Instituto Mauá de Tecnologia

Projeto Semestral - Sistemas de Controle I

Simulação de controle de posição e velocidade de um

motor DC com parâmetros interativos e pré-calculados

Leonardo Oneda Galvani

Guilherme Nami Bortolozi

Henrique Fortuna Accorinti

Matheus Ferreira Palú

São Caetano do Sul

2023

Sumário

[1. Introdução 3](#_Toc150275992)

[2. Funcionamento 3](#_Toc150275993)

[2.1. Hardware 3](#_Toc150275994)

[2.2. Interface 5](#_Toc150275995)

[3. Identificação do Sistema 7](#_Toc150275996)

[4. Validação do Sistema 12](#_Toc150275997)

[5. Proposta de controle do sistema 14](#_Toc150275998)

[6. Controle Embarcado 18](#_Toc150275999)

[7. Conclusões 24](#_Toc150276000)

# Introdução

O projeto consiste no desenvolvimento de um simulador interativo que controlará a posição angular e velocidade de uma roda de inércia acoplada ao motor DC. O projeto foi desenvolvido utilizando os conhecimentos das matérias estudadas na 4ª série do curso de Engenharia de Controle e Automação do Instituto Mauá de Tecnologia, sendo estas: Sistemas de Controle, Programação Orientada a Objetos e Banco de Dados, Instrumentação, Microcontroladores. Este simulador permitirá experimentar e compreender conceitos importantes relacionados a essas disciplinas.

Para isso, o projeto foi dividido em duas fases. A primeira fase consiste no desenvolvimento do ambiente virtual, subdividida em duas etapas: a primeira etapa abrange o controle interativo dos ganhos de um controlador PID, que controlará a posição em graus (°) e a velocidade em rotações por minuto (RPM), e a segunda etapa abrange a análise o comportamento do sistema com controladores pré-projetados para controle apenas da posição em graus (°). Em ambas as etapas haverá a visualização dos resultados das simulações. A segunda fase do projeto consiste na construção do *hardware* necessário para realizar o que for requisitado pelo simulador, envolvendo a especificação dos componentes e a estrutura entre eles e, por fim, a programação para o sistema funcionar de acordo com o especificado pelo usuário do simulador.

# Funcionamento

Antes de entrar nos estudos da teoria de controle que evolve o projeto, será abordado o funcionamento das etapas que envolvem o ambiente virtual, os componentes presentes e a lógica entre eles.

## *Hardware*

Como mencionado anteriormente, o item principal do projeto é motor DC, e nele temos uma roda de inércia acoplada no eixo de saída de uma caixa de redução. O motivo para implementá-la no sistema foi criar uma carga para este a fim de ter uma maior dificuldade para ser controlado. No mesmo motor existe, acoplado no eixo de entrada da caixa de redução, o próprio rotor do motor e um encoder de efeito hall, como na seguinte imagem:

Uma imagem contendo no interior, mesa, computador, escritório

Descrição gerada automaticamente

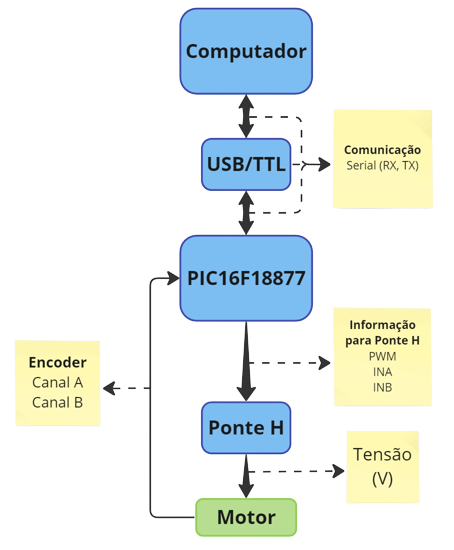
Quando pensamos em controle, pensamos em grandezas analógicas primeiramente. Por exemplo, um motor pode ter diferentes velocidades e, por conta disso, é preciso controlar a tensão que é aplicada no motor. Portanto, será necessário o uso de uma ponte H, que controlará a tensão e o sentido do motor por meio de dois pinos digitais e um PWM (Pulse Width Modulation). A ponte H utilizada no projeto está ilustrada abaixo:

Tela de um aparelho eletrônico

Descrição gerada automaticamente com confiança baixa

Visto que para controlar a tensão e a direção do motor é necessário pinos digitais e um PWM, foi inserido no projeto um microcontrolador, o PIC16F18877. Ele controlará a ponte H enviando os dados de acordo com o controlador implementado. Além de estar conectado na ponte H, o microcontrolador ainda será responsável por contar os pulsos gerados no encoder e identificar, por meio do canal A e B, o sentido de rotação do motor e ao final atribuir ao seu valor corretamente.

Neste momento, o *hardware* já teria capacidade de executar o controle, porém a ideia do projeto é fazer um simulador. Para isso, o *hardware* existente deve ser conectado ao computador. Assim, foi utilizado a comunicação serial entre o microcontrolador e o computador, de acordo com o diagrama de comunicação entre *hardwares* apresentados a seguir:



## Interface

Como dito anteriormente, o simulador conta com duas funcionalidades principais: o controle de posição e controle de velocidade. Para acessar os recursos disponíveis, foi desenvolvido uma interface gráfica usando a linguagem de programação Python, inicializando com a tela “Menu” mostrada a seguir:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Nela, há as opções “Posição” e “Velocidade”, que abrem janelas que permitem configurar e visualizar a simulação, como mostrado na figura a seguir:

Tabela

Descrição gerada automaticamente

Além dessas duas opções, ainda há a opção de simular alguns controladores calculados, por meio da aba “Controladores”. Esta funcionalidade será abordada com mais detalhes no decorrer do relatório.

Uma imagem contendo Tabela

Descrição gerada automaticamente

# Identificação do Sistema

A primeira etapa para projetar um controlador é saber o que ele deve controlar. Em termos matemáticos, devemos conhecer a função de transferência que será trabalhada. No caso do projeto, deve-se encontrar a função de transferência do sistema motor e da roda de inércia.

Por se tratar de um motor DC, sua modelagem é bem desenvolvida, sendo possível encontrar diagramas completos, como o mostrado na figura a seguir:

Diagrama, Esquemático

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

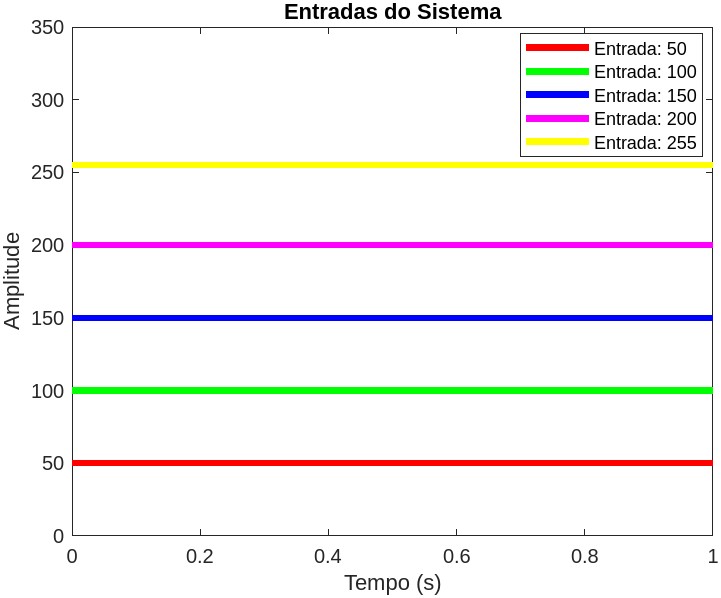
Ao estudar o modelo e suas simplificações, é possível aproximar o modelo do motor para controle de velocidade em um sistema de primeira ordem, e para controle de posição em um sistema de segunda ordem, descritos a seguir:

Existem algumas formas de obter essa função de transferência. Uma delas é a modelagem do motor, que usa as características físicas e elétricas para encontrar a função. Porém, o motor utilizado não tem as informações necessárias para a sua modelagem, tornando o método inviável para esta aplicação. Portanto, para obtenção da sua função de transferência foi optado o método do ensaio, ou levantamento da curava, do motor.

Para esse ensaio, foi decidido que serão realizados 5 degraus com os seguintes valores:

|  |  |
| --- | --- |
| **PWM (8-bit)** | **Porcentagem (%)** |
| 50 | 19,6% |
| 100 | 39,2% |
| 150 | 58,8% |
| 200 | 78,4% |
| 255 | 100% |

Com esses dados, é possível definir a entrada do ensaio (U):



Esses valores foram escolhidos de acordo com a resolução do PWM no microcontrolador. Com as entradas definidas é possível realizar o ensaio. A ideia foi coletar a quantidade de pulsos do encoder no decorrer de 3 segundos, o suficiente para estabilizar o sistema. Com esses dados será realizado 4 abordagens, duas abordagens analíticas, usando os dados de posição e velocidade e duas técnicas computacionais, usando o *software* MatLab, com os mesmos dados.

Ao fim dos ensaios, os seguintes dados foram registrados:

Com os dados registrados, foram definidas as unidades das grandezas a fim de reduzir o processamento necessário no microcontrolador. Dessa maneira, a unidade de velocidade será de pulsos por segundo (pulsos/s) e posição em pulsos (pulsos).

Começando pela análise da velocidade, os dados foram tratados, chegando ao seguinte gráfico:

Com esses dados, foi feita a análise separada de cada saída, seguindo o modelo:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Usando os dados e os conhecimentos aprendidos na matéria de Sistemas de Controle I, obtemos as seguintes funções de transferência e seu modelo médio:

|  |  |  |
| --- | --- | --- |
| Função de Transferência | Constante de tempo (s) | Ganho |
|  | 0,3 | 20 |
|  | 0,3 | 34 |
|  | 0,27 | 26,67 |
|  | 0,19 | 23 |
|  | 0,18 | 21,176 |
|  | 0,248 | 24,97 |

Usando outro método de obtenção, pelo *software* MatLab e sua parte de identificação de sistemas, chegamos nas seguintes funções:

|  |  |  |
| --- | --- | --- |
| Função de Transferência | Constante de tempo (s) | Ganho |
|  | 0,321 | 20,128 |
|  | 0,353 | 33,697 |
|  | 0,295 | 26,26 |
|  | 0,21 | 22,776 |
|  | 0,16 | 21,04 |
|  | 0,2678 | 24,7802 |

Com os resultados obtidos, é possível observar que as funções de transferência médias dos dois métodos têm parâmetros próximos, sendo o erro percentual do ganho de 0,76% e da constante de tempo de 7,4%. Portanto, é possível considerar um modelo médio pela análise da velocidade:

Além da análise dos dados da velocidade, que permite estimar a função de transferência, é proposto neste projeto outra forma de obtenção da curva característica: a partir dos dados de posição do motor, para que no final seja possível comparar os resultados das duas. Portanto, será realizado o mesmo processo feito na velocidade, e posterirormente a análise com o auxílio do MatLab, seguindo o modelo:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

|  |  |  |  |
| --- | --- | --- | --- |
| Analítica | | | *Software* |
| Função | Cte. De tempo | Ganho | Função |
|  | 0,3 | 19,93 |  |
|  | 0,31 | 33,68 |  |
|  | 0,27 | 26,17 |  |
|  | 0,2 | 23,4 |  |
|  | 0,16 | 20,78 |  |
|  | 0,248 | 24,794 |  |

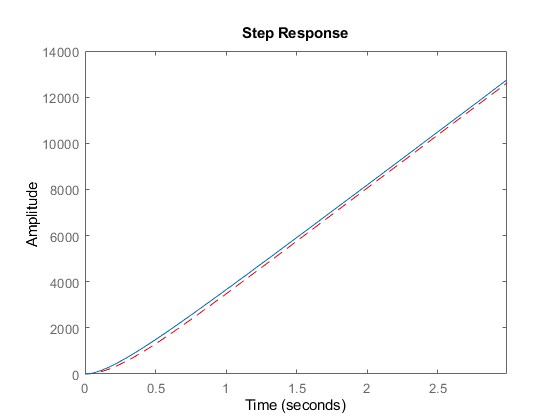
A primeira conclusão importante observada a partir da comparação dos métodos é a função de transferência sem o integrador livre obtida usando as ferramentas do MatLab, diferente do modelo simplificado obtido analiticamente. O modelo sem o integrador livre será utilizado para o cálculo analítico do controlador PID, que será abordado futuramente.

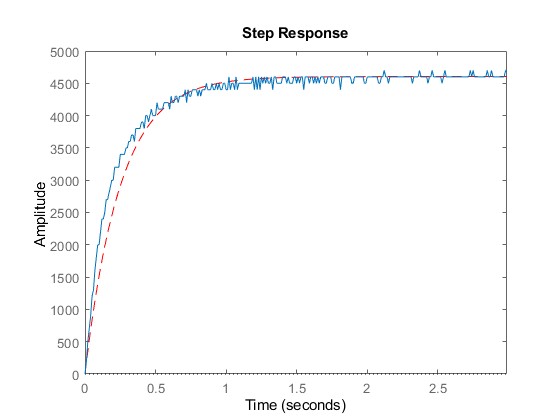
A segunda conclusão possível é: considerando que a função de transferência da velocidade integrada resulta na função de transferência da posição para a mesma entrada, chegamos com funções de transferência semelhantes, com um erro da constante de tempo em 3,8% e no ganho em 0,33%.

Com todos os dados obtidos e calculados é possível obter um modelo médio do motor estudado:

# Validação do Sistema

Agora que temos a função de transferência que representa matematicamente o sistema, podemos comparar a sua resposta com o que o sistema real realizou. A seguir são mostrados alguns testes com as curvas real e simuladas.





A curva azul representa o que o motor realmente realizou, e em vermelho tracejado está a simulação pela função de transferência obtida pelos ensaios.

# Proposta de controle do sistema

Como dito anteriormente, o projeto possui a parte de controle interativo que é baseado em um controlador PID e a parte de controladores projetados. Sobre a segunda parte, o objetivo principal é comparar controladores calculados e projetados com auxílio do MatLab. Estes são: avanço de fase, controlador PID e um controlador PD, sendo o último projetado no MatLab.

Primeiramente, todos os controladores devem seguir o mesmo diagrama de blocos:

Diagrama

Descrição gerada automaticamente

O primeiro controlador projetado será o avanço de fase para o controle da posição do motor. Para isso, foram determinados os requisitos do projeto: um tempo de assentamento (2%)) de 1 segundo e um sobressinal () de 1%. Sabendo disso, é possível começar o projeto do controlador.

Sabendo que:

O primeiro passo para projetar o controle é definir os polos desejados:

Forma

Descrição gerada automaticamente com confiança média

Depois de calcular os polos desejados, é possível começar o cálculo do controlador. O primeiro passo é assumir o zero do controlador igual a parte real do polo desejado, ou seja, . Portanto, temos:

O próximo passo é utilizar o critério de fase para encontrar o valor de :

Com o polo do controlador calculado, encontra-se o ganho do controlador, . Para isso, será usado o critério do módulo:

Depois de obter seus parâmetros, foi obtido o seguinte controlador:

Para verificar se o controlador calculado respeita os parâmetros do projeto é possível simulando a sua resposta no MatLab:

Gráfico

Descrição gerada automaticamente

Outro controlador proposto é o PID, mas para este será necessário utilizar outro modelo, aquele que contêm dois polos diferentes de 0, como mencionado anteriormente:

O controlador PID pode ser escrito como visto abaixo, e ao final do processo os valores de Kp, Ti e Td devem ser definidos.

O primeiro passo é encontrar os valores de Ti e Td a fim de “cancelar” os polos da planta. Para isso, é necessário normalizar o denominador de :

Com o denominador normalizado é possível estabelecer o seguinte sistema:

Nese momento, foi obtido os valores de Ti e Td, faltando calcular o valor de Kp. Porém, antes disso, deve-se olhar o diagrama do lugar raízes admitindo :

Gráfico, Gráfico de linhas, Gráfico de caixa estreita

Descrição gerada automaticamente

É possível verificar que, independentemente do valor do , a resposta do sistema nunca será instável e nunca oscilará, portanto, a única influência de está no tempo de resposta do sistema. Portanto, já satisfaz os requisitos do projeto.

Admitindo o controlador projetado, temos o seguinte controlador e a resposta simulada no MatLab:

Gráfico

Descrição gerada automaticamente

O resultado obtido é satisfatório para projeto do controlador.

Além desses dois projetos de controlador, avanço de fase e PID, será proposto uma comparação com um controlador PD projetado usando ferramentas do MatLab, o RLTOOL. Otimizando o resultado de acordo com os requisitos do projeto e tomando cuidado com o ganho do controlador, chegamos no seguinte controlador e na seguinte saída da planta controlado por esse controlador:

Gráfico

Descrição gerada automaticamente

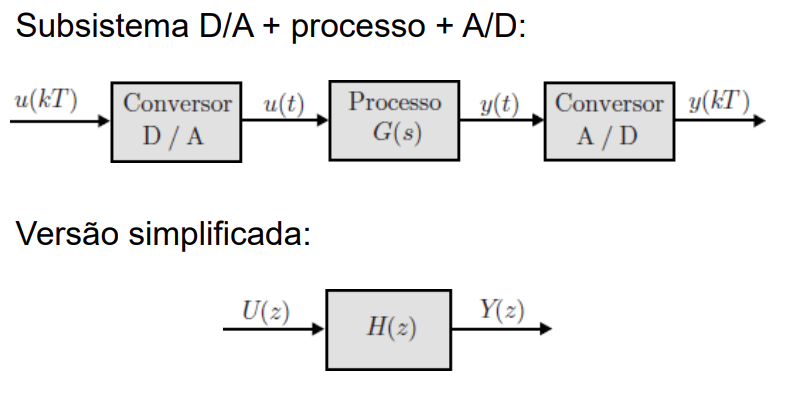
# Controle Embarcado

A fim de embarcar um controlador em um microcontrolador ou CLP por exemplo, o controlador que foi projetado em tempo contínuo precisa ser discretizado. Isso significa que o controlador atualizará em certos tempos definidos por T, conhecido como tempo de amostragem. Abaixo encontra-se um exemplo de um sistema discretizado:

Gráfico

Descrição gerada automaticamente

Para isso, utilizaremos a função de transferência do nosso controlador e a discretização será feita usando a transformada Z e a simplificação dos diagramas com os conversores analógicos/digitais e digitais/analógicos, como na imagem e equação abaixo:



com

No primeiro momento, o cálculo será demonstrado utilizando o controlador de avanço de fase. Porém, os outros dois controladores serão construídos utilizando a ferramenta do MatLab.

Para realizar a transformada Z foi utilizado o método da expansão em frações parciais:

Considerando um tempo de amostragem para os controladores de 10ms, , chegamos a seguinte função discreta:

Para otimizar o processo foi utilizada a função c2d, que resulta na função já discretizada a partir da função de transferência em tempo contínuo e o tempo amostral. O resultado está descrito nas seguintes funções:

O próximo passo é encontrar a equação de diferenças utilizando a manipulações algébricas e a propriedade do atraso, . O cálculo será novamente desenvolvido para o controlador de avanço de fase, e os outros dois controladores terão exatamente o mesmo cálculo.

Sendo “u” a entrada, o erro, e “y” a saída do sistema, o PWM, aplicada no motor. Aplicando o mesmo cálculo para os outros controladores achamos as seguintes equações de diferenças:

Com essas equações de diferenças é possível implementá-las no microcontrolador e testá-las a fim de compará-las com os resultados simulados, como segue nos seguintes gráficos:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Gráfico

Descrição gerada automaticamente

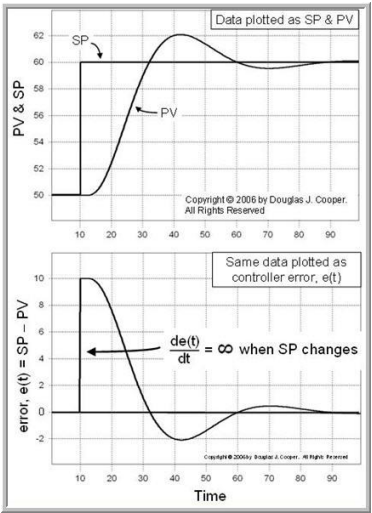
Gráfico

Descrição gerada automaticamente

A linha vermelha tracejada representa a curva simulada e a linha azul representa a curva do sistema real.

Outro controle discretizado que é importante ser mencionado está compreendido na primeira parte do projeto, no qual os ganhos de um controlador PID são interativos. Nessa parte, existe um controlador discreto implementado da seguinte forma:

A componente integrativa é um somatório do erro multiplicado a variação do tempo, , e a componente derivativa é uma diferença dividida pelo mesmo . Essa diferença geralmente é feita entre o erro atual e anterior, porém esta técnica pode causar uma derivada que tende ao infinito. Por esse motivo, utiliza-se a variação da variável do processo, ou seja, ao invés de fazer a diferença entre os erros, usa-se a variável que está sendo controlada, no caso a posição. Então, é feita a subtração entre a posição atual e a posição anterior. Ao realizar a seguinte transformação a derivada terá o comportamento invertido. Para ajustar isso, a parcela da derivada será subtraída ao invés de somada, possível ver na seguinte comparação:



Dessa forma, obtemos a seguindo função:

# Conclusões

Como primeira conclusão, é possível observar diferenças na resposta do sistema. Existem alguns motivos para isso: um deles é a folga presente na caixa de redução, que interfere na dinâmica do sistema. Não é possível eliminá-la completamente, então seria necessário utilizar outra caixa de redução a fim de aproximar ainda mais os resultados obtidos.

Outro motivo relevante levantado é a zona morta do motor. Trata-se de um intervalo de tensões entre 0V e uma certa tensão em que o motor não se mexe por questões mecânicas, que incluem principalmente o atrito estático e momentos.

Apesar dessas pequenas diferenças observadas, é visível que o projeto atendeu às expectativas, sendo possível comparar a eficiência entre controladores e métodos, e entender todos os conceitos envolvendo as matérias abordadas.