

Desenvolvimento de Plataforma Auto-Nivelada

João V. B. Mergener, Leonardo Zambotto da Silva e Luíza Bortolon Heck

Resumo—A utilização de plataformas que não têm sua posição modificada quando submetidas à alterações do ambiente tem sido cada vez mais necessária. Um grande exemplo disso pode ser observado nas plataformas petrolíferas. Nesse trabalho, é construído e analisado um modelo que simula essa situação, com o uso de uma plataforma em pequena escala, feita em MDF. A leitura dos dados de posição é realizada utilizando um acelerômetro (MPU-6050) e o equilíbrio é atingido a partir da movimentação de servomotores, com a plataforma Arduino sendo o meio de controle e comunicação. Ao final, a intenção é que a estrutura se mantenha na mesma posição, ou muito próxima, mesmo que sua base seja movimentada.

Index Terms—Plataforma Auto-Nivelada, Robô Equilibrista, Instrumentação, Acelerometria



1 INTRODUÇÃO

MANTER uma estrutura sempre em um determinado nível, independente da interferência do ambiente, é uma tarefa de suma importância em diversas aplicações. Esse conceito de auto nivelção pode ser utilizado desde situações simples, como a estabilização de uma câmara ou de um quadricóptero, até situações de alta complexidade, como em prédios à prova de terremotos e plataformas marítimas.

Para esses e outros casos, é apropriado utilizar o módulo MEMS (*micro-electrical mechanical system*) com acelerômetros e giroscópios. Esses sensores coletam informações do ambiente pela medição de fenômenos mecânicos. Essas informações são processadas pelos componentes microeletrônicos e direcionam uma resposta para os dispositivos atuadores [1] que, no caso das plataformas de auto nivelamento, será uma resposta de posição. Essa resposta é enviada como um ângulo para o atuador a fim de que haja a compensação da inclinação da plataforma.

Este trabalho descreve a teoria e implementação de uma plataforma auto estabilizante com componentes simples, sendo a maioria deles de baixo custo e de fácil aquisição. O objetivo dessa plataforma é manter um objeto sempre no mesmo nível, independente das alterações sofridas pela sua base. Para a implementação, além do módulo MEMS mencionado anteriormente, também foram utilizados servomotores,

um Arduino UNO, uma estrutura produzida em MDF (madeira compensada), mini proto-boards e jumpers.

Dez, 2018

2 METODOLOGIA

Para o correto funcionamento da plataforma, será efetuada a seguinte sequência de operações: inicialmente, deve-se montar uma estrutura, leve o suficiente para que os servomotores possam movimentá-la, e o microcontrolador deve ser conectado ao sensor inercial, servomotores e ao módulo de acionamento dos servomotores, além de uma fonte de alimentação externa de 5 V, conforme pode ser observado na Figura 1

Após a montagem da parte física, utiliza-se o código desenvolvido em C. Nele, os valores do sensor inercial são adquiridos pelo microcontrolador. Estes sinais recebem um condicionamento, podendo este ser filtro de Kalman, filtro complementar ou então sem condicionamento nenhum.

O microprocessador, após ter estes valores filtrados, realiza o cálculo dos ângulos da plataforma e calcula a diferença entre o ângulo de referência e o atual nas direções x e y. Este erro de posição é informado a um controlador, o qual calcula a nova posição dos servos. Os servomotores são então acionados, por meio do

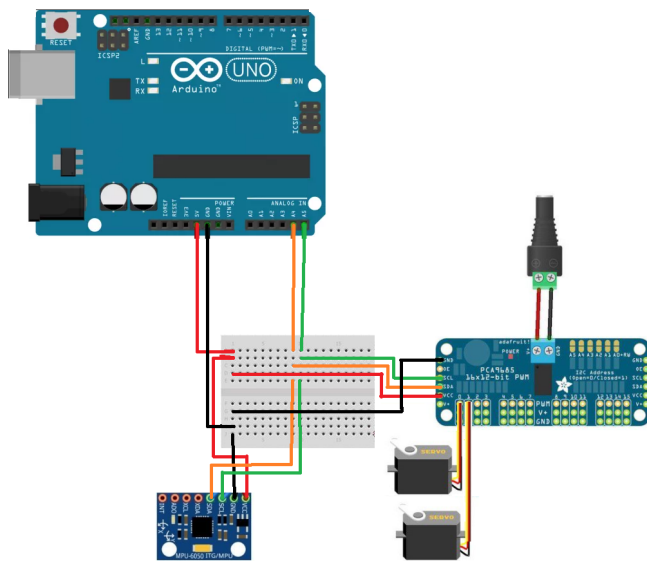


Figura 1: Esquemático mostrando as ligações do circuito eletrônico da plataforma

módulo de acionamento, a fim de compensar essa diferença de ângulo.

As subseções a seguir tratarão da construção e funcionamento da plataforma auto-nivelada.

2.1 Princípio de Funcionamento de um Acelerômetro e Giroscópio

A fim de obter uma medida da posição espacial e movimento da plataforma, usa-se uma IMU - *Inertial Measurement Unit*. O módulo usado possui as funções de acelerômetro e giroscópio, além de um sensor de temperatura.

O acelerômetro utilizado é do tipo MEMS capacitivo. A Figura 2 exemplifica o funcionamento deste tipo de sensor: quando aplicada uma força à uma massa interna, devido à inércia ocorre a aproximação entre as placas de um sistema capacitivo, alterando a capacitância, que é a variável medida. Deste modo, a aceleração (produto da força pela massa) é medido de maneira indireta.

O giroscópio utilizado é do tipo vibratório MEMS. Quando este dispositivo é rotacionado em algum de seus eixos, o efeito Coriolis cria vibrações, que por sua vez são detectadas por placas capacitivas, similarmente ao acelerômetro. Seu funcionamento pode ser observado na Figura 3.

Capacitive accelerometer

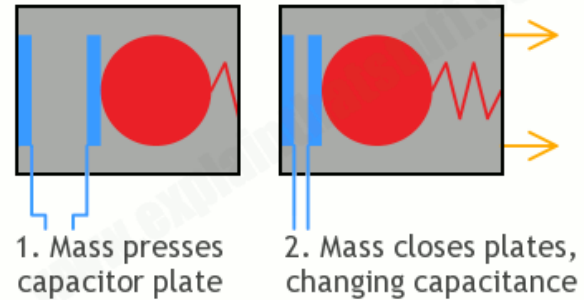


Figura 2: Exemplificação do funcionamento interno de um acelerômetro.[2]

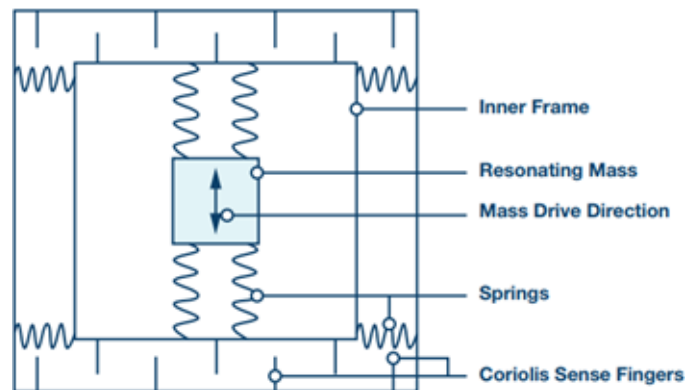


Figura 3: Exemplificação do funcionamento interno de um giroscópio.[3]

2.2 Sensor Utilizado

A unidade inercial utilizada foi a MPU-6050. Este módulo possui funções de giroscópio e acelerômetro integradas, tensão de operação de 2,375 V a 3,46 V e suporta comunicação I2C. Possui um DMP (*Digital Motion Processor*) incluso e conta com um *buffer* FIFO de 1024 bytes, o qual ajuda a reduzir o consumo de energia, ao ler o valor de dados em *bursts* - bateladas de informações. Também possui interrupções para detecção de movimento.

A função de giroscópio é realizada por três giroscópios MEMS vibratórios independentes, os quais possibilitam a captação de velocidade nos eixos X-, Y- e Z-. O sinal gerado é processado, gerando uma tensão proporcional à taxa angular exercida. Esta voltagem é digitalizada usando um conversor analógico-digital de 16 bits para cada eixo. Também é possível programar digitalmente o fundo de escala de 250 a 2000 graus por segundo e a taxa de aquisição

do ADC de 3,9 até 8000 amostras por segundo, além de ser possível a seleção de filtros passa-baixas. Possui uma sensibilidade de 131 LSB/s

A função de acelerômetro, similarmente à de giroscópio, é composta por 3 massas de prova, uma para cada eixo, e possui um ADC de 16 bits. Quando posta sobre uma superfície plana, essa IMU deve medir 0 g nos eixos X- e Y- e 1 g no eixo Z-. O fator de escala é calibrado em fábrica e é independente da tensão aplicada. O fundo de escala pode ser ajustado de 2 g até 16 g, e possui uma sensibilidade de 16384 LSB/g.

Mais informações sobre o sensor utilizado podem ser conferidas no *datasheet*, fornecido pelo fabricante [11].

2.3 Microcontrolador Utilizado

O microcontrolador utilizado foi o Arduino Uno. Este microcontrolador possui uma memória flash de 32 KB e uma RAM de 2 KB. Ele também opera a uma frequência de 16 MHz. Possui 14 pinos I/O digitais e 6 pinos de entrada analógica com um ADC de 10 bits de resolução.

É necessário utilizar uma fonte externa na faixa de 5V a 12V, já que uma tensão abaixo de desta faixa pode deixar a tensão de funcionamento da placa (5V) instável e quando alimentada com tensão acima de 12V, o regulador de tensão da placa pode superaquecer e danificá-la.

Foi utilizado o protocolo de comunicação I2C, tendo o Arduino como *Master* e a IMU e *driver* de acionamento como *Slave*. A conexão é feita por padrão nos pinos A4 e A5 do Arduino e nos pinos SDA e SCL da IMU e do *driver*, sendo utilizadas funções da biblioteca padrão *Wire* do Arduino para execução do protocolo.

2.4 Atuadores Utilizados

Para realizar a movimentação da plataforma, serão utilizados micro servos 9g SG90 da TowerPro. Estes atuadores serão utilizados devido à sua fácil aquisição e interface com o microcontrolador utilizado.

Este atuador tem engrenagens internas de nylon, peso de 14,7 gramas e três pinos, dois para alimentação e um para ajuste de posição. Seu alcance é de 180 graus, opera com tensão

de 4,8 a 6 V e consegue exercer um torque máximo de 2.5 kg.cm, dependendo da tensão aplicada. É acompanhado de braços de nylon para fixação na estrutura desejada [4].

Além disso, esses atuadores têm como variável de controle o ângulo em que o rotor se posiciona e este controle é feito por PWM (*Pulse Width Modulation* - modulação por largura de pulso).

Embora já exista uma biblioteca dedicada à movimentação de servos na plataforma Arduino, experimentos anteriores com os micro servos adquiridos demonstraram a necessidade um ajuste da largura máxima e mínima do pulso utilizado, possivelmente devido a um padrão diferente de fabricação. Este ajuste é feito no código, e implementado de acordo com a biblioteca utilizada [12].

2.5 Condicionamento

A fim de minimizar ruídos e erros de medição espúrios do sensor inercial, pode ser utilizado um condicionamento, o qual comumente se dá pelo filtro de Kalman ou pelo filtro complementar.

2.5.1 Filtro de Kalman

O filtro de Kalman é um método matemático para diminuir o ruído em sinais medidos, via *software*.

Ele produz estimativas dos valores reais de grandezas medidas e valores associados predizendo um valor, estimando a incerteza do valor predito e calculando uma média ponderada entre o valor predito e o valor medido. O peso maior é dado ao valor de menor incerteza [5].

2.5.2 Filtro Complementar

O filtro complementar, diferentemente do filtro de Kalman, faz uma fusão de sensores, utilizando a leitura de dois eixos do acelerômetro e de um eixo do giroscópio. Ele é composto por um filtro passa baixas, que diminui o ruído da medida lida do acelerômetro, e um filtro passa altas, que elimina o erro causado pelo acúmulo de ruídos ao integrar a saída do giroscópio para a obtenção do ângulo. A Figura 4 mostra um esquemático do filtro complementar.

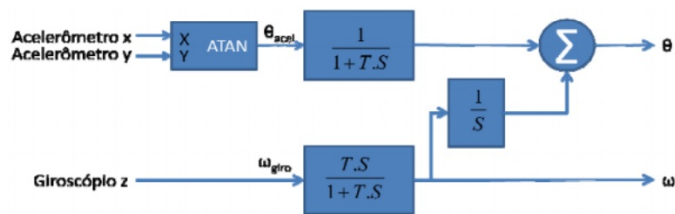


Figura 4: Exemplificação do funcionamento de um filtro complementar.

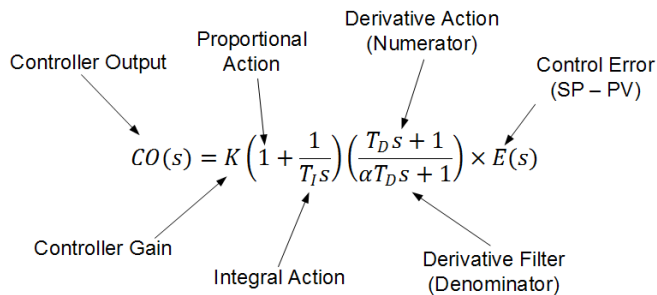


Figura 5: Diagrama de blocos ilustrando o funcionamento de um controlador PID

2.6 Controle

A fim de realizar um ajuste suave de posição, foi implementado no código um controlador, baseado nos métodos de discretização descritos em Bazanella *et al*, 2005. Foram criadas variáveis referentes aos parâmetros desejados, e uma rotina que permite a implementação de um controlador proporcional, proporcional-integral, proporcional-derivativo ou proporcional-integral-derivativo, de acordo com os parâmetros fornecidos pelo usuário. O princípio de funcionamento do controlador implementado pode ser visto na Figura 5.

2.7 Programação

A ideia geral do código utilizado consiste em ler constantemente os valores do sensor inercial e realizar o acionamento dos servos quando ocorrer uma inclinação.

Foram utilizadas versões levemente diferentes do código, cada uma com um tipo de condicionamento, além de uma versão que somente escrevia os dados na porta serial, para a correta visualização dos efeitos de cada filtro e aquisição da resposta ao salto de ângulo dos servomotores.

Cabe ressaltar que na inicialização da plataforma é feita uma calibração, em que ambos os

servos são posicionados em 90 graus e então os valores lidos pelo acelerômetro nesta posição definem os ângulos de referência. Assim, é possível ao usuário calibrar uma posição de referência com um determinado *offset* angular.

Foram utilizadas as bibliotecas:

- Wire.h, biblioteca padrão do Arduino, para realizar a comunicação I2C.
- MPU6050.h, para aquisição e manipulação de dados e configuração da IMU.
- Kalman.h, para implementação do filtro de Kalman.
- Adafruit_PWMServoDriver.h, para controle do módulo de acionamento dos servomotores.

É possível encontrar no *GitHub* tanto a biblioteca MPU6050.h[6] quanto a Kalman.h[7], assim como a Adafruit_PWMServoDriver.h [12].

Todo o código usado no desenvolvimento da plataforma pode ser conferido no *GitHub* de Leonardo Zambotto[8].

2.8 Estrutura

A estrutura da plataforma foi projetada em duas partes, fixa e móvel. A parte fixa tem a função de dar suporte aos componentes eletrônicos e a um dos servomotores e deve ter uma base grande, a fim de conferir maior equilíbrio ao sistema.

Já a parte móvel tem em vista a leveza, de modo que fosse possível ao servomotor movê-la. Além disso, esta parte é na verdade subdividida em duas: um pedaço que encaixa no servomotor da parte fixa, responsável pela movimentação em um dos eixos, e tem um encaixe para outro servomotor. No outro pedaço, há o encaixe para o servomotor responsável pela movimentação no outro eixo, que está fixado no outro pedaço da parte móvel, e uma parte alongada, onde deve ser fixado a unidade inercial.

A estrutura foi desenvolvida e modelada com o auxílio do *software* SolidWorks e confeccionada por corte a laser, no Laboratório de Inovação e Fabricação Digital da Escola de Engenharia (LIFELAB). Uma renderização no SolidWorks é mostrada na Figura 6. As peças da estrutura, assim como um molde para corte à laser, podem ser encontrados em [9].

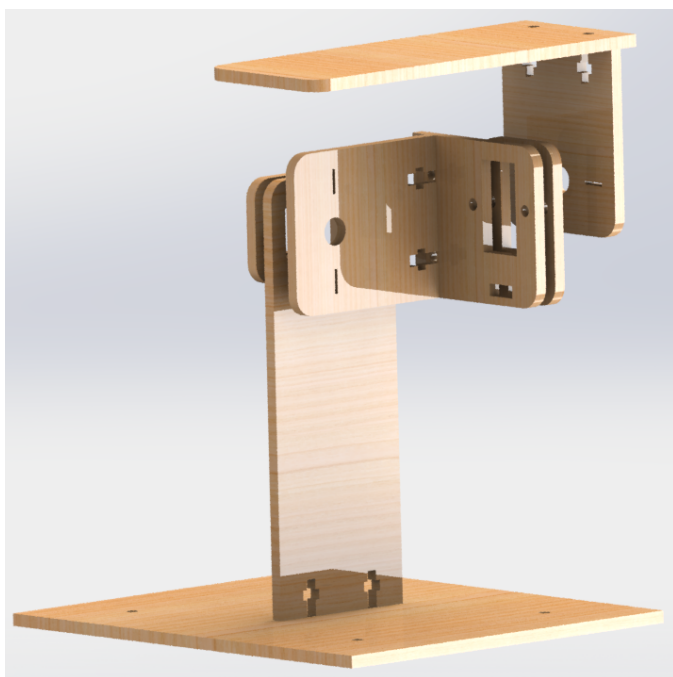


Figura 6: Renderização no SolidWorks da estrutura.

3 RESULTADOS

Nesta seção, serão apresentados resultados obtidos a partir de rotinas de testes, assim como constatações feitas sobre aspectos do projeto após estas. A plataforma já montada, ligada e funcional pode ser conferida na Figura 7

3.1 Considerações sobre a estrutura

Devido a ter mais enfoque em prova de conceito do que em desempenho, a estrutura foi desenvolvida de modo que pudesse ser construída a partir de um baixo orçamento, porém executando bem sua função de suporte aos componentes do projeto. Entretanto, conforme testes foram sendo efetuados, foi possível notar que a estrutura tinha um comportamento ruim frente à vibrações causadas pelo deslocamento da parte estrutural móvel, comandada pelo acionamento dos servomotores.

Uma solução imediata se deu com a conclusão de que para testar o sistema, era necessário segurar a parte fixa da estrutura e movimentá-la, a fim de que a parte móvel fosse se ajustando até que alcançasse a posição de calibração. Com isso, a mão do usuário já providenciava uma atenuação das vibrações.

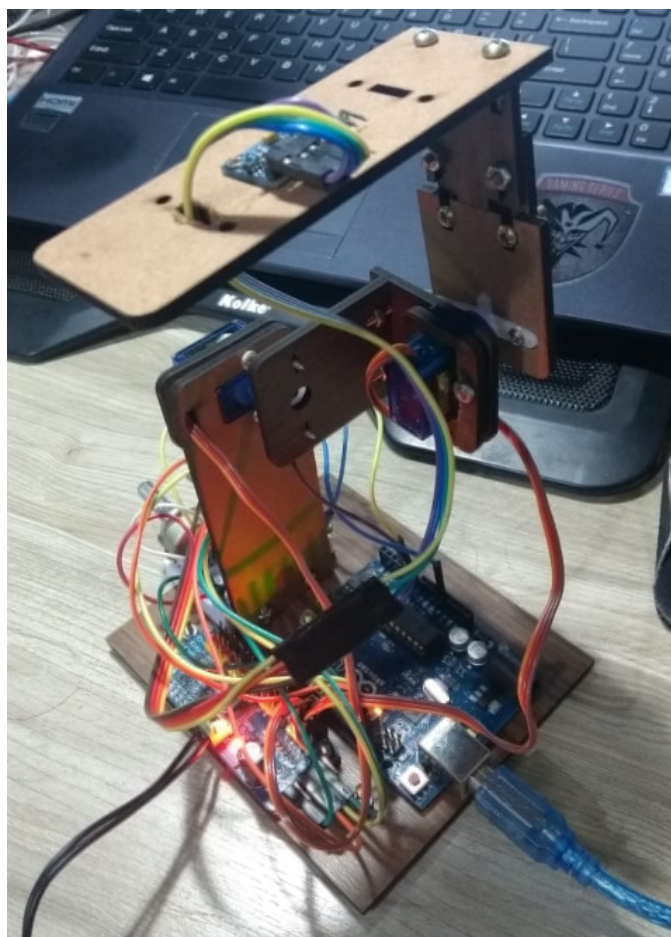


Figura 7: Plataforma concluída.

Além disso, um dos componentes da parte móvel foi adaptado, já que um erro de projeto passou despercebido - um dos servos era impedido fisicamente de percorrer todos os ângulos que poderia. Devido a modularidade do projeto, foi possível realizar esta adaptação com peças sobressalentes, e a parte reparada não apresentou problemas maiores.

3.2 Implementação do filtro de Kalman e filtro complementar

Priorizando mais o condicionamento e filtragem do que a modelagem do sensor inercial em si, foi usada a biblioteca *KalmanFilter* [7], a qual realiza uma implementação do filtro de Kalman já com os parâmetros do sensor inercial utilizado. O filtro complementar também foi implementado, com os valores do giroscópio sendo adquiridos analogamente aos do acelerômetro.

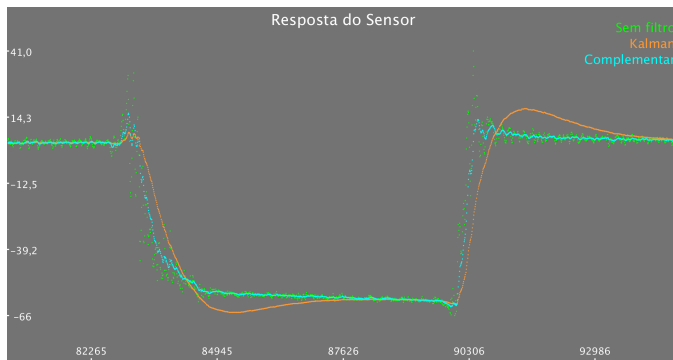


Figura 8: Resposta do sensor aos diferentes tipos de condicionamento.

Como esperado, tanto o filtro de Kalman quanto o complementar obtiveram resultados significativamente melhores em questão de eliminação de ruído, consequentemente diminuindo também a vibração da estrutura por movimentos desnecessários. Uma comparação entre estes dois filtros e os dados não-filtrados pode ser vista na Figura 8.

3.3 Dinâmica dos servomotores

Ainda que os dados apresentem um determinado comportamento, ele sofre variação devido à aplicação, já que o sensor é movimentado por servomotores. Saber o comportamento do sistema com essa dinâmica é de grande ajuda no momento de realizar o controle da planta e ajuste do filtro de Kalman.

Sendo assim, foram adquiridos os ângulos calculados pelas três maneiras abordadas (dados não-filtrados, filtro de Kalman e filtro complementar) de ambos os servomotores utilizados, ao iniciar o servo em 90 graus e mover, após determinado tempo, o servo para zero graus, simulando um *step* no sensor. Também foi obtida a resposta ao mover o servo para 180 graus, já que devido à assimetria da estrutura, é possível gerar mais momento fletor em uma determinada faixa de ângulo.

É possível observar a resposta do sistema a este *step* simulado na Figura 9 para o servo responsável pelo movimento na direção x e na Figura 10 para o responsável pela direção y.

3.4 Ajuste do controlador

Inicialmente, antes de serem obtidas as respostas ao salto dos servomotores, os ganhos do

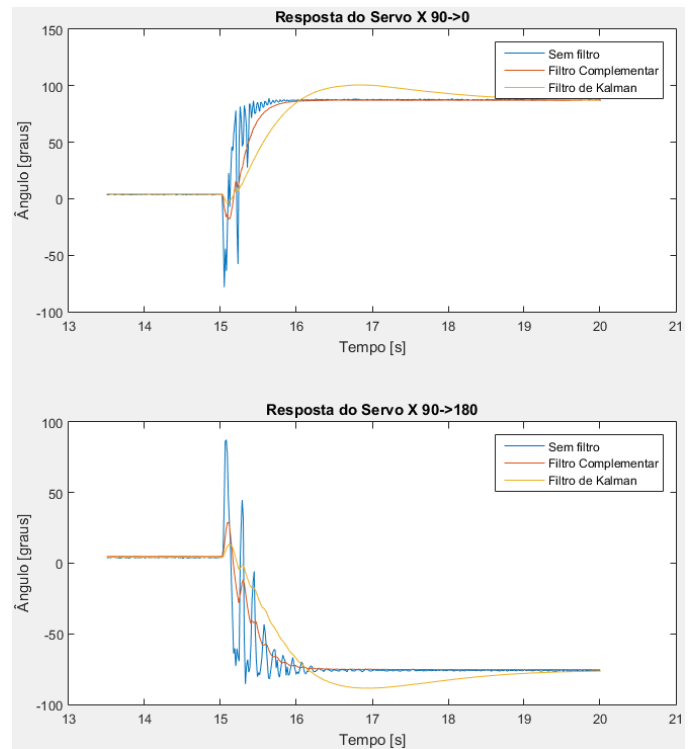


Figura 9: Resposta do servo responsável pelo giro em x.

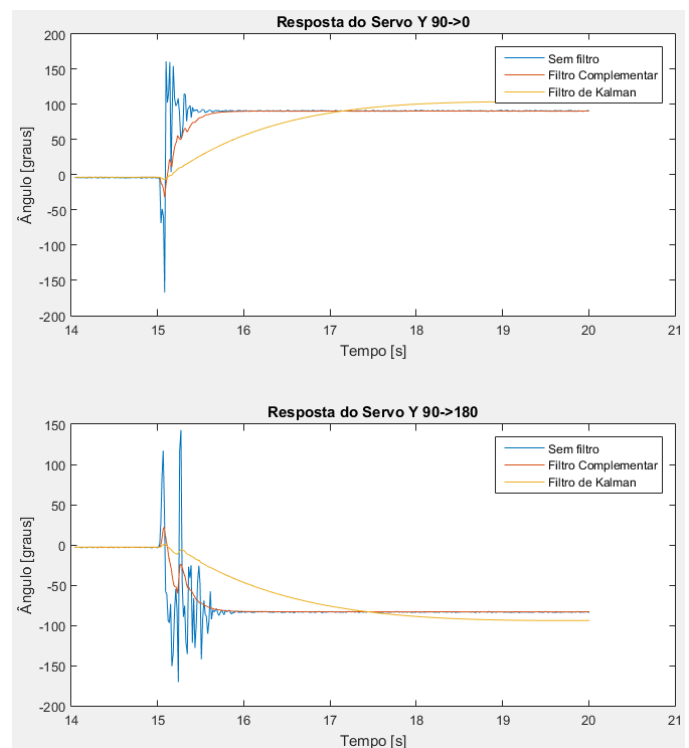


Figura 10: Resposta do servo responsável pelo giro em y.

controlador foram obtidos experimentalmente, observando a resposta na movimentação da plataforma à variações induzidas de ângulo e ajustando os coeficientes do controlador visando primeiramente estabilidade e, se possível, rapidez.

Com isso, chegou-se ao valor de 0,2 para o ganho K e 0,01 para T_i . Estes ganhos são implementados de acordo com a Figura 5. Com estes valores, obteve-se um ajuste estável e com pouco sobrepasso, com uma velocidade moderada.

4 CONCLUSÕES

Ambos os métodos de condicionamento se mostraram eficazes em eliminar o ruído, porém a resposta mais lenta do filtro de Kalman dificultou seu uso com o controlador implementado, fazendo, assim com que o filtro complementar fosse estabelecido como condicionamento padrão da plataforma. Porém esta resposta lenta pode ser devido ao fato de a biblioteca utilizada para sua implementação considerar o modelo apenas do sensor, não fixado na estrutura utilizada.

Como o filtro de Kalman leva em conta o modelo da planta, ao utilizá-lo em uma aplicação como a tratada neste artigo seria necessário uma recalibração dos parâmetros do filtro, o que poderia ser feito já que foi possível obter a resposta ao salto dos servomotores.

Também com essa resposta ao salto, seria possível obter funções de transferência para ambos os servos de 0 a 90 graus e de 90 a 180 graus, possibilitando um ajuste do controlador mais sofisticado, por métodos como lugar das raízes ou LQR.

Embora o problema de vibrações tenha sido moderadamente eliminado devido à operação do usuário, seria recomendado um reprojeto da estrutura, visando a estabilidade frente a vibrações, com elementos móveis mais simétricos. Entretanto, a estrutura desenvolvida foi suficiente para a fase de protótipo.

Uma futura melhoria também seria a troca dos servomotores utilizados por outros com maior resolução de saída ou motores de passo, possibilitando um ajuste maior do que o grau unitário utilizado nesta implementação.

De modo geral, a plataforma desenvolvida consegue compensar sua posição frente à movimentação de sua base de forma satisfatória, com grande estabilidade e relativa rapidez.

REFERÊNCIAS

- [1] https://pt.wikipedia.org/wiki/Sistemas_microeletromec%C3%A2nicos
- [2] C. Woodford, "Accelerometers". Disponível em: <https://www.explainthatstuff.com/accelerometers.html>
- [3] J. Watson, "MEMS Gyroscope Provides Precision Inertial Sensing in Harsh, High Temperature Environments". Disponível em: <http://www.analog.com/en/technical-articles/mems-gyroscope-provides-precision-inertial-sensing.html>
- [4] P. Cheung, http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
- [5] https://pt.wikipedia.org/wiki/Filtro_de_Kalman
- [6] <https://github.com/jarzebski/Arduino-MPU6050>
- [7] <https://github.com/TKJElectronics/KalmanFilter>
- [8] https://github.com/Leo13Z/Plat_Self_Stabilizing/tree/master
- [9] <https://drive.google.com/drive/folders/1q7YPe5NSgFtUKp8LcbRjj-uaghM40MR8?usp=sharing>
- [10] BAZANELLA, Alexandre Sanfelice; DA SILVA JR, João Manoel Gomes. *Sistemas de controle: princípios e métodos de projeto*. Editora da UFRGS, Porto Alegre, 1ª Edição, 2005
- [11] https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%20.pdf
- [12] http://adafruit.github.io/Adafruit-PWM-Servo-Driver-Library/html/class_adafruit___p_w_m_servo_driver.html