

# Sistema didático de sintonia automática e auto-ajuste de controlador PID

José R. de Oliveira, Jr. René D. O. Pereira  
Bismark C. Torrico Fabrício G. Nogueira

\* *Departamento de Engenharia Elétrica, Universidade Federal do Ceará, CE, (e-mails: jose.raimundo@alu.ufc.br, reneolimpio@alu.ufc.br, bismark@dee.ufc.br, fnoqueira@dee.ufc.br).*

**Abstract:** This work presents a PID controller auto-tuning and self-tuning didactic system in the speed control of a direct current (DC) motor. The utilized techniques, the didactic system components and the obtained experimental results are presented. It is also discussed the application of the system in education.

**Resumo:** Este trabalho apresenta um sistema didático de sintonia automática de auto-ajuste de controladores PID no controle de velocidade de um motor de corrente contínua (CC). São apresentados as técnicas utilizadas, os componentes do sistema didático e os resultados experimentais obtidos. Também é discutida a aplicação do sistema na educação.

*Keywords:* Auto-tuning; self-tuning; PID control; DC motor; speed control.

*Palavras-chaves:* Sintonia automática; auto-ajuste; controle PID; motor CC; controle de velocidade.

## 1. INTRODUÇÃO

Em várias aplicações industriais o processo controlado pode mudar no decorrer do tempo. Para sistemas com controladores de parâmetros fixos, isso pode levar à deterioração de desempenho ou até mesmo à instabilidade do sistema. Para lidar com este tipo de problema, técnicas de controle adaptativo têm sido estudadas desde os anos 1950.

Dois tipos de áreas de controle adaptativo se destacam na indústria de controle: o auto-tuning (sintonia automática, em português) e o self-tuning (auto-ajuste, em português) de controladores.

Técnicas de sintonia automática de controladores são importantes no comissionamento de novas plantas, onde o operador apenas efetua um comando para realizar a sintonia inicial do controlador. Elas ficaram populares nos anos 1970 com o uso de computadores digitais na indústria, quando, por exemplo, o método de Ziegler-Nichols passou a ser realizado de forma automática.

O primeiro controlador auto-ajustável foi proposto em Kalman (1958). Mas somente nos anos 1970 esta técnica começaria a ter grandes avanços, com o trabalho de Åström and Wittenmark (1973).

Desse modo, o objetivo deste trabalho é apresentar um sistema didático de sintonia automática e auto-ajuste de controladores PID no controle de velocidade de um motor CC.

O artigo está organizado da seguinte forma: na Seção 2 é apresentado o método de sintonia automática, na Seção 3 é apresentado o método de auto-ajuste, na Seção 4 é apre-

sentado o sistema didático, na Seção 5 são apresentados os resultados experimentais e na Seção 6 as conclusões.

## 2. SINTONIA AUTOMÁTICA DO CONTROLADOR PID

A sintonia automática de um controlador PID pode ser realizada em duas etapas. Primeiro, realiza-se um experimento e, usando um método de identificação, obtém-se um modelo. Depois disso, este modelo é usado juntamente com um método de sintonia para se obter os parâmetros do controlador. Estas etapas e os métodos de identificação e sintonia do controlador adotados são apresentados a seguir.

### 2.1 Identificação em malha aberta

Inicialmente, é necessário realizar a identificação da planta de forma autônoma. Este processo pode ser realizado observando a resposta do sistema à uma entrada conhecida, como um impulso, um degrau, uma rampa, uma senoide ou uma Sequencia Binária Pseudo-Aleatória (SBPA).

Neste trabalho foi utilizado o método proposto em Ohta et al. (1979), baseado em resposta de entrada tipo degrau. O método de identificação será embarcado em um microcontrolador e, assim, são necessários baixo custo computacional e robustez a ruído de medição. O método de Ohta et al. (1979) utiliza apenas operações algébricas simples e integrais de sinais, que são resolvidas rapidamente pelo microcontrolador. Além disso, por ser baseado no cálculo de integrais, ele é robusto aos ruídos de média nula, pois a integral de um sinal com esse tipo de ruído é aproximadamente igual à integral de um sinal sem o ruído.

O Modelo a ser identificado é estável de primeira ordem com atraso de transporte, que pode ser descrito por três parâmetros: ganho estático  $K$ , constante de tempo  $T$  e atraso de transporte  $L$ . Sua função de transferência é dada por

$$P(s) = \frac{K}{Ts + 1} e^{-Ls}. \quad (1)$$

Após a aplicação do degrau na entrada  $u(t)$  e a saída  $y(t)$  atingir o regime permanente, inicia-se a identificação do modelo.

O primeiro parâmetro a ser obtido é o ganho estático, calculado por

$$K = \frac{y(\infty)}{A_u}, \quad (2)$$

onde  $y(\infty)$  é o valor final da saída e  $A_u$  é a amplitude do degrau.

Considera-se a soma da constante de tempo e do atraso de transporte como  $T_0 = T + L$ . Usando a integral  $A_0$ :

$$A_0 = \int_0^{\infty} \{y(\infty) - y(t)\} dt, \quad (3)$$

é possível obter

$$T_0 = \frac{A_0}{y(\infty)}. \quad (4)$$

A integral da saída de  $t = 0$  até  $t = T_0$  é definida por:

$$A_1 = \int_0^{T_0} y(t) dt. \quad (5)$$

A constante de tempo  $T$  é obtida então por

$$T = \frac{A_1}{e^{-1} \cdot y(\infty)} \quad (6)$$

e o atraso de transporte  $L$  por

$$L = T_0 - T. \quad (7)$$

## 2.2 Sintonia do controlador

Existem vários métodos de sintonia de controladores baseados em modelos de primeira ordem com atraso de transporte na literatura de sistema de controle. No entanto, simplicidade e baixo custo computacional são características desejáveis para métodos de sintonia automática. Portanto, neste trabalho, usa-se uma regra de sintonia como o método de sintonia do controlador PID.

Será utilizada a regra de sintonia proposta por Skogestad (2003). Segundo esta regra, para uma planta de primeira ordem com atraso de transporte, é projetado um controlador PI, representado por

$$C(s) = K_c \left( 1 + \frac{1}{T_i s} \right). \quad (8)$$

Os parâmetros do controlador  $K_c$  e  $T_i$  são obtidos por

$$K_c = \frac{T}{K(T_c + L)}, \quad (9)$$

$$T_i = \min\{T, 4(T_c + L)\}, \quad (10)$$

onde  $T_c$  é a constante de tempo de malha fechada desejada, um parâmetro livre de sintonia.

O compromisso entre agressividade da resposta e robustez depende da escolha do valor da constante de tempo de malha fechada  $T_c$ . Valores pequenos geram uma resposta agressiva, enquanto valores maiores geram maior robustez para o controlador. Em Skogestad (2003), é recomendado usar  $T_c = L$  como o melhor compromisso entre agressividade e robustez. No entanto, existe um limite físico mínimo da planta para a constante de tempo de malha fechada desejada. Dessa forma, devido ao atraso de transporte real do motor poder ser dezenas de vezes inferior a constante de tempo em malha aberta, utilizar  $T_c = L$  pode forçar o sistema a uma resposta proporcionalmente muito agressiva. Por isso, quando se trata de motores CC, é melhor relacionar  $T_c$  diretamente a características físicas da máquina, como a constante de tempo de malha aberta  $T$ .

## 3. AUTO-AJUSTE DO CONTROLADOR PID

Enquanto a sintonia automática utiliza a resposta de malha aberta do processo, o auto-ajuste de controladores utiliza a resposta de malha fechada do sistema de controle. Assim como a sintonia automática, o auto-ajuste acontece em duas etapas: identificação do modelo e sintonia do controlador. São descritos a seguir os métodos adotados em cada etapa.

### 3.1 Identificação em malha fechada

Identificação de modelos em malha fechada pode acontecer por meio de lote de dados ou recursivamente a cada período de amostragem. Neste trabalho, usa-se por simplicidade um método baseado em lote de dados.

O método escolhido foi proposto em Veronesi and Visioli (2010), é aplicável a controladores PID e é baseado em integrais de sinais e operações algébricas simples. Segundo este método, a identificação é realizada impondo uma mudança tipo degrau na referência  $r(t)$  e avaliando a resposta do sistema até atingir o regime permanente. O modelo a ser identificado é estável de primeira ordem com atraso de transporte.

O erro  $e(t)$  do sistema em malha fechada e sua integral são definidos, respectivamente, por

$$e(t) = A_r - y(t), \quad (11)$$

$$I_e = \int_0^{\infty} e(t) dt, \quad (12)$$

onde  $A_r$  é a amplitude do degrau na referência.

O ganho estático é então calculado por

$$K = \frac{T_i A_r}{K_c I_e}. \quad (13)$$

Considerando o sinal  $e_u(t)$  e sua integral, respectivamente dados por

$$e_u(t) = K u(t) - y(t), \quad (14)$$

$$I_{eu} = \int_0^{\infty} e_u(t) dt, \quad (15)$$

a soma da constante de tempo e do atraso de transporte é obtida por

$$T_0 = \frac{I_{eu}}{A_r}. \quad (16)$$

Segundo Veronesi and Visioli (2010), o atraso de transporte  $L$  pode ser obtido empiricamente como o tempo em que a saída atinge 2% do seu valor final ( $y(L) = 0.02 \cdot A_r$ ). Uma outra alternativa mais prática é usar uma banda de ruído  $NB$ , sendo o atraso o tempo em que a saída se torna maior que a banda de ruído, desse modo  $y(L) \geq NB$ .

De posse do atraso de transporte, a constante de tempo pode ser calculada por

$$T = T_0 - L. \quad (17)$$

### 3.2 Sintonia do controlador

O método de Skogestad (2003) também será utilizado para o auto-ajuste do controlador PID. Portanto, serão utilizadas (9) e (10) para sintonia de um controlador PI.

## 4. SISTEMA DIDÁTICO DE IDENTIFICAÇÃO E CONTROLE DE VELOCIDADE DE MOTOR CC

O sistema didático desenvolvido consiste de um hardware baseado em microcontrolador e uma interface gráfica em computador. Como todo sistema de controle, ele contém um sensor (*encoder*) para medição da variável controlada, um controlador embarcado em microcontrolador e comunicação para permitir definição da referência e obtenção dos sinais de entrada e saída do processo.

Para desenvolvimento do *firmware* do microcontrolador e da interface gráfica, utilizou-se um computador com sistema operacional Microsoft Windows 10, com acesso à tecnologia de comunicação sem fio *Bluetooth*<sup>®</sup>, com os softwares MPLAB X IDE<sup>®</sup>, PICKit 3 e MATLAB<sup>®</sup>, além de um gravador de microcontroladores PICKit 3 da Microchip. Para uso do sistema didático é necessário apenas um computador com sistema operacional Microsoft Windows<sup>®</sup>.

### 4.1 Hardware e firmware

O controlador foi implementado através de *firmware* embarcado em um microcontrolador dsPIC30F4011 da Microchip. A escolha do microcontrolador foi feita em concordância com os conhecimentos prévios dos alunos que irão utilizar o sistema. Além disso, este microcontrolador é voltado a aplicações de controle de motores, possuindo diversos módulos especiais para estas aplicações. Entre os módulos especiais, foram utilizados o módulo de codificador em quadratura de *encoder* diferencial na forma de contador de 16-bit e o módulo de Modulação de Largura de Pulso (PWM, do inglês *Pulse Width Modulation*). Além destes, foram utilizados o módulo temporizador, o módulo de comunicação Receptor/Transmissor assíncrono universal (UART, do inglês *Universal Asynchronous Receiver Transmitter*) e o módulo controlador de portas.

O Hardware do sistema didático é mostrado na Fig. 1. Além do microcontrolador, outros componentes utilizados são descritos a seguir:

- *driver* de motor CC MC33926 *Motor Driver Carrier* da Pololu Corporation
- Motor CC: tensão nominal de 12 V, velocidade nominal de 80 rpm, caixa de transmissão 131,25:1, torque

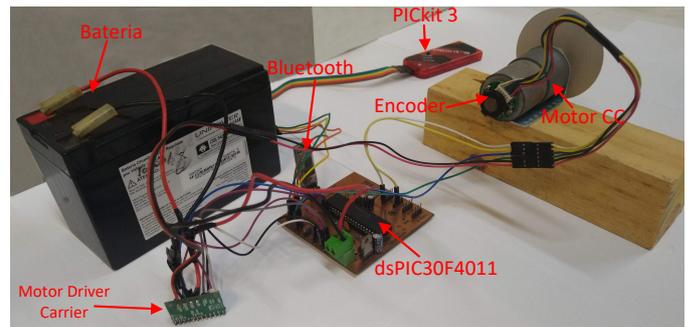


Figura 1. Hardware do sistema didático.

em rotor bloqueado de 1,059 Nm e *encoder* em quadratura com resolução de 64 pulsos por revolução;

- Bateria Chumbo Ácido Regulada por Válvula de 12 V e 7 Ah;
- Placa de circuito para uso do dsPIC30F4011 e periféricos
- Módulo *Bluetooth*<sup>®</sup> HC-06

No *firmware* estão incluídos os algoritmos de comunicação com a interface gráfica, de identificação (malha aberta e malha fechada), de sintonia e do controlador PI discreto.

O controlador PI embarcado é uma equação de diferenças resultante da expressão (8) discretizada pelo método de Tustin. O tempo de amostragem escolhido foi  $T_s = 10ms$ , pois a partir do modelo obtido por Pereira and dos Reis (2014), tem-se que a constante de tempo da planta de controle de velocidade do motor CC é aproximadamente dez vezes o valor de  $T_s$ . O atraso de transporte original da planta é ainda menor que  $T_s$ . Assim, para também se avaliar a identificação deste parâmetro do modelo, foi incluído via *firmware* um acréscimo de atraso de quatro períodos de amostragem.

Com o intuito de reduzir o custo computacional e evitar conversões de unidades desnecessárias, internamente no *firmware* utilizou-se como medida de velocidade a quantidade de pulsos de *encoder* por período de amostragem e como entrada do processo o valor do registrador do módulo de PWM. Desse modo, considerando-se valores nominais do motor CC, 80 rpm equivalem a 112 pulsos/ $T_s$  e 12 V equivalem a 1000 unidades de registrador do módulo de PWM. Na comunicação com a interface gráfica, valores de referência e sinais de entrada e saída e estão nestas unidades.

### 4.2 Interface gráfica

A interface gráfica foi desenvolvida utilizando o Ambiente de Desenvolvimento de Interface Gráfica com Usuário (GUIDE, do inglês *Graphical User Interface Development Environment*) disponível no MATLAB<sup>®</sup>. Ela torna a interação do usuário com o sistema de controle bastante simples, pois permite a utilização completa do sistema sem precisar compreender o protocolo de comunicação, os algoritmos embarcados, a configuração do microcontrolador, o processamento dos dados e etc.

O programa inicia com apenas o painel de comunicação Bluetooth e seus respectivos botões habilitados, além do quadro de avisos, com mensagens de sucesso ou de erro.

A Figura 2 mostra a interface gráfica ao inicializar tendo configurado apenas o *Bluetooth*<sup>®</sup>.

Após estabelecida a comunicação com o microcontrolador, obtém-se acesso ao painel Funções Auto-Tuning. Ele comanda o ensaio de resposta ao degrau com o PWM configurado em 66,6%, salvando a resposta de velocidade na memória interna do microcontrolador. A partir da resposta salva são calculados os parâmetros do modelo de primeira ordem com atraso de transporte. Por fim, a resposta de velocidade e os parâmetros do modelo são enviados para a interface gráfica, onde é possível visualizar e plotar os gráficos em seus eixos.

Ao concluir a identificação, a interface habilita o painel Configurar Controle. Nele, ajusta-se o valor da constante de tempo de malha fechada desejada  $T_c$ , que determina o compromisso entre agressividade e robustez da sintonia do controlador. Há um *Slider* que permite a variação entre o valor mínimo de  $0,3T$  e máximo de  $2T$  para o valor de  $T_c$ . Uma vez escolhido o valor de  $T_c$ , o *firmware* sintoniza o controlador PI. A referência é escolhida a partir de uma caixa texto e permite valores entre -80 e 80 rpm. Antes de enviar a referência para o microcontrolador, a interface gráfica converte seu valor para a unidade de pulsos/ $T_s$ . Após a realização do ensaio em malha fechada com o controlador obtido pelo auto-tuning, os sinais de controle e velocidade são plotados na interface.

Logo após, o painel Funções Self-Tuning é habilitado. Este painel comanda a identificação de um novo modelo utilizando os sinais do primeiro ensaio em malha fechada. A partir deste modelo, é sintonizado um novo controlador e realizado um novo ensaio de malha fechada. Por fim, o painel permite plotar o resultado deste ensaio de malha fechada junto com o ensaio anterior no mesmo gráfico. Além disso, calcula-se a integral do erro absoluto (IAE, do inglês *Integrated Absolute Error*) das respostas de velocidade, um índice de desempenho para comparação objetiva dos dois resultados.

Também é importante comentar que existe um painel que condensa as informações obtidas no decorrer da operação de toda a interface. Nele consta de forma visual os dois modelos de planta identificados e os dois controladores sintonizados.

Tem-se ainda o painel Tempo Real, que consiste em um sistema supervisorio simples operando com o último controlador sintonizado. Por simplicidade, esse sistema atualiza a plotagem da resposta de velocidade a cada segundo, adicionando ao gráfico as última 100 amostras acumuladas. Nele, existe a possibilidade de alterar a referência a qualquer momento.

Por fim, há um campo que possibilita salvar todos os dados manipulados dentro da interface gráfica. Nela, digita-se o nome do arquivo e, ao pressionar o botão Salvar Resultados, é gerado um arquivo com este nome e extensão “.mat” na pasta corrente do MATLAB<sup>®</sup>.

## 5. RESULTADOS EXPERIMENTAIS

Será apresentada uma seqüência típica de experimentos usando a interface gráfica e com o motor operando sem carga. Como resultados serão apresentados os modelos

identificados, os controladores obtidos e índices de desempenho das respostas de velocidade. No texto, variáveis com índice 1 referem-se ao auto-tuning e com o índice 2 ao self-tuning.

O primeiro ensaio foi realizado utilizando o painel Funções Auto-Tuning. Como pode ser visto na Fig. 3, o modelo obtido pelo ensaio de resposta ao degrau em malha aberta foi destacado em verde e os parâmetros calculados foram inclusos. Este modelo é apresentado abaixo:

$$P_1(s) = \frac{0.1156}{0.0991s + 1} e^{-0.05s}. \quad (18)$$

Observa-se também na Fig. 3 a resposta de velocidade em rpm e o degrau de entrada aplicado em unidades de registrador do módulo PWM.

Posteriormente, foi utilizado o painel Configurar Controle, onde é definida a constante de tempo de malha fechada  $T_c$  em função do valor da constante de tempo de malha aberta  $T$ . Foi escolhido  $T_c = 0.8T$ . O controlador é então sintonizado usando os parâmetros do modelo  $G_{p1}$  e o valor de  $T_c$ , sendo apresentado na interface. Os parâmetros calculados do controlador PI foram:

$$K_{c1} = 6.9004, T_{i1} = 0.0991. \quad (19)$$

Ainda no painel Configurar Controle, foi escolhida a referência de 40 rpm para o ensaio de malha fechada utilizando o controlador recém sintonizado pelo método de auto-tuning. Os sinais de controle e de velocidade deste ensaio podem ser vistos na Fig. 4 em linha de cor preta.

Sequencialmente, no painel Funções Self-Tuning, obtém-se um novo modelo baseado na resposta ao degrau de malha fechada anterior. Este modelo pode ser visto na Fig. 4 e é dado por:

$$P_2(s) = \frac{0.1093}{0.0942s + 1} e^{-0.05s}. \quad (20)$$

Baseado no modelo acima, é realizada uma nova sintonia para o controlador PI, obtendo-se

$$K_{c2} = 6.9332, T_{i2} = 0.0942. \quad (21)$$

Assim, é realizado um outro ensaio de malha fechada usando os parâmetros (21) do controlador PI calculados pelo método de self-tuning. Seus resultados são mostrados na Fig. 4, onde as respostas de velocidade e sinal de controle estão em linha de cor vermelha.

Ao final dos experimentos, são mostrados na interface gráfica os valores de IAE de cada resposta em malha fechada:

$$IAE_1 = 5.4643, IAE_2 = 5.2071. \quad (22)$$

### 5.1 Discussão

Didaticamente, como se tratam de técnicas com abordagens diferentes de identificação (malha aberta e malha fechada), o objetivo do experimento não é determinar qual delas é a melhor. O objetivo é mostrar aos alunos que as técnicas de auto-tuning e self-tuning utilizadas produzem resultados semelhantes. Isso pode ser constatado observando os resultados em termos dos parâmetros dos modelos identificados, dos controladores obtidos, das respostas dos ensaios e dos índices de desempenho, que são muito similares para ambos métodos. Portanto, educacionalmente, é importante ressaltar aos estudantes que estas

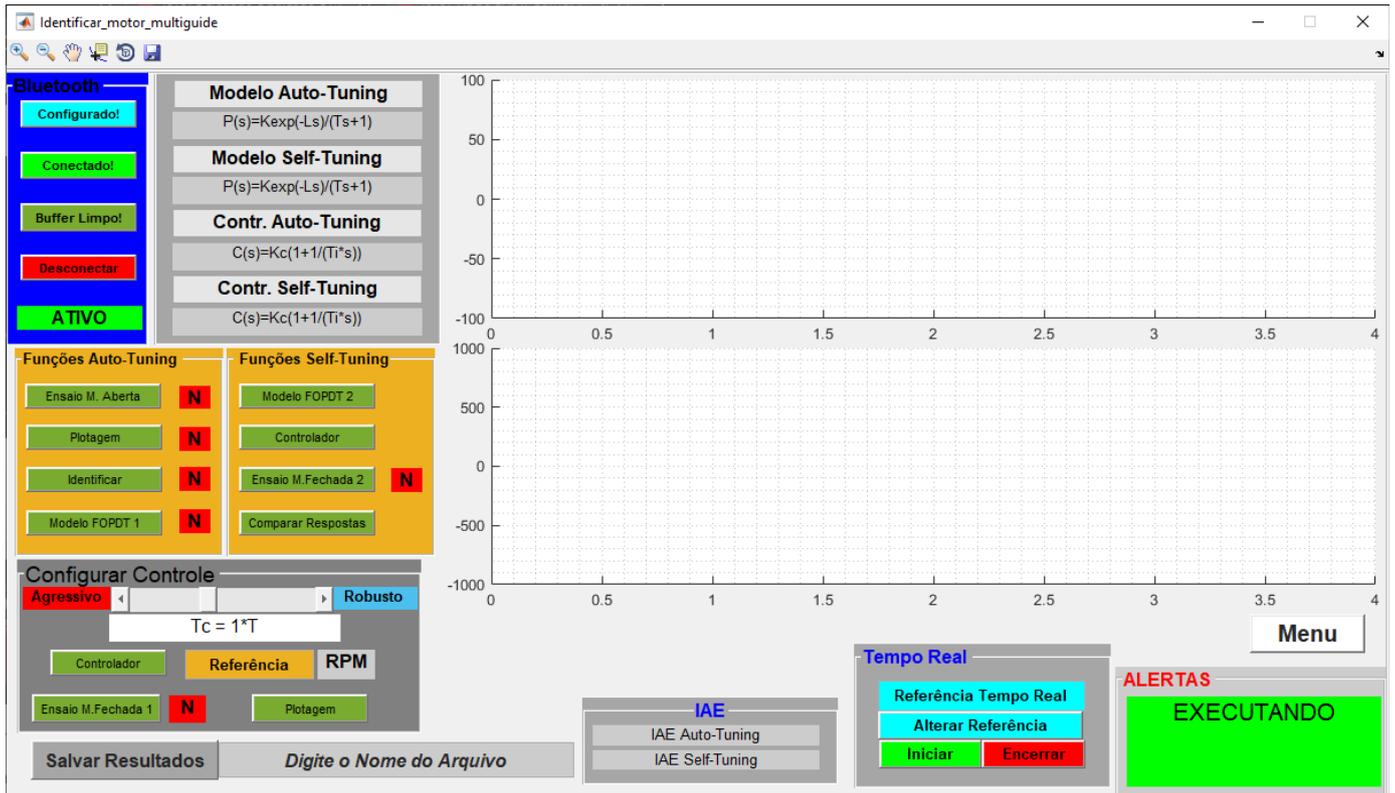


Figura 2. Visão geral da interface gráfica.

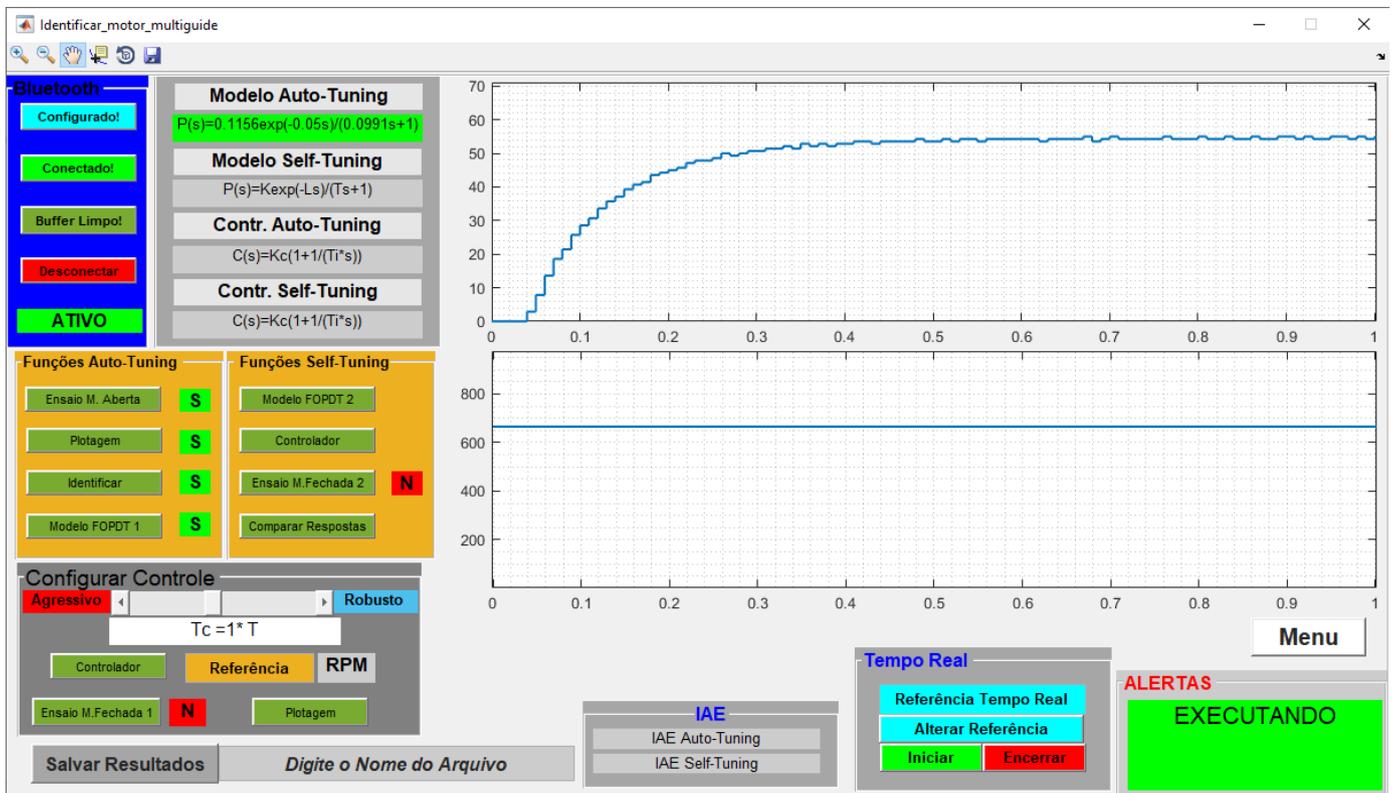


Figura 3. Interface gráfica em ensaio de malha aberta.

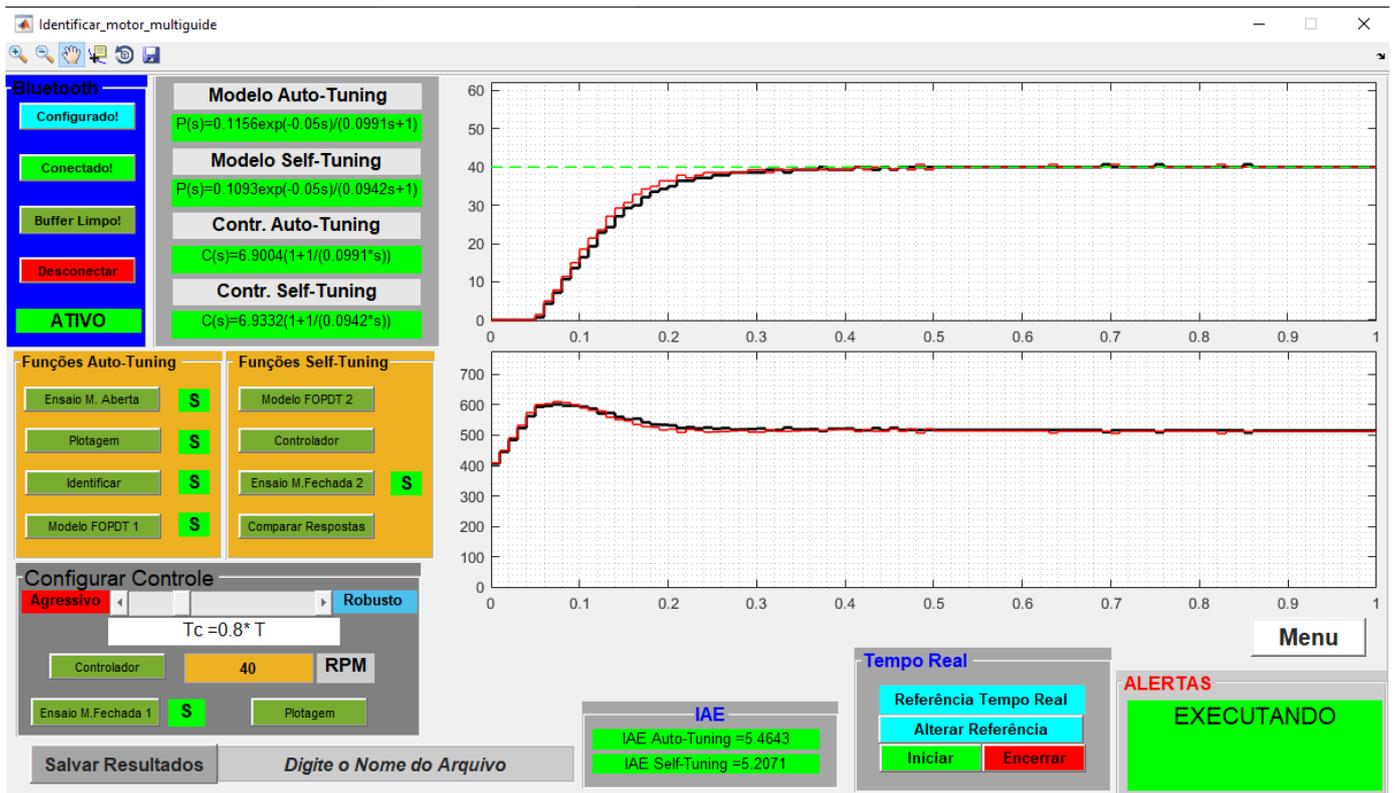


Figura 4. Interface gráfica com resultados de ensaios de auto-tuning e self-tuning.

técnicas podem, na verdade, ser utilizadas em conjunto, em diferente momentos para o sistema de controle. Como, por exemplo, o auto-tuning pode ser usado no momento de comissionamento de uma nova planta, enquanto o self-tuning durante a operação de uma planta já existente.

## 6. CONCLUSÃO

O sistema didático desenvolvido torna simples a realização de experimentos de auto-tuning e self-tuning de controlador PID em uma planta de controle de velocidade de motor CC. Ele condensa várias informações de um sistema de controle em uma única ferramenta. Além disso, o sistema didático dá aos alunos uma dimensão da interdisciplinaridade que envolvem as aplicações de controle.

O conjunto de dados obtido pela interface gráfica pode ser salvo, permitindo a reutilização desses dados didaticamente. Assim, cada etapa dos processos de auto-tuning e self-tuning pode ser reproduzida pelo aluno por meio de simulações. Isso torna o sistema didático uma ferramenta eficaz no ensino e na consolidação de conceitos básicos da teoria de controle, como identificação de sistemas e sintonia de controladores.

Como trabalho futuro, será construído um sistema de carga variável acoplado ao eixo do motor. Desse modo, será possível aos alunos obter diferentes modelos tanto em malha aberta como em malha fechada. Além disso, será possível perceber a deterioração das respostas de malha fechada após a mudança da planta e também as vantagens da aplicação de self-tuning como solução para o problema.

Outra importante melhoria será a implementação de uma nova interface gráfica usando ferramentas de software livre,

como as baseadas em linguagem de programação Python. Isso reduzirá os custos do projeto e permitirá aos alunos o uso e o desenvolvimento de forma independente.

## AGRADECIMENTOS

Os autores agradecem à CAPES e à FUNCAP pelo financiamento desta pesquisa.

## REFERÊNCIAS

- Åström, K.J. and Wittenmark, B. (1973). On self tuning regulators. *Automatica*, 9(2), 185 – 199.
- Kalman, R.E. (1958). Design of self-optimizing control system. *Trans. ASME*, 80, 468–478.
- Ohta, T., Tanaka, H., Tanaka, K., Sannomiya, N., and Nishikawa, Y. (1979). A New Optimization Method of PID Control Parameters for Automatic Tuning by Process Computer. *IFAC Proceedings Volumes*, 12(7), 133–138.
- Pereira, R.D.O. and dos Reis, L.L.N. (2014). Avaliação de desempenho de controladores PID 2-DOF em controle de velocidade de um motor CC. In *Anais do XX Congresso Brasileiro de Automática*, 4151–4158. Belo Horizonte.
- Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13(4), 291–309.
- Veronesi, M. and Visioli, A. (2010). An industrial application of a performance assessment and retuning technique for PI controllers. *ISA Transactions*, 49(2), 244–248.