Disciplina Controle e Servomecanismo

Professor Orides Morandin Jr.

Projeto do robô cartesiano no eixo Z
Relatório inicial de tarefas

Henrique Yuji Tati, RA: 496294
Leonardo Tavares Oliveira, RA: 628174
Leonardo Utida Alcantara, RA: 628182
Lucas Novaes Bezerra, RA: 594954
Matheus Vrech Silveira Lima, RA: 727349
Victor Bruno Sanches Cassano, RA: 552429
Victor Augusto Tavares, RA: 628140

Turma A, Sexta-feira

São Carlos, 2019

1. Introdução

O projeto do robô cartesiano é um projeto de engenharia completo para o controle de um elevador que deve se movimentar até uma posição desejada.

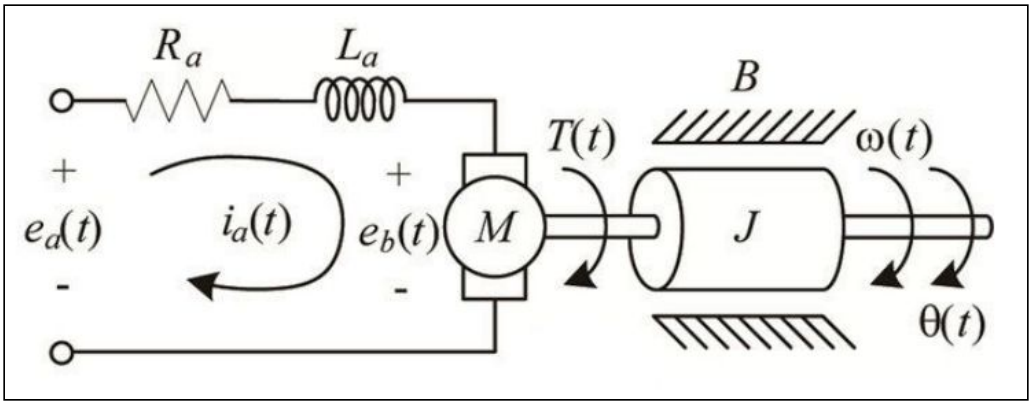
Foi realizado o levantamento das etapas e dos dados necessários para todo o desenvolvimento e dividido em grupos focados em tarefas específicas que, ao fim, foram agrupadas para gerar o resultado final.

A análise de sensibilidade da função de transferência Simulação do PID e análise das variáveis Kp, Ki e Kd foi o primeiro tópico a ser estudado, passando para o projeto mecânico, o projeto eletrônico, a definição de parâmetros de desempenho dos resultados obtidos, o projeto do software.

Cada etapa foi trabalhada de forma dedicada por uma parte específica do grupo e ao fim todos também ficaram responsáveis por uma proposta de melhoria de projeto, que foi a modificação do diagrama do controle, aumentando as variáveis que o sistema leva em consideração para melhorar a acurácia do resultado final do controle do eixo, referente ao seu tempo em realizar sua tarefa e ao seu erro de posição final.

2. Análise de sensibilidade da função de transferência

2.1 Motor CC



O circuito do motor é composto por uma entrada $e_a(t)$, que representa a força eletromotriz medida em Volts; uma resistência R_a e uma indutância L_a, que representa a indutância do enrolamento da armadura. Quando uma tensão é aplicada, a corrente *i* vai gerar um campo magnético, e este por sua vez vai agir na bobina do motor provocando um torque T(t) que gera uma rotação com velocidade $\omega(t)$. Além disso existem variáveis contrárias a este movimento como o atrito B e o momento de inércia J. Estes dois valores possuem componentes internas e externas ao motor, porém elas serão tratadas em conjunto pelo fato que as variáveis externas possuem valores muito maiores que a interna fazendo

2.2 Modelagem matemática e diagrama de blocos

A função de transferência do motor pode ser obtida a partir da manipulação de relações já conhecidas dos blocos que compõem o circuito do motor. Aplicando a lei de Kirchhoff na malha da figura mostrada na seção anterior, obtemos a seguinte equação:

$$e_a(t) = R_a \cdot i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t)$$

Se aplicarmos a lei de Faraday, chegaremos em:

$$e_b(t) = K_2 \cdot \omega(t)$$

onde K_2 **é a força contra eletromotriz**.

Quando um motor é atravessado por uma corrente i_a, um campo magnético será produzido, gerando um torque no eixo do motor. No caso do motor, a armadura (que é percorrida por corrente) cruza as linhas de campo, produzindo um torque τ_m, cujo módulo é:

$$\tau_m(t) = K_1 \cdot i_a(t)$$

onde K_1 **é a constante de torque do motor**.

Sabe-se também que a **inércia do sistema** τ_i é dada por:

$$\tau_i(t) = B\omega(t) + J\frac{d\omega(t)}{dt}$$

onde J é o **momento de inércia** e B é **o coeficiente de atrito** viscoso.

Como a soma dos torques do sistema é igual ao torque gerado pelo motor, e dados os valores de τ_m e τ_i, conclui-se que:

$$\tau_m(t) = \tau_d(t) + B\omega(t) + J\frac{d\omega(t)}{dt}$$

onde τ_d é o **torque introduzido pela carga**.

Finalmente, é necessário que as equações obtidas acima sejam passadas para o domínio da frequência. Para isso, pode-se aplicar a **transformada de Laplace.** Dessa forma teremos as seguintes equações:

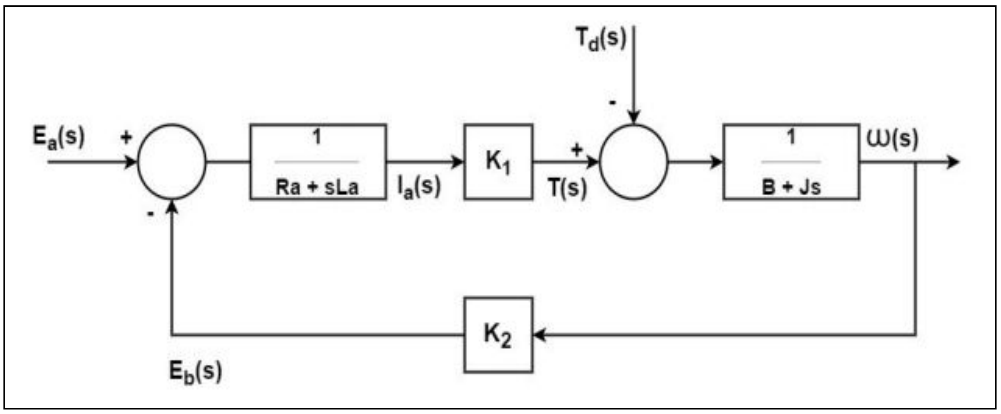
$$e_a(s) = (R_a + L_a s)\, i_a(s) + e_b(s)$$

$$e_b(s) = K_2 \cdot \omega(s)$$

$$\tau_m(s) = K_1 \cdot i_a(s)$$

$$\tau_m(s) = \tau_d(s) + (Js + B)\omega(s)$$

A partir destas funções podemos montar a função de transferência final a partir de blocos de outras funções que representam partes do sistema. Com isso conseguimos montar o diagrama de blocos mostrado abaixo contendo os blocos do sistema do motor com as funções de transferência utilizadas em cada parte.



O primeiro bloco diz respeito ao circuito da armadura (**parte elétrica**), que contém a função de transferência que gera a corrente $i_a(s)$ do circuito. Essa corrente, como discutido anteriormente, provoca um campo magnético, que por sua vez, atua na bobina do motor **(torque τ_s)**. O torque gerado nessa etapa, representado na saída do segundo bloco do diagrama, multiplica a corrente $i_a(s)$ pela constante do motor K_1.

Em seguida, tem-se um comparador (segundo círculo do diagrama) para representar a **perturbação τ_d do sistema**, que é principalmente o **torque de atrito do motor gerado pela sua rotação, e outras eventuais perturbações** que possam interferir no funcionamento do sistema.

O terceiro bloco, que representa a carga (**parte mecânica**), recebe o torque resultante e produz uma velocidade angular $w(s)$. Vale notar que o módulo da velocidade angular depende do **momento de inércia total do rotor e da carga (J),** e ao **atrito viscoso do rotor e da carga(B).**

Finalmente, existe um bloco no ramo de realimentação, no qual o valor da força contra eletromotriz (K_2) é **multiplicada pela velocidade angular e comparada com a entrada do sistema (*setpoint*) no primeiro comparador do sistema**.

No entanto, é desejável controlar, além da velocidade do motor, a posição. Para isso, deve-se encontrar uma relação entre a velocidade angular $\omega(s)$ com a posição angular $\theta(s)$.

Sabe-se que:

$$\omega(t) = \frac{d\theta(t)}{dt}$$

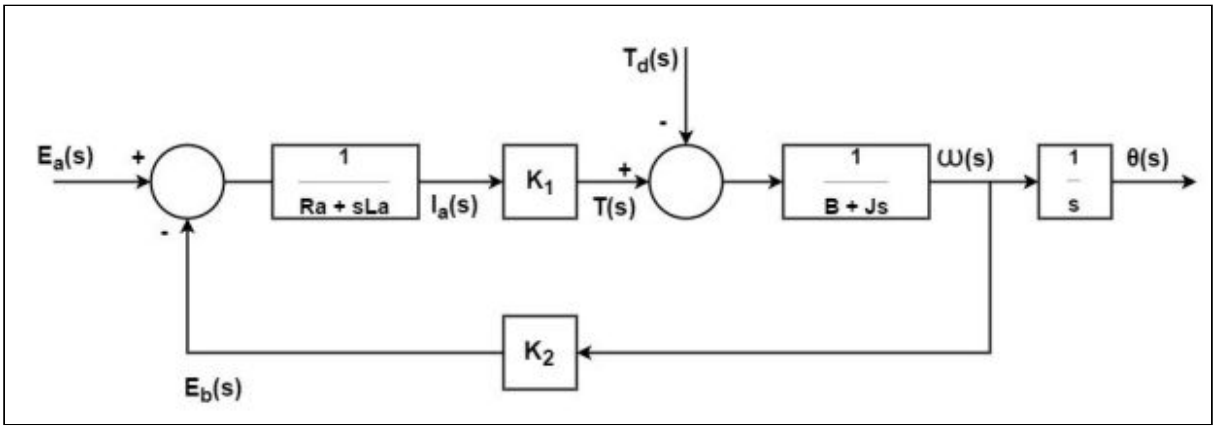
Integrando ambos os lados:

$$\int \omega(t)dt = \theta(t)$$

Aplicando a transformada de Laplace, para passar para o domínio da frequência, obtemos a seguinte equação:

$$\theta(s) = \frac{1}{s}\omega(s)$$

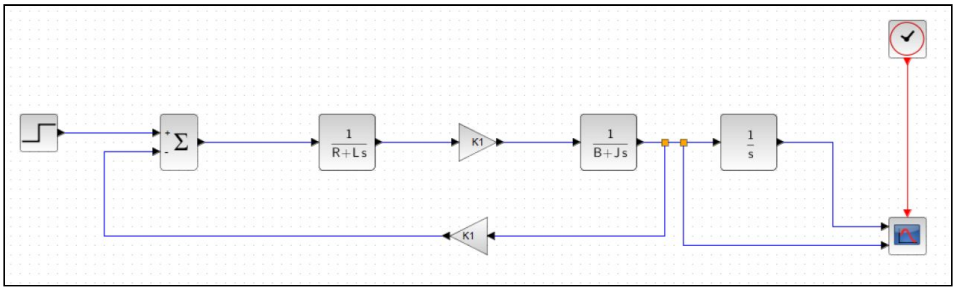
Portanto, para desenhar o diagrama de blocos de controle da velocidade e posição do motor, basta adicionar um bloco $\frac{1}{s}$ à malha ilustrada na figura a seguir.



O valor $\theta(s)$ é a posição angular, dada em radianos. Sendo assim, deve-se dividir esse valor por 2π (uma volta em radianos), já que a razão $\frac{\theta(s)}{2\pi}$ representa a fração de uma volta que foi dada.

2.3 Análise de sensibilidade dos parâmetros da função de transferência

De modo a entender melhor a influência de cada um dos parâmetros da função na velocidade final do motor, foi realizada uma fatoração da função de transferência em blocos como explicado acima e foi montado o circuito mostrado na figura abaixo no software de simulação do SciLab para entender como cada parâmetro da função afeta a saída do mesmo.



O circuito acima pode ser dividido em 2 partes, sendo elas: a **componente elétrica** do motor (representada pelos blocos que contém as variáveis R, L e K1) e a **componente mecânica** (representada pelo bloco que contém as variáveis B e J). Como os valores de B e J são afetados por variáveis externas, como perturbações no sistema e variações na carga, sabemos que estes valores no geral são muito maiores que os valores de R e L, fazendo com que a saída do sistema seja controlada apenas pela componente mecânica do motor. Podemos identificar isto reescrevendo a função de transferência em relação às constantes de tempo elétrica e mecânica da seguinte forma:

$$G_1(s) = \frac{K_t}{R_a B(1+\tau_e s)(1+\tau_m s) + K_t K_e}$$

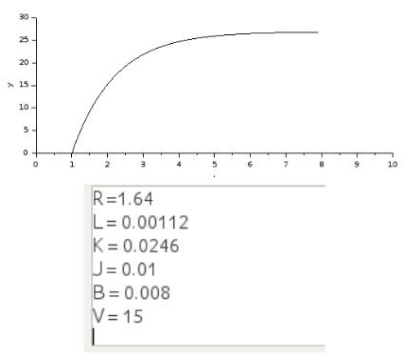
,onde $\tau_e = L_a / R_a$ é a constante de tempo elétrica e $\tau_m = J / B$ é a constante de tempo mecânica. Se temos valores muito baixos para a indutância, a constante de tempo elétrica acaba sendo desprezível.

Para realizar a análise de sensibilidade dos parâmetros da função de transferência utilizamos as informações de um motor real da fabricante Maxon, como foi sugerido pelo professor em aula. O motor utilizado foi o mostrado na imagem abaixo e possui os seguintes parâmetros: R = 1.64 Ω, L = 1.12 mH e K = 24.6 mNm/A.

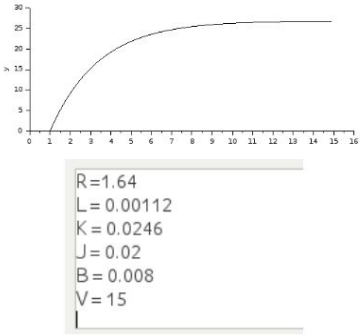
Com os valores de R, L e K configurados, foram feitas diversas simulações variando os valores de B e J para identificar a influência de cada um dos parâmetros na velocidade final do motor. Inicialmente valores bem baixos foram utilizados para identificar a velocidade máxima que o motor atingiria e gradualmente aumentamos os valores de B e J (individualmente em cada teste) e verificamos como a curva de velocidade se comportava com cada um desses valores.

Testes com a variável de inércia (J)

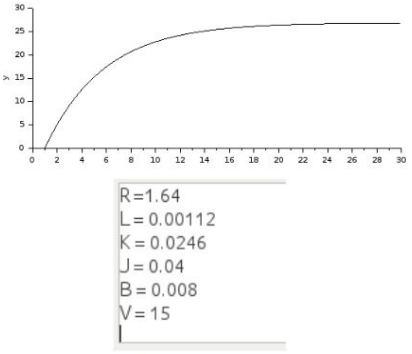
A figura abaixo mostra os resultados da análise descrita acima para a variável J. Nestes testes todas as outras variáveis tiveram seus valores fixados e apenas a variável J teve seu valor modificado, sempre dobrando seu valor. A partir destes gráficos é possível notar que a variável de inércia afeta no tempo de resposta do sistema de maneira linear e diretamente proporcional ao valor de J, ou seja, se dobrarmos o valor de J, o tempo de resposta do sistema também dobra.



TR próximo a 6,75s



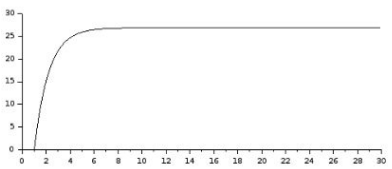
TR próximo a 12,5s



TR próximo a 24s

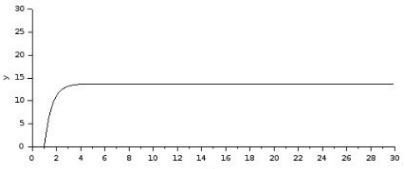
Testes com a variável de atrito (B)

Testes similares foram feitos para a variável B e neste caso podemos identificar que quando a variável B tem seu valor dobrado, o valor final da velocidade em regime cai de maneira proporcional e linear, porém inversamente proporcional, ou seja, se o valor de B dobra o valor final da velocidade em regime cai pela metade.



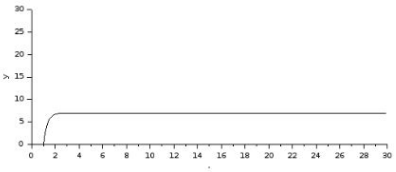
R=1.64
L=0.00112
K=0.0246
J=0.01
B=0.008
V=15

Regime em 27 rpm



R=1.64
L=0.00112
K=0.0246
J=0.01
B=0.016
V=15

Regime em 14 rpm



R=1.64
L=0.00112
K=0.0246
J=0.01
B=0.032
V=15

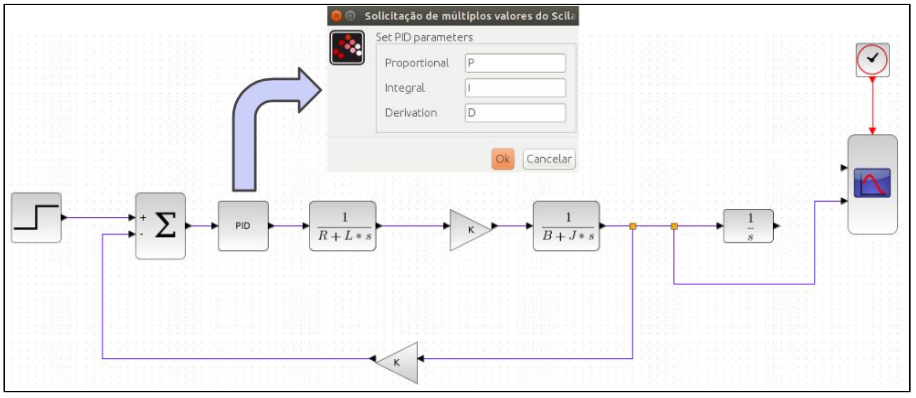
Regime em 7 rpm

Por fim, podemos notar que a variável J aumenta o tempo de resposta do motor, conforme seu valor aumenta e a variável B diminui a velocidade final do motor conforme seu valor aumenta. Como estas duas variáveis estão relacionadas principalmente a carga do sistema (pois os valores externos são muito maiores que os internos do próprio motor), é muito importante entender como elas afetam a velocidade do motor para entender melhor como controlar o sistema com diversas cargas diferentes.

3. Simulação do PID e análise das variáveis Kp, Ki e Kd

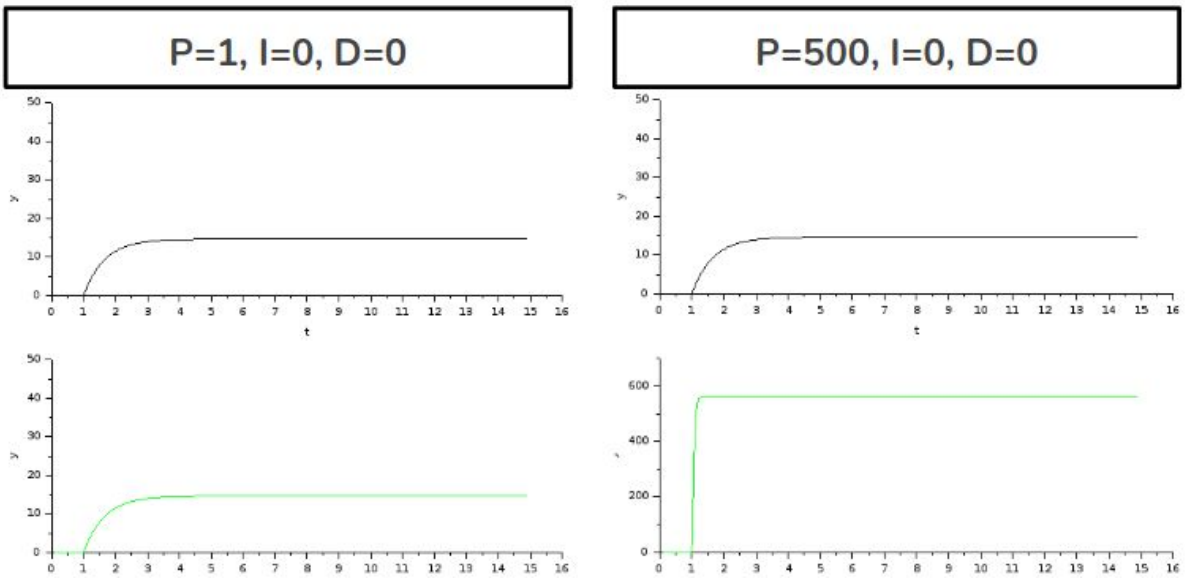
3.1 Inclusão do PID no diagrama de blocos

Um bloco PID pode ser adicionado ao sistema para ajustar a velocidade para o set point, corrigindo os erros gerados pela carga (variáveis J e B). Para simular o comportamento de um sistema de motor DC controlado via PID foi adicionado o bloco de PID do SciLab no circuito. Com isso foi possível verificar a influência das variáveis Kp, Ki e Kd do PID na saída do sistema. As análises foram direcionadas a entender a relação destas 3 variáveis com os seguintes parâmetros de desempenho: **erro em regime**, **tempo de resposta**, **tempo de acomodação** e **overshoot**. As definições destas variáveis estão melhor descritas na seção 5. O bloco de PID adicionado no circuito do motor é mostrado na figura abaixo, juntamente com a janela de definição das variáveis Kp, Ki e Kd.



Para entender melhor como cada variável do PID afeta a saída do sistema, foram feitos experimentos similares aos da análise de sensibilidade dos parâmetros da função de transferência. Como o sistema de um motor real tem a componente mecânica muito mais influente que a componente elétrica, a influência da componente elétrica é muito baixa, fazendo com que a curva da saída do sistema se comporte como a de um **sistema de primeira ordem,** ou seja, não ocorre overshoot. Para que fosse possível avaliar a influência nos 4 parâmetros de desempenho citados acima, precisamos modificar os valores de R e L para que estes fiquem próximos, respectivamente, dos valores de B e J. Com isso a saída do

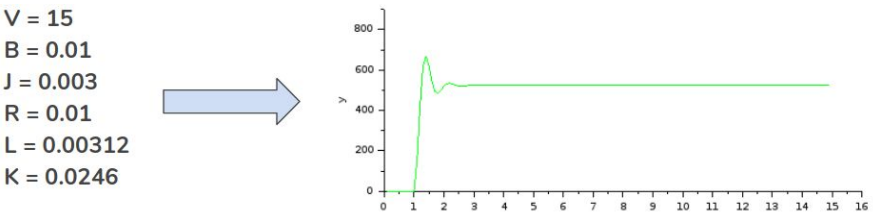
sistema se comportará como a de um sistema de segunda ordem e poderemos entender melhor as influências causada pelo PID. A figura abaixo mostra o primeiro experimento, ainda utilizando as variáveis de um motor real, e dessa forma sem overshoot.



Nesta figura podemos verificar que nesta situação, em teoria, apenas o Kp é necessário para zerar o erro em regime e o tempo de subida, porém seria inviável atingir valores tão altos de Kp como necessário, porém esta análise já mostrou a influência de Kp em 2 parâmetros de desempenho.

3.2 Metodologia e simulação

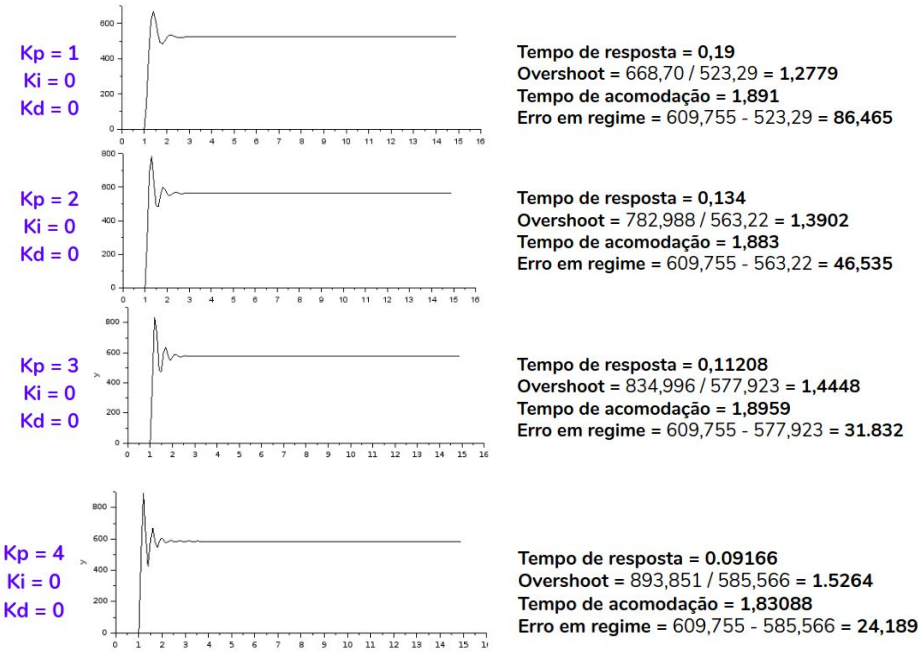
Para analisar as variáveis do PID utilizamos os seguintes parâmetros para obter uma saída na forma de um sistema de segunda ordem. A figura abaixo mostra as variáveis utilizadas e a curva gerada pelo circuito sem o PID.



Utilizando a curva mostrada acima podemos realizar os experimentos variando os valores de Kp, Ki e Kd para verificar a influência de cada um na saída do sistema. Primeiramente iniciaremos com a análise da variável Kp, onde fixaremos os valores de Ki e Kd em 0 iremos analisar a saída da velocidade com Kp nos valores 1, 2, 3 e 4 e serão analisados os parâmetros de desempenho que estão melhor explicados na seção 6. Após isto uma análise similar será realizada com a variável Ki, onde fixaremos o valor de Kp em 1 e o valor de Kd em 0 e analisaremos a saída da velocidade em função dos parâmetros de desempenho novamente. Por fim será feito o mesmo com a variável Kd, porém fixando Kp em 3 e Ki em 0.5. Ao final de todo este processo será montada uma tabela que mostra as relações destas variáveis com saída da função de transferência.

3.3 Resultados PID

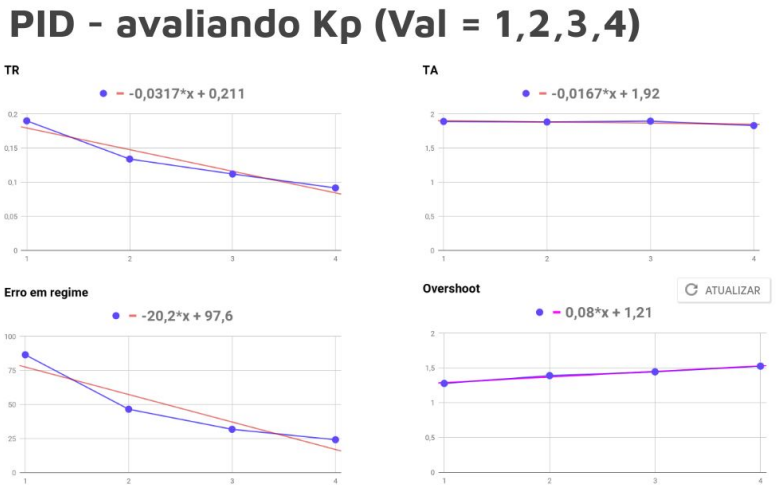
As figuras abaixo mostram as análises feitas relacionadas a variável Kp.

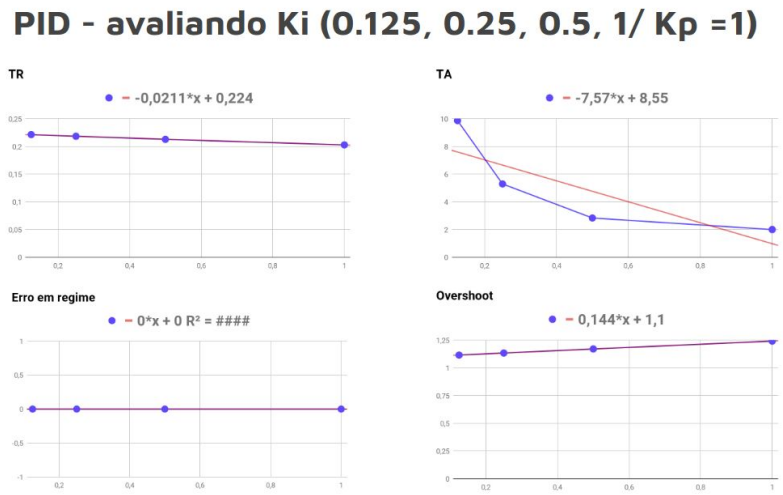


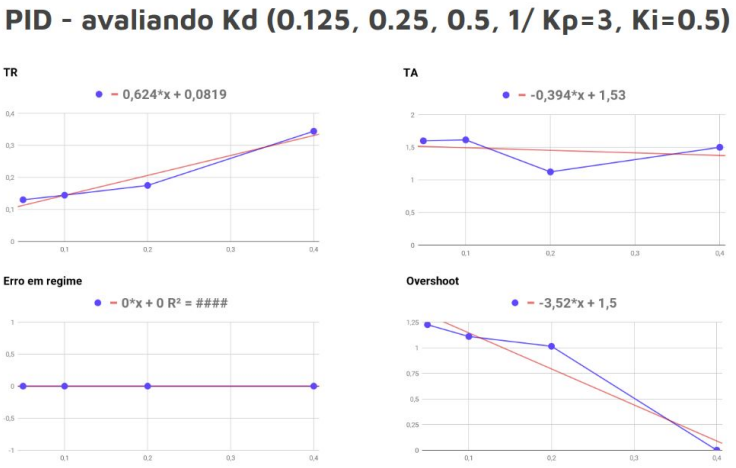
Análise do Kp.

Com os valores obtidos da análise foram gerados gráficos que representam os pontos de dados obtidos e traçadas linhas de tendência para identificar a dependência das relações que fossem lineares. Em casos de dependências exponenciais, os valores obtidas variam muito para conseguirmos definir uma

função concreta para o mesmo. Nos gráficos a seguir a curva em azul representa os pontos de dados obtidos e a reta vermelha representa a linha de tendência linear destes pontos. As figuras abaixo mostram, respectivamente, os gráficos resultantes da análise de Kp, Ki e Kd.







Ao final de todas as simulações, a seguinte tabela foi criada mapeando as relações de cada variável do PID com cada parâmetro de desempenho com relação a qual a influência (não numérica causada na saída).

Variável	Tempo de resposta	Overshoot	Tempo de acom.	Erro em regime
Kp	Diminui	Aumenta	Pouca mudança	Diminui
Ki	Diminui	Aumenta	Aumenta	Diminui
Kd	Pouca mudança	Diminui	Diminui	Não afeta

Além disso a seguinte tabela foi criada de modo a tentar mostrar a influência numérica encontrada ou de maneira mais precisa, quando a mesma não pode ser representada numericamente.

Variável	Tempo de resposta	Overshoot	Tempo de acom.	Erro em regime
Kp	**Decaimento exponencial**	**0,0878**	**Sem variação significante *** **	**Decaimento exponencial**
Ki	**-0,0211**	**0,144**	**Crescimento exponencial***	**Elimina**

Kd	**0,624 *** **	**Decaimento exponencial****	Diminui ****	**Não afeta**

(*): Nos testes apresentou decaimento, porém com valores mais altos de Ki começa a gerar um crescimento exponencial do tempo de acomodação.
(**): É capaz de eliminar o overshoot em alguns casos.
(***): Não há padrão definido ou pouca mudança - [http://faculty.mercer.edu/jenkins_he/documents/TuningforPIDControllers.pdf]
(****): Não foi possível realizar um estudo conclusivo do valor numérico, porém é esperado que o mesmo diminua o tempo de acomodação exponencialmente.

4. Projeto mecânico

4.1 Análise

O projeto possui um eixo vertical com uma barra roscada e uma base acoplada em uma porca encaixada na barra.

A barra rotaciona no eixo Z através do giro do motor que está em cima dela. Cada giro do motor, que contém uma resolução de 400 pontos em seu encoder, faz com que a barra movimente a base verticalmente em 4mm por giro do motor.

Na parte traseira está o circuito que faz todo o controle do projeto.

4.2 Sugestão de Melhorias

Após análise da estrutura atual do projeto, pôde-se observar que diversas partes foram feitas de uma forma simplificada e/ou improvisada ao construir o conjunto. Isso deve-se, principalmente, aos fatos de um possível baixo fundo de investimento mas também por conta da demanda que temos no laboratório não ser tão grande. Entretanto, se é desejado um alto desempenho do conjunto por inteiro, não devemos pensar apenas no desenvolvimento de software e eletrônico, mas também nos componentes mecânicos. Por isso essa etapa visa avaliar os principais componentes as sugerir algumas mudanças que poderiam influenciar em um melhor desempenho do projeto final.

4.2.1 Acoplamento de Eixo

Esse componente tem como objetivo unir e prover interação entre dois sistemas mecânicos. No caso do nosso projeto, esses sistemas são o Motor (que proverá o movimento rotacional) e o mancal/guia linear (item 3.2.2).

O principal motivo de adicionar um acoplamento de eixo de melhor qualidade é para evitar problemas, principalmente em relação ao motor pois, além desse componente sincronizar com mais precisão os dois eixos e reduzir as vibrações, ele ainda absorve choques e previne que o motor utilize força demais, que em alguns casos pode gerar danos ao próprio.

Imagem 3.1: Acoplamento de eixo flexível

Atualmente utilizamos um simples tubo plástico que apenas funciona como um conector entre os dois sistemas.

4.2.2 Mancal/Guia Linear

O mancal (ou guia) linear é uma peça essencial ao nosso projeto. Ele é o grande responsável pela movimentação da plataforma no eixo Z, mantendo a estabilidade do conjunto. Ele é um componente que, se feito com qualidade, garante uma movimentação com erro mínimo muito por conta do coeficiente de atrito baixo no conjunto.

Entretanto, o mancal não atua sozinho na movimentação de um sistema, ele apenas garante a estabilidade e eficiência desejada, enquanto o acionamento é feito por um outro componente, como por exemplo um fuso (item 3.2.3).

Sugestão: utilizar um mancal linear com trilho.

mancal linear com trilho

4.2.3 Fuso de Esferas

O fuso é um dos componentes mecânicos essenciais para o funcionamento do projeto. Ele é o grande responsável por transformar o movimento rotacional provido pelo motor em movimento linear (no caso, o eixo Z). Ele é capaz de suportar altas cargas enquanto apresenta baixo torque de partida (evitando "trancos" na movimentação) e baixa necessidade de manutenção.

Fuso de esferas

4.2.4 Base e Plataforma

Aqui temos uma possível sugestão de mudança que é, de certa forma, mais simples e barata, entretanto também impacta diretamente tanto na eficiência quanto na segurança do projeto.

Primeiramente, ao analisar a base inferior, notamos que apenas uma pequena área está firme à placa de madeira utilizada, o que resulta em uma certa instabilidade no conjunto como um todo, podendo comprometer até a segurança da estrutura ao colocar um peso muito grande em teste. A sugestão aqui seria talvez utilizar uma área maior de fixação ou até mesmo uma estrutura com hastes que dessem maior estabilidade.

Sobre a plataforma, vemos que não há absolutamente nenhum mecanismo que "prende" o objeto à plataforma, podendo causar certa instabilidade, principalmente durante o funcionamento. A sugestão aqui também é simples: colocar algum tipo de gradil ou apenas algum ponto de fixação no centro da plataforma (como um parafuso que mantivesse o objeto a ser testado na posição desejada durante todo o trajeto).

4.2.5 Análise de Viabilidade

Após todas as sugestões, precisamos levar em conta agora alguns fatores para definir a viabilidade dessas dentro do nosso projeto. Algumas principais são: valor da peça e capacidade de aplicação.

Analisando cada uma das sugestões, pode-se concluir logo em primeiro momento que o acoplamento de eixo é uma peça essencial para a melhora e segurança do nosso projeto. A partir de algumas pesquisas online foi possível encontrar diversos modelos na faixa de 14 a 50 reais, o que é algo possível para nós.

Outra peça que elevaria o nível do projeto a um patamar de qualidade muito superior é o fuso de esferas, entretanto, isso exigiria um trabalho muito maior na instalação, visto que alguns modelos encontrados na internet são compostos pelo conjunto inteiro (parafuso e até mancal), o que complicaria o nosso trabalho. Outro contraponto é o valor, alguns desses que são o conjunto completo custam entre 200 e 300 reais.

Finalmente, sobre as bases, em primeiro momento é possível aplicar um revestimento de borracha à base do elevador, diminuindo em muito a possibilidade de movimentação do objeto durante o movimento, além de ser a opção mais barata e simples.

Já na base inferior, a indicação inicial seria aumentar a área de contato pregada à superfície de madeira. Não é também a solução ideal, mas traz já um avanço simples e barato em relação ao conjunto atual.

5. Projeto eletrônico

5.1 Componentes eletrônicos

Na parte de sensores para identificação base de posição do eixo estão três sensores indutivos, que identificam quando há algum componente ferromagnético próximo. Esses sensores enviam um sinal para um acoplador óptico, que é responsável por fechar ou abrir o circuito de uma tensão de 5V para um pino do arduino.

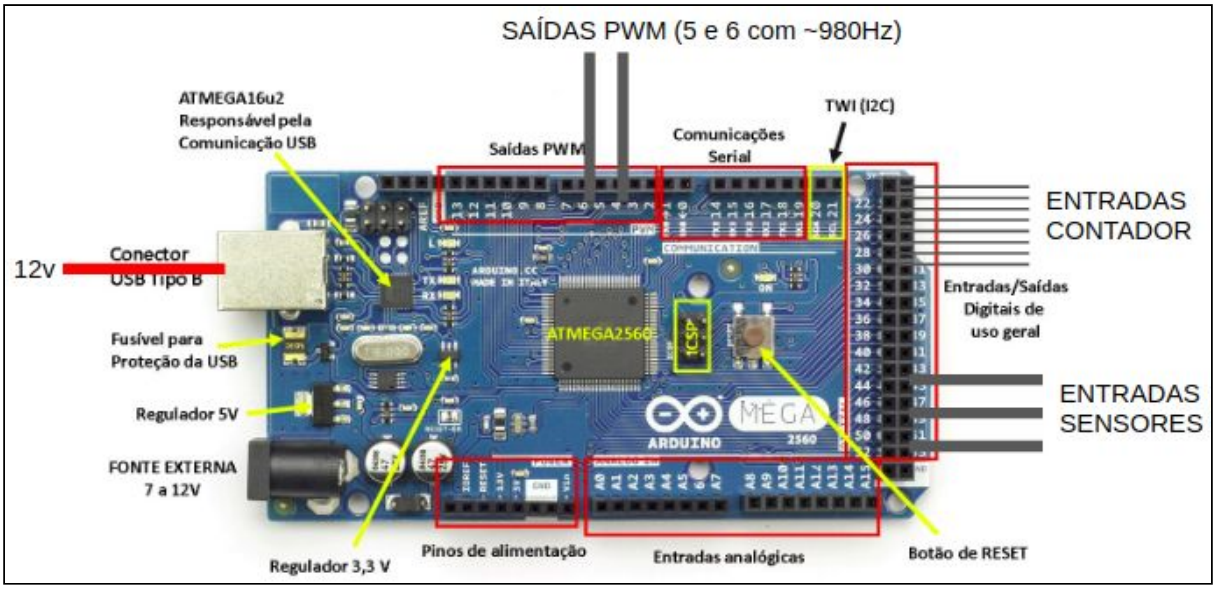
O arduino é responsável pelo controle da rotação do motor e receber o feedback dessa rotação. Ele envia para o driver do motor quanto e em que direção o motor gira, o driver então identifica o que o arduino está querendo e passa a quantidade necessária para o motor da tensão que está recebendo de uma fonte.

O motor possui um encoder que envia sua informação para um circuito contador, que então envia o valor final para o arduino.

5.2 Arduino

O Arduino Mega 2560 deve receber uma alimentação de 12V e seus pinos de entrada e saída chegam até 5V.

Para o projeto, são necessários como entrada 8 pinos digitais do contador e mais 3 pinos digitais do acoplador óptico. Como saída requer os pinos 5 e 6 do PWM, que possuem uma frequência de aproximadamente 980 Hz, o dobro de frequência das demais saídas PWM.



5.3 Motor

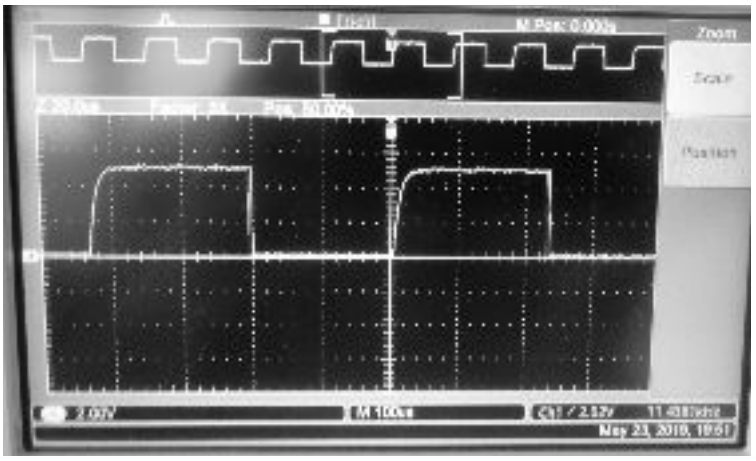
O motor de corrente contínua utilizado é de 30V e 1.1A, que possui um encoder de 400 ranhuras com uma frequência de até 12 khz.

Sua alimentação é feita pelo driver do motor, para determinar sua velocidade e sentido de rotação.

5.4 Circuito de contagem do Encoder

5.4.1 Características do sinal de saída do encoder

A partir do encoder, o sinal de saída obtido foi uma onda quadrada com leve ruído afetando principalmente a borda de subida, conforme a figura. Considerou-se que não haveria necessidade de retificação do sinal pois ele não interferiria na contagem. A frequência máxima obtida para o sinal foi de 12KHz correspondente a uma tensão máxima de 30V.



5.4.2 Estratégia de contagem

Através da observação do número de pulsos e do sentido de movimento do motor podemos inferir a o deslocamento relativo. A proporção de pulsos e deslocamento obtida foi de 80:1 mm. Os valores foram obtidos da análise do passo do parafuso acoplado ao motor.

Com os valores obtidos através do contador podemos considerar uma janela de leitura de 15 a 20 ms considerando que temos 400 ranhuras no disco do encoder girando a 12KHz:

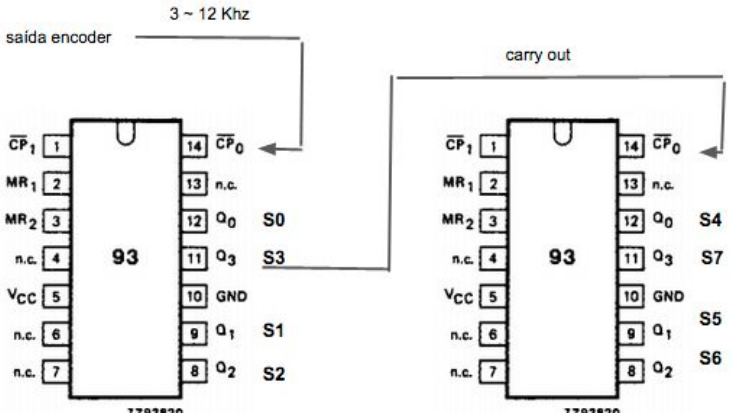
com 15ms: $\frac{12KHz}{100\ medições}$ *= 120 pulsos*

com 20ms: $\frac{12KHz}{67\ medições}$ *= 182 pulsos*

Ambos valores são inferiores a 255 pulsos que corresponde ao valor máximo de overflow do contador. A contagem total é obtida através da análise de duas janelas consecutivas: se o módulo da contagem aumenta o eixo se move em sentido positivo. Caso contrário, se move em sentido negativo. (ver 8.2)

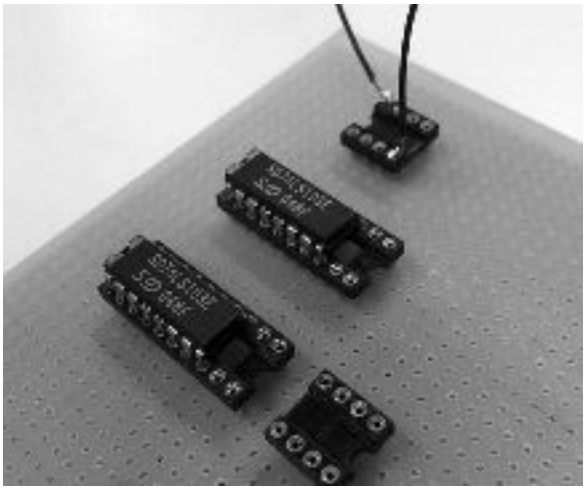
5.4.3 Circuito contador

Foram utilizados dois contadores de 4 bits 74LS192/193 em série para gerar um contador assíncrono de 8 bits (0-255).



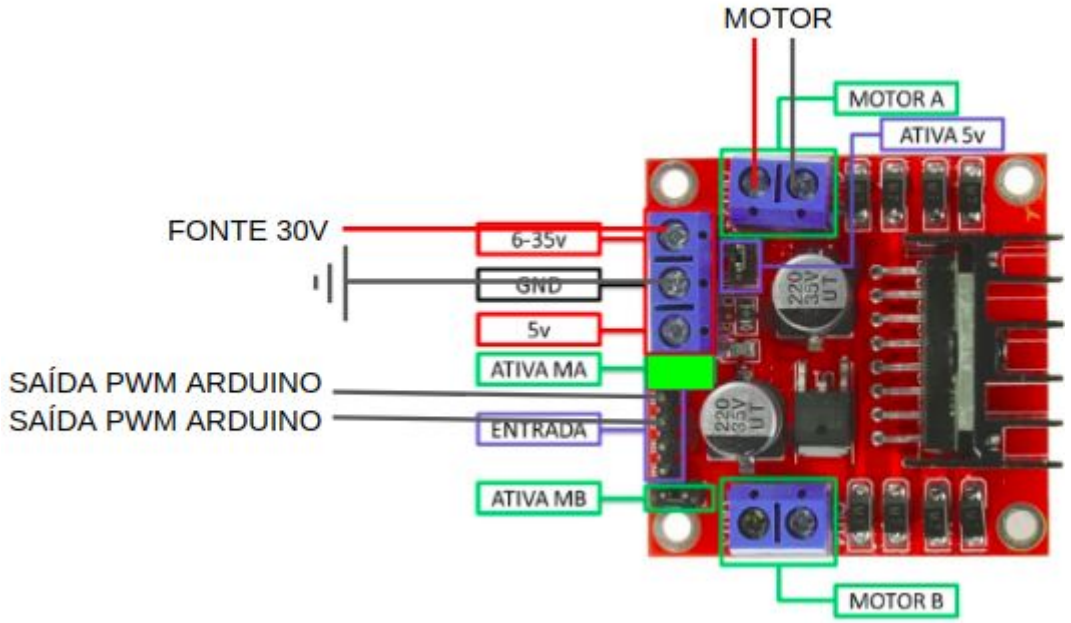
Nessa configuração, o bit S_3 é o carry out da entrada do contador seguinte, transferindo a frequência de entrada CP_0. Com essa cascata, as saídas paralelas S_0 a S_7 correspondem aos dígitos binários do contador.

Utilizando wire-wrapping obteve-se um circuito da figura. Observou-se que amplitudes elevadas de onda produziam ruídos significativos no contador.



5.5 Driver de potência do Motor

O driver utilizado é o L298N, que contém uma ponte H de 0 a 5V, levando de 0 a 35V em até dois motores. Desse modo, é possível controlar a potência e a direção de giro do motor através de uma porta PWM do arduino.



5.6 Sensor Indutivo

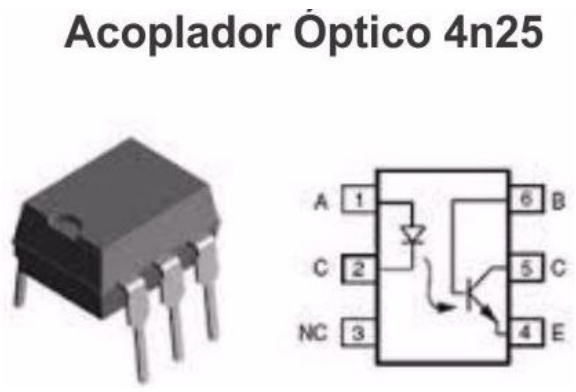
Os sensores indutivos de 18mm (Schmersal ifl-5-18-10n) estão distribuídos pelo eixo da forma: um “marco zero” do sistema de referências assim como nas extremidades do mesmo. O sensor recebe uma alimentação de 24v e é normalmente fechado. Ao ter algum material ferromagnético próximo, ele abre o circuito e não envia nada no fio de sinal.

5.7 Acoplador Óptico

O acoplador óptico comuta dois circuitos de forma isolado, fazendo com que o sensor indutivo consiga informar o arduino, que recebe 5v, do seu sinal de 24v.

Os pinos, de 1 a 6, são ligados respectivamente pelos seguintes componentes: sinal do sensor, terra do sensor com um resistor de 10 kohm, não

conectado, terra do arduino, coletor do arduino com um resistor de 1 kohm, fonte de 5v.



5.8 Componentes secundários

Outros componentes que podem ser inseridos no projeto são botões e LEDs.

Botões podem fazer com que o usuário envie comandos básicos ao projeto, como desligar a energia e voltar ao ponto zero.

LEDs funcionam como feedback para informar o usuário, por exemplo, de que o elevador chegou no valor pedido ou no ponto zero.

5.9 Alimentação dos componentes

É preciso alimentar o driver do motor com 30v, o sensor indutivo com 24v, o arduino com 12v e os acopladores ópticos e o contador com 5v.

5.10 Arquitetura do projeto eletrônico

A parte eletrônica contém diversos componentes que precisam se comunicar, portanto foram inicialmente testados e então definida suas ligações e dependências. Estes componentes foram listados e então criado um diagrama de blocos de modo macro com a relação entre eles.

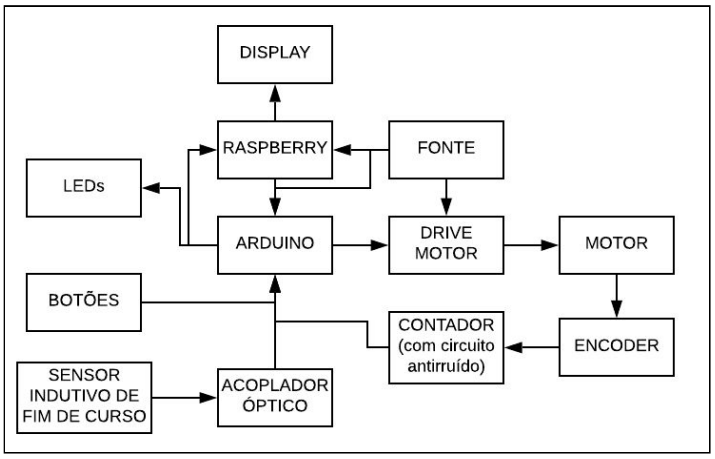


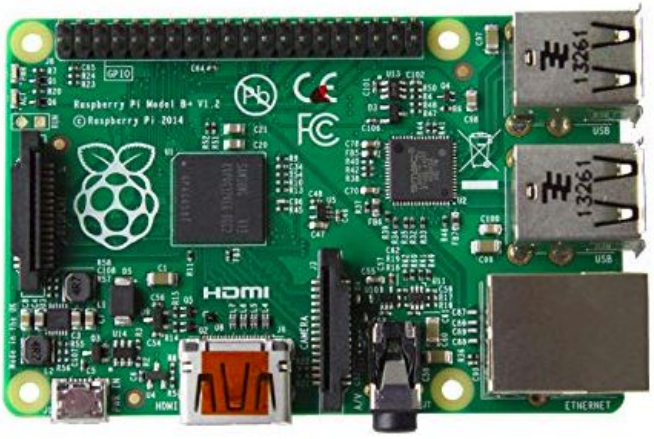
Diagrama macro da relação entre os componentes

O arduino, junto ao driver do motor e ao acoplador óptico, estão em uma placa de circuito com ligações e pinos soldados, unificando em um só lugar estes componentes e a comunicação em si.

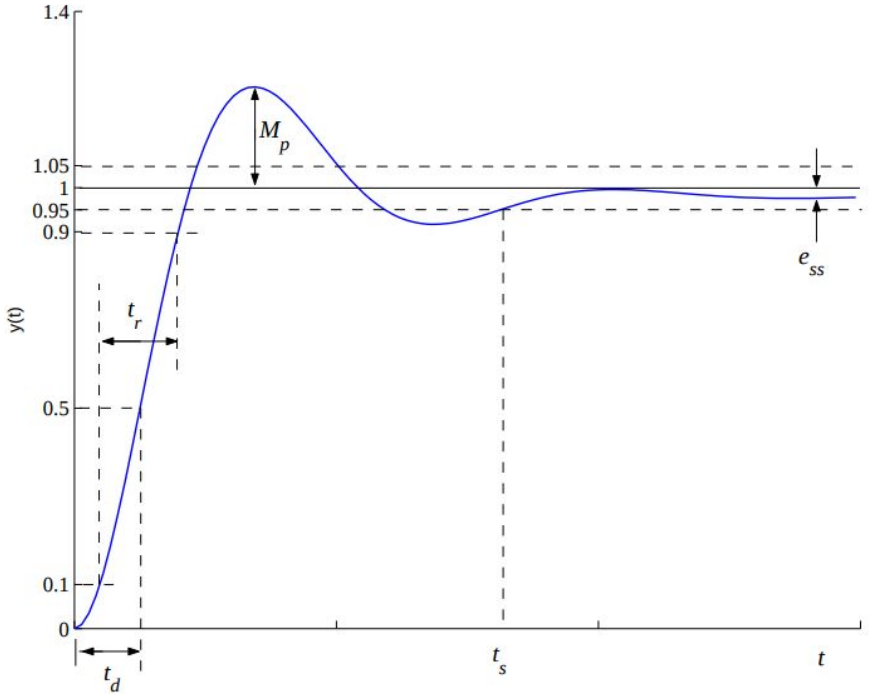
5.11 Evolução do projeto: Raspberry

O poder de processamento de um arduino é baixo e não paralelo. Com uma placa mais potente e robusta daria mais liberdade ao projeto e mais integrações para oferecer ao usuário uma interface.

Uma evolução é a inserção de uma Raspberry Pi, que gerenciará todo controle do projeto e responsável por uma interface.



6. Definição de parâmetros de desempenho



A forma de medir e validar a correta execução do sistema utiliza como base os parâmetros de desempenho do PID. Conforme pode ser visto na figura acima, existem 5 parâmetros, descritos abaixo:

- **Tempo de subida "tr"** (*rise time*): é o tempo necessário para a sinal de saída variar de 10% a 90% (sistemas sobre-amortecidos) ou de 0% a 100% (sistemas subamortecidos) do valor final;
- **Tempo de acomodação "ta"** (ou *settling time* **"ts"**): é o tempo gasto para o sinal acomodar na faixa de ±5% do valor final;
- **Sobre-sinal máximo percentual "Mp"** (*overshoot*): diferença entre o valor máximo de pico atingido e o valor final em percentual do valor final;
- **Tempo de atraso "td"** (*delay time*): é o tempo para o sinal alcançar 50% do valor final;
- **Erro em regime "ess":** é a diferença entre o sinal de saída e o sinal de referência, em regime permanente.

7. Definição e execução do conjunto de testes

7.1 Mecânica

Nessa etapa, o objetivo é testar alguns dos principais componentes que participam da movimentação e funcionamento do projeto: a plataforma e o motor (referentes à capacidade do conjunto); e o sensor (referente à detecção das posições inicial e final).

Os testes serão totalmente práticos, visando menor esforço mas também tomando cuidado para não forçar o projeto além de sua capacidade.

No caso do motor e da plataforma, faremos testes com diversos pesos e objetos diferentes com o objetivo de compreender a faixa de valores para a carga que podemos utilizar em segurança.

Já para o sensor, devemos fazer o teste de distância de detecção e material a ser utilizado. Primeiramente, iremos escolher um material a ser detectado pelo sensor ferromagnético; em seguida faremos testes da distância segura que podemos posicionar o sensor em relação ao material selecionado, para enfim fixar no sistema todos os componentes garantindo sua máxima funcionalidade.

7.2 Eletrônica

No Arduino Mega é preciso realizar testes de leitura e envio de sinais digitais e PWM.

Para o driver do motor, deve-se determinar se suas ligações ou componentes internos estão funcionando, portanto são feitos testes do valor de saída de 0 a 35V e verificação do correto funcionamento do PWM, e da ponte H através do valor de entrada em sentido horário e anti-horário.

O Acoplador óptico 4N25 deve prover uma saída correta ao abrir e fechar o sistema de 0 a 24V para 0 a 5V (leva do sensor pro arduino).

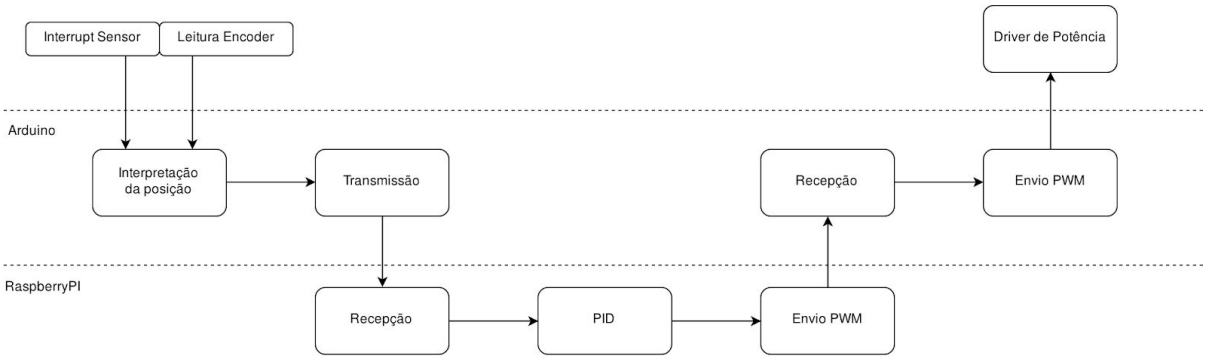
No sensor indutivo é necessário verificar se está funcionando corretamente em modo normalmente fechado, se está detectando um material ferromagnético próximo e respondendo corretamente à detecção, abrindo o envio do sinal.

8. Projeto do Software

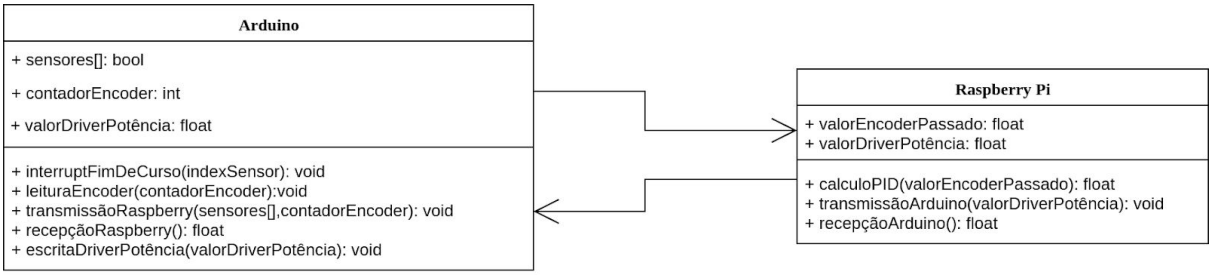
8.1 Arquitetura de Software

A arquitetura de software do sistema foi montada levando em consideração diversos aspectos que ajudarão na integração das partes de diversos grupos por meio da modularização das principais funções do sistema. Dessa forma, é esperado que haja um sistema com baixíssimo acoplamento e alta coesão.

Para representar o sistema como um todo, com suas camadas e interações entre módulos, foi criado o diagrama de blocos abaixo apresentado. O mesmo possui como característica uma definição em altíssimo nível dos principais módulos necessários em cada sistema, assim como os envios de dados entre os mesmos.

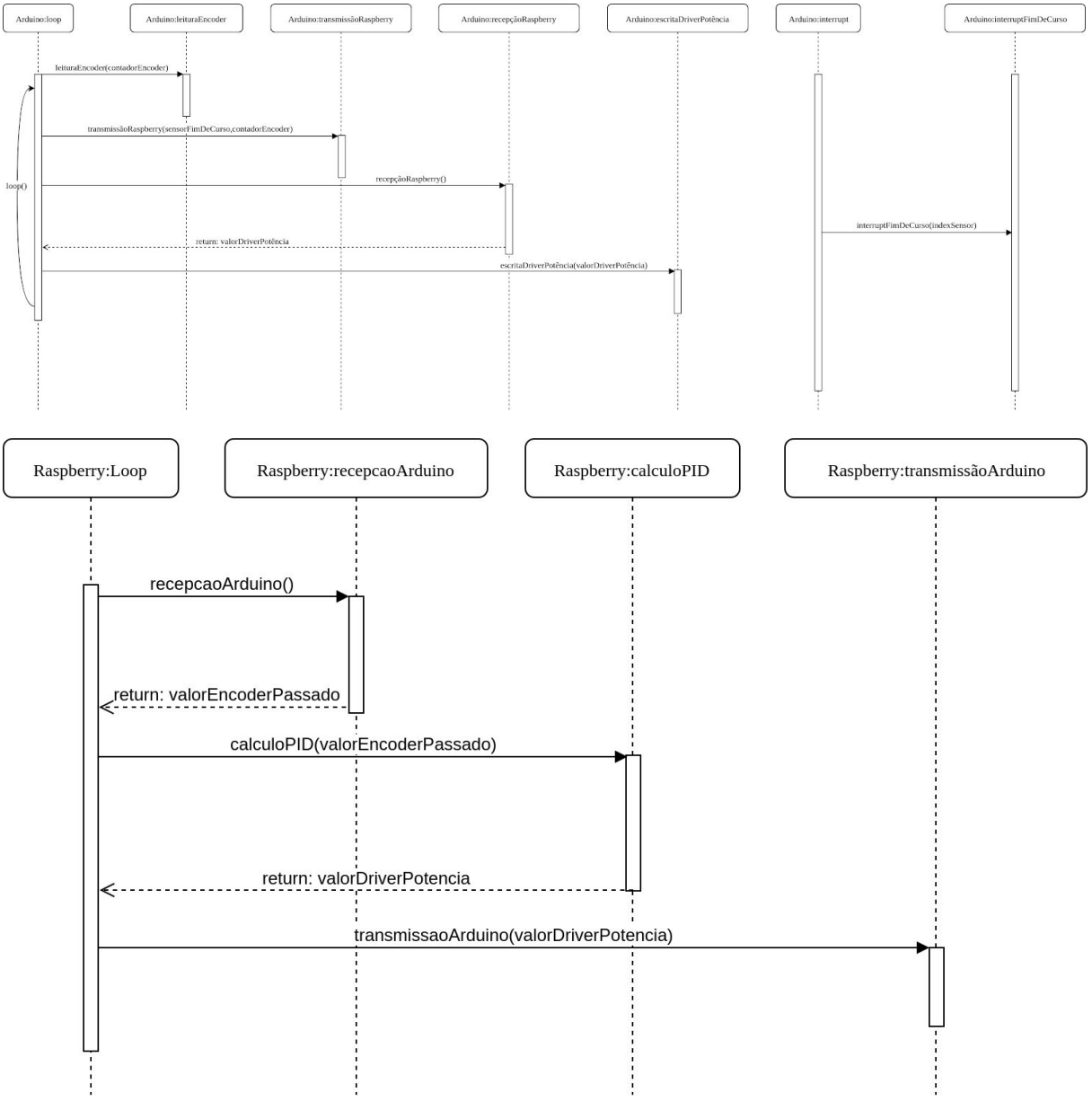


Posteriormente foram criadas representações em mais baixo nível do sistema explicando cada função necessária para a leitura dos sensores e escrita nos atuadores. Como pode ser visto no Diagrama de Classes abaixo, o Arduino será responsável pela leitura dos sensores e envio do sinal PWM ao driver de potência. Já a Raspberry Pi será responsável por realizar o processamento dos valores enviados pelo Arduino e enviar de volta o valor a ser escrito no driver de potência.

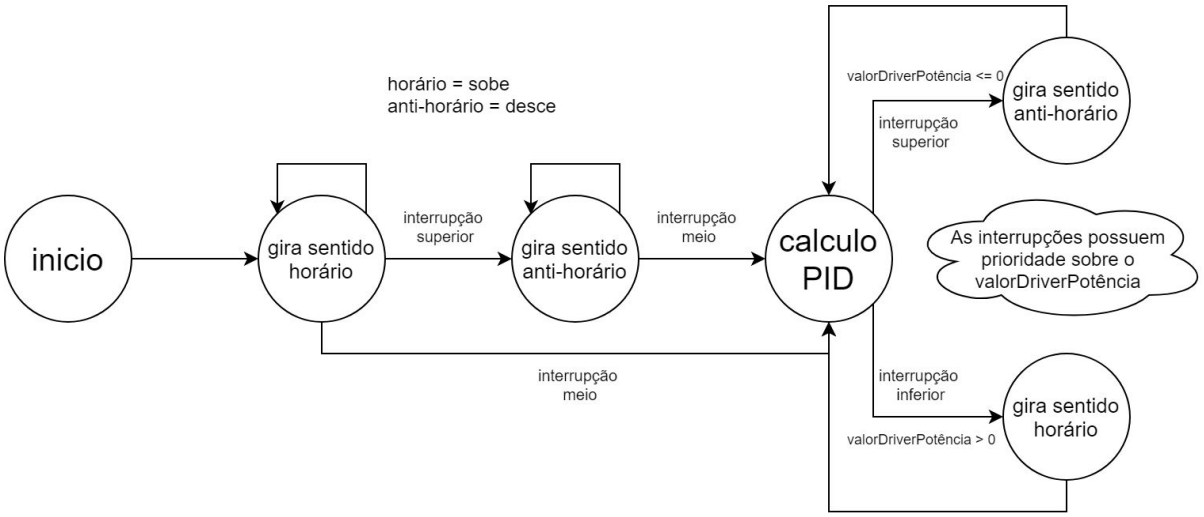


Seguindo a linha de ir especificando em mais baixo nível o sistema, foram criados dois diagramas de classes do Arduino e da Raspberry Pi, onde é possível

ver a sequência de chamadas de funções, assim como as estruturas de loop em ambos os sistemas e os sistema de interrupções assimétricas presentes no Arduino.

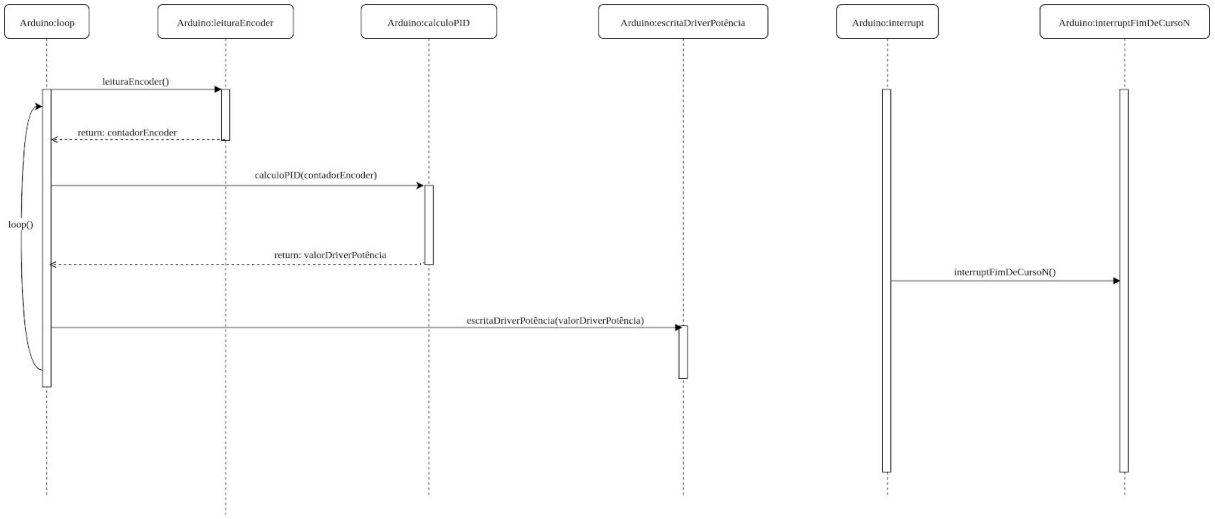


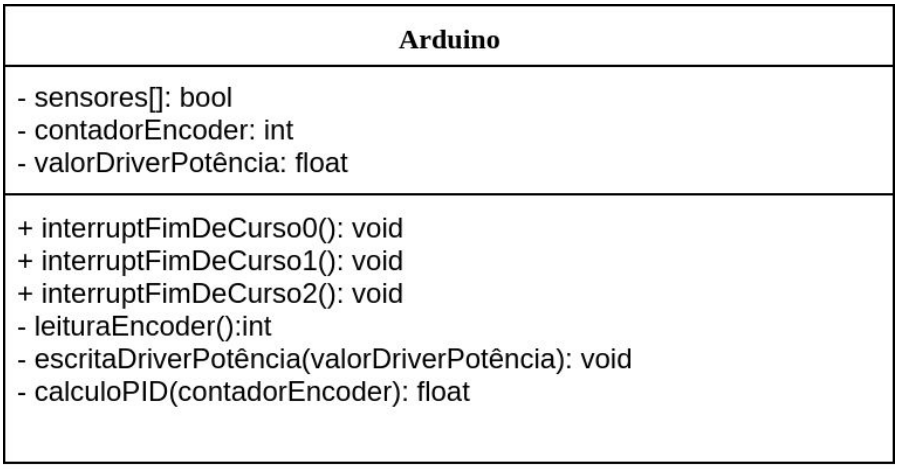
Porém, devido a problemas com o cronograma e a mudanças de escopo do projeto de forma a facilitar a integração, foi realizada a remoção da placa Raspberry Pi do sistema, mantendo apenas o Arduino como central de processamento. De forma a explicar em alto nível a execução do sistema, foi decida a criação de uma máquina de estados em altíssimo nível, de forma a poder auxiliar a pessoas leigas o entendimento do sistema.



Assim, partindo para mais baixo nível nós temos que, a princípio,o software montado no Arduino lê o valor do contador feito no encoder e processa esse dado para obter os valores de posição e velocidade do elevador. Em seguida,ele usa o módulo de PID para efetuar o processamento desses dados a partir da função de transferência obtida previamente. Por final,o Arduino irá transmitir o resultado do PID para o driver de potência através de um PWM,que por sua vez,irá definir a potência o qual o motor irá girar.

Desta forma, foram refeitos os diagramas de sequência e de classe previamente explicados utilizando apenas o Arduino para todas a funções principais.





8.2 Software de integração com o Encoder

O método proposto para a integração com o hardware contador por meio de uma interrupção a que é chamada a cada intervalo fixo de tempo, o qual recebe a contagem em um barramento paralelo do contador, converte o valor em decimal para facilitar tratamento, usa a diferença entre contagens para projetar a posição do robô.

Será utilizada a biblioteca TimerOne para gerar a interrupção a cada intervalo de tempo que chamará uma função principal, que por sua vez chamará três funções menores responsáveis pela leitura do contador, conversão dos valores e atualização dos valores de posição e velocidade.

Para os cálculos dos valores previamente mencionados será realizado um cálculo proporcional utilizando como base a característica do sistema de que a cada giro de 0.9 graus gera uma nova contagem e que 360 graus é referente a 4 mm.

Como serão utilizadas interrupções periódicas, uma das necessidades de tal componente do sistema será a realização de um *profiling* da aplicação, de forma a ver qual o tempo ótimo que garante a integridade dos dados e boa execução do PID. Além disso, a cada interrupção será utilizada uma rotina que fará o tratamento do valor do contador para os casos de *overflow*, também auxiliando na garantia da integridade dos dados.

8.3 Software controlador PID

Para a implementação do controlador PID no Arduino será utilizada uma biblioteca escrita em C responsável por utilizar os valores lidos do contador do encoder para gerar a saída necessária ao motor.

As constantes derivadas das simulações da função de transferência e PID no Simulink e terminadas na simulação previamente realizada serão utilizadas como

valores iniciais para o sistema. Tais constantes serão *tunadas* e melhoradas de forma iterativa, observando o comportamento do sistema e incrementando ou decrementando as constantes.

8.4 Software de controle do motor

Devido às características elétricas do motor DC utilizado, será necessário a utilização de um driver de potência para poder prover a energia necessária para acionar o motor.

O driver de potência têm como característica “transformar” os 5V do Arduino em 30V no motor DC. Dessa forma, será necessário no Arduino um módulo responsável por fazer o cálculo do duty cycle necessário para o PWM de forma a prover ao motor DC a tensão exata para adquirir a velocidade desejada.

Além disso, serão utilizados dois fios para definir o sentido de rotação dos motores, de forma que com base do sinal do valor de saída do PID (positivo ou negativo), serão gerados os sinais correspondentes a tal sentido de rotação.

8.5 Software de recebimento de valores dos sensores

Além do contador do encoder serão utilizados 3 sensores de fim de curso para demarcar o ponto 0 do sistema, assim como os limites superiores e inferiores. De forma a otimizar o tempo de processamento do sistema e garantir a segurança do mesmo, serão utilizadas portas compatíveis com interrupções para a leitura de tais valores.

As interrupções utilizadas serão as de “borda de subida”, de forma que a função de tratamento seja chamada apenas uma vez após acionamento do sensor. Cada sensor terá como função de tratamento uma função própria, na qual uma variável booleana associada ao sensor será setada como *true*.

Desta forma é possível garantir que o tratamento da interrupção será realizada em pouquíssimo tempo, deixando o tratamento das variáveis em si para a função *loop* do Arduino, onde tais interrupções serão utilizadas para garantir a integridade do sistema.

9. Proposta de melhoria de projeto

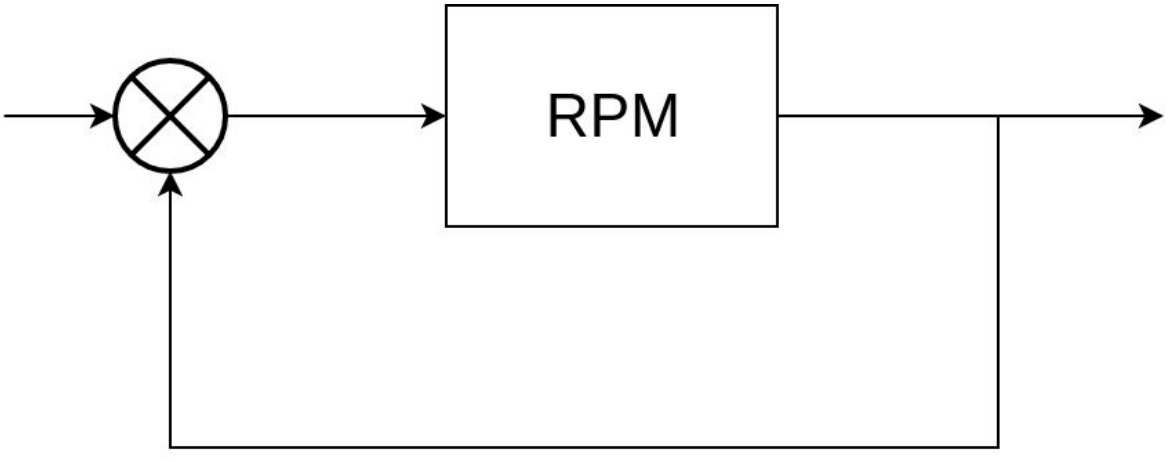
9.1 Aumento de precisão do controle do eixo Z

O projeto previamente proposto servirá de versão inicial para melhorias incrementais da qualidade por meio da adição de características ao sistema. A melhoria almejada para o projeto será dada pela utilização de mais blocos para alcançar o aumento de precisão do controle do eixo.

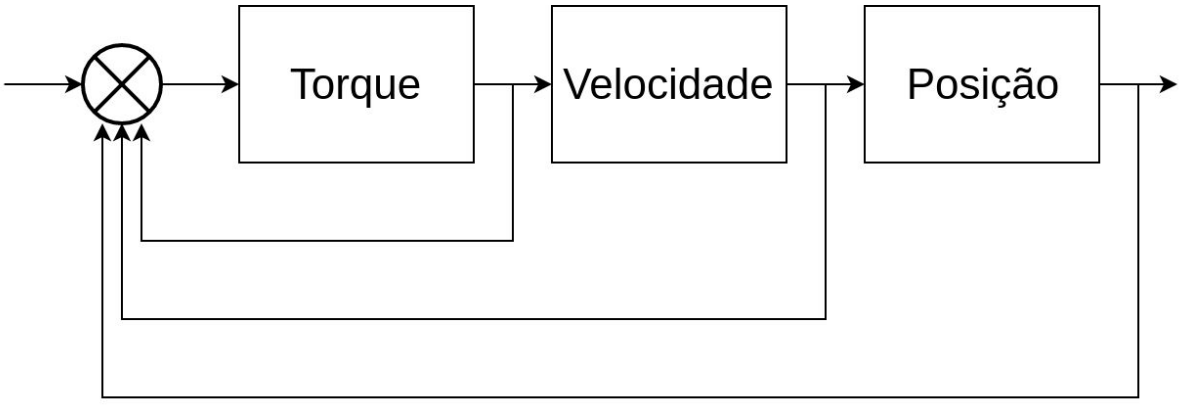
Para isto será necessário definir sensores novos a serem utilizados para obter cada uma das saídas dos blocos com boa resolução a fim de maximizar a precisão da saída obtida. Dentre os sensores possíveis para o sistema, os que serão utilizados serão o Sensor de torque, Sensor de velocidade e Sensor de posição.

Além disso, não serão necessárias mudanças do tipo de algoritmo de controle, sendo utilizado um PID simples para o projeto. Desta forma, provavelmente não será necessária mudança das placas utilizadas, podendo manter a Raspberry Pi para processamento e no máximo um microcontrolador mais avançado para leitura dos sensores.

No projeto atual, a variável de referência para o cálculo do PID é a apenas a posição, calculada através dos valores recebidos pelo contador do sinal do encoder. Para aperfeiçoar o controle, também seria utilizado os dados do torque, da velocidade e da posição, nesta ordem de obtenção dos dados, como forma de ‘pipeline’, o que esperamos que tenda à uma resposta mais rápida do controle.



Na nova proposta, como é utilizado o torque, a velocidade e a posição, estamos aproveitando todas as informações que nosso sistema pode fornecer e assim poder encontrar um controle mais robusto para o robô cartesiano no eixo Z.

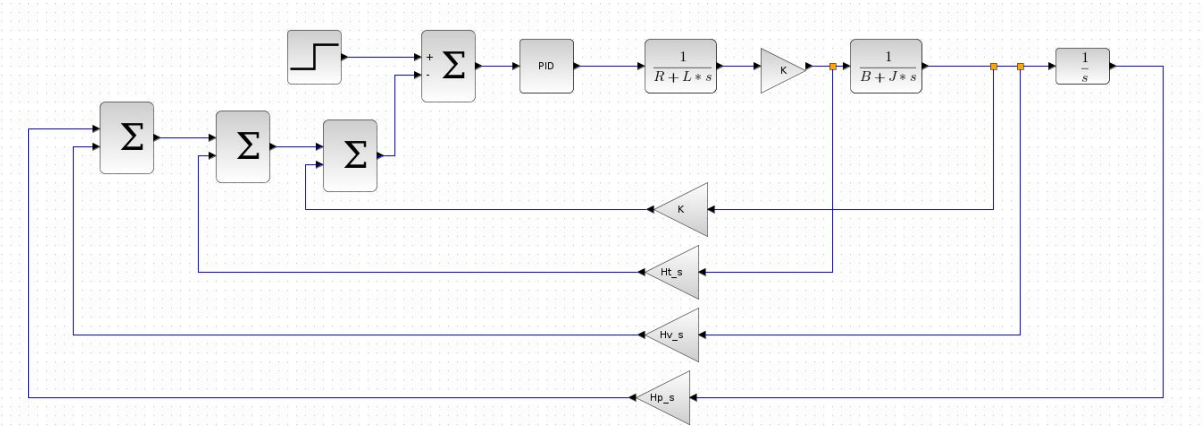


O dado da posição já é obtida no projeto pelo arduino ao receber o valor do contador vindo do encoder. Além disso, com esse mesmo dado recebido é possível obter a velocidade que o motor está girando. Deste modo, não precisamos de nenhum componente extra do que já temos para obter esses dados.

O valor do torque, por outro lado, precisa de um novo sensor para ser obtido. Uma solução escolhida foi a de inserir um sensor de efeito Hall no fio do motor para medir a corrente que está passando por ali. O sensor consegue captar o valor do campo magnético à sua volta e consequentemente a corrente que está passando pelo fio e sua variação. Como a corrente é diretamente proporcional ao torque, já conseguimos uma referência do torque, mesmo que em uma proporção diferente.

Ao final, todos esses dados devem ser normalizados e ponderados, sendo requerido um estudo mais aprofundado e detalhado de efeito e importância de cada um no controle do projeto.

Como há um tempo de delay para o dado chegar na realimentação, pois passa por algum sistema para tratamento antes, foram inseridas funcões multiplicadoras que apresentam esses sistemas de delay. Ao simular esse novo projeto, chamamos essas funções de atraso de Ht(s), Hv(s) e Hp(s), referentes ao sistema do torque, da velocidade e da posição, respectivamente, em relação ao tempo. Ao simular, atribuímos esses valores à 1, considerando essas ligações como resistivas, sem atraso no circuito.



Simulamos primeiramente a realimentação do torque e da velocidade, o que gerou um erro de regime maior somente a realimentação da velocidade, portanto essa nova modelagem se comportou de forma diferente da original, mostrando que é possível obter resposta diferente do sistema com a nova abordagem e, com isso, tentar melhorar o resultado do controle.

Ao tentar simular com as três realimentações, obtivemos dificuldade de modelar o controle, uma vez que a posição fornecida após o bloco “1/s” é a posição geral de movimentação, enquanto para a realimentação precisamos da posição relativa ao offset desejado de parada do eixo Z.

Mesmo não simulando com todas as variáveis, ou seja, utilizando apenas a dupla realimentação, já foi verificado que é válido montar um sistema com mais de uma variável de base para o cálculo do PID e que mais testes devem ser realizados para comparar o desempenho destas modificações em relação à implementação original.