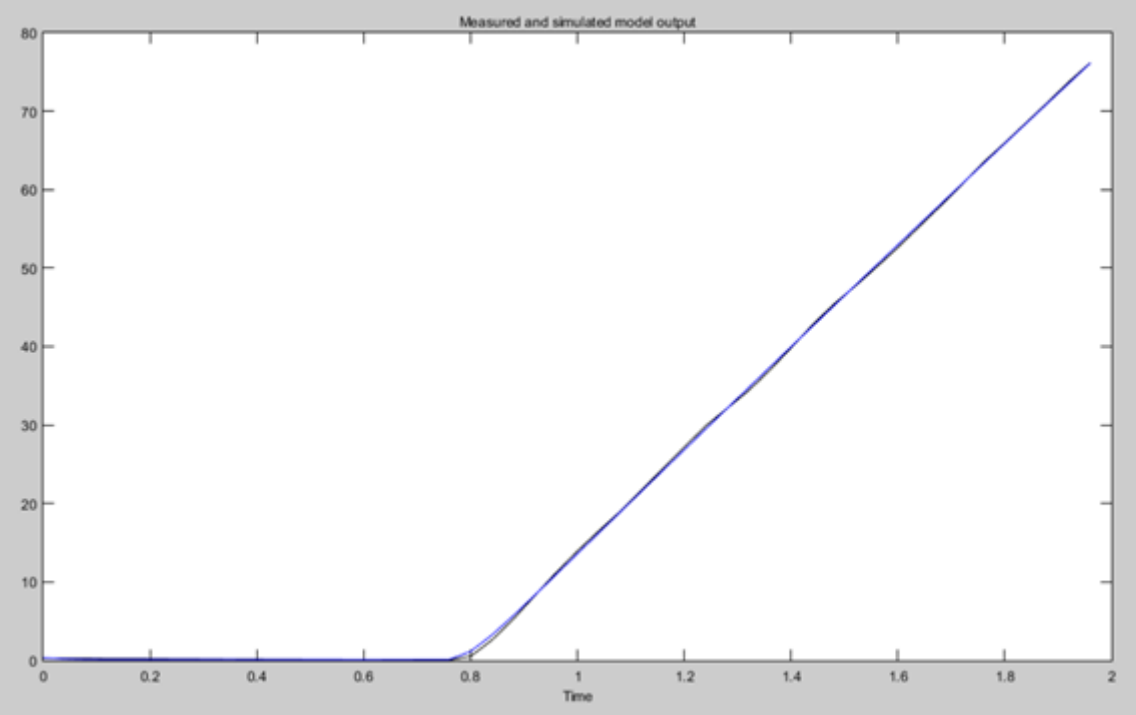

Figura 15 Comparação do modelo pitch projetado

```
Gs =

          43.77
  ---------------------
  s^2 + 26.12 s + 1.168

Continuous-time transfer function.
```

Figura 16 Função de transferência do pitch

A partir da FT, mostrada na figura 16, obtemos o conjunto de variáveis para as matrizes A, B, C e D do espaço de estado para que possamos utilizar a função acker do Matlab e determinar os autovalores de forma a montar um sistema de controle através da função Control[0] para compensar a diferença do pitch.

4.4.2 Identificação do Roll

A identificação do modelo de controle do Roll foi idealizada a partir da variável Control[1] e da variável Angle[1]. A figura 17 relaciona um degrau de entrada na variável u1 e a rampa de saída na variável y1 a partir da relação entre o servo motor e a respectiva diferença angular com a LOS. Desta forma é possível determinar a dinâmica desse subsistema.

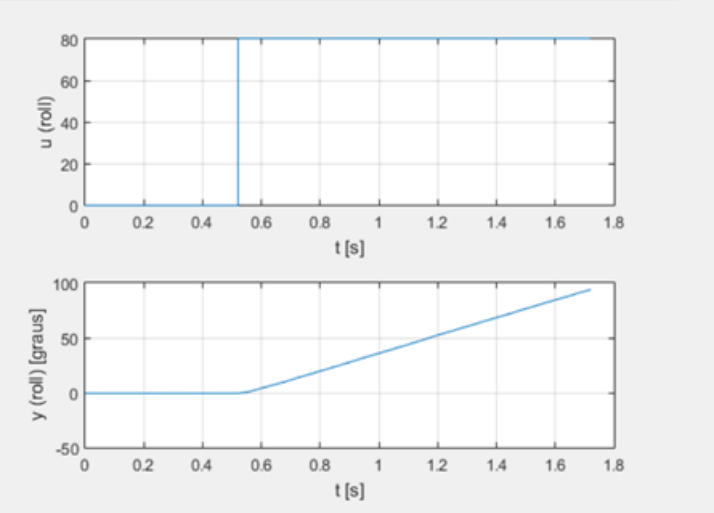

Figura 17 Identificação do modelo do roll

Utilizando o ident do Matlab, importamos as variáveis u1 e y1 e identificamos o modelo de saída correspondente a dinâmica do roll com um valor estimativo em porcentagem, conforme demonstrada na figura 18.

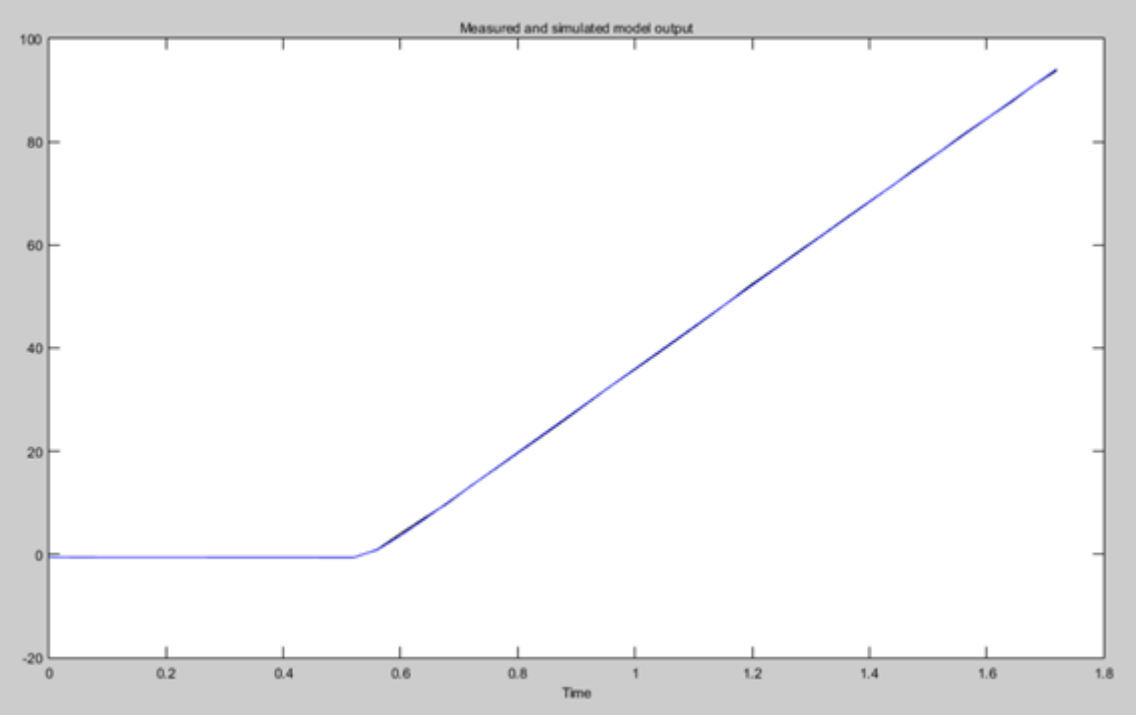

Figura 18 Comparação do modelo roll projetado

A figura 19 mostra a FT do controle do roll e a partir dela extraiu-se o conjunto de variáveis para as matrizes A, B, C e D do espaço de estado, da mesma forma que o controle do pitch.

Assim, a partir da função acker do Matlab, determina-se os autovalores para montar o sistema de controle a partir da função Control[1], para atuar sobre a diferença de roll.

```
Gs =

          33.66
-----------------------
s^2 + 33.06 s + 0.3728
```

Figura 19 Função de transferência de roll

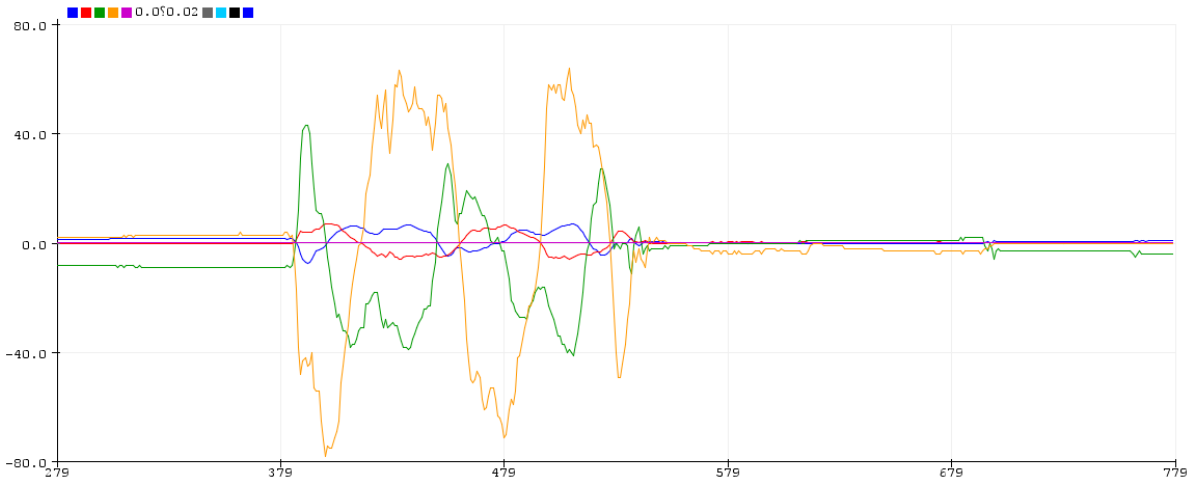

Figura 20 Comparação entre angulo e ganho do controle

5 CONCLUSÃO

O objetivo do projeto era a implementação do sistema de controle da plataforma estabilizadora avalizando a sua viabilidade. O controle do protótipo mostrou-se eficiente para a tarefa de estabilização de objetos. Naturalmente esta aplicação limita-se a objetos com baixa massa para que os servo motores possam responder bem dentro das suas limitações de torque.

A atuação do sistema de controle apresentou um comportamento suavizado, mesmo exercendo variações bruscas na plataforma. Apresenta-se um algoritmo de controle simplificado para implementação e conceito em futuras aplicações em projetos da área.

Com o desenvolvimento e acessibilidade dos embarcados e acessórios, ligada a crescentes publicações de plataformas open source , em breve aumentará a demanda de projetos DIY (do inglês, faça você mesmo). No entanto, para o desenvolvimento do presente trabalho, um nível mais avançado de entendimento técnico é necessário em virtude das particularidades na

identificação comportamental de cada componente para se obter melhor estabilização. Desta maneira, ainda não é possível realizar um tutorial genérico para que outras pessoas venham a realizar protótipos com dispositivos similares.

Possíveis melhorias podem ser realizadas no sistema, considerando que a placa de processamento possui outras portas para controle de mais um eixo de atuação, além de que o protocolo I2C permite integrar mais dispositivos para aquisição de dados de posicionamento, entre outros.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ACIONAMENTO DE SERVOMECANISMO - Tiago Donizete de Moraes – Itatiba, 2011.
BRAGA, I.; Uso de unidades de medidas inerciais para obtenção dos parâmetros de atitude e azimute .Revista Sociedade Brasileira de Telecomunicações, São Paulo, v.1, n.1, p. 834, 2017.

CASTELLANI, Bruno. Controladores de Processos industriais. Mecatrônica Atual, Editora Saber, São Paulo, 2004.

CIA, A. e. Acelerômetro e Giroscópio MPU6050. 2015. Disponível em: < https://www.arduinoecia.com.br/acelerometro-giroscopio-mpu6050-arduino/> Acessado em: 02-09-2019.
Disponível em: <https://www.embarcados.com.br/arduino-mega-2560/> Acesso em: 20 nov. 2019.
DRONEBOT. Using Servo Motors with the Arduino. Disponível em:

<https://dronebotworkshop.com/servo-motors-with-arduino/> Acesso em: 05 nov. 2019.

ESTABILIZAÇÃO INERCIAL DE CÂMERA PAN-TILT DE UM ROV, Victor Frangipani de Oliveira Lima. Agosto 2013.
HILKERT, J., 2008, “Inertially Stabilized Platform Technology: Concepts and Principles”, IEEE Control System Magazine, v. 28, n. 1, pp. 26–46.

LI, Qinq e YAO, Caroline. Real Time Concepts for Embedded Systems, CMP Book, 2003.

MEINECKE, JONAS. LOPES, PAULO. DESENVOLVIMENTO DE PROTÓTIPO PARA PRÓTESE TRANSFEMURAL COM JOELHO HIDRÁULICO ELETRONICAMENTE CONTROLADO, Trabalho de Conclusão de Curso (Tecnólogo em Mecatrônica Industrial) - Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. 2015. 63p.

MOURA, Rafael. Desenvolvimento de um sistema de orientação espacial inercial. 2013. Trabalho de Conclusão de Curso (Graduação de Engenharia elétrica e de Computação) - Universidade de São Paulo, Escola de Engenharia de São Carlos, 2016.

PERMADI, T. A.; HALOMOAN, J.; HADIYOSO, S. Balancing System of Tray on Waiter Robot Using Complementary Filter and Fuzzy Logic. Industrial Automation, Information and Communications Technology (IAICT). 2014, Bali. p. 15-21, 2014.

SANTANA, Douglas. Pontifícia Universidade Católica de São Paulo. Navegação terrestre usando unidade de medição inercial de baixo desempenho e fusão sensorial com filtro de Kalman adaptativo suavizado, 2011. 230p, il Tese (Doutorado).

Santos, J.E.S. (2007). Controle Preditivo não linear para Sistemas de Hammerstein. Tese de doutorado. Programa de pós-graduação em Engenharia Elétrica. Universidade Federal de Santa Catarina. (p.01).
Scheffer-Dutra, C.B., Núñez-Reyes, A. & Bordons, C. (2002) Controle Preditivo com Restrições Aplicado a Uma Planta Solar de Climatização. XVI Congresso Brasileiro de Autómatica, Natal, RN, p.2798-2803.

SOUSA, Rafaela. "Segunda Revolução Industrial"; Brasil Escola. Disponível em: https://brasilescola.uol.com.br/historiag/segunda-revolucao-industrial.htm. Acesso em 10 de novembro de 2019.
SOUZA, Fábio. **Arduino MEGA 2560**.

TAKAHASHI, Vitor Gonçalves. Desenvolvimento de uma unidade de medição inercial para determinação de orientação de um quadricóptero. 2016. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) - Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2016.

U.S.A. InvenSense Inc. MPU-6000 and MPU-6050 Product Specification Revision 3.4. 2013. Disponível em: <https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf>. Acesso em: 3 set. 2019.

URDHWARESHE, R.; BAKSHI, M.; NAIKNAVARE, P.; NAIK, S. Design and Implementation of IMU Sensor Fusion and PID Control in Quadrotor. IPASJ International Journal of Electronics and Communication (IIJEC). v. 2, n. 9, p. 56- 63, Set. 2014.

WOODMAN, O. J. An Introduction to Inercial Navigation. University of Cambridge, August 2007.

APÊNDICE (OPCIONAL)

CÓDIGO DE PROGRAMAÇÃO DO ARDUINO MEGA 2560

```
//Carrega a biblioteca Wire
#include <Wire.h>
#include <Servo.h>

//Declaracao dos objetos
Servo servopitch;
Servo servoroll;

//Endereco I2C do MPU6050
const int MPU = 0x68;

//Pulso de degrau
#define ki 50
#define kf 80
```

```
//Conversao
#define A_R 16384.0 // 32768/2
#define G_R 131.0 // 32768/250

#define roll0 -3.7114
#define pitch0 0.996
#define gyrox0 -0.4498
#define gyroy0 0.1112

//Servo
#define motpitch0 1500 //repouso em us
#define motroll0 1490 //repouso em us

int k=0;

int Control[2] = {0, 0};

//conversao de rad para graus 180/PI
#define RAD_A_DEG = 57.295779

//Variaveis para armazenar valores dos sensores
int AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;

//Angulos
float Acc[2];
float Gy[3];
float Angle[3];
float A = 0.96;

String valores;
String val;

long tiempo_prev;
float dt;

void setup()
{
  Serial.begin(9600);
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);

  //Inicializa o MPU-6050
  Wire.write(0);
  Wire.endTransmission(true);

  //Servo
  servopitch.attach(11);
  servoroll.attach(10);
}

void loop()
{
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);

  //Solicita os dados do sensor
```

```
  Wire.requestFrom(MPU, 14, true);

  //Armazena o valor dos sensores nas variaveis correspondentes
  AcX = Wire.read() << 8 | Wire.read(); //0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY = Wire.read() << 8 | Wire.read(); //0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ = Wire.read() << 8 | Wire.read(); //0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp = Wire.read() << 8 | Wire.read(); //0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX = Wire.read() << 8 | Wire.read(); //0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY = Wire.read() << 8 | Wire.read(); //0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ = Wire.read() << 8 | Wire.read(); //0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

  //A partir dos valores do acelerometro, se calcula os angulos Y, X
  //respectivamente, com a formula da tangente
  Acc[1] = atan(-1 * (AcX / A_R) / sqrt(pow((AcY / A_R), 2) + pow((AcZ / A_R), 2))) * RAD_TO_DEG;
  Acc[0] = atan((AcY / A_R) / sqrt(pow((AcX / A_R), 2) + pow((AcZ / A_R), 2))) * RAD_TO_DEG;

  //Calculo do angulo do Giroscopio
  Gy[0] = GyX / G_R;
  Gy[1] = GyY / G_R;
  Gy[2] = GyZ / G_R;

  dt = (millis() - tiempo_prev) / 1000.0;
  tiempo_prev = millis();

  //Aplicacao do Filtro Complementar
  Angle[0] = A * (Angle[0] + (Gy[0] - gyrox0) * dt) + (1 - A) * (Acc[0] - pitch0);
  Angle[1] = A * (Angle[1] + (Gy[1] - gyroy0) * dt) + (1 - A) * (Acc[1] - roll0);

  // Leis de Controle
    Control[0] = -5.8226*Angle[0]-0.1344*Gy[0];
    Control[1] = -10.7688*Angle[1]-0.1496*Gy[1];

  //Mostrar os valores na serial
  valores = String(Angle[0]) + " " + String(Angle[1]) + " " + String(Control[0]) + " " + String(Control[1]) +
" " + String(dt); ;
  //val = String(Acc[0]) + " " + String(Acc[1]);
  //Serial.println (val);

  Serial.println(valores);

  //Atualiza pwm dos motores
  servopitch.writeMicroseconds(motpitch0 + Control[0]);
  servoroll.writeMicroseconds(motroll0 + Control[1]);

  //Aguarda 300 ms e reinicia o processo
  k=k+1;
  delay(40);
}
```