

# Desenvolvimento de biblioteca para Arduino empregando o MRAC para fins didáticos, demonstrado no controle de velocidade de motor tacogerador

Juan V. B. dos Santos \* José D. F. de O. Luna \*\*  
Kariston D. Alves \*\*\*

\* *Faculdade de Engenharia de Controle e Automação, Instituto  
Federal, RO, (e-mail: juan.santos@estudante.ifro.edu.br)*

\*\* *Faculdade de Engenharia de Controle e Automação, Instituto  
Federal, RO, (e-mail: jose.luna@ifro.edu.br)*

\*\*\* *Faculdade de Engenharia de Controle e Automação, Instituto  
Federal, RO, (e-mail: kariston.alves@ifro.edu.br)*

---

**Abstract:** In the learning environment, the great benefit of carrying out practical activities is observed. With that in mind, this work aims to promote the practice of advanced control for students and researchers, using the Arduino through a library for implementing the MRAC (adaptive control by reference model). For this work, all the necessary steps were taken to take MRAC from mathematical formulas on paper to the form of code, ranging from mathematical understanding, the way it can be represented in time-limited simulation programs and its adaptation to other environments of programming. That in the end, it has as a product the library for Arduino, available on GitHub for public access, capable of using the MRAC in a simple way with the purpose of being a simpler gateway to adaptive control content in non-linear systems, and how example of use will be applied in the speed control of a tachogenerator motor. This project has the ability to provide, under the ideals of open science, a tool for educators from any educational institution, to work, in a practical way, skills and competencies related to adaptive control, with emphasis on the MRAC.

**Resumo:** No ambiente de aprendizagem, observa-se o grande benefício da realização de atividades práticas. Entretanto, dependendo do tema pode haver uma íngreme curva de aprendizado. Pensando nisso, esse trabalho tem como objetivo promover a prática de controle avançado para estudantes e pesquisadores, utilizando-se do Arduino através de uma biblioteca para implementação do MRAC (controle adaptativo por modelo de referência). Para este trabalho, foi percorrido todo o caminho necessário para tirar o MRAC das fórmulas matemáticas em papel para a forma de código, indo desde o entendimento matemático, a forma como pode ser representado em programas de simulação de tempo limitado e a sua adaptação para outros ambientes de programação. Que ao final, possui como produto a biblioteca para Arduino, disponibilizada no GitHub para acesso público, capaz de empregar o MRAC de forma simples com propósito de ser uma porta de entrada mais simples ao conteúdo de controle adaptativo em sistemas não-lineares, e como exemplo de uso será aplicado no controle de velocidade de um motor tacogerador. Esse projeto tem a capacidade de fornecer, sob os ideais de ciência aberta, uma ferramenta para educadores de qualquer instituição de ensino, para trabalhar, de forma prática, habilidades e competências relacionadas ao controle adaptativo, com destaque para o MRAC.

*Keywords:* MRAC; Arduino; open science; GitHub; nonlinear systems.

*Palavras-chaves:* MRAC; Arduino; ciência aberta; GitHub; sistemas não-lineares.

---

## 1. INTRODUÇÃO

Durante o desenvolvimento de projetos de controle é comum se deparar com dois principais objetivos de controle: Regulatório e servo.

No Controle Regulatório, o SP (*SetPoint*, palavra inglesa para Ponto Objetivo ou Meta) é fixo e tem como objetivo rejeitar ou minimizar os efeitos das perturbações, como afirma Campos and Teixeira (2010).

Na Figura 1 (adaptado de Campos and Teixeira (2010)) é possível observar o SP fixo, a MV (Variável Manipulada, em inglês) sendo modificada para que o PV (Valor do Processo, em inglês) seja capaz de acompanhar o objetivo.

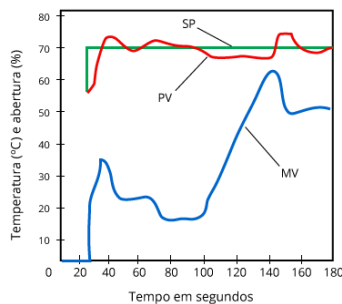


Figura 1. Exemplo de controle regulatório. Fonte: Adaptado de Campos and Teixeira (2010).

Já no Controle Servo, o SP é variável ao longo do tempo, sendo necessário que a variável controlada tanto acompanhe cada novo SP, quanto rejeite perturbações, como explica Campos and Teixeira (2010).

Na Figura 2 (adaptado de Campos and Teixeira (2010)) é possível observar como o SP é modificado ao longo do tempo, e como a MV precisa agir para garantir que o PV acompanhe o objetivo.

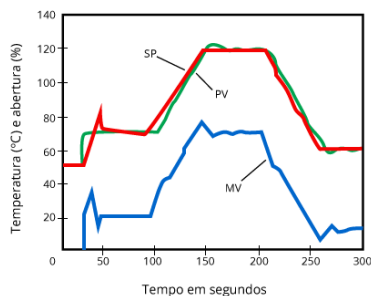


Figura 2. Exemplo de controle servo. Fonte: Adaptado de Campos and Teixeira (2010).

A partir dessa condição, são definidos diferentes objetivos, dependendo da aplicação desejada, sendo uma das principais decisões o tempo de acomodação, que pode tornar o sistema mais robusto (tempo de acomodação maior com sobressinal menor ou nulo) ou agressivo (tempo de acomodação menor com alto sobressinal).

Em outras palavras, um controle robusto significa que garante ao sistema características importantes como estabilidade, rejeição de perturbações, ou índices mínimos de desempenho frente variações que possam ocorrer no

sistema, como explica Fischman (1993) e Tambara (ano). De forma contrária, caso um controlador cause mudanças significativas no sistema em um curto tempo, pode-se dizer que é um controle agressivo.

Além dessas considerações, é necessário observar a linearidade ou não-linearidade da planta para realizar o controle. "Sistemas lineares: São fenômenos ou dispositivos cujo comportamento dinâmico pode ser descrito por equações diferenciais (ou recursivas) lineares", Trofino (ano). Isso significa que, a dinâmica do sistema não se altera para todos os pontos de operação. Por outro lado, sistemas não-lineares apresentam dinâmica variável ao longo dos SP, como afirma Callai et al. (2007).

Apesar da diferença nos objetivos de controle, em sistemas lineares uma mesma técnica de controle funciona tanto para o controle servo, como para o regulatório, visto que nesses sistemas a dinâmica da planta é constante para todo SP, como confirmou Trofino (ano).

Entretanto, em sistemas não-lineares o controle servo não pode ser realizado da mesma forma que o regulatório, pois nesses sistemas a dinâmica da planta é diferente para cada SP, como atestou Callai et al. (2007).

### 1.1 Problemática

Para o caso de controle servo em sistemas não-lineares, há a possibilidade da aplicação de técnicas de controle adaptativo. Como disse Tambara (ano) (*apud* Astrom (1987)):

Controle adaptativo pode ser definido como uma técnica de controle que possui a capacidade de mudar seu comportamento de acordo com as modificações dos parâmetros, da dinâmica de um processo ou por distúrbios que afetam este sistema.

Ou seja, o funcionamento do controle adaptativo se baseia em adaptar o sistema de controle se amparando na variação da dinâmica do sistema controlado.

No entanto, o controle adaptativo possui uma íngreme curva de aprendizado para aqueles que buscam aprender ou ensinar, que a depender da situação pode ser um motivo a mais para desistência. Por exemplo, professores universitários podem evitar entrar nesse tópico durante aulas de controle avançado, pois seriam necessárias muitas aulas para que os alunos compreendessem e fossem capazes de projetar utilizando essa técnica.

### 1.2 Justificativa

Levando em conta essa dificuldade, como é possível facilitar a entrada no aprendizado do controle adaptativo? Uma alternativa para responder a essa pergunta é desenvolver um facilitador que possa ser utilizado no ensino introdutório do controle adaptativo, por exemplo uma biblioteca em que o usuário informe os dados básicos do sistema a ser controlado e consiga acompanhar e analisar a evolução do controlador durante o processo, garantido a vivência prática necessária para contextualizar a teoria de controle moderno.

Além disso, por se tratar de uma ferramenta introdutória, uma boa alternativa é criar esta biblioteca no ambiente do Arduino, um microcontrolador amplamente utilizado por projetos em áreas como robótica e eletrônica. Como é dito pela Arduino (2021), seu produto facilita que pessoas pelo mundo todo possam acessar tecnologias avançadas quem interagem com o mundo físico. Sendo possível observar o seu alcance pelos inúmeros artigos que utilizam de seu *hardware* e *software* em suas pesquisas, o qual é o confirmado pela fala de Evans et al. (2013):

O Arduino tem sido adotado como a ferramenta preferida das comunidades de desenvolvedores e fabricantes interessados na construção e prototipagem de seus próprios projetos.

Há também opções de controladores como o ESP8266 e o Raspberry. O ESP pode ser programado pela mesma IDE (Ambiente de Desenvolvimento Integrado, em inglês) do Arduino, portanto a biblioteca seria funcional em ambos. E o Raspberry possui um preço elevado quando comparado com os demais microcontroladores.

E para tornar acessível essa facilidade a um maior número de pessoas, uma alternativa seria disponibilizar essa biblioteca em um site muito utilizado por pesquisadores da área de programação, tanto na língua portuguesa como inglesa, como o GitHub, um *site* repositório dos mais diversos conteúdos que envolva informática, como modelos 3D, códigos, arquivos de circuitos elétricos, bibliotecas independentes para Arduino e outros controladores, entre outros.

### 1.3 Objetivo geral

Desenvolver uma biblioteca para uso em dispositivos Arduino, que será capaz de implementar o MRAC (*Model Reference Adaptive Control*, Controle Adaptativo por Modelo de Referência) em sistemas de primeira ordem. E seguindo o ideal de ciência aberta, será disponibilizada gratuitamente no GitHub. Como forma de demonstração, será apresentado o seu uso prático em um módulo eletrônico de controle de velocidade de rotação de motor DC (Corrente contínua, em inglês) por um motor tacogerador.

### 1.4 Objetivos específicos

De maneira a alcançar o desejado objetivo deste trabalho, é necessário que alguns passos sejam realizados. Dentre eles tem-se a necessidade do entendimento fundamental do método MRAC, para assim ser capaz de entender como outros autores na área utilizaram este método em suas pesquisas e projetos. Com base nisso, adentrar no estudo sobre desenvolvimento de bibliotecas de Arduino e de como cálculos matemáticos mais complexos podem ser traduzidos em código. E finalizando essa parte de estudo inicial, será realizada uma revisão do circuito eletrônico do módulo eletrônico utilizado de exemplo. Tudo isso servirá de apoio para o desenvolvimento adequado deste projeto, que também disponibilizará o produto no *site* GitHub, em português brasileiro e inglês americano de forma a beneficiar um maior número de pessoas envolvidas em projetos com controle adaptativo.

## 2. REFERENCIAL TEÓRICO

[CITAÇÃO FALANDO DE ALGUNS DOS CONTROLES ADAPTATIVOS EXISTENTES].

Dentro do controle adaptativo, há métodos, como escalonamento de ganhos (também conhecida como técnica de mapeamento), reguladores auto-ajustáveis (STR), entre outros.

Dentre eles há o método por modelo de referência, o MRAC. Neste método, o controlador se modifica a fim de fazer com que a saída real da planta seja igual a saída de um modelo de referência, como explicou Tambara (ano) ao dizer "Um sistema de controle por modelo de referência é aquele em que o comportamento dinâmico do sistema em malha fechada é, idealmente, idêntico ao de um modelo de referência escolhido".

Na Figura 3 é apresentada a dinâmica geral de funcionamento do MRAC: O controlador envia sinal a planta, que tem sua saída comparada com a saída do modelo, com base nessa diferença, os parâmetros do controlador são modificados até que saída da planta seja igual a saída do modelo.

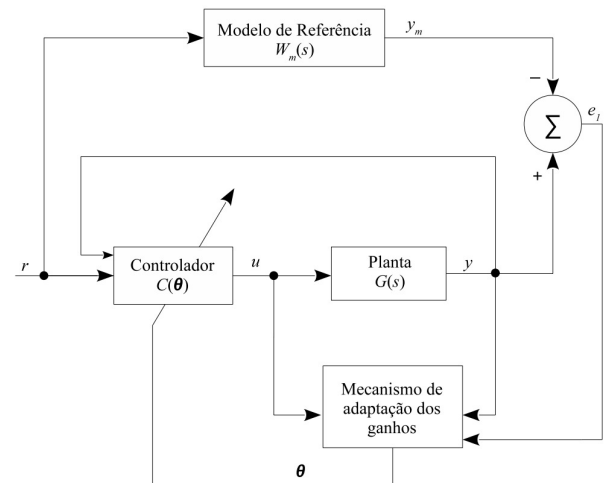


Figura 3. Modelo de controle adaptativo por modelo de referência (MRAC). Fonte: Tambara (ano).

O controle adaptativo pode ser distinguido em 2 formas: Controle direto ou indireto. No controle indireto os parâmetros da planta são estimados, e a partir disso o controlador é ajustado, como disse Bruce (2021) e Tambara (ano). Já no controle direto, não há identificação da planta, mas o controlador continua a se modificar a fim de zerar o erro entre a saída da planta e saída do modelo, como também disse Bruce (2021) e Tambara (ano).

Na Figura 4 é apresentada a dinâmica geral de funcionamento do controle indireto. É possível observar como os cálculos de adaptação do controlador se baseiam na estimativa de parâmetros da planta.

Na Figura 5 é apresentada a dinâmica geral de funcionamento direto, onde não há a necessidade de estimação dos parâmetros da planta para realizar a adaptação do controlador.

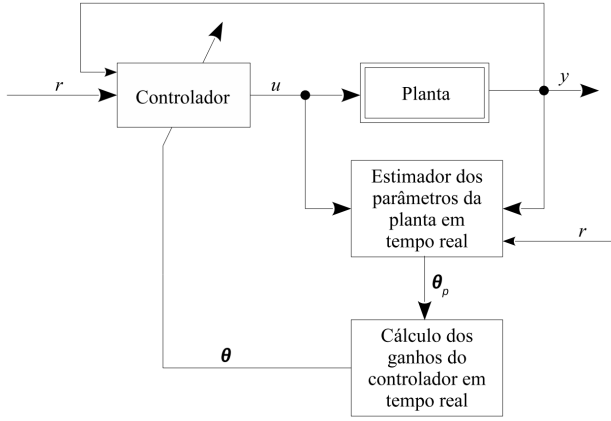


Figura 4. Modelo de controle adaptativo indireto. Fonte: Tambara (ano).

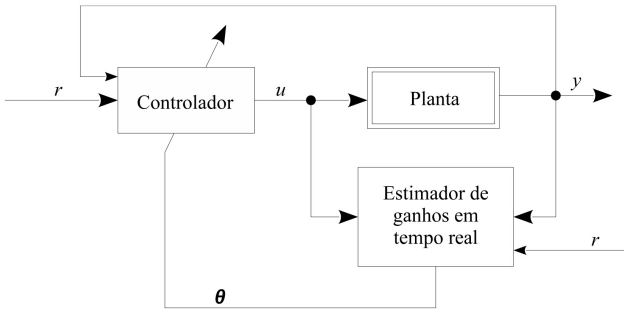


Figura 5. Modelo de controle adaptativo direto. Fonte: Tambara (ano).

Ainda dentro do MRAC, é possível projetar seu controle utilizando-se a regra do MIT ou pelo método de Lyapunov. Pela regra do MIT, os parâmetros desconhecidos da planta são substituídos pelas suas estimativas em cada instante de tempo, como disse Silva (2017). Já o método de Lyapunov é utilizado para garantir a convergência e estabilidade de sistemas com controle adaptativo, como disse Souza (2016).

De maneira simplificada pode-se explicar a regra do MIT começando pela Equação 1 em que  $J$  é uma função custo que é calculada pelo quadrado do erro, pois assim considera como mais erro tanto o erro para mais quanto para menos. E sendo  $\theta$  o parâmetro de ajuste do controlador, como explicou Mallick and Mondal (2019a) e Mallick and Mondal (2019b).

Em seguida, observa-se a Equação 2, que pode ser traduzida em: A variação do parâmetro de ajuste do controlador ao longo do tempo é igual a variação da função custo ao longo do tempo ponderada pela taxa de adaptação do controlador ( $\gamma$ ) que também é igual a variação do erro ao longo do tempo ponderado pela taxa de adaptação, como apresentou Mallick and Mondal (2019a) e Mallick and Mondal (2019b).

$$J(\theta) = \frac{e^2}{2} \quad (1)$$

$$\frac{d\theta}{dt} = -\gamma \frac{\delta J}{\delta \theta} = -\gamma e \frac{\delta e}{\delta \theta} \quad (2)$$

## 2.1 Decisão de pesquisa

Nesta pesquisa utilizou-se o controle direto do MRAC pela regra do MIT, pois como o objetivo é tornar a curva de entrada no controle adaptativo menos íngreme para aqueles que buscam aprender, optou-se pelas formas simples e compreensíveis do MRAC, como é o caso da regra do MIT em relação o método de Lyapunov e o controle direto em relação o controle indireto.

Além disso, conforme apresentado a seguir, segundo melhor conhecimento dos autores, a maioria das aplicações do MRAC foi realizada apenas em simulação, em *softwares* como o MATLAB, uma menor quantidade chega a utilizar microcontroladores como o Arduino no projeto físico, porém nenhuma pesquisa foi encontrada desenvolvendo uma biblioteca.

## 2.2 MRAC em outras áreas

De maneira geral, diversos autores da atualidade utilizam-se do método adaptativo MRAC para o desenvolvimento de seus projetos. Podemos citar o trabalho de Pawar and Parvat (2015) ao projetar um pêndulo invertido com controle adaptativo a fim de servir como base de pesquisa para controle de braços robóticos, sistema de lançamento de satélites, entre outros.

Além desse, é observado também sua aplicação em sistemas pneumáticos de para lavagem de garrafas, como apresentou Mushiri et al. (2017), a fim de evitar desperdícios, por se tratar se um operação com variação constante nos parâmetros de temperatura e pressão.

Observa-se também a aplicação do MRAC em manufatura aditiva, conhecida popularmente como impressão 3D, como é apresentado por Candy (2021) ao demonstrar a utilidade do MRAC em uma variedade de sub processos na manufatura aditiva.

Um exemplo, cada vez mais presente, são os carros autônomos, parte dessa tecnologia utiliza um sistema de controle de distancia entre entre veículos, que necessita funcionar bem em situações diversas como inclinação da rua, pista molhada, velocidade da via, peso do carro, entre outros. Para isso também é utilizado o MRAC, como é apresentado por Sapiee and Sudin (2010).

## 2.3 MRAC na literatura

Na literatura também é observado a relevância que o MRAC tem, como exemplo o livro 'Controle Adaptativo Teoria e Aplicação' por Rodrigo Tambara (ano) e o livro 'Adaptive Control' por Karl Astrom e Bjorn Witternmark (1995) que apresentam o controle adaptativo de forma clara e descritiva.

Também vale destaque o artigo *Robust Model Reference Adaptive Control* de Kreisselmeier and Anderson (1986) que é citado por centenas de outros trabalhos.

## 2.4 Análise de performance

Nesta pesquisa foram mais proveitosos os trabalhos de Mallick and Mondal (2019a) e Mallick and Mondal (2019b)

por apresentarem com detalhes desde o funcionamento do MRAC, os cálculos para a regra do MIT e método de Lyapunov, até as formulas matemáticas utilizadas na análise de performance que também foram utilizadas neste trabalho.

Nestas pesquisas, a comparação de performance entre o MRAC pelo método de Lyapunov e o MRAC junto do controle PID (Proporcional, Integral e Derivativo, em inglês) foi baseada nos resultados obtidos pelas seguintes formulas: IAE (Integral da magnitude Absoluta do Erro, em inglês), ISE (Integral do Erro ao Quadrado, em inglês), ITAE (Integral do Tempo de Erro Absoluto, em inglês) e ITSE (Integral do Tempo de Erro ao Quadrado, em inglês). Que são calculadas pelas seguintes equações:

$$IAE = \int_0^{\infty} |e(t)| dt \quad (3)$$

$$ISE = \int_0^{\infty} [e(t)]^2 dt \quad (4)$$

$$ITAE = \int_0^{\infty} t |e(t)| dt \quad (5)$$

$$ITSE = \int_0^{\infty} t [e(t)]^2 dt \quad (6)$$

## 2.5 Artigo predecessor

Com destaque também para o artigo de Souza et al. (ano), ainda não publicado, do qual foi utilizado o módulo didático produzido, a fim de servir como exemplo de funcionamento da biblioteca desenvolvida, como se observa na Figura 6.



Além disso, segundo melhor conhecimento dos autores, nenhuma pesquisa foi encontrada desenvolvendo uma biblioteca para Arduino aplicando o MRAC.

### 3. DESENVOLVIMENTO

Neste capítulo será apresentado as etapas do desenvolvimento deste projeto, com o objetivo de deixar claro ao leitor como se deu a construção do trabalho, como também meios da recriação desta pesquisa.

#### 3.1 MRAC aplicado no MATLAB

Começando pelo livro 'Controle adaptativo' de Tambara (ano), onde é apresentado os conceitos básicos como controle adaptativo, robustez, controle direto e indireto. Segue então para a revisão de tópicos como a transformada de Laplace, aproximação de Euler (tópico que será aprofundado no próximo subcapítulo deste artigo), modelagem por função de transferência tanto no tempo contínuo como no discreto. Em seguida se aprofunda no tópico de estimação de parâmetros, onde o tema é exposto e são apresentados exercícios resolvidos utilizando-se o *software* MATLAB.

Em seguida começa o tópico de controle por modelo de referencia, onde é iniciado a explicação sobre o MRAC, seu funcionamento, aplicação em plantas tanto de primeira ordem como maiores e de diferentes grau relativo. No final do tópico são apresentados exercícios simples aplicando o que foi ensinado durante o tópico, também com resolução utilizando o MATLAB.

Baseado nesses exercícios e códigos de MATLAB, foi possível entender um pouco mais na pratica, mesmo que por simulação, o conceito de controle adaptativo e iniciar o desenvolvimento deste trabalho. Mais precisamente, foi utilizando como base principal o segundo código do exercício da seção 4.5.1, onde é projetado um controlador MRAC aplicado a uma planta de primeira ordem.

Visto que, apesar de motores DC possuem tanto uma dinâmica física (inércia, atrito, entre outros) quanto elétrica (corrente nos condutores, geração de campo magnético, efeito indutivo, entre outros), a dinâmica física se mostrou dominante nos testes realizados com o módulo de Souza et al. (ano), sendo possível considerar a planta do módulo tacogenerador como um sistema de primeira ordem.

#### 3.2 Adaptação de código do MATLAB para Arduino

Apesar do código de MATLAB funcionar como deveria, este programa não funciona da mesma forma que o compilador de códigos do Arduino.

Em primeiro lugar porque a dinâmica de ambos os ambientes são diferentes, ou seja, enquanto o MATLAB funciona em uma quantidade de tempo fechada, no Arduino o código pode funcionar continuamente sem limite.

Segundo porque tanto a linguagem de programação quanto o uso de funções em cada um são diferentes, por exemplo, no MATLAB há uma função específica para converter uma função do tempo contínuo para o discreto, já no Arduino foi necessário escrever todo o código necessário para essa conversão, utilizando o Método de Euler na discretização de sistemas explicada em Tambara (ano).

Para isso se tomou como base a explicação de Tambara (ano) e conceitos matemáticos para reescrever a FT de forma que fosse possível utilizar em código.

Portanto, tendo como base que uma Função de Transferência de primeira ordem pode ser escrita como se vê na Equação 7.

$$\frac{Y(s)}{U(s)} = \frac{K}{\tau s + a} \quad (7)$$

O método de Euler para discretização de sistemas é apresentado da Equação 8 à Equação 18.

$$\tau s Y(s) + a Y(s) = K U(s) \quad (8)$$

Lembrando que a conversão do domínio de Laplace para derivadas se dá por  $s = \frac{d}{dt}$ , o que leva a Equação 8 para Equação 9.

$$\tau \frac{dy}{dt} + ay = Ku \quad (9)$$

Relembrando-se que uma derivada é descrita como se apresenta na Equação 10.

$$\frac{dy}{dt} = \frac{y(t+1) - y(t)}{T_s} \quad (10)$$

Que aplicado na Equação 9 resulta na Equação 11.

$$\frac{dy}{dt} = \frac{-a}{\tau} y + \frac{Ka}{\tau} u \quad (11)$$

$$\frac{y(t+1) - y(t)}{T_s} = \frac{-a}{\tau} y(t) + \frac{Ka}{\tau} u(t) \quad (12)$$

$$y(t+1) - y(t) = \left( \frac{-a}{\tau} y(t) + \frac{Ka}{\tau} u(t) \right) T_s \quad (13)$$

$$y(t+1) = y(t) + \left( \frac{-a}{\tau} y(t) + \frac{Ka}{\tau} u(t) \right) T_s \quad (14)$$

Recordando que a conversão do domínio do tempo contínuo para tempo discreto ( $z$ ) é dada por  $y(t) = y(z)$  e  $y(t+1) = zy(z)$ , logo a Equação 14 se transformada na Equação 15.

$$zY(z) = Y(z) - \frac{aT_s}{\tau} Y(z) + \frac{KaT_s}{\tau} U(z) \quad (15)$$

$$zY(z) - Y(z) + \frac{aT_s}{\tau} Y(z) = \frac{KaT_s}{\tau} U(z) \quad (16)$$

$$Y(z) \left( z - 1 + \frac{aT_s}{\tau} \right) = \frac{KaT_s}{\tau} U(z) \quad (17)$$

Desta forma se encerra a discretização da FT na Equação 7 do domínio de Laplace para o tempo discreto com a Equação 18.

$$\frac{Y(z)}{U(z)} = \frac{\left(\frac{KaTs}{\tau}\right)}{z + \left(-1 + \frac{aTs}{\tau}\right)} \quad (18)$$

Que em código pode se dar como:

$$Y(z)/U(z) = (K*a*Ts/tau)/(z+(-1+a*Ts/tau)) \quad (19)$$

Onde  $Y(z)$  é a saída da planta,  $U(z)$  o sinal de controle,  $K$  o ganho da FT,  $a$  pode-se dizer que representa a dinâmica da planta onde 'a' menor representa um planta com menos resistência à mudança,  $T_s$  é o intervalo de tempo entre cada amostra,  $\tau$  é a resposta ao degrau da FT, e sendo  $z$  parte do plano discreto apenas se utiliza o valor de sua constante que é um.

### 3.3 Da IDE para o ambiente de bibliotecas Arduino

Agora no ambiente de bibliotecas Arduino, ainda foram necessárias pequenas adaptações no código, ainda que tanto na IDE quando na biblioteca seja escrito em linguagem C++, os ajustes foram necessários, pois devido ao propósito educativo da biblioteca foram criadas subfunções para realizar cada parte do código.

De maneira geral, a biblioteca funciona da seguinte forma: Inicialmente o usuário escolhe um nome para o dispositivo utilizado e informa todas as informações básica para que o controle adaptativo funcione: Portas do Arduino utilizadas, dados da FT modelo, taxa de adaptação do MRAC e SP desejado.

A partir disso é possível utilizar subfunções para modificar cada um destes parâmetros independentemente (exceto as portas utilizadas no microcontrolador, já que não é esperado que o usuário troque as portas durante o funcionamento). Além de ter disponível 2 formas de observar os dados do sistema: Uma forma clara para leitura em texto no *Serial Monitor* do Arduino, e outra adaptada para apresentação em gráfico no *Serial Plotter* também na IDE do Arduino.

Para um melhor entendimento de como utilizar a biblioteca desenvolvida, observa-se a Figura 8 onde é possível ver o que é possível realizar com a biblioteca e o que é necessário para que ela funcione.

Para melhor visualização do Módulo de atualização de dados do controlador apresentando na Figura 8, estende-se na Figura 9 o que é possível modificar e como fazer.

Para melhor visualização do Módulo de apresentação de dados do controlador apresentando na Figura 8, estende-se na Figura 10 o que é possível expor e como fazer.

### 3.4 Arduino e módulo tacogerador

Como apresentado anteriormente, o módulo tacogerador desenvolvido por Souza et al. (ano) serviu como objeto de testes tanto do código inicial na IDE do Arduino, como para a biblioteca criada.

Utilizando-se então do microcontrolador Arduino Uno, foram utilizadas as mesmas conexões realizadas no trabalho de Souza et al. (ano) para conexão com o módulo: GND (abreviação para Aterramento, em inglês), para referência de 0 Volts; uma entrada analógica, para leitura da tensão

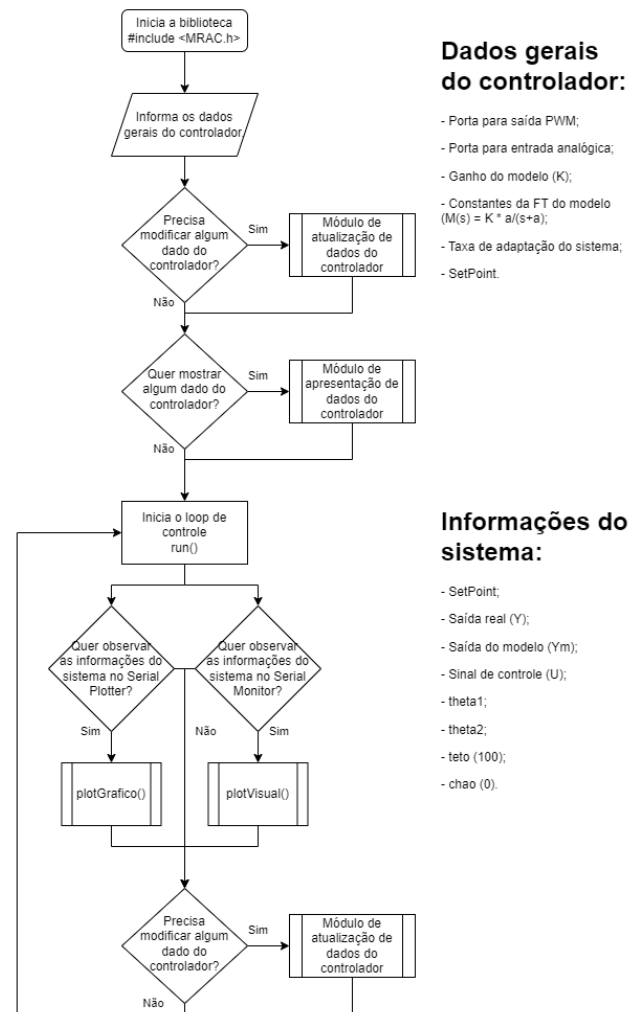


Figura 8. Fluxograma geral de uso da biblioteca. Fonte: De autoria própria.



Figura 9. Fluxograma do Módulo de atualização de dados do controlador. Fonte: De autoria própria.

do segundo motor; e uma saída PWM, para envio do sinal de controle de velocidade do primeiro motor, como mostra na Figura 7.

Para o uso da biblioteca é necessário informar: A constante no numerador na FT (Função de Transferência) modelo, a primeira e segunda constante no denominador da FT, a taxa de adaptação do MRAC, o SP desejado em porcentagem

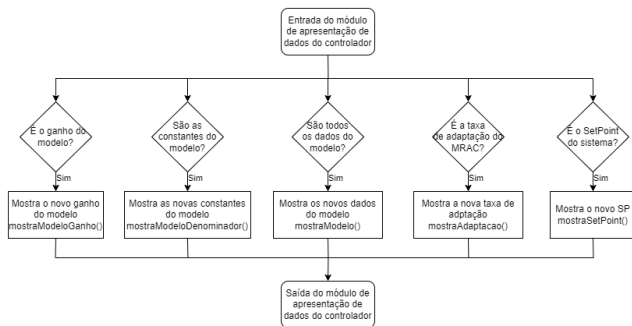


Figura 10. Fluxograma do Módulo de apresentação de dados do controlador. Fonte: De autoria própria.

gem da velocidade do motor, a porta utilizada para envio do sinal de controle de velocidade do primeiro motor e a porta utilizada para leitura da tensão gerada pela rotação do segundo motor.

## 4. RESULTADOS E CONCLUSÕES

Para fazer a comparação de performance entre o controle PI e o MRAC, utilizou-se do mesmo método apresentado por Mallick and Mondal (2019a) e Mallick and Mondal (2019b), pois dessa forma é possível observar quantitativamente qual controle manteve a velocidade mais próxima do SP.

Além disso, cabe aos futuros projetistas e leitores, a análise de contexto, por exemplo, o quanto de sobre-sinal é aceitável no caso de controle de nível de um tanque de água em que o SP chega próximo do limite máximo do reservatório. Ou seja, ponderar a respeito de qual comportamento do controle é mais adequado para cada situação.

Utilizando-se das Equações 3 à 6, que calculam o erro total do controle, apresentadas por Mallick and Mondal (2019a) e Mallick and Mondal (2019b) e aplicando no módulo tacogerador de Souza et al. (ano) foi possível chegar aos seguintes resultados:

### 4.1 Controle adaptativo

[APRESENTAR A PERFORMANCE DO CONTROLE ADAPTATIVO EM VARIOS PONTOS DE OPERAÇÃO (UMA UNICA IMAGEM MOSTRANDO O SP EM DEGRAIS INDO DE 0 A 100 E A SAIDA REAL, UMA IMAGEM PARA CADA VALOR DA TAXA DE ADAPTAÇÃO)]

Em relação o valor da taxa de adaptação, é um valor adimensional que vem da análise e teste do projetista, como explicou Tambara (ano). Observou-se que valores altos na taxa tornar o controlador muito agressivo no sinal de controle, se assemelhando a um controle *on-off*, enquanto valores baixos faziam com que o controlador demorasse mais para se adaptar.

Feita a análise chegou-se aos resultados da Tabela [REFERENCIA DA TABELA ABAIXO].

[TABELA DE PERFORMANCE DO MRAC]

### 4.2 Controle PI

[APRESENTAR A PERFORMANCE DO CONTROLE PI DO ARTIGO DO MODULO UTILIZADO EM VARIOS PONTOS DE OPERAÇÃO (UMA UNICA IMAGEM MOSTRANDO O SP EM DEGRAIS INDO DE ALGO ENTRE 0 A 100 E A SAIDA REAL)]

Feita a análise chegou-se aos resultados da Tabela [REFERENCIA DA TABELA ABAIXO].

[TABELA DE PERFORMANCE DO PI]

### 4.3 Comparação

Comparando a performance dos dois controles que se apresenta na Tabela [REFERENCIA DA TABELA ABAIXO], é possível observar que o MRAC se sobressai ao controle PI em pontos distantes do ponto de linearização utilizada para o PI, porém o MRAC precisa de ciclos para melhor se adaptar às rampas de subida e descida do SP, além disso é importante definir a taxa de adaptação corretamente, pois caso fique muito alta o controlador será muito agressivo que resulta em uma baixa performance.

[TABELA CONJUNTA DE PERFORMANCES MRAC E PI]

## 5. CONSIDERAÇÕES FINAIS

Esta pesquisa teve como propósito criar uma biblioteca para Arduino com o objetivo de facilitar o aprendizado do controle adaptativo, usando o método do MRAC direto pela regra do MIT. Neste estudo se percorreu desde os cálculos do MRAC em matemática direta, passando para a forma que pudesse ser utilizada em *softwares* de simulação de tempo limitado, como o MATLAB. Indo então para a adaptação do código de MATLAB para a IDE do Arduino e por final o ambiente de bibliotecas Arduino. Resultando nesta biblioteca que apresentou seus resultados comparando o esperado do modelo e o resultado real.

Apesar desta pesquisa ter como objetivo o viés educativo como uma biblioteca de Arduino, este mesmo trabalho pode ser utilizado como base em projetos independentes, MVPs (Mínimo Produto Viável, em inglês) e projetos de teste em empresas.

Durante o desenvolvimento deste projeto, observou-se partes que poderiam ser mais aprofundadas em trabalhos futuros. Como o estudo sobre a taxa de adaptação, que se demonstrou extremamente influenciadora na performance do controle; expandir a biblioteca para que seja possível utilizar outras formas do MRAC como o controle indireto e pelo método de Lyapunov, ou mesmo para outros controles adaptativos.

De maneira geral, este trabalho cumpriu seu objetivo de criar uma biblioteca que implementasse o MRAC no Arduino para ser apresentado de forma simples às pessoas que estão iniciando na área do controle adaptativo. Espera-se que muitos alunos possam se encantar com o controle adaptativo e a melhoria que isso pode trazer aos sistemas não-lineares pelo mundo.



## REFERÊNCIAS

- Arduíno (2021). About arduíno. URL <https://www.arduino.cc/en/about>. Último acesso em: 06 de Dez. de 2022.
- Astrom, K. (1987). Adaptive feedback control. *Proceedings of the IEEE*, 75, 185–217.
- Astrom, K. and Wittenmark, B. (1995). *Adaptive Control*. Addison-Wesley, Mineola.
- Bruce, J. (2021). *Indirect Model Reference Adaptive Control with online aircraft parameter estimation*. Master’s thesis, Embry-Riddle Aeronautical University, Daytona Beach, FL.
- Callai, T., Coelho, A., and Coelho, L. (2007). Controle nebuloso adaptativo por modelo de referência: projeto e aplicação em sistemas não-lineares. *Controle & Automação*, 18, 479–489.
- Campos, M. and Teixeira, H. (2010). *Controles típicos de equipamentos e processos industriais*. Edgard Blucher Ltda., São Paulo.
- Candy, J. (2021). Model reference adaptive control (mrac) for additive manufacturing. Technical report, Lawrence Livermore National Laboratory.
- Evans, M., Noble, J., and Hochenbaum, J. (2013). *Arduíno em ação*. Novatec, São Paulo.
- Fischman, A. (1993). *Controle robusto de sistemas lineares incertos - teoria e aplicações*. Master’s thesis, Universidade Federal de Santa Catarina, Florianópolis, SC.
- Kreisselmeier, G. and Anderson, D. (1986). Robust model reference adaptive control. *Transactions on automatic control*, 31, 127–133.
- Mallick, S. and Mondal, U. (2019a). Comparative performance study of lyapunov based mrac technique and mrac augmented with pid controller for speed control of a dc motor. *International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, 2.
- Mallick, S. and Mondal, U. (2019b). Performance study of different model reference adaptive control techniques applied to a dc motor for speed control. *International Conference on Trends in Electronics and Informatics (ICOEI)*, 3, 770–774.
- Manimozhi, M. and Rajathi, A. (2021). Design of mrac and modified mrac for dc motor speed control. *International Journal of Nonlinear Analysis and Applications*, 12, 1863–1871.
- Mushiri, T., Mahachi, A., and Mbohwa, C. (2017). A model reference adaptive control (mrac) system for the pneumatic valve of the bottle washer in beverages using simulink. *International Conference on Sustainable Materials Processing and Manufacturing*, 364–373.
- Nguyen, M., Vuong, D., and Nguyen, T. (2020). The mrac based - adaptive control system for controlling the speed of direct current motor. *Indonesian Journal of Electrical Engineering and Computer Science*, 19, 723–728.
- Pawar, R. and Parvat, B. (2015). Design and implementation of mrac and modified mrac technique for inverted pendulum. *International Conference on Pervasive Computing (ICPC)*.
- Sapiee, M. and Sudin, S. (2010). Road vehicle following system with adaptive controller gain using model reference adaptive control method. *International Journal of Simulation, Systems, Science & Technology (IJSSST)*, 11, 24–32.
- Silva, M. (2017). *Controle Adaptativo Aplicado a um Veículo Aéreo Não Tripulado*. Ph.D. thesis, Universidade Federal de Ouro Preto.
- Souza, A., Marinho, A., Santos, A., and Luna, J. (ano). Um módulo motor-tacogerador de baixo custo para ensino de controle automático. Artigo ainda não publicado.
- Souza, R. (2016). *Estratégia de controle adaptativo para estabilização de um quadricóptero na presença de variação de massa*. Master’s thesis, Universidade Federal de Uberlândia, Uberlândia, MG.
- Tambara, R. (ano). Controle adaptativo teoria e aplicação. Material didático da Universidade Federal de Santa Maria (UFSM), em Edição do Autor. Disponível em [https://intranet.ctism.ufsm.br/gsec/Publicacoes/Controle\\_Adaptativo\\_Prof\\_Rodrigo\\_1\\_02\\_19.pdf](https://intranet.ctism.ufsm.br/gsec/Publicacoes/Controle_Adaptativo_Prof_Rodrigo_1_02_19.pdf). Último acesso em: 16 de Ago. de 2022.
- Trofino, A. (ano). Sistemas lineares. Material didático da Universidade Federal de Santa Catarina. Disponível em [https://www.academia.edu/19191808/Apostila\\_Sistemas\\_Lineares](https://www.academia.edu/19191808/Apostila_Sistemas_Lineares). Último acesso em: 16 de Ago. de 2022.
- Xiong, A. and Fan, Y. (2007). Application of a pid controller using mrac techniques for control of the dc electromotor drive. *International Conference on Mechatronics and Automation*, 2616–2621.