



## PLATAFORMA DIDÁTICA PARA O ENSINO DE CONTROLE

Christopher Sauer<sup>1</sup>, Elisiane Pelke Paixão<sup>2</sup>, João Vitor das Chagas Silva<sup>3</sup>,  
Manuel M. P. Reibold<sup>4</sup>

<sup>1</sup>UNIJUÍ/DCEENG, kikisauer@gmail.com

<sup>2</sup>UNIJUÍ/DCEENG, elisianep251@gmail.com

<sup>3</sup>UNIJUÍ/DCEENG, joaovitor753pw@gmail.com

<sup>4</sup>UNIJUÍ/DCEENG, manolo@unijui.edu.br

**RESUMO:** O Controle de Sistemas destaca-se por ser uma área de grande relevância científica, propondo soluções para as mais diversas áreas tecnológicas. O alto grau de abstração no sistema de ensino atual da disciplina, no entanto, dificulta o entendimento por parte dos estudantes. Este trabalho apresenta a proposta de uma plataforma didática que visa ao aprimoramento do ensino, fazendo a conexão entre teoria e prática. A metodologia se desenvolve a partir da construção da plataforma e implementação de um controlador PID, e posterior análise dos resultados por meio de uma interface gráfica. Os resultados obtidos são uma contribuição ao sistema de ensino da academia.

**Palavras-chave:** controle, ensino, PID.

### 1 INTRODUÇÃO

O presente trabalho trata do ensino de Controle de Sistemas na universidade, através do uso de uma plataforma didática. A disciplina destaca-se por ser uma área de grande relevância científica, propondo soluções para as mais diversas áreas tecnológicas. Dessa forma, torna-se essencial que os alunos de cursos de Engenharia dominem tal conhecimento. Isso pressupõe o ensino, não apenas da teoria, mas também de sua aplicação. Uma plataforma didática surge como auxílio importante no contexto do ensino.

A plataforma consiste em uma gangorra com motores em cada extremidade. A variação da velocidade dos motores varia o ângulo da gangorra, sendo possível selecionar um ângulo desejado e observar a resposta do sistema conforme o ajuste acontece. A análise da resposta e de suas características permite compará-la a parâmetros de desempenho ideais.

No sistema atual de ensino, a disciplina de Controle trabalha com um alto grau de abstração. Isso dificulta o entendimento por parte dos estudantes da real utilidade do tópico de seu estudo, ficando a teoria desconectada da prática. O uso de uma plataforma didática fará a conexão abstrato-concreto, o que de forma retroativa irá melhorar o entendimento da disciplina.

Nesse contexto, este trabalho investiga a implementação de um controlador PID (Proporcional, Integral e Derivativo) em uma plataforma didática, visando o estudo da resposta do sistema. A estratégia passa pelo uso de uma interface gráfica que permite a observação e análise do resultado. A contribuição deste trabalho está em facilitar o entendimento dos estudantes de Controle, promovendo a motivação e melhorando o desempenho dos mesmos na disciplina.

O texto deste artigo está organizado da seguinte forma: na seção 2 estabelecem-se conceitos básicos de controle e do controlador PID. A seção 3 contém a metodologia, a qual mostra a estrutura da plataforma e a implementação do controlador. Na seção 4 são apresentados os resultados e na seção 5 sua discussão. Por fim, a seção 6 traz as conclusões e perspectivas.

## 2 CONTROLADORES PID

A finalidade de um sistema de controle é manter um sinal específico em sua saída, ou seja, sobre a planta ou processo que está sendo controlado. Um sistema de controle em malha fechada é aquele que possui realimentação - o sinal da saída é realimentado para o controlador, e este aplica o sinal de controle até que a saída atinja o valor desejado. Dessa forma, um sistema de controle em malha fechada possibilita a constante correção do sinal aplicado à planta.

Há várias maneiras de se implementar um controlador em malha fechada, sendo que o controlador de três termos, ou PID, é o mais utilizado na indústria. Trata-se de um controlador composto por três termos aos quais o sinal de erro é aplicado: proporcional, integral e derivativo. A forma mais comum de descrever o controlador PID é através da expressão (1):

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

Onde:  $u(t)$  é o sinal de controle,  $e(t)$  é o sinal de erro e  $K_p$ ,  $K_i$  e  $K_d$  são os ganhos de cada termo. Cada um dos termos do controlador contribui individualmente para o sinal de controle final: o termo proporcional opera sobre o valor atual do erro, o termo integral sobre o valor passado do erro e o termo derivativo sobre a taxa de variação do erro, ou seja, seu valor futuro. A

representação do controlador PID em diagrama de blocos é ilustrada na Figura 1.

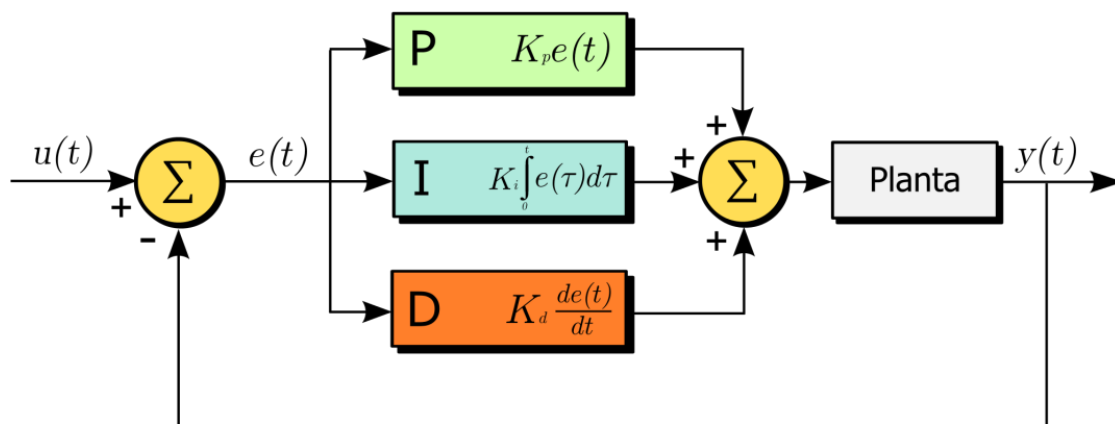


Figura 1 – Diagrama de blocos do controlador PID

Um conceito essencial no estudo de um sistema de controle é o de desempenho. De acordo com Ogata (2010), as especificações de desempenho podem ser dadas em termos de requisitos de resposta transitória e requisitos de regime estacionário. Para uma entrada degrau, há alguns parâmetros que caracterizam o sistema como: máximo *overshoot*, tempo de subida, tempo de acomodação e erro estacionário. Há ainda requisitos em termos de resposta em frequência.

Ainda de acordo com Ogata (2010), o processo de selecionar parâmetros do controlador que garantam dada especificação de desempenho é conhecido como sintonia do controlador. O desempenho de um controlador PID está diretamente ligado ao ajuste de seus três termos. Tal ajuste é feito variando-se os ganhos  $K_p$ ,  $K_i$  e  $K_d$  até que se obtenha a resposta ideal.

Dentre os métodos de sintonia que não exigem o conhecimento do modelo matemático do sistema, pode-se citar o método de Ziegler e Nichols (1942), que utiliza a resposta ao degrau, e vale-se de regras empíricas para a obtenção dos parâmetros do PID. Astrom e Hagglund (1984) desenvolveram o método do relé, que utiliza oscilações limitadas para obter os parâmetros, também conhecido como *autotuning*.

Conhecendo-se o modelo matemático da planta, é possível projetar a posição dos pólos da função de transferência em malha fechada do controlador, a fim de atingir as especificações de desempenho. Esse método é conhecido como alocação de pólos.

Dentro da área de ensino, pode-se citar o estudo de Kheir e Astrom (1996). Em seu trabalho, tratam do balanço entre ensino teórico e a intuição física, focando-se na melhora do currículo de Controle. Analisa-se também a utilidade de ferramentas de simulação e atividades de laboratório. O estudo visa um futuro mais interdisciplinar para a área de ensino de Controle, com maior proximidade entre teoria e prática.

### 3 METODOLOGIA

A pesquisa realizada neste trabalho caracteriza-se por ser analítica, quantitativa e qualitativa, e de caráter experimental. Utiliza o método dedutivo, partindo da teoria para um caso específico, a observação sistemática e a realização de testes. A metodologia utilizada no estudo se inicia pela construção da plataforma didática, seguida pelo desenvolvimento do *software*, que implementa a malha de controle de um PID digital.

As diferentes estruturas em que o controlador pode ser encontrado leva a modificações na sua lógica para dirimir problemas de aplicação prática. Assim, pode-se citar a existência das estruturas Ideal, Série e Paralela, em que cada uma arranja os termos P, I e D de maneira distinta. Da mesma forma, é possível mudar o arranjo para solucionar o chamado *wind-up* do termo integral e o impulso derivativo.

Seguindo-se na especificação da pesquisa, é preciso definir a implementação do controlador PID. No caso desse estudo, é a digital – um sistema que opera com sinais analógicos do mundo externo e os processa como sinais discretos, através da conversão A/D. Dessa forma, torna-se necessário utilizar aproximações numéricas da integral e da derivada. Isso fornece a lei do PID na forma de uma equação de diferenças, com um tempo de amostragem fixo. Essa equação é posteriormente implementada no microcontrolador.

Estabelecida a parte inicial da pesquisa, o próximo passo é a construção da plataforma propriamente dita. A estrutura é feita com canos de PVC, ligados de tal forma a dar estabilidade à plataforma, e em sua parte superior conecta-se a gangorra por meio de rolamento, o que permite um movimento suave e livre de atrito significativo. A gangorra é composta de alumínio, o que garante que seu peso seja de pouca influência na movimentação da mesma, e conta com adaptadores nas extremidades onde são parafusados os motores. A gangorra ainda conta com um fixador para o sensor no seu centro, o que garante que o sensor esteja bem posicionado e livre de vibrações. O *hardware* é composto de dois motores *brushless* Turnigy, de 1400 Kv, dois ESC's Turnigy Plush de 30 A e um sensor MPU6050, que contém acelerômetro de 3 eixos e giroscópio de 3 eixos. Por fim, o Arduino Uno como microcontrolador.

O algoritmo de controle é implementado na linguagem C padrão Arduino, e utiliza a biblioteca PID desenvolvida por Brett Beauregard. A leitura do sensor é feita via protocolo i2c. O sensor MPU6050 possui integrado no *chip* um processador digital de movimento DMP (*Digital Motion Processing*), que faz a fusão dos dados de acelerômetro e giroscópio. Assim, obtém-se em sua saída os dados de movimento em 3 eixos: rolagem (*roll*), arfagem (*pitch*) e guinada (*yaw*). Isso elimina a necessidade de processamento extra por meio de filtros. A leitura dos dados do sensor é feita com auxílio da biblioteca i2cdev, desenvolvida por Jeff Rowberg. No caso da plataforma, apenas um ângulo é necessário para fazer o controle, e pelo posicionamento específico do sensor, esse ângulo é o de rolagem. O PID recebe um ângulo desejado como *setpoint* e o ângulo lido pelo sensor, e processa o sinal de erro, que é a diferença entre

os dois. A saída do controlador é então mapeada para o *duty cycle* do sinal PWM (*Pulse Width Modulation*) que é enviado aos ESC's (*Electronic Speed Controller*). Esse sinal PWM é gerado por programação direta do *timer1* do ATMEGA328p, que é o microcontrolador do Arduino. Isso permite escolher a frequência desejada para o PWM, que depende do ESC utilizado. Dessa forma, a malha de controle é estabelecida pelo programa.

A obtenção dos dados é feita através de um programa desenvolvido por Brett Beauregard, na linguagem Processing, que é compatível com a linguagem do Arduino. É possível visualizar os gráficos do *setpoint*, sinal de erro e saída do controlador em tempo real, enquanto as mudanças ocorrem no sistema. O programa também permite a escolha do *setpoint* e dos parâmetros do controlador, Kp, Ki e Kd, enviando os dados ao Arduino. A ferramenta gráfica permite observar parâmetros de desempenho de um sistema, como *overshoot*, tempo de subida, tempo de acomodação e erro em regime permanente. A possibilidade de enviar novas constantes ao PID permite observar a mudança gerada na resposta do sistema em tempo real.

## 4 RESULTADOS

A Figura 2 apresenta a plataforma didática como primeiro resultado obtido. É possível ver a gangorra com o sensor posicionado, o Arduino Uno e os motores *brushless*.

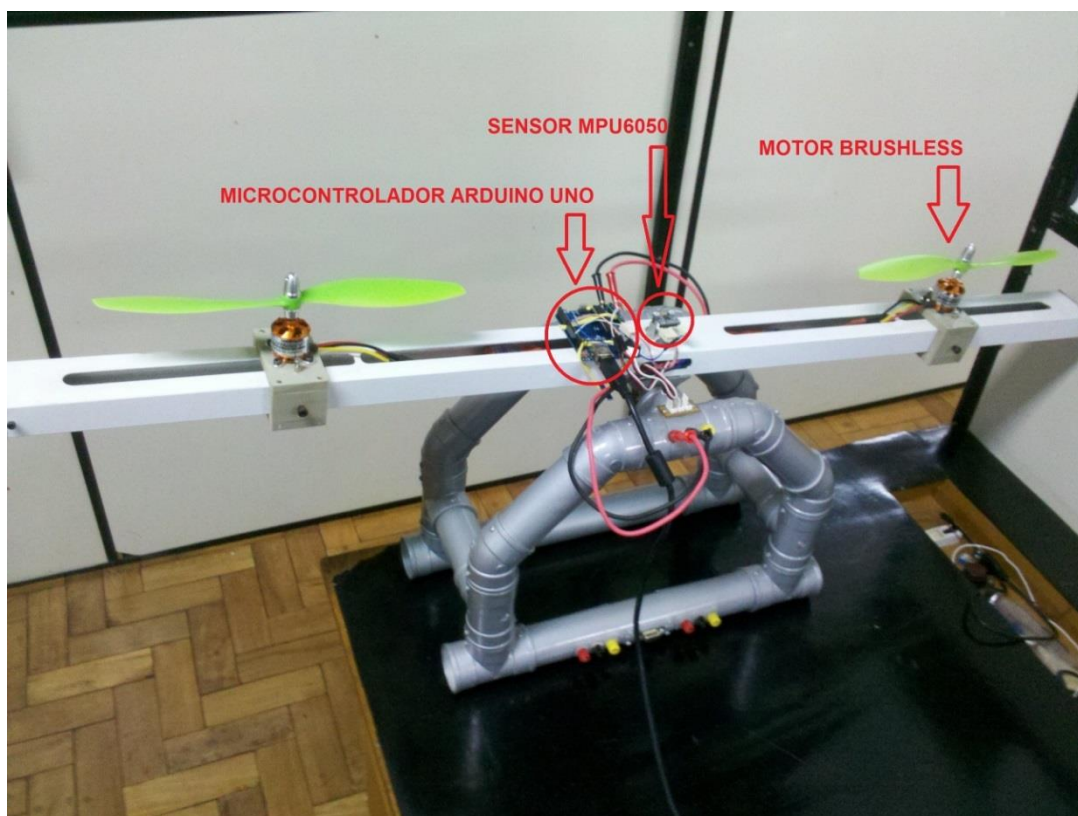


Figura 2 – Plataforma desenvolvida

Como segundo resultado tem-se o *software*, que processa os dados do sensor e calcular a saída do PID. O módulo DMP do sensor possui um *buffer* de memória FIFO (*First In, First Out*), que é preenchido com dados a uma taxa fixa de 10ms. Quando o *buffer* está completo, é gerada uma interrupção, permitindo ao microcontrolador receber os dados na mesma taxa. Isso garante que o tempo de execução do código do controlador seja fixo.

A Tabela 1 permite ver a variação real da saída do sistema. O sinal do sensor indica o quão próxima a saída está em relação ao *setpoint*. Também é possível ver o tempo de execução de cada loop do PID.

Tabela 1 – Resultados de execução do loop de controle PID

| <b>Setpoint (graus)</b> | <b>Input - sinal do sensor (graus)</b> | <b>Intervalo de tempo (ms)</b> |
|-------------------------|----------------------------------------|--------------------------------|
| 0                       | 0.06                                   | 10                             |
| 0                       | 0.04                                   | 9                              |
| 0                       | 0.02                                   | 10                             |
| 0                       | 0.02                                   | 10                             |
| 0                       | -0.03                                  | 10                             |
| 0                       | -0.02                                  | 10                             |
| 0                       | 0.01                                   | 9                              |
| 0                       | 0.02                                   | 10                             |

A saída do controlador é mapeada para um sinal PWM adequado ao ESC, que por sua vez gera o sinal adequado para o motor. A granularidade obtida com o sinal PWM é de 1/20000, resultante da programação do registro do *timer1* do microcontrolador que define a frequência. Garante-se assim boa precisão do sinal, mesmo que o *duty cycle* fique limitado de 10% a 20%. A frequência obtida para essa programação do *timer1* foi de 100Hz. Isso está de acordo com as especificações do ESC para seu correto acionamento.

O programa desenvolvido em Processing para visualização dos dados gráficos permite alterar a velocidade com que os dados passam pela tela, e também os limites máximo e mínimo das janelas de visualização. A comunicação do programa com o Arduino se dá via serial, e ocorre nos dois sentidos, permitindo que o programa envie dados ao Arduino e altere o estado de algumas variáveis do controlador, como o *setpoint* e os parâmetros Kp, Ki e Kd.

Na Figura 3 é possível ver o detalhe dos comandos de controle do programa que permitem escolher os diferentes parâmetros, sendo possível enviá-los ao Arduino clicando no botão inferior.

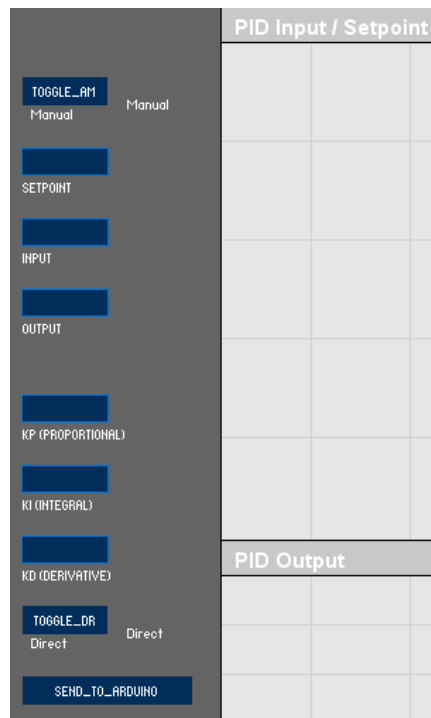


Figura 3 – Detalhe da janela do software de interface gráfica e controle

Na Figura 4 é possível ver o software em operação. A janela superior apresenta o *setpoint*, em verde, e o sinal do sensor, em vermelho. A janela inferior apresenta a saída do controlador, em azul, antes de ser mapeada para o PWM.

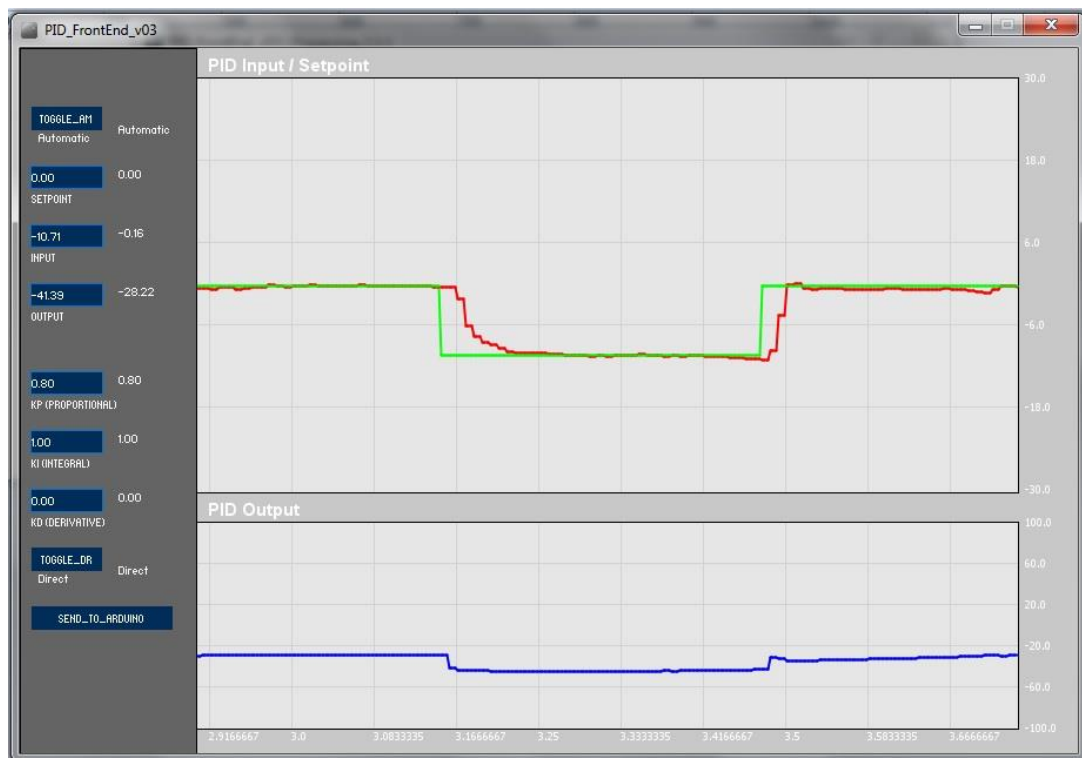


Figura 4 – Resultado de um teste de mudança de *setpoint*

## 5 DISCUSSÃO

Os resultados apresentados na Tabela 1 mostram a precisão do controlador, ao manter o sinal de saída muito próximo do *setpoint*, assim como a variação do tempo de execução do *loop*. Como pode ser verificado na Figura 4, a resposta do sistema não é ideal; isso se dá porque o controlador PID não foi sintonizado, sendo que os parâmetros utilizados ( $K_p = 0,8$ ;  $K_i = 1$ ;  $K_d = 0$ ) foram escolhidos empiricamente.

O gráfico obtido na Figura 4 demonstra a resposta do sistema a uma entrada degrau. É possível observar os parâmetros de desempenho do controlador e associar o resultado ao que foi visto no estudo da teoria de Controle. O estudante poderá analisar a parte física – a gangorra – e teórica – a resposta gráfica – simultaneamente.

Inicialmente utilizou-se o Arduino Pro Micro ao invés do Arduino Uno. Por possuir um microcontrolador ATMEGA32u4 que controla as funções de programa e comunicação USB no mesmo *chip*, ele mostrou-se incapaz de receber e enviar dados ao *software* simultaneamente. Não se pode afirmar se trata-se de limitação de *hardware* ou *software*, mas esse percalço levou ao uso do Arduino Uno. Outro problema enfrentado foi o conflito entre o uso do *timer1* do ATMEGA328p para geração de PWM e a leitura dos dados do DMP. Tal conflito ocorre apenas quando o sensor está muito afastado do microcontrolador, ou seja, para cabos longos. Quando se reduz a distância para alguns centímetros, o conflito não mais persiste. Isso limita o posicionamento do controlador, sendo que ele foi fixado na gangorra.

Entre as possíveis melhorias está a inclusão da sintonia para o controlador PID e a adição de uma interface que permita controlar essa sintonia, propiciando aos estudantes a possibilidade de aplicar diferentes métodos à plataforma e fazer observações críticas e comparações. Outra possível melhoria é o uso de um console externo que permita o controle da plataforma, com botões e *display*, o que seria uma adição à interface computacional, permitindo o uso da plataforma para os casos em que não se tem um computador disponível.

Como comparação, pode-se citar o trabalho de Martin e Kassab Junior (2006), que descrevem um trocador de calor como ferramenta didática. Foi desenvolvido um software didático para auxiliar os estudantes de controle, possibilitando o uso para os diversos cursos de Engenharia.

## 6 CONCLUSÃO

O uso da plataforma didática na disciplina de Controle de Sistemas visa aprimorar o ensino no âmbito dos cursos de Engenharia. O uso de um controlador PID permite a implementação de uma malha fechada de controle, e a interface gráfica auxilia na análise da resposta do sistema. Os resultados demonstram a validade da metodologia, assim como a utilidade da plataforma proposta. As implicações são de interesse, tanto para educadores, como para estudantes, pois ambos obterão benefício pelo uso da plataforma. As perspectivas são de que a idéia possa ser reconhecida em seu valor, e que a preocupação com o ensino de qualidade possa ser retomada, particularmente na disciplina de Controle de Sistemas, no âmbito acadêmico.

## 7 REFERÊNCIAS

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 5 ed. São Paulo: Pearson Education do Brasil, 2010.

ZIEGLER, J. G. e NICHOLS, N. B. **Optimum Settings for Automatic Controllers**. The American Society of Mechanical Engineers, p. 759-768, 1942.

ASTROM, K. J. e HAGGLUND, T. **Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins**. International Federation of Automatic Control, v. 20, n. 5, p. 645-651, 1984.

KHEIR, N. A., ASTROM, K. J., AUSLANDER, D., CHEOK K. C., FRANKLIN G. F., MASTEN M. e RABINS M. **Control Systems Engineering Education**. Automatica, v. 32, n. 2, p. 147-166, 1996.

MARTIN, Paulo Alexandre e KASSAB JUNIOR, Fuad. **Uso de Trocador de Calor como Ferramenta Didática para o Ensino de Modelagem e Sistemas de Controle**. Revista de Ensino de Engenharia, v. 25, p. 3-9, 2006.