

# Controle por modelo preditivo de velocidade para um motor de corrente contínua

A. Martins \* D. Xavier \*\* G. Chaves \*\*\* W. Oliveira \*\*\*\*

Departamento de Engenharia de Controle e Automação, Universidade Federal de Ouro Preto, MG

\* e-mail: anderson.martins@aluno.ufop.edu.br

\*\* e-mail: douglas.xavier@aluno.ufop.edu.br

\*\*\* e-mail: gabriela.chaves@aluno.ufop.edu.br

\*\*\*\* e-mail: wesley.santana@aluno.ufop.edu.br

---

**Abstract:** Classical control techniques have long been applied to systems that have a single input and output, however, with the emergence of modern control theory, it has become possible to control systems with multiple variables and/or variants over time. Model predictive control (MPC) has been used since the 1980s and has shown satisfactory results. One of its greatest advantages among the other methods is the fact that it allows a decrease in the response time taking into account future states. In this paper, two controllers are designed, a PID and a MPC, which will be applied to a DC motor. When comparing the two methods, it can be seen that, although both have short and satisfactory response times, for this project, the classical control method presents an over-signal in its response while the predictive presents greater rejection of disturbances, allowing the system to work in a softer way.

**Resumo:** Há muito tempo técnicas de controle clássico são aplicadas para sistemas que possuem uma única entrada e saída, no entanto, com o surgimento da teoria de controle moderna, tornou-se possível controlar sistemas com múltiplas variáveis e/ou variantes no tempo. O controle por modelo preditivo (MPC) vem sendo utilizado desde os anos 80 e apresentando resultados satisfatórios. Uma de suas maiores vantagens dentre os demais métodos, é o fato de permitir uma diminuição no tempo de resposta levando em consideração estados futuros. Neste trabalho são projetados dois controladores, um PID e um MPC, que serão aplicados a um motor CC. Ao comparar os dois métodos, pode-se perceber que, embora ambos apresentem tempos de resposta curtos e satisfatórios, para esse projeto, o método de controle clássico apresenta sobre-sinal em sua resposta enquanto o preditivo apresenta maior rejeição a distúrbios, permitindo que o sistema funcione de forma mais suave.

**Keywords:** Classical control, modern control, modeling; PID; MPC; response time.

**Palavras-chaves:** Controle clássico; controle moderno; modelagem; PID; MPC; tempo de resposta.

---

## 1. INTRODUÇÃO

O controle de sistemas físicos, econômicos, biológicos e sociais é um tema com grande impacto na sociedade e que vem sendo cada vez mais discutido com o avanço das tecnologias. Atualmente, com o aumento do número de máquinas presentes em um processo e com a criação de novos equipamentos, se faz cada vez mais necessário um controle preciso desses processos, afim de evitar erros que podem ser críticos para a atividade.

O controle de processos industriais é realizado por um controlador. Esse equipamento recebe os valores da variável que se deseja controlar em um processo e determina um sinal de correção a ser aplicado à planta em busca de corrigir e otimizar. Dentro de uma fábrica, esse equipamento é de vital importância pois faz com que o processo mantenha valores desejados na saída do sistema.

Sistemas de controle clássico funcionam muito bem em processos com uma única entrada e uma única saída (SISO) e invariantes no tempo, porém, muitas das vezes, devido a complexidade dos processos, esse tipo de controlador não obtém muito sucesso. Nesses casos mais complexos se vê necessário a utilização de sistemas de controle modernos, capazes de lidar com múltiplas variáveis e sistemas variantes no tempo.

O objetivo desse trabalho é projetar um controlador por modelo preditivo aplicado a um motor de corrente contínua, além disso, deseja-se analisar diferentes métodos de controle e compará-los.

## 2. REVISÃO BIBLIOGRÁFICA

Nessa seção serão avaliados trabalhos relevantes diretamente relacionados com o controle de sistemas, suas limitações e possibilidades.

Um estudo de caso foi realizado em Barbosa and Silvestre (2018) e teve como objetivo analisar o uso de diferentes estratégias de controle numa planta que controla nível e temperatura. O processo e as variáveis de controle são explicados inteiramente no trabalho, no entanto, o processo tem múltiplas entradas e múltiplas saídas (MIMO). As três estratégias analisadas foram o controle PID, o PI Fuzzy e o preditivo. Para exibir o desempenho dos métodos foram realizadas diversas simulações. Como conclusão o autor observou que todos os métodos chegaram ao setpoint, porém o controlador PI Fuzzy obteve o melhor desempenho seguido pelo controlador preditivo e depois o PID convencional.

No trabalho apresentado em Morales et al. (2013) foi aplicado um controle preditivo para o controle do processo de neutralização de pH. O estudo proposto nesse artigo analisa o controle preditivo aplicado a um processo não linear e que possui um alto tempo morto. Ao longo do artigo o autor aborda todas as etapas do projeto e dá ênfase à vantagem que o MPC tem ao trabalhar com sistemas que possuem restrições, tanto na entrada como na saída. No final do trabalho, ele testa o seu projeto no algoritmo matriz de controle dinâmico quadrática (QDMC) e, após realizar uma comparação com um controle PID, chega à conclusão de que o controle MPC apresenta um melhor resultado nos diferentes pontos de operação da planta.

Já no estudo realizado em de Oliveira and Lages (2006), o autor faz o uso do controlador MPC em um robô braçejador subatuado. Nesse trabalho o autor disserta sobre a dificuldade de se obter as expressões do modelo e, por isso, opta por utilizar a técnica de estimação on-line para modelar o sistema. Para o cálculo da função objetivo do robô, o autor utiliza a técnica de linearização, já que a mesma leva em consideração também a posição do efetuador final, em vez de considerar somente as coordenadas de junta. Ao final, o autor parte para a fase de testes, onde tem como objetivo controlar o movimento do robô que imita o movimento de um macaco. Como conclusão, o autor observou que o controle MPC alcançou resultados satisfatórios no seu objetivo, além de poder executar o movimento desejado por vários ciclos.

Limites das variáveis de controle podem ocorrer em variados sistemas e provocar uma piora no desempenho do sistema que não foi prevista durante o projeto do controlador. Em Bohn and Atherton (1995) os autores explicam e discorrem sobre o efeito *windup*. No trabalho, os autores dizem que quando há saturação no atuador, incrementos posteriores no sinal de controle não contribuirão para uma resposta mais rápida do sistema. Logo, a continuação da integração do erro do sistema faz com que o termo integral adquira valores elevados sem qualquer efeito sobre a saída da planta. Com isso, o erro deve ter sinal negativo durante um longo intervalo de tempo para que o termo integral possa ser trazido de volta ao estado estacionário, o que ocasiona um elevado sobre-sinal e um tempo de acomodação relativamente longo.

Para conseguir um desempenho satisfatório e reduzir o efeito citado acima, são desenvolvidas diversas técnicas chamadas *anti-windup*. Em NETO (2005) o autor estuda diferentes técnicas tais como: *Back Calculation*, técnica Condicional, técnica da integração condicional, técnica

Proposta por Visioli, integração limitada e *Feedforward*. A integração condicional consiste em manter a saída da parte integrativa constante quando o sinal de controle está saturado, caso contrário, essa saída será incrementada normalmente.

### 3. METODOLOGIA

Neste trabalho pretendemos estudar e aplicar técnicas de um Controlador por Modelo Preditivo ou *Model Predictive Control* (MPC) em um motor de corrente contínua, com a finalidade de controlar sua velocidade. Carvalho et al. (2015) utiliza esse tipo de controlador e obtém bons resultados, logo, desejamos desenvolver esse controlador no software MATLAB, realizando, a priori, uma simulação para posteriormente aplicar o controle no motor.

O primeiro passo para o desenvolvimento desse projeto consiste na montagem de um circuito eletrônico para acionamento de um motor de corrente contínua (CC) de 6V e comunicação com o MATLAB. Para o circuito eletrônico de acionamento, pretendemos utilizar um módulo de acionamento de motor de corrente contínua, que nos permitirá, de forma simples, acionar o motor através de um microcontrolador. Para o circuito de comunicação, utilizaremos o microcontrolador Arduino NANO, que, através do protocolo de comunicação serial, trocará informações com o MATLAB, enviando o valor de velocidade do motor captado pelo sensor e recebendo o valor do sinal de controle a ser aplicado na planta. Para a alimentação de todo esse circuito eletrônico, utilizaremos uma fonte de 5V.

Para a etapa de modelagem, Carvalho et al. (2015) utiliza métodos matemáticos tendo em vista que os autores têm conhecimento dos parâmetros físicos do motor necessários para a modelagem. No entanto, o fabricante do motor que será utilizado para esse trabalho não disponibiliza todos os parâmetros necessários para realizar a modelagem. Uma das possíveis abordagens para esse problema seria a utilização de experimentos e cálculos matemáticos para a obtenção dos parâmetros necessários para a modelagem. Um caminho mais simples, e que provavelmente será o escolhido, seria alcançar a modelagem desse sistema através de uma modelagem caixa cinza, utilizando métodos empíricos.

Uma vez obtida a função de transferência do sistema, torna-se necessário compreender melhor o controlador MPC. Como visto em Reis (2018), esse tipo de controlador tem dois principais parâmetros: horizonte de predição ( $H_p$ ) e horizonte de controle ( $H_c$ ).  $H_p$  refere-se à quantidade de predições que serão realizadas através do modelo,  $H_c$  é a quantidade de sinais de controle preditos que serão aplicados, comumente aplica-se apenas o primeiro sinal de controle, descartando-se todos os outros. Para alcançar esses parâmetros, existem diferentes caminhos, nesse trabalho realizaremos alguns testes com variados parâmetros, começaremos utilizando os parâmetros citados em Carvalho et al. (2015). Utilizamos para os testes  $H_p=10$  e  $H_c=1$ .

Dando continuidade no desenvolvimento do projeto, daremos início à etapa de simulação. Para isso, utilizaremos como ferramenta de simulação e modelagem o MATLAB. A princípio, montaremos a planta de controle no software, em seguida utilizaremos o controlador determinado

anteriormente e, a partir daí, realizaremos experimentos para alcançar valores mais efetivos para os parâmetros do controlador. No momento em que esses parâmetros forem escolhidos, serão realizadas outras simulações, com o intuito de criar uma base de dados para, posteriormente, comparar com os dados do projeto realizado na prática.

Para realizar o projeto na prática, utilizaremos o controlador no próprio MATLAB, que enviará o sinal de controle para o microcontrolador Arduino NANO que aplicará o sinal de controle no módulo de acionamento do motor através de um sinal de Modulação de Largura de Pulso ou *Pulse Width Modulation* (PWM). O microcontrolador também será responsável por interpretar os dados captados pelo encoder e enviar a leitura de velocidade do motor para o controlador no MATLAB. Toda comunicação será realizada através de um protocolo Transmissor/Receptor Universal Assíncrono ou *Universal Asynchronous Receiver/Transmitter* (UART).

Ao final, caso tudo corra dentro dos prazos pré-definidos, poderá ser realizado um controle Proporcional Integral e Derivativo (PID) aplicado ao mesmo projeto, com o intuito de analisar e comparar os resultados obtidos nos dois métodos de controle.

Para o desenvolvimento do projeto, serão necessários os seguintes materiais:

- Mini motor CC 6V com caixa de redução;
- Encoder AB magnético para micromotor com 12 pulsos/revolução;
- Fonte de tensão contínua 5V;
- Módulo de acionamento para motor CC L298N;
- Microcontrolador Arduino NANO;
- Protoboard 1580 furos;
- Jumpers.

Além disso, serão necessários os softwares:

- MATLAB;
- Arduino IDE;
- Proteus 8 Professional.

#### 4. DESENVOLVIMENTO

Para dar início ao trabalho, iniciamos o projeto do circuito de controle e potência. Com esse objetivo, utilizamos o software Proteus 8 Professional para projetar o circuito com os componentes idealizados anteriormente, uma demonstração do projeto pode ser visualizada nas figuras 1 e 2. A partir desse planejamento, demos início à montagem do circuito físico.

O circuito é composto por um *encoder* (figura 3) que tem como função fechar a malha de controle do sistema, trabalhando como um sensor de velocidade, esse componente está acoplado a um motor 6V (figura 4) que está conectado a um módulo de acionamento (figura 5) responsável por enviar o sinal do circuito de controle vindo do microcontrolador para o circuito de potência.

Na segunda etapa, evoluímos para a montagem do circuito projetado na protoboard, que pode ser visualizado na figura 6.

Em seguida, foi realizada a conexão do circuito com o computador. Para isso, utilizamos a IDE do Arduino para

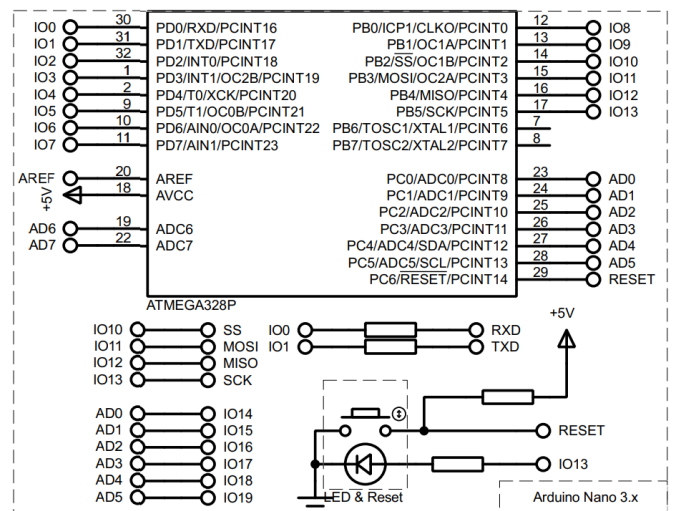


Figura 1. Diagrama do circuito elétrico.  
Fonte: Autoria própria.

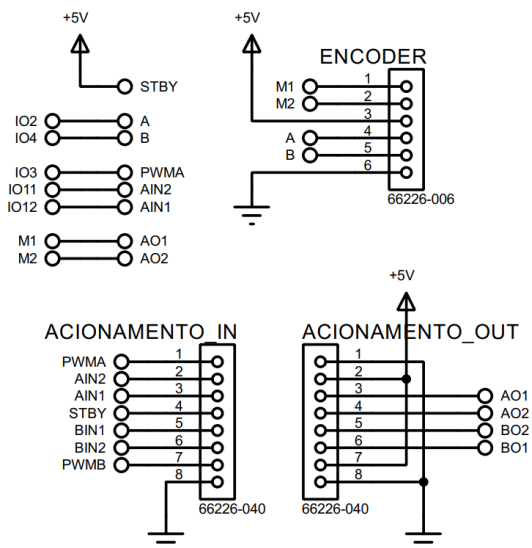


Figura 2. Diagrama do circuito elétrico.  
Fonte: Autoria própria.

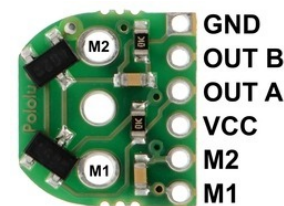


Figura 3. Encoder magnético Pololu.  
Fonte: Pololu.

iniciar a programação do microcontrolador Arduino Nano (figura 7). O código implementado foi utilizado para captar os dados de velocidade do motor em uma frequência de amostragem pré-determinada e envia-los para o MATLAB através da comunicação serial. O tempo de amostragem deve ser 8 vezes menor que o tempo de subida do sistema, dado isso escolhemos a frequência de amostragem igual a 100Hz e vimos que era suficiente para amostrar nosso

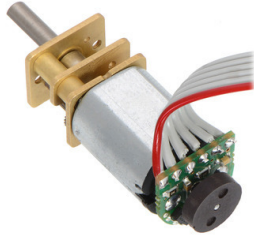


Figura 4. Motor com redução e encoder magnético.  
Fonte: Pololu.

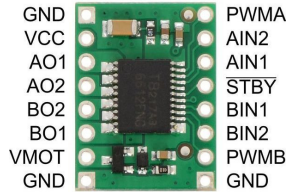


Figura 5. Módulo de acionamento.  
Fonte: Pololu.

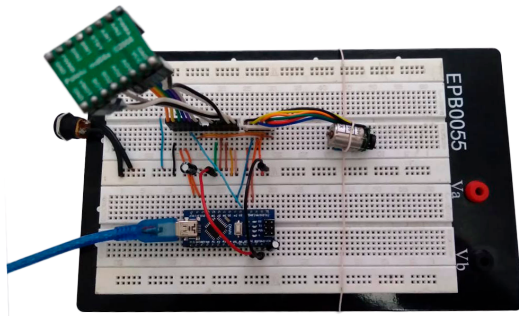


Figura 6. Circuito montado na protoboard.  
Fonte: Autoria própria.

sistema. Essa etapa do código é realizada através de uma interrupção por estouro do *timer*. A comunicação inversa, ou seja, do computador para a planta é outra etapa importante desse processo e o arduino é o responsável por receber as mensagens e aplicá-las no ciclo de trabalho (*duty cycle*) do PWM.

Com a comunicação estabelecida demos início à programação no MATLAB onde foi feita a captação dos dados do sensor para uma análise detalhada da malha de controle. Para a modelagem do sistema utilizamos o método da curva de reação proposto por Ziegler et al. (1942). Em resumo, o método consiste em aplicar um degrau com amplitude de 10% a 20% do fundo de escala ao sistema em malha aberta e a partir da curva de resposta obtida, traçar uma reta tangente ao seu ponto de inflexão, para então, obter os parâmetros atraso de transporte ( $L$ ), constante de tempo ( $T$ ) e ganho ( $K$ ) conforme mostrado na figura 8. Esse método aproxima o sistema da planta a um sistema de primeira ordem com atraso de transporte.

Após obter a curva de reação do motor (figura 9), conseguimos calcular os parâmetros  $K$ ,  $T$  e  $L$  e assim obter a função

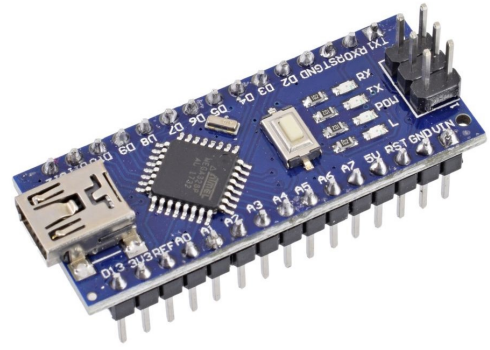


Figura 7. Arduino NANO.  
Fonte: Arduino.

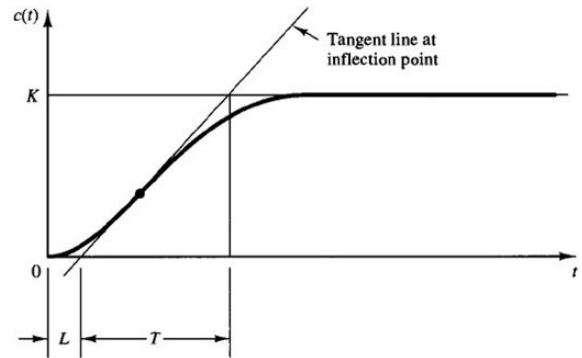


Figura 8. Curva de reação.  
Fonte: Universidade Gama Filho.

de transferência (1) referente ao nosso sistema. Com a função de transferência em mãos, foi possível executar simulações com o sistema modelado afim de obter respostas do sistema ao degrau. Através disso, realizamos a validação da modelagem, como mostrado na figura 10. Podemos perceber que a modelagem foi muito fiel ao sistema real.

$$G(s) = \frac{K e^{(-Ls)}}{Ts + 1}, L = 0.0172, K = 1.568, T = 0.06163 \quad (1)$$

Diferente do citado no capítulo de metodologia desse trabalho, primeiramente projetamos um controlador proporcional, integral e derivativo (PID) visto que temos maior proximidade, para depois iniciarmos os testes relacionados ao controle por modelo preditivo.

Dessa forma, iniciamos o projeto do controlador PID que é representado pela equação 2. Nessa etapa do projeto realizamos testes com diferentes métodos numéricos de sintonia de um controlador. O primeiro teste foi feito utilizando o método proposto em Ziegler et al. (1942), em seguida, testamos o controlador PID utilizando o método descrito em Cohen (1953). Em ambos os casos, adquirimos valores muito altos para os ganhos proporcionais ( $k_p$ ) e integrativos ( $k_i$ ) e, analisando a curva de resposta, percebemos que esses valores elevados não alcançaram bons resultados em nossa planta.

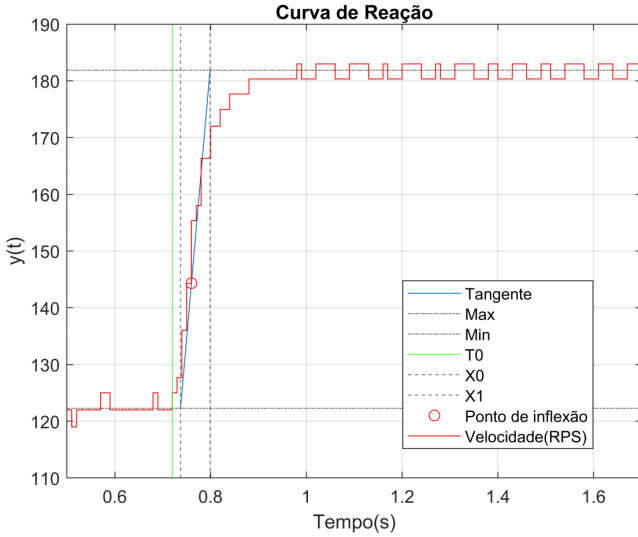


Figura 9. Curva de reação obtida do sistema real.  
Fonte: Autoria própria.

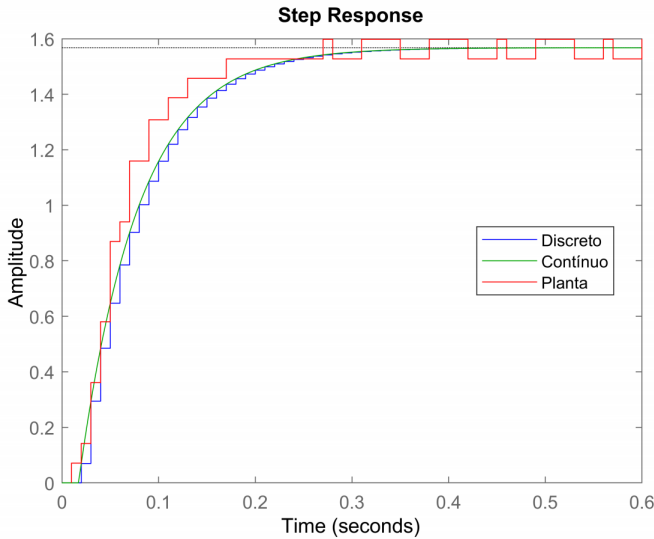


Figura 10. Validação do modelo.  
Fonte: Autoria própria.

Sendo assim, optamos por utilizar uma função de sintonia para controladores PID disponível no MATLAB, chamada "pidtune" e descrito em The MathWorks (1994-2021c). Essa função é utilizada para sintonizar controladores que serão utilizados em plantas lineares. Com isso, calculamos os ganhos e obtivemos valores menores, sendo eles  $k_p = 1,1186$ ,  $k_i = 19,3065$  e  $k_d = 0$ . Depois disso, implementamos no próprio MATLAB o controlador PID, que envia os valores da variável controlada para o arduino, que os envia para a planta e retorna ao MATLAB os valores captados pelo sensor.

$$C(t) = K_p + K_i \int e(t) dt + K_d \frac{\partial e(t)}{\partial t} \quad (2)$$

Após sintonizar e implementar o controlador PID, partimos para o projeto do MPC. Nesse passo, utilizamos outras duas funções também disponíveis pelo MA-

TLAB chamadas "mpc" e "mpcmove" que são descritas em The MathWorks (1994-2021b) e The MathWorks (1994-2021a) respectivamente. A função "mpc" é responsável por criar o objeto do MPC a partir da função de transferência da planta, do tempo de amostragem, do horizonte de predição ( $H_p$ ), do horizonte de controle ( $H_c$ ) e das limitações da variável de controle. O MPC não tem método de sintonia para os parâmetros  $h_p$  e  $h_c$ , por isso, através de testes variamos esses parâmetros até chegarmos a resultados satisfatórios. Já a função "mpcmove" é responsável por computar a ação de controle ótima no instante de tempo  $t$  e atualizar o estado do controlador, ela faz isso recebendo como parâmetros o objeto do controlador, o estado atual estimado no instante  $t$ , a saída medida pelo sensor também no instante  $t$  e o sinal de referência desejado. O sinal de controle e o valor da velocidade medida são enviados e recebidos da planta da mesma forma que foi feito no controlador PID.

É necessário enfatizar nesse momento do trabalho que ambas as estratégias de controle retornam valores reais para serem aplicados à malha. Contudo, esses valores foram tratados para que fossem inteiros e pudessem ser utilizados como (*duty cycle*) do PWM no arduino. Outro tratamento que tivemos que realizar está relacionado ao limite dessa variável de controle que deve variar entre 0 e 255. No PID, limitamos a variável de controle utilizando uma estrutura condicional, onde valores fora da faixa de atuação são ignorados. Essa restrição no sinal de controle provoca um efeito indesejado chamado *windup*, que foi corrigido utilizando uma técnica descrita em NETO (2005) chamada "Integração Condicional", que limita a atuação da parcela integrativa caso a saída do controlador esteja saturada. Já no MPC esse limite pode ser passado como parâmetro fazendo assim um controlador preditivo com restrições, o efeito *windup* nesse controlador não existe pois não temos a parcela integrativa como temos no PID.

## 5. RESULTADOS E DISCUSSÕES

Após a conclusão de todos os procedimentos descritos na seção anterior, já podemos rodar modelos simulados do nosso sistema com controlador PID e também MPC.

Para o controlador PID com os parâmetros calculados, obtemos a resposta mostrada no gráfico da figura 11. Podemos observar um atraso de transporte de 0,018s, sobre-sinal de 5,57% e, para um critério de até 2% do sinal de referência, o sistema estabiliza em 0,11s. Na figura 12, vemos a curva do sinal de controle aplicado na simulação.

Foi feito também a simulação do sistema com controlador MPC e a resposta do sistema e o sinal de controle estão mostrados na figura 13. Pode ser observado que, diferente do controlador PID, a resposta do sistema simulado com controlador MPC não possui sobre-sinal e o atraso de transporte é de 0,01s. Para um critério de 2%, o sistema estabilizou em 0,15s.

Apesar das pequenas diferenças nos resultados obtidos com a simulação dos dois sistemas, ambos são igualmente satisfatórios, já que apesar do MPC se mostrar um pouco mais lento, não possui sobre-sinal. A partir desse ponto de vista, a depender da aplicação, um ou outro controlador pode ser mais adequado. É importante ressaltar que, o



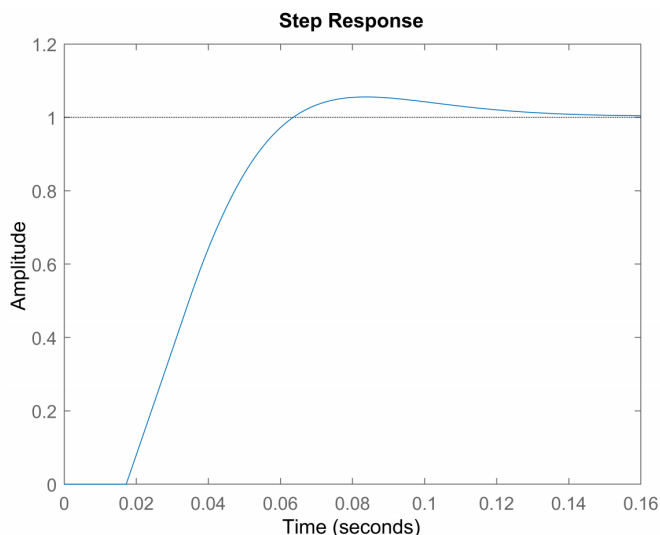


Figura 11. Resposta do sistema simulado com PID.  
Fonte: Autoria própria.

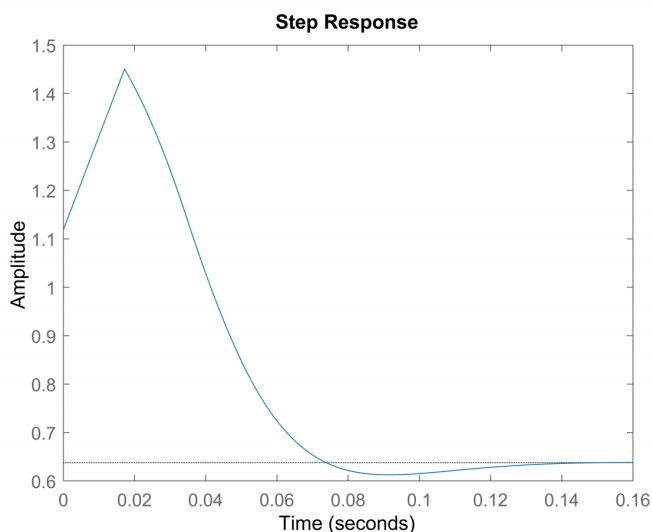


Figura 12. Sinal de controle do sistema simulado com PID.  
Fonte: Autoria própria.

projeto do controlador PID não leva em consideração a limitação de sinal de controle que temos na planta real, logo a resposta desse controlador provavelmente será prejudicada na prática.

Depois de analisarmos as simulações com os dois métodos de controle, vamos agora analisar se, na prática, os resultados são semelhantes. Utilizamos um sinal de referência variável que nos permite analisar melhor o comportamento do sistema em diferentes faixas de atuação.

A resposta do sistema com controlador PID, assim como o sinal de referência estão plotados na figura 14. De maneira geral o controlador funcionou bem em toda faixa de atuação analisada, mantendo a velocidade do motor bem próxima ao sinal de referência e o erro próximo a zero (figura 15). A variação observada no sinal de erro está mais associada à imprecisão do sensor de velocidade do que ao controlador propriamente dito. Porém, essa variação

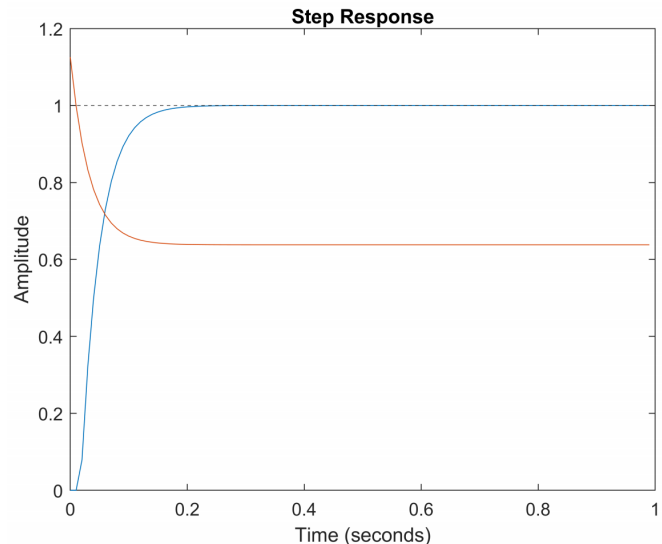


Figura 13. Simulação do sistema com controlador MPC.  
Fonte: Autoria própria.

nos impede de utilizar critérios de acomodação baixos como os utilizados na etapa de simulação. Sendo assim, adotamos um critério de até 8% do sinal de referência para encontrar o tempo de acomodação. É importante fixar um dos degraus aplicados para fazer essas análises, e, por isso, escolhemos o degrau aplicado no instante 9s. Dessa forma observamos que assim como o sistema simulado, a planta real também apresenta sobre-sinal, e nesse caso foi de 14,5%. O tempo de acomodação foi de 0,17s e o atraso de transporte de 0,03s.

Outro ponto que pode ser levado em consideração é o sinal de controle aplicado na planta (figura 16). Devido as características do controlador PID responder de acordo com o sinal de erro, o ruído presente no sinal de erro é transmitido para o sinal de controle e obtemos portanto um sinal igualmente ruidoso aplicado ao motor.

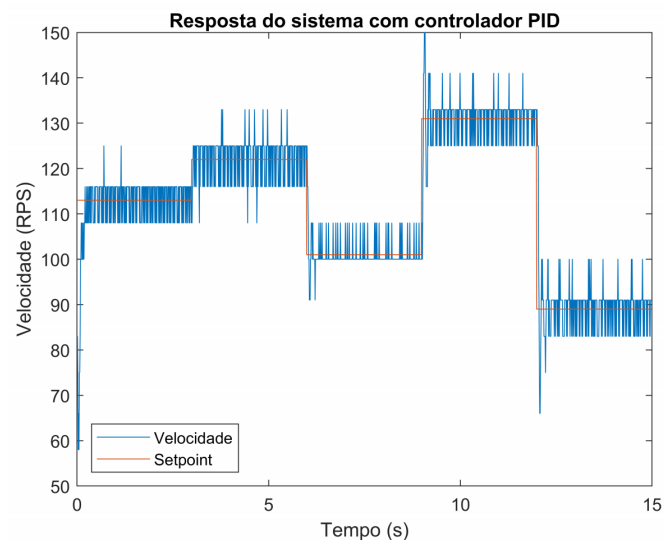


Figura 14. Resposta do sistema com controlador PID.  
Fonte: Autoria própria.

No gráfico da figura 17 referente à resposta do MPC e o sinal de referência, podemos ver que o controlador

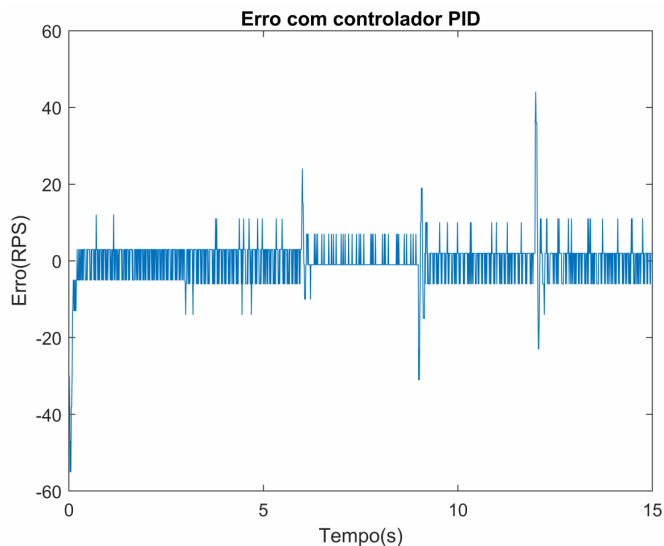


Figura 15. Erro do sistema com controlador PID.  
Fonte: Autoria própria.

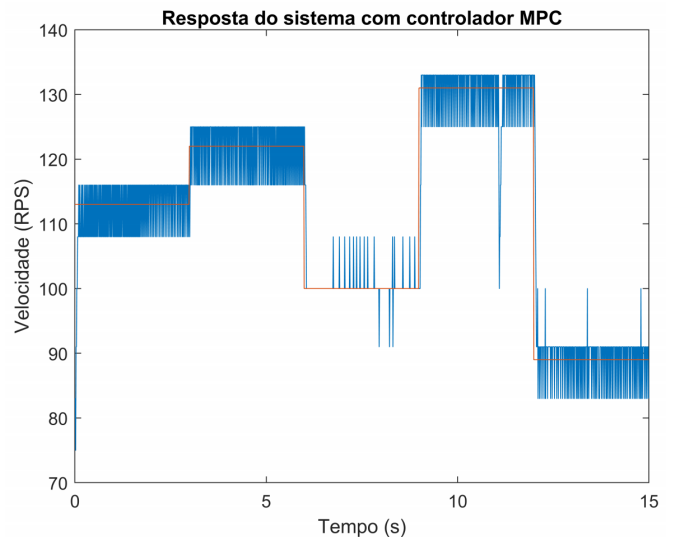


Figura 17. Resposta do sistema com MPC.  
Fonte: Autoria própria.

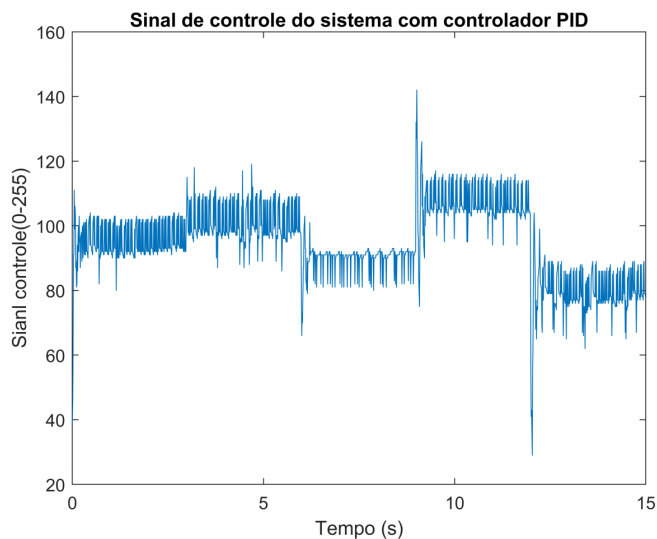


Figura 16. Sinal de controle do sistema com controlador PID.  
Fonte: Autoria própria.

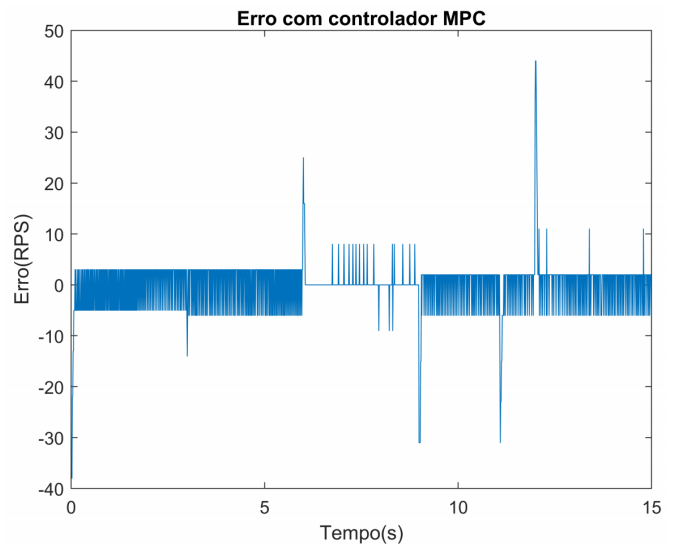


Figura 18. Erro do sistema com MPC.  
Fonte: Autoria própria.

## 6. CONCLUSÕES

agiu muito bem em toda faixa de atuação. Seguindo os mesmos critérios utilizados para analisar o controlador PID, percebemos que semelhante a sua simulação, o MPC não apresentou sobre-sinal, seu tempo de acomodação foi de 0,07s e o atraso de transporte de 0,04s. Devido as características do MPC, o sinal de controle (figura 19) se mostrou muito mais suave e menos ruidoso se comparado ao do controlador PID. Essa diferença mostra que o MPC funciona muito bem até mesmo com sistemas ruidosos, o que muitas das vezes é um problema para o controlador PID.

Outro ponto que pode ser observado sobre o comportamento do MPC, é sua capacidade de, através da comparação entre a resposta predita e a resposta real, fazer melhorias no modelo, tornando o controlador mais preciso a cada iteração. Isso pode explicar o porque do MPC ter se saído um pouco melhor na prática do que na simulação.

Neste trabalho foi realizado o desenvolvimento de um MPC, através do *software* MATLAB, com o intuito de se controlar a velocidade de um motor. O desenvolvimento deste projeto promoveu grandes aprendizados acerca do tema.

Depois de obtido a modelagem do sistema estudado neste projeto, foi possível projetar um controlador PID e, após isso, elaborar um MPC, com objetivo de comparar os resultados dos diferentes métodos.

Comparando o resultado obtido pelos dois métodos, pôde-se perceber que ambos apresentam tempos de resposta curtos e satisfatórios (PID = 0,17s e MPC = 0,07s). No entanto, o controlador PID apresentou um sobre-sinal de 14,5%. Ao analisar o MPC, percebeu-se que ele não apresenta sobre-sinal e possui uma maior rejeição a ruídos.

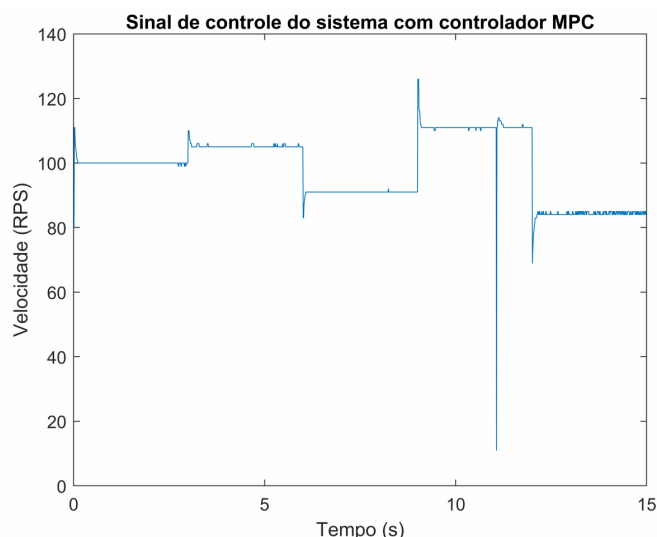


Figura 19. Sinal de controle do sistema com MPC.

Fonte: Autoria própria.

Concluimos então que é difícil realizar a comparação entre esses dois métodos, cada controlador deve ser utilizado analisando seu desempenho de acordo com determinada aplicação.

## REFERÊNCIAS

- Barbosa, G. and Silvestre, A. (2018). Estratégias de controle avaliadas a uma planta de nível e temperatura.
- Bohn, C. and Atherton, D.P. (1995). An analysis package comparing pid anti-windup strategies. *IEEE Control Systems Magazine*, 15(2), 34–40. doi:10.1109/37.375281.
- Carvalho, D., Calixto, W., Ganzaroli, C., Dias, R., and Couto, L. (2015). Controlador preditivo otimizado aplicado ao controle de velocidade de motor cc.
- Cohen, G. (1953). Theoretical consideration of retarded control. *Trans. Asme*, 75, 827–834.
- de Oliveira, V.M. and Lages, W.F. (2006). Controle preditivo de um robô bracejador subatuado utilizando linearização. In *XXVI Congresso da Sociedade Brasileira de Computação—III Encontro de Robótica Inteligente*, 83–92. Citeseer.
- Morales, C., Lourenço, L., and Garcia, C. (2013). Controle preditivo aplicado a uma planta piloto de neutralização de ph. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 1(1).
- NETO, A.H. (2005). *Técnicas anti-windup em estruturas de controle PID, RST e GPC*. Master's thesis, Universidade Federal de Santa Catarina, Florianópolis.
- Reis, L.A. (2018). *Controle preditivo por modelo de um circuito simulado de remoagem de minério de ferro*. Master's thesis, Universidade Federal de Ouro Preto, Ouro Preto.
- The MathWorks, I. (1994-2021a). Compute optimal control action and update controller states. [https://www.mathworks.com/help/mpc/ref/mpc.mpcmove.html?searchHighlight=mpcmove&s\\_tid=srchtitle](https://www.mathworks.com/help/mpc/ref/mpc.mpcmove.html?searchHighlight=mpcmove&s_tid=srchtitle). Acessado em: 08 abr. 2021.
- The MathWorks, I. (1994-2021b). Model predictive controller. [https://www.mathworks.com/help/mpc/ref/mpc.html?searchHighlight=mpc&s\\_tid=srchtitle](https://www.mathworks.com/help/mpc/ref/mpc.html?searchHighlight=mpc&s_tid=srchtitle). Acessado em: 08 abr. 2021.
- The MathWorks, I. (1994-2021c). Pid tuning algorithm for linear plant model. <https://www.mathworks.com/help/control/ref/lti.pidtune.html>. Acessado em: 08 abr. 2021.
- Ziegler, J.G., Nichols, N.B., et al. (1942). Optimum settings for automatic controllers. *trans. ASME*, 64(11).