

# *Desenvolvimento de um robô auto equilibrante para aplicações didáticas de sistemas de controle*

Bruna Caroline Thomaz<sup>1</sup>, Ivan Jorge Gabe<sup>2</sup>

TCC2 - Curso de Engenharia de Controle e Automação

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Farroupilha

Discente<sup>1</sup>, orientador<sup>2</sup>

Farroupilha, Brasil

brunacthomaz@gmail.com<sup>1</sup>, ivan.gabe@farroupilha.ifrs.edu.br<sup>2</sup>

**Abstract**—O objetivo deste artigo é desenvolvimento de um robô auto equilibrante de baixo custo para fins didáticos que contará com um sistema de controle permitindo sua movimentação linear e rotacional. Será utilizado um controlador PID para coordenar sua trajetória e um algoritmo de Filtro de Kalman para atenuação de ruídos no acelerômetro e giroscópio obtendo assim a medida do deslocamento velocidade angular, dois motores de corrente contínua com encoders integrado às suas carcaças determinam o deslocamento e velocidade linear, possibilitando assim o equilíbrio do robô. A construção do robô foi realizada conforme projeto mecânico e o Filtro de Kalman apresentou bom desempenho assim como o controlador PID, porém não se obteve correspondência entre os ganhos obtidos de forma simulada do PID fazendo uso do modelo matemático do sistema, tendo de efetuar os ajustes dos ganhos empiricamente para implementação no projeto físico do robô. Os testes para o equilíbrio em malha fechada foram realizados com sucesso, o robô se mantém equilibrado mesmo com pequenas perturbações externas além de apresentar bom controle de trajetória.

**Palavras-chave**—robô auto equilibrante; Filtro de Kalman; PID.

## I. INTRODUÇÃO

Com o avanço da tecnologia percebe-se a introdução crescente da robótica no cotidiano da sociedade, na área da robótica móvel existem os veículos auto equilibrantes que são caracterizados pela habilidade de se manterem em equilíbrio em apenas duas rodas e girar sobre o próprio eixo. Devido à sua grande flexibilidade de movimentação, instintividade de operação e tamanho compacto os veículos auto equilibrantes podem ser utilizados em diversas aplicações [1]. A empresa Segway, por exemplo, disponibiliza alguns produtos nessa área, visando o transporte de pessoas.

Apesar disso, existe ainda uma certa dificuldade de aprendizagem de sistemas de controle dinâmico, ocasionado pela falta de entendimento teórico do problema ou até mesmo pela impossibilidade de implementação prática do mesmo. Por vezes, o aluno não exercita de forma prática o conhecimento adquirido nas aulas, já que não existe nenhum equipamento didático que proporcione este tipo de exercício no IFRS. Pensando nisto, o presente projeto possui o propósito de exercitar o conhecimento do aluno para o desenvolvimento de sua criatividade e raciocínio lógico na resolução de problemas de área de controle dinâmico, dessa maneira é importante que o indivíduo construa o seu próprio conhecimento a partir da iteração com o ambiente em que vive, propiciando o

desenvolvimento e uma maior significação da aprendizagem [2].

Propondo esta interação entre os alunos e a área de controle dinâmico, o objetivo deste artigo é o desenvolvimento de um robô auto equilibrante de baixo custo para fins didáticos. O mesmo contará com estruturas de controle que permitirão sua movimentação linear e rotacional. Será utilizado o algoritmo do Filtro de Kalman para atenuação de ruídos no acelerômetro e giroscópio, obtendo assim as medidas do deslocamento e velocidade angular, um controlador PID será empregado para a estabilização do robô. Dois motores de corrente contínua com encoders integrado às suas carcaças determinam o deslocamento e velocidade linear.

## II. REVISÃO BIBLIOGRÁFICA

Há um número significativo de trabalhos sobre o auto equilíbrio de robôs, existem também muitos protótipos de robôs capazes de se equilibrar que foram construídos para uso acadêmico. Para fins de síntese, esta revisão selecionou apenas três destes trabalhos que concentram as possibilidades e dificuldades das abordagens que estes trabalhos propõem.

Em [3], que baseando-se na modelagem de um pêndulo invertido desenvolveu um robô auto equilibrante de duas rodas, com o propósito de investigar o uso do Filtro de Kalman para a atenuação de ruídos e de examinar o desempenho de um regulador quadrático linear (LQR), um controlador proporcional integral derivativo (PID) foi utilizado por ele para o controle da trajetória do robô.

Referência [3] retrata a instabilidade do seu sistema quando aplicada uma entrada do tipo impulso na função de transferência fazendo com que o ângulo de inclinação e a posição do robô aumentem indefinidamente resultando na queda do robô [4]. Para resolver esta situação duas técnicas foram empregadas por [3]: o controle de realimentação de estados com ganhos projetados pela técnica LQR e por Alocação de Polos.

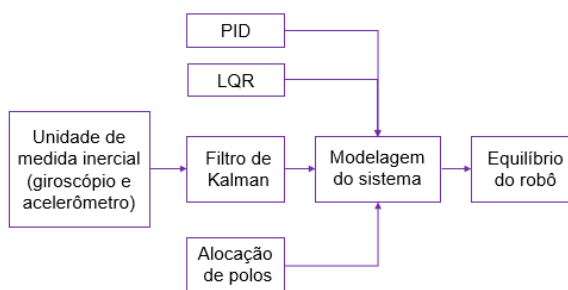


Fig. 1. Diagrama do sistema proposto por [3]

Referência [4] descreve detalhadamente o controlador PID implementado em seu robô que juntamente com a utilização da função chamada lugar geométrico das raízes disponível no MATLAB ajusta os ganhos do controlador para obter o desempenho desejado.

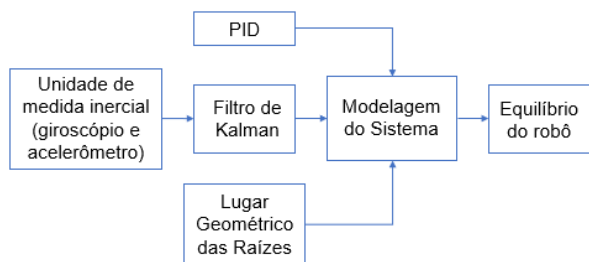


Fig. 2. Diagrama do sistema proposto por [4]

Referências [3] e [4] detalham, com diferentes abordagens a modelagem matemática dos sistemas implementados, as equações de movimento do robô, as forças atuantes sobre cada parte do sistema, a possibilidade de linearização, as equações de estado e a função de transferência do sistema já linearizado e também propõem a utilização do MATLAB para as simulações.

O objetivo de [1] é o desenvolvimento de um robô auto equilibrante para servir de plataforma para estudo e aplicação de sistemas de controle, realizando a leitura dos sinais dos sensores (através dos conversores analógico-digitais do Arduino) e transforma-os em valores com unidades utilizáveis no sistema. Para validar seu projeto, [1] utilizou um controlador PID, apoiado em métodos heurístico baseado na observação do comportamento, visto que para a sua aplicação, [1] considera que não há a necessidade do conhecimento do modelo matemático do sistema a ser controlado e que após a implementação do PID basta ajustar as variáveis manualmente relacionadas aos ganhos proporcional, integral e derivativo diretamente na planta e observar a sua resposta.

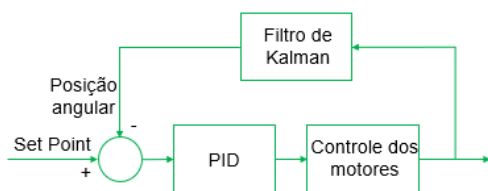


Fig. 3. Diagrama do sistema proposto por [1]

Todos os autores sugerem a utilização do Filtro de Kalman para atenuação de ruídos e outras incertezas na unidade de medida inercial (IMU) utilizada para verificação do deslocamento e velocidade angular do corpo do robô. A ideia é realizar a combinação das medições do acelerômetro e do giroscópio de maneira a contornar sua imprecisão. Referências [1] e [4] aplicam um controlador PID para resolver problemas de instabilidade do sistema, já [1] utiliza um regulador linear quadrático (LQR) para resolver este mesmo problema.

Contudo, o melhor resultado obtido foi o do trabalho realizado por [3], pois seu robô conseguiu manter-se equilibrado com pequenas interferências externas. O robô de [1] obteve um bom resultado, conseguindo manter-se equilibrado sem interferências externas. Já o robô de [4] não possuiu implementação prática, apenas simulada.

Apesar da realização de diversos trabalhos na área, existe uma grande carência nos estudos analíticos de robôs auto equilibrantes envolvendo problemas de controle dinâmico, e, a associação das análises de modelo com resultados experimentais e técnicas de controle adequadas.

O projeto proposto será composto por um sistema mecânico, sensores e um controlador que almejam manter o equilíbrio do robô em regime permanente, um esquemático do projeto é apresentado na Fig. 4. Entretanto, para que o projeto seja viável a dinâmica do robô deve ser descrita por modelo matemático para facilitar o desenvolvimento de um sistema de controle eficiente. O controlador PID será escolhido para o controle de equilíbrio e trajetória devido a sua robustez e facilidade de projeto. O algoritmo do Filtro de Kalman será utilizado para a obtenção da leitura do deslocamento angular. Todas as variáveis destas estruturas serão projetadas a partir do modelo matemático do sistema, sendo ajustadas para atingir o desempenho desejado no protótipo.

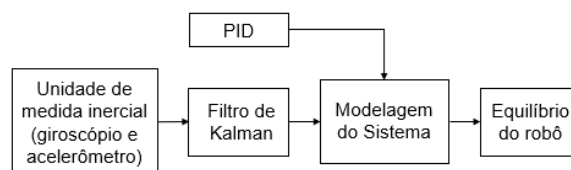


Fig. 4. Diagrama do sistema proposto

### III. FUNDAMENTAÇÃO TEÓRICA

O robô auto equilibrante proposto é composto por um sistema mecânico, dois motores CC de 12 V com encoders integrados às suas respectivas carcaças, uma unidade de medida inercial formada por um acelerômetro e um giroscópio de três eixos cada e um controlador Arduino Uno.

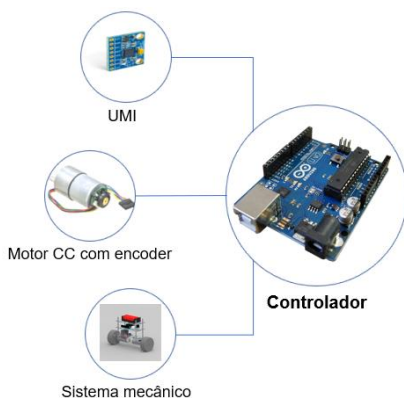


Fig. 5. Diagrama robô auto equilibrante

#### A. Arduino Uno

A placa Arduino Uno possui seu próprio ambiente de desenvolvimento baseado na linguagem C, deixando a programação bem intuitiva. Seu processador é um ATmega328, possui 6 portas analógicas, 14 portas digitais, sua velocidade de clock é de 26 MHz. A tensão de operação da placa é de 5 V, já a tensão para alimentação da placa pode variar de 7 V a 12 V.



Fig. 6. Controlador Arduino Uno

#### B. Motor CC com encoder GM25-370

O motor CC de 12 V, 3050 RPM sem carga e 2620 RPM com carga, corrente de 0,59 A, conta com um encoder acoplado à sua carcaça permitindo medir com exatidão a distância percorrida pelo robô. O motor dispõe de uma pequena placa na parte de trás onde estão instalados os sensores de efeito *hall* que são responsáveis por ler as informações relativas à rotação do motor. A placa possui seis conexões disponíveis: duas para a alimentação do motor, denominadas M- e M+, duas para a alimentação do encoder e duas conexões restantes para as saídas dos sinais do encoder C1 e C2 correspondente ao canal A e B.



Fig. 7. Motor CC com encoder

#### C. Ponte H L298N

Visto que o motor CC escolhido para o projeto exige mais corrente do que a suportada pelos pinos do controlador,

emprega-se um módulo ponte H L298N. Possui uma tensão de operação entre 5 e 35 V, pode controlar dois motores CC ou um motor de passo, sua corrente máxima de operação é 2 A por canal e disponibiliza uma potência máxima de 25 W.



Fig. 8. Ponte H L298N

#### D. Unidade de medida inercial MPU-6050

A unidade de medição inercial MPU-6050 escolhida para o projeto é composta por um giroscópio e um acelerômetro de três eixos cada, totalizando seis graus de liberdade.

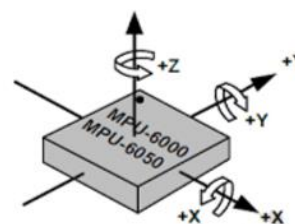


Fig. 9. MPU-6050

O sensor possui um processador de movimento digital, responsável por realizar cálculos complexos com os sensores embutidos. Além disso, possui um recurso adicional, um sensor de temperatura, que permite medições entre  $-40$  e  $85$  °C.

O protocolo de comunicação utilizado pela UMI é o I<sup>2</sup>C que por meio dos pinos de dados seriais (SDA) e relógio serial (SCL), transportam informações entre os dispositivos conectados ao barramento. Cada dispositivo é reconhecido por um endereço único (seja um microcontrolador, driver LCD, memória ou interface de teclado) e pode operar como um transmissor ou receptor, dependendo da função do dispositivo. Um driver de LCD pode ser apenas um receptor, enquanto uma memória pode receber e transmitir dados [12]. O sensor possui alta precisão devido ao conversor analógico digital de 16 bits para cada canal e captura os canais X, Y e Z ao mesmo tempo. O giroscópio possui uma escala programável pelo usuário no intervalo de  $\pm 250$ , 500, 1000, 2000°/s, apresenta melhor desempenho no ruído de baixa frequência e tem uma sensibilidade de 131 LSB/°/s. O acelerômetro também possui uma escala programável de  $\pm 2$ ,  $\pm 4$ ,  $\pm 8$ ,  $\pm 16$ g e sensibilidade de 16384 LSB/g e sua alimentação pode variar entre 3 e 5 V.

#### E. Servomotor SG90

O acelerômetro e o giroscópio da UMI precisam ser verificados para a garantia da confiabilidade das medições, então o servo SG90 será utilizado para esta auxiliar nesta verificação, acoplado a UMI no servo e fazendo-o rotacionar em uma faixa específica. O servo dispõe de um ângulo de rotação de 180 °, uma tensão de operação que pode variar de 3

à 7 V, um torque de 1,2 kg.cm para uma alimentação de 4,8V e 1,6 kg.cm para uma alimentação de 6,0V.



Fig. 10. Servomotor SG90

#### F. Módulo Bluetooth HC-05

Para o controle de direção do robô através de um aplicativo chamado Joystick Bluetooth Commander, o módulo bluetooth HC-05 foi adicionado ao projeto. Ele tem um alcance de 10 m de distância, suporta tanto o modo mestre quanto o modo escravo e sua alimentação pode variar de 3,3 a 5 V. Este módulo possui seis terminais sendo dois para alimentação, dois utilizados pelo protocolo de comunicação RS-232 que permite recepção e envio de dados (Rx e Tx), um pino de status e um pino de configuração



Fig. 11. Módulo bluetooth HC-05

#### G. Joystick bluetooth Commander

Se trata de um aplicativo desenvolvido para Android que emula a funcionalidade *Joystick* virtual via *Bluetooth* disponível de forma gratuita na Google Play. Por padrão, o dispositivo transmite dados a cada 50 ms e possui seis botões configuráveis.

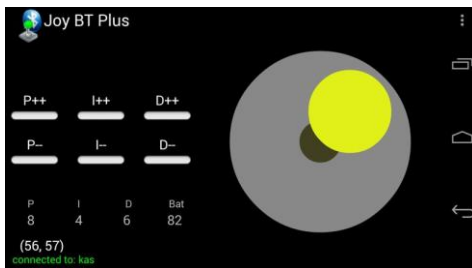


Fig. 12. Joystick bluetooth Commander

#### H. Filtro de Kalman

O Filtro de Kalman é um algoritmo de filtragem utilizado para estimar o estado de um processo visando reduzir o seu erro quadrático médio. Ele é empregado para atenuar os ruídos do giroscópio e do acelerômetro, da UMI adotada na construção do robô.

O acelerômetro fornece para o robô a posição angular do corpo, com um grau de precisão elevado quando o robô está com pequenas variações em relação a sua posição de equilíbrio. Já o giroscópio é responsável pelo fornecimento de medidas

precisas da posição angular do corpo quando está em movimento. Por sua vez, o filtro combina as medições do acelerômetro e do giroscópio, contornando a faixa de imprecisão um do outro, para que seja possível a obtenção dos ângulos de rotação do corpo do robô.

O filtro incorpora todas as informações que podem ser fornecidas e processadas estimando assim o valor atual das variáveis de interesse com o uso de um conhecimento do sistema e dinâmica de medição, a descrição estatística dos ruídos do sistema, medição erros, incerteza na dinâmica do modelo e qualquer informação disponível sobre a condição inicial das variáveis de interesse. Ele não exige que todos os dados anteriores sejam mantidos no armazenamento e reprocessados toda vez que uma nova medição é realizada. Com este comportamento, o filtro de Kalman é implementado como um algoritmo no processador [3].

O filtro é subdividido em duas etapas, predição e correção, baseado em técnicas recursivas do sistema representado no espaço de estados, sendo uma estimação sobre a dinâmica do sistema. Durante a primeira etapa é realizada uma predição sobre a dinâmica do modelo e no segundo passo uma correção, atuando na covariância do erro. Neste sentido, o Filtro de Kalman funciona como um estimador e otimizador do estado  $x_k$  com a medição de  $z_k$  [5].

Considere a representação:

$$x_k = A_{k-1}x_{k-1} + Bu_{k-1} + w_{k-1}z_k = Hx_k + v_k \quad (1)$$

Sendo:

$k$  o tempo atual;

$x_k$  o estado atual giroscópio no tempo  $k$ ;

$A_{k-1}$  e  $B_{k-1}$  entradas de controle do sistema;

$u_k$  representa as medidas retiradas do MPU-6050;

$w_k$  o ruído do sistema;

$z_k$  é a medida real acelerômetro;

$H$  o modelo de observação;

$v_k$  o ruído de medição.

Onde:

$$A = \begin{bmatrix} -1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad (2)$$

$$B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \quad (3)$$

$$x_k = \begin{bmatrix} \theta_{giro} \\ \dot{\theta}_{giro} \end{bmatrix} \quad (4)$$

A matriz A e B são obtidas através das propriedades construtivas do MPU-6050 e fornecidas pelo fabricante.  $x_k$  retrata a saída real do giroscópio, representado por  $\theta_{giro}$  e o escorregamento, representado por  $\dot{\theta}_{giro}$  no tempo  $k$  em graus por segundo ( $^\circ/s$ ).

$$H = [1 \quad 0] \quad (5)$$

$w_k$  e  $v_k$  são considerados ruído branco gaussiano não correlacionados, oriundos de erros de modelagem e dos sensores da medição, respectivamente. Com isso, tais ruídos em cada amostra tem uma distribuição normal com média zero e covariância da matriz  $Q$  do ruído do acelerômetro e  $R$  a matriz de medição do processo. Estas matrizes podem ser atualizadas a cada execução do filtro e sem prejuízo podem ser consideradas constantes do sistema. A matriz  $Q$  pode ser obtida na folha de dados do sensor. A matriz  $R$  deve ser o quadrado do desvio padrão do sensor, também obtida na folha de dados ou pode ser obtida de forma experimental mantendo o sistema estacionário e registrando os dados por aproximadamente cinco minutos [5].

$$Q = \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_{\theta_{giro}} \end{bmatrix} \cdot \Delta t \quad (6)$$

$$R = E[v_k v_k^T] = E[\sigma_k^2] \quad (7)$$

$$\bar{v}_k = x_k - \bar{x}_k \quad (8)$$

$$v_k = x_k - \hat{x}_k \quad (9)$$

As variáveis  $\hat{x}_{k-1|k-1}$  denotam valores de tempo “k-1” com dados de estado estimado previamente. Variáveis  $\hat{x}_{k|k-1}$ , exprimem estados no instante de tempo “k” com base no estado prévio “k-1”. Já as notações  $\hat{x}_{k|k}$  explicitam estimações de estado no instante “k” com base no estado atual. Obteve-se, durante estimação, o erro covariante no instante de tempo “k” com base no estado anterior “k-1” ou erro covariante preditivo,  $P_{k|k-1}$ , e, durante a atualização, o erro covariante no instante “k” com base no estado atual ou erro covariante corrigido,  $P_{k|k}$  [5]:

$$P_{k|k-1} = E[v_k v_k^T] \rightarrow P_{k|k-1} = A \cdot P_{k-1|k-1} \cdot A^T + Q \quad (10)$$

$$P_{k|k} = E[v_k v_k^T] \rightarrow P_{k|k} = P_{k|k-1} - K_k \cdot H \cdot P_{k|k-1} \quad (11)$$

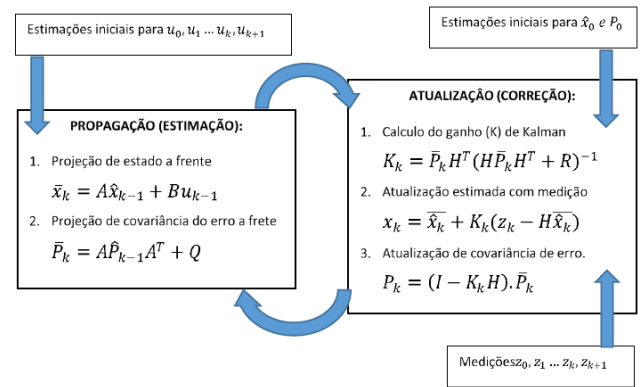
A etapa de atualização, ou correção, corresponde ao cálculo do ganho, determinado pela (12), atualização do estado estimando pela (13), seguida da atualização do erro de covariância dado pela (14) que correspondem a segunda etapa do algoritmo.

$$K_k = P_{k|k-1} \cdot H^T (H \cdot P_{k|k-1} \cdot H^T + R)^{-1} \quad (12)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H \cdot \hat{x}_{k|k-1}) \quad (13)$$

$$P_{k|k} = (1 - K_k H) P_{k|k-1} \quad (14)$$

A Fig 11 demonstra o funcionamento das duas etapas do algoritmo.



Fonte: Adaptado de Welch e Bishop (2006).

Fig.13. Diagrama do funcionamento do algoritmo do Filtro de Kalman

Nas linhas 1 e 2 do algoritmo, a predição  $\bar{x}_k$  e a matriz  $\bar{P}_k$  são calculadas representando a predição no instante de tempo posterior, mas antes de se incorporar a mensuração  $z_k$ . Esta predição é obtida incorporando-se o sinal de controle  $u_k$ . A média é atualizada usando a versão determinística da função de transição de estados, com a média  $\hat{x}_{k-1}$  substituída pelo valor do estado  $x_{k-1}$ . A atualização da covariância considera o fato de que os estados dependem dos estados anteriores através da matriz  $A_k$ . Esta matriz é multiplicada duas vezes na covariância, uma vez que a covariância é uma matriz quadrática.

A predição é subsequentemente transformada na predição desejada nas linhas 4 a 6 do algoritmo, ao se incorporar a medição  $z_k$ . A variável  $K_k$ , computada na linha 3, é chamada de ganho de Kalman. Ele especifica o quanto que a medição será incorporada na estimativa do novo estado. A linha 4 manipula a média, ao ajustá-la em proporção ao ganho de Kalman,  $K_k$ , o desvio da medição atual,  $z_k$  e a predição da medição de acordo com a probabilidade da medição. Finalmente, a nova covariância da predição posterior é calculada no item 5, de acordo com informações do ganho resultantes da medição [5].

### 1. Modelagem linear de um motor de corrente contínua

Devido à necessidade do projeto de um controle preciso da saída do motor é de suma importância a representação do seu modelo de espaços de estados. Então este modelo será utilizado na modelagem dinâmica do robô para o fornecimento de uma relação entre a tensão de entrada dos motores e o torque de controle necessário para equilibrar o robô [3]. A modelagem apresentada é baseada no trabalho do [3] e do [4].

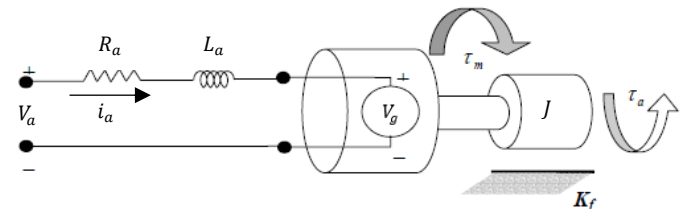


Fig. 14. Modelo para representação de um motor de corrente contínua

Uma tensão  $V_a$  é aplicada nos terminais de alimentação do motor, então uma corrente  $I_a$  é gerada em sua armadura. Por sua vez, o motor produz um torque  $\tau_m$ , dado em  $Nm$ , que é proporcional à corrente multiplicada pela constante de torque do motor  $k_m$ , dada em  $\frac{Nm}{A}$ .

$$\tau_m = k_m \cdot i_a \quad (15)$$

Já a armadura é modelada por uma resistência constante  $R_a$  ( $\Omega$ ) em série com uma indutância do enrolamento da armadura  $L_a$  ( $H$ ). A fonte de tensão  $V_g$  representa a tensão gerada na armadura, que se opõe a corrente da armadura  $I_a$  ( $A$ ). A primeira de uma série de equações básicas que modelam o motor de corrente contínua é obtida escrevendo uma equação de tensão na malha da armadura [7].

$$V_a - R_a \cdot i_a - L_a \cdot \frac{di_a(t)}{dt} = V_g \quad (16)$$

Supondo que a corrente de campo seja constante e ignorando as mudanças de fluxo de campo devido à reação da armadura e outros efeitos secundários o fluxo de campo se torna constante [7].

$$V_g = k_e \cdot \omega \quad (17)$$

Em que:

$k_e$ : Constante de proporcionalidade eletromotriz  $\left(\frac{V \cdot s}{rad}\right)$ ;

$\omega$ : Velocidade angular do eixo  $\left(\frac{rad}{s}\right)$ .

Substituindo a (17) na (16) e reorganizando, obtém-se a seguinte expressão:

$$L_a \cdot \frac{di_a(t)}{dt} = V_a - R_a \cdot i_a - k_e \cdot \omega \quad (18)$$

Para um modelo simplificado do motor a indutância do enrolamento da armadura é desprezível.

$$i_a = \frac{V_a}{R_a} - \frac{k_e \cdot \omega}{R_a} \quad (19)$$

A soma de todos os torques produzidos no eixo está linearmente relacionada à aceleração do eixo pela carga inercial  $J$  da armadura:

$$\sum M = \tau_m - k_f \cdot \omega - \tau_a = J \cdot \alpha \quad (20)$$

Onde  $k_f \cdot \omega$ : conjugado de resistência mecânica e particularmente o  $k_f$  é o atrito viscoso  $\left(\frac{Nm \cdot s}{rad}\right)$ .  $\tau_a$ : torque aplicado ( $Nm$ ),  $J$ : inércia do motor em  $(Kg \cdot m^2)$  e  $\alpha$ : aceleração angular  $\left(\frac{rad}{s^2}\right)$ .

Substituindo a (15) na (20) e reorganizando obtém-se a aceleração angular  $\alpha$ :

$$\alpha = \frac{d\omega(t)}{dt} = \frac{k_m}{J} \cdot i_a - \frac{k_f}{J} \cdot \omega - \frac{\tau_a}{J} \quad (21)$$

Já que um modelo simplificado do motor é o suficiente para o projeto,  $k_f$  é desprezível.

$$\frac{d\omega(t)}{dt} = \frac{k_m}{J} \cdot i_a - \frac{\tau_a}{J} \quad (22)$$

Substituindo a (19) na (22) obtemos uma aproximação a aceleração instantânea do motor.

$$\frac{d\omega(t)}{dt} = \frac{k_m}{J \cdot R_a} \cdot V_a - \frac{k_m \cdot k_e}{J \cdot R_a} \cdot \omega - \frac{\tau_a}{J} \quad (23)$$

Finalmente a dinâmica do motor é então representada pelo modelo de espaço de estados. A posição angular e a aceleração angular são as variáveis de estado. As entradas são representadas pela tensão aplicada e o torque aplicado. Já a saída é a posição angular do robô.

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{k_m \cdot k_e}{J \cdot R_a} \end{bmatrix} \cdot \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{J \cdot R_a} & -\frac{1}{J} \end{bmatrix} \cdot \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \quad (24)$$

$$y = [1 \quad 0] \begin{bmatrix} \theta \\ \omega \end{bmatrix} + [0 \quad 0] \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \quad (25)$$

### J. Modelagem dinâmica para as rodas do robô

O comportamento do robô pode ser influenciado por qualquer perturbação externa e reflete diretamente no torque do motor. Desta forma, o modelo matemático, das rodas esquerda e direita, necessita acomodar essas forças. A Fig. 13 mostra o diagrama de corpo livre para a roda direita. Para a roda esquerda utiliza-se o mesmo diagrama, alterando apenas  $H_{rd}$  por  $H_{re}$ .

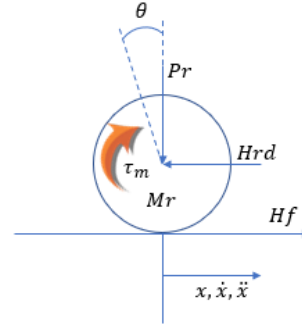


Fig.15. Diagrama de corpo livre para a roda direita

Empregando a Primeira Lei de Newton para descrever o princípio fundamental da dinâmica, somando-se as forças horizontais:

$$\sum F_x = M \cdot a \quad (26)$$

$$M_r \cdot \ddot{x} = H_f - H_{rd} \quad (27)$$

Onde:

$M_r$ : massa da roda ( $Kg$ );

$\ddot{x}$ : aceleração linear  $\left(\frac{m}{s^2}\right)$ ;

$H_f$ : força de atrito entre a roda e o chão ( $N$ );

$H_{rd}$ : força de reação entre a roda e o corpo ( $N$ ).

Para a modelagem do sistema em rotação, considera-se a Fig. 14.

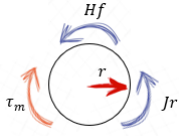


Fig. 16. Diagrama do sistema em rotação das rodas

Somando as forças de rotação da roda:

$$\sum M_0 = I \cdot \alpha \quad (28)$$

$$J_r \cdot \alpha = \tau_m - H_f \cdot r \quad (29)$$

Onde:

$I$ : momento de inercia ( $Kg \cdot m^2$ );

$\alpha$ : aceleração angular ( $\frac{rad}{s^2}$ );

$J_r$ : momento de inercia da roda ( $Kg \cdot m^2$ );

$\tau_m$ : torque aplicado do motor ( $N \cdot m$ );

$H_f$ : força de atrito entre a roda e o chão ( $N$ );

$r$ : raio da roda;

O torque do motor pode ser expresso pela equação:

$$\tau_m = J \cdot \frac{d\omega(t)}{dt} + \tau_a \quad (30)$$

$\tau_a$ : torque aplicado ( $Nm$ ).

Substituir a (23) na (30):

$$\tau_m = J \cdot \left( \frac{k_m}{J \cdot R_a} \cdot V_a - \frac{k_m \cdot k_e}{J \cdot R_a} \cdot \omega - \frac{\tau_a}{J} \right) + \tau_a \quad (31)$$

Reorganizando a equação, obtém-se o torque de saída para as rodas.

$$\tau_m = \frac{k_m}{R_a} \cdot V_a - \frac{k_m \cdot k_e}{R_a} \cdot \omega \quad (32)$$

Substituindo a (32) na (29) e reorganizando

$$H_f = \frac{k_m}{R_a \cdot r} \cdot V_a - \frac{k_m \cdot k_e}{R_a \cdot r} \cdot \omega - \frac{J_r \cdot \alpha}{r} \quad (33)$$

Substituindo a (33) na (27) obtém-se a equação de movimento de uma das rodas.

$$M_r \cdot \ddot{x} = \frac{k_m}{R_a \cdot r} \cdot V_a - \frac{k_m \cdot k_e}{R_a \cdot r} \cdot \omega - \frac{J_r}{r} \cdot \alpha - H_{rd} \quad (34)$$

Já que o movimento linear está agindo no centro da roda, a rotação angular pode ser transformada em movimento linear, ou seja:

$$\dot{x} = \omega \cdot r$$

Utilizando a transformação linear e substituindo na (34), obtém-se a equação da roda direita (35) e da roda esquerda (36):

$$M_r \cdot \ddot{x} = \frac{k_m}{R_a \cdot r} \cdot V_a - \frac{k_m \cdot k_e}{R_a \cdot r^2} \cdot \dot{x} - \frac{J_r}{r^2} \cdot \ddot{x} - H_{rd} \quad (35)$$

$$M_r \cdot \ddot{x} = \frac{k_m}{R_a \cdot r} \cdot V_a - \frac{k_m \cdot k_e}{R_a \cdot r^2} \cdot \dot{x} - \frac{J_r}{r^2} \cdot \ddot{x} - H_{re} \quad (36)$$

Adicionando a (35) na (36):

$$2 \cdot \left( M_r + \frac{J_r}{r^2} \right) \cdot \ddot{x} = 2 \cdot \left( \frac{k_m}{R_a \cdot r} \right) \cdot V_a - 2 \cdot \left( \frac{k_m \cdot k_e}{R_a \cdot r^2} \right) \cdot \dot{x} - (H_{rd} + H_{re}) \quad (37)$$

### K. Modelagem dinamica corpo do pêndulo

O modelo matemático associado ao corpo do pêndulo também precisa acomodar forças de perturbações. A Fig. 15 mostra o diagrama de corpo livre do corpo do corpo do pêndulo.

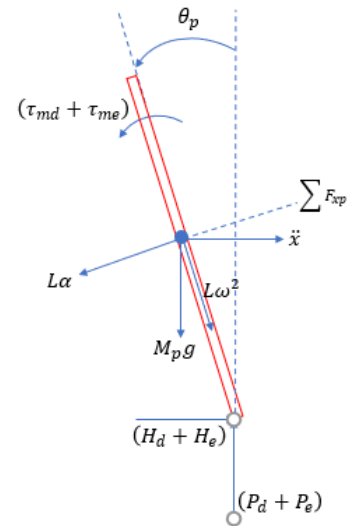


Fig. 17. Diagrama de corpo livre para o corpo do robô

Onde:

$\theta_p$ : Ângulo de rotação do chassi ( $rad$ );

$(\tau_{md} + \tau_{me})$ : Torque do motor direito e esquerdo ( $Nm$ );

$L$ : Comprimento do centro da roda ao centro de gravidade do robô ( $m$ );

$\alpha$ : Aceleração angular ( $\frac{rad}{s^2}$ );

$\omega$ : Velocidade angular ( $\frac{rad}{s}$ );

$\ddot{x}$ : Aceleração linear ( $\frac{m}{s^2}$ );

$M_p$ : Massa do corpo ( $m$ );

$(H_d + H_e)$  e  $(P_d + P_e)$ : Força de reação entre a roda e o corpo ( $N \cdot m$ ).

Utilizando novamente a Segunda Lei de Newton e utilizando a (26), a soma das forças na direção horizontal:

$$H_d - H_e - M_p \cdot L \cdot \ddot{\theta}_p \cdot \cos(\theta_p) + M_p \cdot L \cdot \dot{\theta}_p^2 \cdot \sin(\theta_p) = M_p \cdot \ddot{x} \quad (38)$$

Somando as forças verticais do pêndulo:

$$\sum F_{xp} = M_p \cdot \ddot{x} \cdot \cos(\theta_p) \quad (39)$$

$$(H_d + H_e) \cdot \cos(\theta_p) + (P_d - P_e) \cdot \sin(\theta_p) - M_p \cdot g \cdot \sin(\theta_p) - M_p \cdot L \cdot \ddot{\theta}_p = M_p \cdot \ddot{x} \cos(\theta_p) \quad (40)$$

Somando as forças de rotação utilizando a (28).

$$-(H_d + H_e) \cdot L \cdot \cos(\theta_p) - (P_d - P_e) \cdot L \cdot \sin(\theta_p) - (C_L + C_r) = I_p \cdot \alpha_p \quad (41)$$

O torque aplicado no motor, conforme descrito na (32) e já aplicando a transformação linear:

$$\tau_{md} + \tau_{me} = 2 \cdot \frac{k_m}{R_a} \cdot V_a - 2 \cdot \frac{k_m \cdot k_e}{R_a} \cdot \frac{\dot{x}}{r} \quad (42)$$

Substituindo a (41) na (42).

$$[-(H_d + H_e) \cdot L \cdot \cos(\theta_p) - (P_d - P_e) \cdot L \cdot \sin(\theta_p)] - 2 \cdot \frac{k_m}{R_a} \cdot V_a + 2 \cdot \frac{k_m \cdot k_e}{R_a \cdot r} \cdot \dot{x} = I_p \cdot \alpha_p \quad (43)$$

Multiplica-se a (40) por  $(-L)$ :

$$[-(H_d + H_e) \cdot L \cdot \cos(\theta_p) - (P_d - P_e) \cdot L \cdot \sin(\theta_p)] + M_p \cdot g \cdot L \cdot \sin(\theta_p) + M_p \cdot L^2 \cdot \alpha_p = -M_p \cdot L \cdot \ddot{x} \cdot \cos(\theta_p) \quad (44)$$

Substituindo-se a (43) na (44).

$$2 \cdot \frac{k_m}{R_a} \cdot V_a - 2 \cdot \frac{k_m \cdot k_e}{R_a \cdot r} \cdot \dot{x} + M_p \cdot g \cdot L \cdot \sin(\theta_p) + (I_p + M_p \cdot L^2) \cdot \alpha_p = -M_p \cdot L \cdot \ddot{x} \cdot \cos(\theta_p) \quad (45)$$

Substituindo-se a (38) na (37):

$$2 \cdot \left(M_r + \frac{J_r}{r^2}\right) \cdot \ddot{x} = 2 \cdot \left(\frac{k_m}{R_a \cdot r}\right) \cdot V_a - 2 \cdot \left(\frac{k_m \cdot k_e}{R_a \cdot r^2}\right) \cdot \dot{x} - M_p \cdot L \cdot \ddot{\theta}_p \cdot \cos(\theta_p) + M_p \cdot L \cdot \dot{\theta}_p^2 \cdot \sin(\theta_p) - M_p \cdot L \cdot \ddot{x} \quad (46)$$

(45) e (46) são as equações não lineares do sistema. Estas equações podem ser linearizadas assumindo  $\theta = \pi + \phi$ , onde  $\phi$  representa um pequeno ângulo a partir da direção ascendente vertical.

Esta simplificação foi empregada para permitir que os controladores de estado linear pudessem ser implementados.

Sendo assim:

$$\cos(\theta_p) = \cos(\pi + \phi) \approx -1$$

$$\sin(\theta_p) = \sin(\pi + \phi) \approx -\phi$$

$$\left(\frac{d\theta(t)}{dt}\right)^2 = \left(\frac{d(\pi + \phi)}{dt}\right)^2 = 0$$

Linearizando e reorganizando a (45) e (46):

$$\ddot{\phi} = \frac{M_p \cdot L}{M_p \cdot L^2 + I_p} \ddot{x} + 2 \cdot \frac{k_m \cdot k_e}{(M_p \cdot L^2 + I_p) R_a \cdot r} \cdot \dot{x} + \frac{M_p \cdot g \cdot L}{M_p \cdot L^2 + I_p} \cdot \phi - 2 \cdot \frac{k_m}{(M_p \cdot L^2 + I_p) \cdot R_a} \cdot V_a \quad (47)$$

$$\ddot{x} = -\frac{2 \cdot k_m \cdot k_e}{R_a \cdot r^2 \cdot (2 \cdot M_r + 2 \cdot \frac{J_r}{r^2} + M_p)} \cdot \dot{x} + \frac{M_p \cdot L}{(2 \cdot M_r + 2 \cdot \frac{J_r}{r^2} + M_p)} \cdot \ddot{\phi} + \frac{2 \cdot k_m}{R_a \cdot r \cdot (2 \cdot M_r + 2 \cdot \frac{J_r}{r^2} + M_p)} \cdot V_a \quad (48)$$

Enfim a dinâmica do corpo do pêndulo é representada pelo modelo de espaço de estados. A posição linear, velocidade linear, posição angular e velocidade angular descrevem as variáveis de estado. A entrada é representada pela tensão aplicada no motor.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_e (M_p L r - I_p - M_p L^2)}{R r^2 \delta} & \frac{M_p g L^2}{\delta} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e (r \beta - M_p L)}{R r^2 \delta} & \frac{M_p g L \beta}{\delta} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m (I_p + M_p L^2 - M_p L r)}{R r^2 \delta} \\ 0 \\ \frac{2k_m (M_p L - r \beta)}{R r \delta} \end{bmatrix} V_a \quad (49)$$

Onde:

$$\beta = 2M_r + \frac{2J_r}{r^2} + M_p$$

$$\delta = I_p \beta + 2M_p L^2 \left(M_r + \frac{J_r}{r^2}\right)$$

#### L. Controlador PID

O controlador proporcional integral derivativo (PID) é um controlador que vêm sendo amplamente difundido na literatura em diversas aplicações, que vão de controle de processos químicos [7]. A utilidade dos controles PID estão na sua aplicabilidade geral à maioria dos sistemas de controle [9]. A combinação das ações proporcional, integral e derivativa portam o objetivo de gerar um só sinal de controle aproveitando

as características de cada uma das ações obtendo uma melhora significativa do comportamento transitório e em regime permanente do sistema controlado [8].

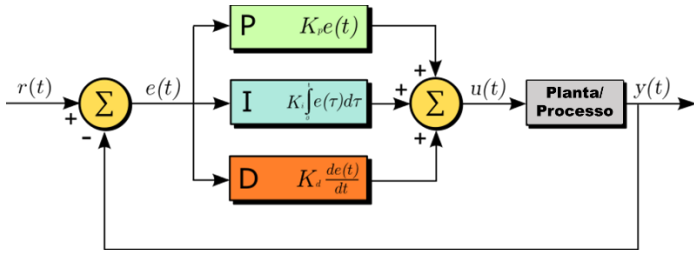


Fig.18. Estrutura do controlador PID

O sinal de controle gerado pelo controlador PID é assim genericamente dado como:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (50)$$

Sendo:

$K_p$ : ganho proporcional;

$K_i$ : ganho integral;

$K_d$ : ganho derivativo;

$e$ : erro;

$t$ : tempo;

$\tau$ : tempo de integração;

A ação proporcional produz um valor de saída proporcional ao valor do erro atual, um ganho proporcional alto resulta em uma grande mudança na saída podendo tornar o sistema instável. Se o ganho proporcional for muito baixo, a ação de controle pode ser muito pequena ao responder a distúrbios do sistema.

O termo integral é proporcional à magnitude do erro e à duração do erro. A integral em um controlador PID é a soma do erro instantâneo ao longo do tempo (tempo de integração) e fornece o deslocamento acumulado que deveria ter sido corrigido anteriormente. O erro acumulado é então multiplicado pelo ganho integral e adicionado à saída do controlador. Se o valor  $K_i$  for baixo, o sistema pode levar um tempo considerável para atingir o valor de referência. Entretanto, se o valor  $K_i$  for muito alto, o sistema pode tornar-se instável.

A ação derivativa prevê o comportamento do sistema e, portanto, melhora o tempo de estabilização e a estabilidade do sistema. A ação derivativa atua em intervalos regulares, isso indica que a ação derivativa não deve ser utilizada em processos nos quais o sistema deve responder rapidamente a uma perturbação, nem em processos que apresentem muito ruído no sinal de medido, pois levaria o processo à instabilidade.

Projetar e ajustar um controlador PID parece ser conceitualmente intuitivo, mas pode ser difícil na prática, se objetivos múltiplos (e frequentemente conflitantes), como transientes curtos e alta estabilidade, forem alcançados. Os controladores PID geralmente fornecem controle aceitável usando afinações padrão, mas o desempenho geralmente pode

ser melhorado com um ajuste cuidadoso, o desempenho pode ser inaceitável com um ajuste ruim. Geralmente, os projetos iniciais precisam ser ajustados repetidamente por meio de simulações de computador até que o sistema de malha fechada se comporte conforme desejado [9].

Utilizando a Transformada de Laplace em (50), obteve-se a expressão no domínio da frequência para o controlador PID:

$$u(t) = K_p + \frac{K_i}{s} + K_d s \quad (51)$$

Uma vez projetado no tempo contínuo o mesmo deve ser discretizado para fins de implementação digital.

#### IV. PROPOSTA

##### A. Implementação do Filtro de Kalman

Com o intuito de corrigir o erro em regime permanente do giroscópio, e o erro do acelerômetro, que amplificou altas frequências sob trepidação, empregou-se o algoritmo de filtragem de Kalman. Para este processo, foram utilizados um controlador Arduino Uno, um sensor MPU-6050 e um servomotor SG90. A alimentação do circuito foi realizada por uma fonte externa de 5 V para garantir menor interferência na leitura do sensor MPU-6050. Abaixo na Fig. 17 é possível observar o circuito construído para o Filtro de Kalman.

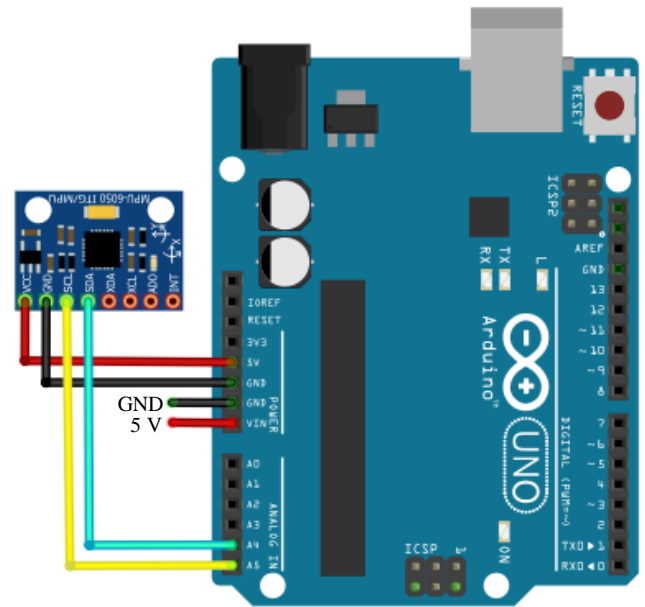


Fig. 19. Circuito teste do Filtro de Kalman

O sensor MPU-6050 foi acoplado ao servomotor controlado por um Arduino para ficar oscilando de  $-60^\circ$  a  $60^\circ$ . Subsequentemente, um programa desenvolvido no Arduino utilizando uma biblioteca específica para o Filtro de Kalman foi executado e o resultado pode ser conferido na Fig. 19 associado diretamente a (4), o giroscópio, representado na cor azul, nos mostra o escorregamento na sua medida ao longo do tempo amostrado. Para vibrações abruptas as medidas reais do acelerômetro  $z_k$ , que está representado pela cor vermelha, pode-se verificar um comportamento ruidoso, principalmente

na mudança de direção do servomotor. Contudo, o Filtro de Kalman, na cor preta, foi o que demonstrou melhor resultado de medida, rejeitando os comportamentos indesejados do escorregamento do giroscópio e das oscilações do acelerômetro. Se somente as medidas do acelerômetro ou do giroscópio fossem utilizadas no projeto, provavelmente causaria um descontrole no robô auto equilibrante resultando em sua queda.

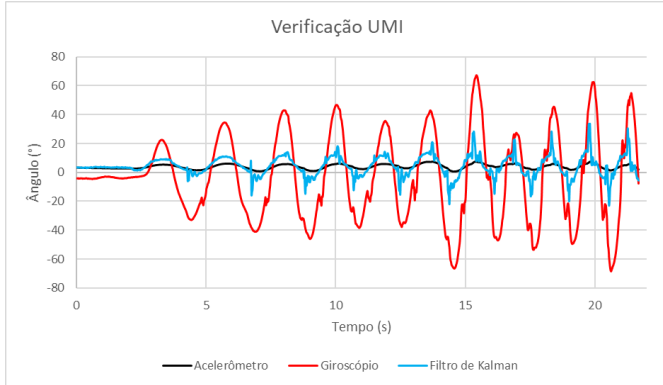


Fig. 20. Resultado das medidas do giroscópio, na cor azul, do acelerômetro, na cor vermelha e da fusão dos dois sensores, com o Filtro de Kalman empregado, na cor preta

### B. Projeto Mecânico.

Por intermédio da plataforma online Grabcad, que disponibiliza projetos de forma gratuita, obteve-se o projeto 3D da estrutura robô auto equilibrante. O robô possui uma base e uma prateleira sustentada por quatro pilares. Na base ficam presos a UMI, os motores e em seus eixos acoplam-se as rodas permitindo a movimentação do robô. Na prateleira superior, ficam posicionados os controladores, a ponte H e o módulo bluetooth. Algumas alterações nas medidas projeto 3D da plataforma online foram realizadas, partindo da medida do raio das rodas de 34 mm, já que foi adquirido o conjunto roda-motor com um raio diferente do projeto da plataforma. Após ajustado no projeto o tamanho das rodas, foi corrigido, proporcionalmente, o tamanho do corpo do robô. Então o sistema mecânico projetado para do robô possuirá 870 g de massa total, largura de 150 mm, com uma altura total de 120 mm e uma profundidade de 90 mm, podendo ser observado na Fig 20.

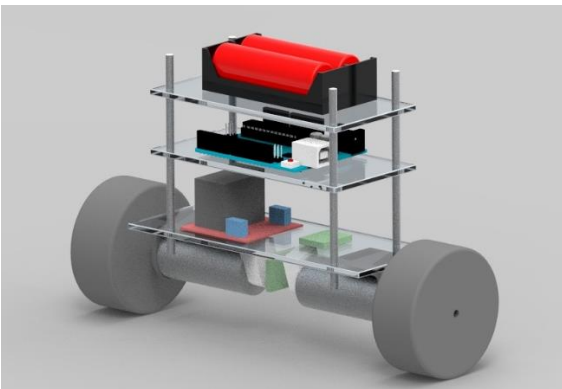


Fig. 21. Projeto mecânico robô auto equilibrante

### C. Projeto Mecânico executado

Com o projeto mecânico concluído, iniciou-se a construção física do veículo. O sistema mecânico real do robô possui 833 g de massa total, sendo a massa do corpo  $M_p$  de 768 g e a massa da roda  $M_r$  de 65 g, largura de 152 mm, com uma altura total de 119 mm e uma profundidade de 90 mm, as rodas possuem raio  $r$  de 34 mm, a distância entre os centros da roda e o centro de gravidade do robô  $L$  de 52,5 mm.

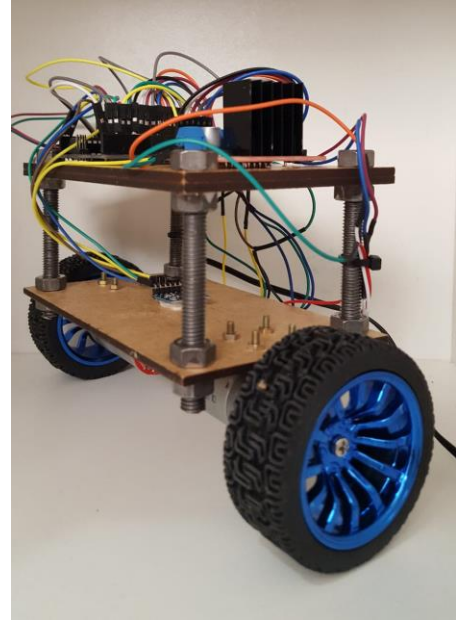


Fig. 22. Sistema mecânico robô auto equilibrante

### D. Controlador PID

Para a implementação do PID são indispensáveis as seguintes características do sistema para a substituição no modelo em espaço de estados da (49):

- $k_m = 0,006123 \frac{Nm}{A}$
- $k_e = 0,006087 \frac{V}{s \cdot rad}$
- $R = 3,5 \Omega$
- $I_p = \frac{M_p(altura^2 + largura^2)}{12} = 0,001891584 \frac{kg}{m^2}$
- $J_r = \frac{M_r r^2}{2} = 3,757 \times 10^{-5} \frac{kg}{m^2}$
- $\beta = 2M_r + \frac{2I_r}{r^2} + M_p = 0,9630$
- $\alpha = I_p \beta + 2M_p L^2 \left( M_r + \frac{I_w}{r^2} \right) = 0,0022$

Substituindo as variáveis do modelo de espaço de estados para o robô dada pela (49) obtêm-se:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \phi \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -32,5059 & 7,1376 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -93,395 & 170,4749 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 5,7864 \\ 0 \\ 16,6253 \end{bmatrix} V_a$$

Observando os autovalores da matriz dinâmica do sistema acima, verificamos que os autovalores do sistema estão em  $[0 \ -33,2203 \ -11,7618 \ 12,4762]$  verificando que se

trata de um sistema instável, visto que existe um polo no semiplano direito.

Se um modelo matemático da planta pode ser obtido, então é possível aplicar várias técnicas de projeto na determinação dos parâmetros do controlador que vão impor as especificações do regime transitório e do regime permanente do sistema de malha fechada [9]. Para este projeto a ferramenta SISOtool disponível no MATLAB foi utilizada para ajustar o PID interativamente determinando assim um novo regime transitório e permanente.

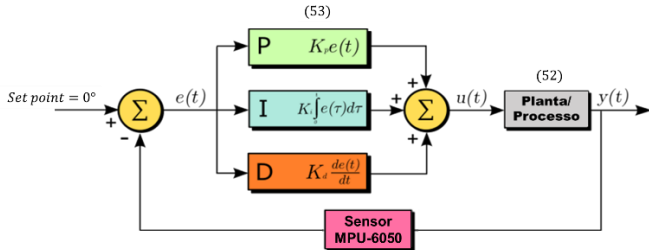


Fig. 23. Controlador PID implementado no projeto

A função de transferência da planta que relaciona a tensão aplicada nos motores e o ângulo do pêndulo, mostrada na (52), foi inserida no controle de arquitetura da ferramenta:

$$tf_{Planta} = \frac{0,349 s - 1,213 e^{-18}}{s^3 + 0,02175 s^2 - 170,5 s - 3.261} \quad (52)$$

No Root Locus Editor acrescentou-se um compensador C de 1000, um integrador na origem e dois zeros em -2 e -8 para que fosse possível atingir a estabilidade do sistema para uma entrada do tipo impulso. Com estas modificações a planta alcança estabilização em aproximadamente 5 segundos tornando possível o equilíbrio do robô.

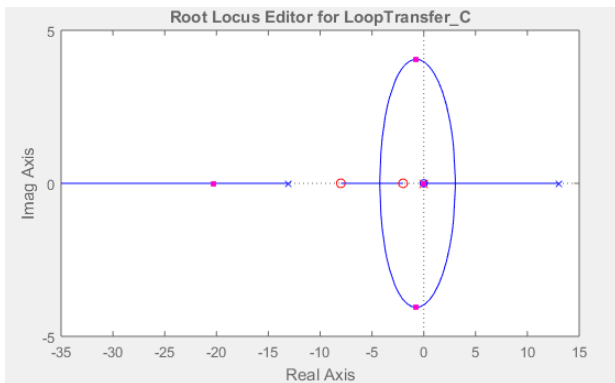


Fig. 24. Lugar geométrico das raízes

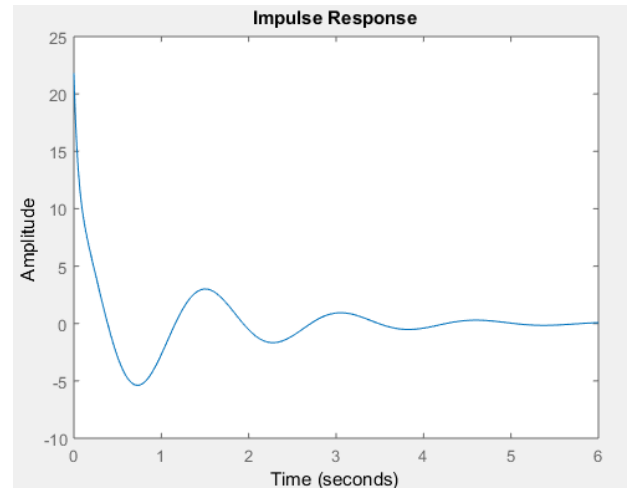


Fig. 25. Resposta a entrada do tipo impulso contínua.

Os valores  $K_p = 625$ ,  $K_i = 1000$  e  $K_d = 62.5$  foram obtidos através do SISOtool para sua implementação no algoritmo do Arduino. A seguinte função de transferência representa o controlador PID:

$$tf_{PID} = \frac{62,5 s^2 + 625 s + 1000}{s} \quad (53)$$

Sinais de tempo discreto surgem se o sistema envolver uma operação de amostragem de sinais de tempo contínuo. O sinal amostrado é  $x(0)$ ,  $x(T)$ ,  $x(2T)$ , ..., onde  $T$  é o período de amostragem. Tal sequência de valores decorrentes da operação de amostragem é geralmente escrita como  $x(kT)$  [13]. Neste caso, com o intuito de obter um equivalente discreto em malha fechada para o controlador contínuo já projetado foi empregado o método de aproximação de Tustin através do comando `c2d(SYSC,TS,METHOD)` inserido no MATLAB, onde SYSC representa a função obtida no SISOtool, em TS é inserido o tempo de amostra em segundos e METHOD se insere o método escolhido para discretização, onde:

- $SYSC = \frac{62,5 s^2 + 625 s + 1000}{s}$
- $TS = 0,05$
- $METHOD = \text{tustin}$

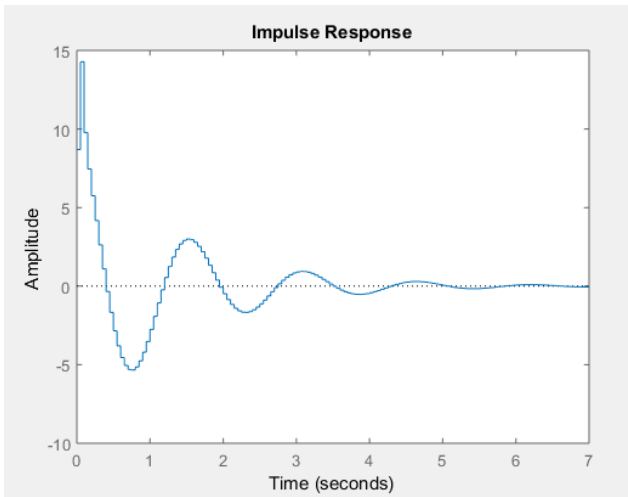


Fig. 26. Resposta discretizada a entrada do tipo impulso

## V. RESULTADOS

### A. Circuito.

Empregando os materiais descritos anteriormente as seguintes conexões foram executadas no Arduino Uno e na ponte H L298N:

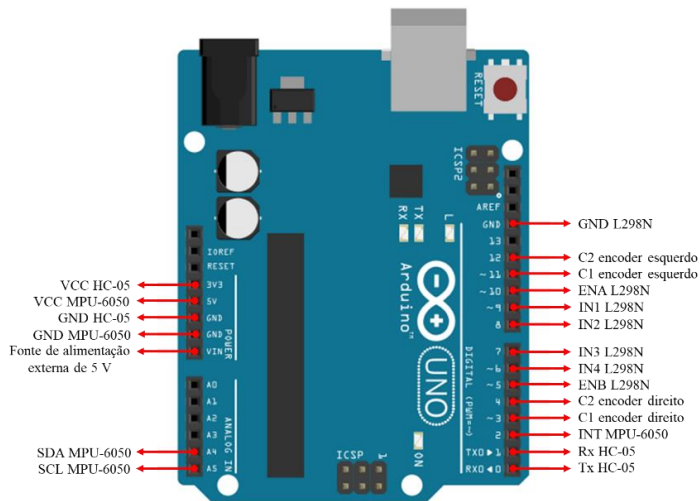


Fig. 27. Diagrama das conexões do Arduino Uno

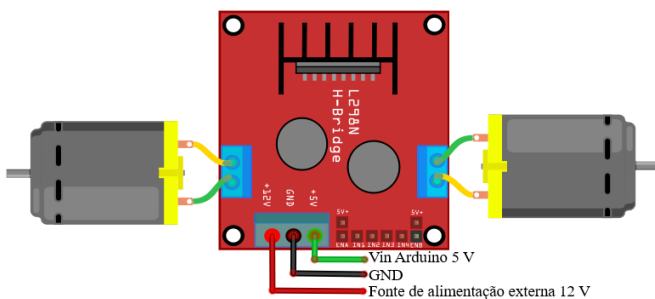


Fig. 28. Diagrama das conexões ponte H L298N

### B. Algoritmo implementado no Arduino Uno

De uma forma geral o funcionamento do algoritmo do robô é simples, ou seja, se o robô estiver inclinando para frente, o controlador PID ajustará os motores para mover as rodas na direção para frente até obter o ponto de equilíbrio (set point). Caso contrário, se o robô estiver inclinando para trás, o controlador PID ajustará os motores para mover as rodas na direção para trás até obter o ponto de equilíbrio.

Entretanto, a implementação do algoritmo de controle de equilíbrio baseou-se na simulação no MATLAB do controlador PID. Porém os ganhos valores  $K_p = 625$ ,  $K_i = 1000$  e  $K_d = 62.5$  obtidos a partir do resultado da simulação não foram aplicados com sucesso no algoritmo, já que não ocasionaram em um resultado efetivo devido a saturação da tensão de movimentação dos motores provocando a queda do robô e na não regulação do mesmo. Esta saturação pode ter se dado tanto por erros da implementação digital do controlador PID bem como nas diferenças de modelo matemático e hardware implementado. A partir da observação empírica e ajuste de ganhos partindo dos ganhos inicialmente projetados, se chegou em  $K_p = 9$ ,  $K_i = 110$  e  $K_d = 0,17$  que acarretaram no equilíbrio do robô dentro dos requisitos previamente estipulados.

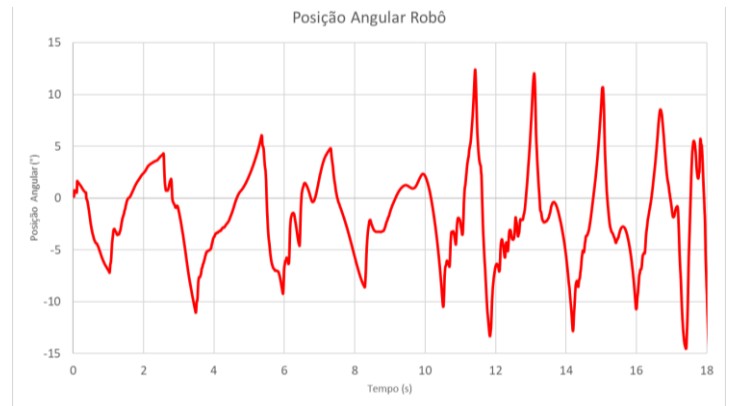


Fig. 29. Posição angular do robô obtido experimentalmente

## VI. CONCLUSÃO

O projeto mecânico e a integração de componentes foram realizados com sucesso. Do ponto de vista de software e do sistema de controle, o sensor foi integrado com efetividade podendo-se extrair os sinais pelo Filtro de Kalman conforme descrito no artigo. Nos testes para o equilíbrio em malha fechada foram aplicados pequenos esforços externos na carcaça do robô que se manteve equilibrado. Em alguns momentos o robô apresentou falha no protocolo de comunicação I<sup>2</sup>C no Arduino Uno, levando a perda de seu equilíbrio, mas de um modo geral apresentou resultado satisfatório tanto de equilíbrio como de direção pelo aplicativo Joystick BT Commander.

Durante o processo de ajustes e desenvolvimento, foi utilizada uma biblioteca para Arduino para implementação do controlador PID o que tornou mais fácil a busca de ganhos que possibilitavam o equilíbrio do robô durante a etapa de

desenvolvimento. Entretanto, o desconhecimento dos detalhes de firmware desta estrutura não permite avaliar o projeto realizado do controlador PID utilizando o modelo matemático encontrado gráfico de lugar das raízes na ferramenta SISOtool do MATLAB. Não houve tempo hábil para avaliar as discrepâncias entre o modelo e a planta e de saturação da ação de controle encontradas quando se utilizou o controlador projetado.

Uma vez que o sistema de atuação e monitoramento funciona perfeitamente, fica como sugestão para trabalho futuro a análise de discrepâncias do modelo do hardware construído e a implementação de um algoritmo PID próprio que permita a aplicação direta dos ganhos projetados, além do possível uso de outras técnicas de controle, como a realimentação de estados.

#### REFERÊNCIAS

- [1] G. Lyo Torige, Construção de um protótipo de um veículo auto-equilibrante, Pontifícia Universidade Católica do Rio Grande do Sul, Departamento de Engenharia, 2013.
- [2] J. Piaget, Psicologia e Pedagogia: A resposta do grande psicólogo aos problemas do ensino. RJ Forense Universitária, p. 184, 2010.
- [3] R. Chi Ooi, Balancing a Two-Wheeled Autonomous Robot, University of Western Australia, Faculty of Engineering and Mathematical Sciences, 2003.
- [4] M. R. Bageant, Balancing a Two-Wheeled Segway Robot, Massachusetts Institute of Technology University, Department of Mechanical Engineering, 2011.
- [5] W. dos Santos Oliveira, e E. Nunes Gonçalves, Implementação em C: Filtro de Kalman, fusão de sensores para determinação de ângulos Universidade Federal de São João del-Rei, Departamento de Engenharia, 2016
- [6] A. E. Fitzgerald, C. Kingsley Jr. Máquinas Elétricas. Amgh Editora, 2014.
- [7] K. Uygun, H. W. T. Matthew e Y. Huang, An optimal control approach to flux balance analysis, Industrial & Engineering Chemistry Research, v. 45, n. 25, p. 8554, 2006.
- [8] João Manoel Gomes da Silva Jr e Alexandre Sanfelice Bazanella, Ajuste de Controlador PID. UFRGS, 2000.
- [9] Universidade de Michigan, Introduction: PID Controller Design. UM, 2018
- [10] Katsuhiko Ogata, Engenharia de Controle Moderno. Pearson, 2011.
- [11] Norman Nise, Engenharia de Sistemas de Controle. LTC, 2012.
- [12] NXP Semicondutores, UM10204 I<sup>2</sup>C-bus specification and user manual. NXP, 2014.
- [13] Katsuhiko Ogata, Discrete-time control systems. PHI, 1995.