**Projeto Semestral - Sistemas de Controle**

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

Sumário

[1 Introdução 3](#_Toc149727066)

[2 Funcionamento 3](#_Toc149727067)

[2.1 *Hardware* 3](#_Toc149727068)

[2.2 Interface 5](#_Toc149727069)

[3 Identificação do Sistema 6](#_Toc149727070)

[4 Validação do Sistema 12](#_Toc149727071)

[5 Proposta de controle do sistema 13](#_Toc149727072)

[6 Controle Embarcado 16](#_Toc149727073)

[7 Conclusões 17](#_Toc149727074)

# 1 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 4° Série do curso de engenharia de controle e automação do Instituto Mauá de Tecnologia, sendo estas: Programação Orientada a Objetos e Banco de Dados, Instrumentação, Microcontroladores e Sistemas de Controle. Este simulador permitirá experimentar e compreender conceitos importantes relacionados a essas disciplinas.

Para isso existirá algumas fases deste projeto, sendo elas: o desenvolvimento do ambiente virtual, a qual existirá duas etapas, sendo a primeira a parte de 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 em analisar o comportamento do sistema com controladores pré-projetados para controle apenas da posição em graus (°) sendo que em ambas as etapas haverá a visualização dos resultados das simulações. A segunda fase consiste na construção do *hardware* necessário para realizar o requisitado pelo simulador, envolvendo a especificação dos componentes e a estrutura entre eles e pôr fim a programação para o sistema funcionar de acordo com o especificado pelo usuário do simulador.

# 2 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.

## 2.1 *Hardware*

Como mencionado anteriormente o item principal do projeto é motor DC, 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 e 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, um encoder de efeito hall, como na seguinte imagem:

Uma imagem contendo no interior, mesa, computador, escritório

Descrição gerada automaticamente

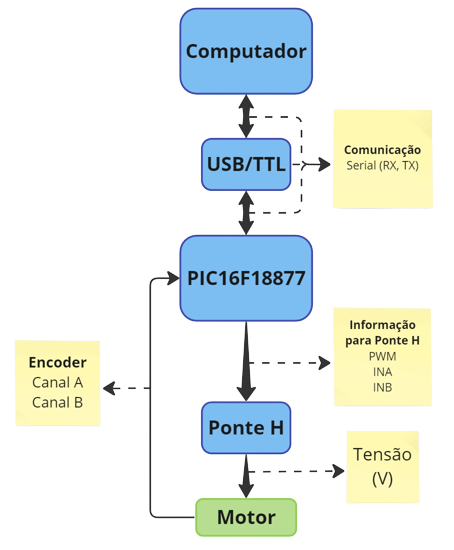
Quando pensamos em controle pensamos em grandezas analógicas primeiramente, por exemplo um motor pode ter diferentes velocidade, por conta disso temos que controlar a tensão que é aplicada no motor, portanto será necessário o uso de uma ponte H, que controlará a tensão e o sentido do motor por meio de dois pinos digitais e um PWM (Pulse Width Modulation), sendo a seguinte ponte H:

Tela de um aparelho eletrônico

Descrição gerada automaticamente com confiança baixa

Agora visto que para controlar a tensão e a direção do motor é necessário pinos digitais e um PWM, foi inserido no projeto um microcontrolador, no caso o PIC16F18877, ele que controlará a ponte H mandando 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 tem que se conectar ao computador, para isso foi utilizado a comunicação serial entre o microcontrolador e o computador, ficando o esquema final:



## 2.2 Interface

Como dito anteriormente o simulador tem duas principais áreas, para acessar estas foi desenvolvido uma interface gráfica usando a linguagem de programação Python, inicializando com a tela Manu:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Nela você terá as opções “Posição” e “Velocidade” que te levaram para o grupo onde poderá configurar e visualizar a simulação:

Tabela

Descrição gerada automaticamente

Além dessas duas opções ainda terá a opção de ir na aba “Controladores” nela você poderá simular alguns controladores calculados, sendo este o que será bordado no decorrer do relatório.

Uma imagem contendo Tabela

Descrição gerada automaticamente

# 3 Identificação do Sistema

A primeira etapa para conseguir projetar um controlador é saber o que ele deve controlar, em termos matemáticos, devemos saber a função de transferência que será trabalhada, no caso do projeto, temos que encontrar a função de transferência do sistema motor mais roda de inércia.

Como se trata de um motor DC, a sua modelagem é bem desenvolvida, é possível encontrar diagramas completos, como por exemplo a figura a seguir:

Diagrama, Esquemático

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

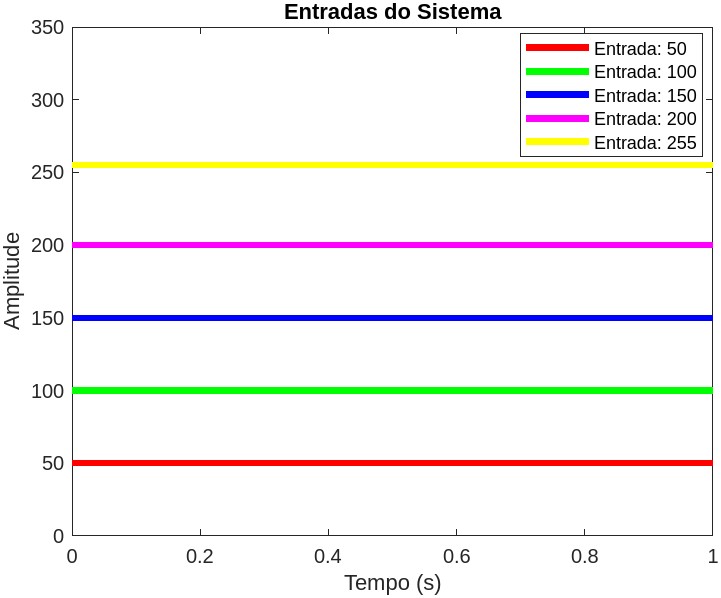
Estudando o modelo e suas simplificações é possível aproximas o modelo do motor para controle de velocidade um sistema de primeira ordem e para controle de posição um sistema de segunda ordem.

Existe algumas formas para obter esta função de transferência sendo uma delas a modelagem do motor, esta usa as características físicas e elétricas para encontrar a função, porém o motor utilizado não tem as informações necessárias para a sua modelagem, se tornando inviável este método, 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 será realizado 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 podemos definir a entrada do ensaio (U):



Com as entradas definidas podemos realizar o ensaio, a ideia foi coletar a quantidade de pulsos do encoder no decorrer de 3 segundo pois era 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.

Com os ensaios chegamos nos seguintes dados:

Com esses dados foi definido as unidades das grandezas, com o motivo de reduzir o processamento necessário no microcontrolador, a unidade de velocidade será de pulsos por segundo (pulsos/s) e posição em pulsos (pulsos).

Para começar vamos fazer a análise da velocidade para isso os dados foram tratados, ficando assim:

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

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Usando os dados e os conhecimentos aprendidos na matéria de Sistemas de Controle 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 |

Agora será usado outro método de obtenção, utilizando o *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 podemos ver que as funções de transferência média dos dois métodos tiveram parâmetros próximos, sendo o ganho um erro de 0,76% e a constante de tempo de 7,4%, portando é possível considerar um modelo médio pela análise da velocidade:

Além da análise dos dados da velocidade e assim 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 comparemos os resultados das duas. Portanto será realizado o mesmo processo feito na velocidade, a análise analítica depois com o auxílio do MatLab, sendo aquela seguindo o seguinte 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 |  |

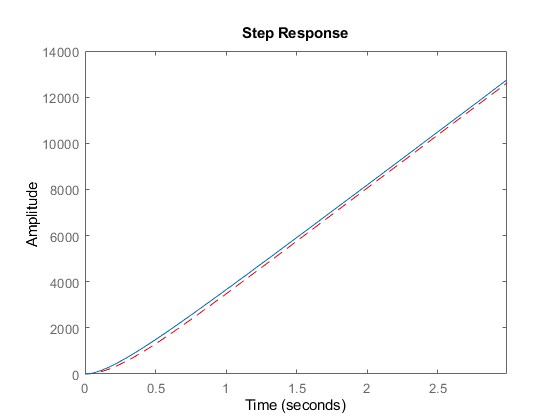
A primeira conclusão importante é que o uso da ferramenta do MatLab ele retorna uma função de transferência sem o integrador livre, diferente do modelo simplificado obtido analiticamente. O modelo sem o integrador livre será utilizado para o cálculo analítico do controlador PID, que será abordado futuramente.

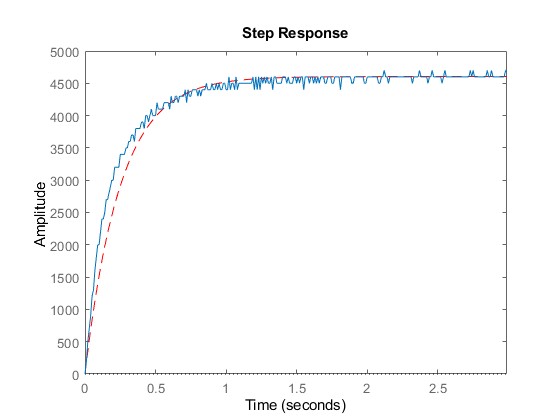
A segunda conclusão possível é que considerando que a função de transferência da velocidade integrada resulta na função de transferência da posição para a mesma entrada, chegamos com funções de transferência semelhantes, com um erro da constante de tempo de 3,8% e no ganho de 0,33%.

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

# 4 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 mostra alguns testes com as curvas real e simuladas.





Sendo a curva azul o que o motor realmente realizou e o vermelho tracejado simulado pela função de transferência obtida pelos ensaios.

# 5 Proposta de controle do sistema

Como dito anteriormente o projeto terá a parte de controle interativo que é baseado em um controlador PID e terá a parte de controladores projetados, e nessa última parte tem como objetivo comparar esses controladores, que envolvem 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 vão 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 foi determinado os requisitos do projeto, no caso foi 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 deste é assumir o zero do controlador igual a parte real do polo desejado, ou seja, , portanto nesse ponto temos o seguinte:

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

Com o polo do controlador calculado, o próximo passo é encontrar o ganho do controlador, , para isso será usado o critério do módulo:

Portanto 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 da seguinte forma, no final deste os valores de Kp, Ti e Td tem que 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:

Neste momento foi obtido os valores de Ti e Td, agora faltaria calcular o valor de Kp, porém antes disso, temos que olhar o diagrama do lugar raízes admitindo :

Gráfico, Gráfico de linhas, Gráfico de caixa estreita

Descrição gerada automaticamente

É possível ver que independentemente do valor do a resposta do sistema nunca será instável e nunca oscilará, portanto, a única influência seria no tempo de resposta do sistema, por tanto para os requisitos do projeto já satisfaz.

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

Gráfico

Descrição gerada automaticamente

Sendo 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 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

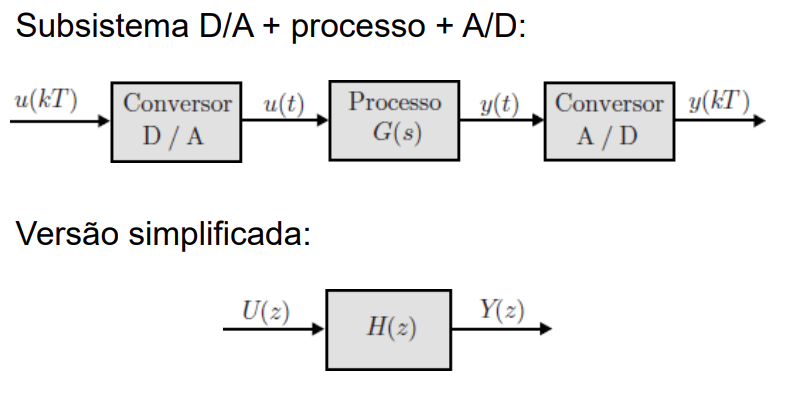
# 6 Controle Embarcado

Para embarcar um controlador em um microcontrolador ou CLP, por exemplo, o controlador que foi projetado em tempo contínuo precisa ser discretizado, ser discretizado significa que o controlador atualizará em certos tempos definidos por T, conhecido como tempo de amostragem, um exemplo de um sistema discretizado:

Então para isso Gráfico

Descrição gerada automaticamente

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



No primeiro momento o calculo será demonstrado utilizando o controlador de avanço de fase, porém os outros dois controladores serão utilizados 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, onde com a função de transferência em tempo contínuo e o tempo amostral, resulta na função já discretizada. Resultando nas seguintes funções:

Para KT

# 7 Conclusões

COMPARAÇÕES e FOLGAS