

SISTEMA PARA AUXÍLIO DE PROJETO DE CONTROLADOR PID UTILIZANDO MICROCONTROLADOR ATMEGA328

Gilvan Farias da Silva¹
Raiza Gabriele Lima dos Santos²

RESUMO

Este trabalho propõe a implementação de um controlador Proporcional-Integral-Derivativo (PID) em um microcontrolador e sua integração com um programa supervisor. O programa supervisor é utilizado para auxiliar na modelagem de um sistema dinâmico, bem como exibir informações sobre a operação do controlador PID. Foi utilizada a placa de desenvolvimento Arduino Nano, que contém o microcontrolador ATmega328, para implementação do algoritmo PID. Para desenvolvimento do programa supervisor, foi utilizada a linguagem computacional Python e as bibliotecas Tkinter, Matplotlib e Pyserial. Para avaliar o desempenho do sistema desenvolvido, foi feito um estudo de caso com um conversor CC-CC do tipo abaixador (*buck*). O conversor foi modelado pelo método de resposta ao degrau, em seguida foi projetado um controlador Proporcional Integral (PI) pelo método de Ziegler Nichols. O controlador foi, então, ajustado com auxílio do programa supervisor para melhor desempenho.

Palavras-chave: Controlador PID, Sistema Supervisor, Conversor *Buck*.

INTRODUÇÃO

Controladores proporcional-integral-derivativo (PID) são bastante populares na indústria. O sucesso dos controladores do tipo PID é devido à sua simplicidade, eficiência e grande aplicabilidade. Indústrias o utilizam para controle de nível, vazão, temperatura e demais variáveis que precisem ser controladas. A forma de implementação evoluiu desde os controladores pneumáticos até os eletrônicos. Atualmente muitos controladores PID são implementados com microprocessadores. Isto possibilita a implementação de algoritmos de auto-sintonia e o desenvolvimento de controladores adaptativos (LEMES, 2010). É possível ainda comunicar o controlador com sistemas de aquisição de dados permitindo seu monitoramento.

Para implementação de um PID usando um microprocessador são necessários alguns fundamentos da teoria de controle discreto. Caso a planta a ser controlada seja do tipo tempo contínuo, o controlador pode ser projetado com auxílio da transformada de Laplace, sendo em seguida convertido para a transformada Z por meio de um processo de discretização. Outra

¹Mestrando do Curso de Engenharia Elétrica da Universidade Federal da Bahia - UFBA, gilvan.farias.silva@email.com;

²Graduada do Curso de Ciências Contábeis da Universidade Federal da Bahia - UFBA, raiza-gabriele@hotmail.com

abordagem é discretizar o sistema composto pela planta, sensor e atuador e projetar o controlador diretamente por meio da transformada Z. Existem vários métodos para converter um sistema de tempo contínuo para tempo discreto. O método de Tustin é o utilizado neste trabalho por não apresentar o problema de sobreposição de espectro (*aliasing*) (MOREIRA, 2014).

Em algum momento durante um curso de Engenharia Elétrica, um estudante pode ter a necessidade de criar um sistema de controle, seja como parte da avaliação de uma disciplina ou outro motivo ligado ao aprendizado. Caso o foco da atividade seja o aprendizado da teoria de controle, tarefas como projetar um circuito eletrônico e programar um microcontrolador podem tomar tempo e desviar o foco do objetivo principal.

Uma busca pela internet por exemplos de controladores desenvolvidos por estudantes ou entusiastas mostra que, existe um padrão entre os projetos. Para projetos desenvolvidos com microcontroladores, os parâmetros do controlador fazem parte do código fonte, que é compilado e gravado. Caso os parâmetros precisem de ajuste, eles são alterados no código fonte, que precisa ser recompilado e gravado no microcontrolador. Existe ainda a possibilidade de o projetista não ter familiaridade com programação de microcontroladores, atrasando a finalização do projeto.

Considerando as necessidades deste público, é plausível a construção de um dispositivo que facilite a criação de um sistema de controle. Tarefas como, fazer aquisição de dados da curva de resposta ao degrau, implementar o algoritmo de um controlador e alterar os parâmetros podem ser facilitadas.

Para avaliar o funcionamento do controlador, é útil conhecer os valores de referência, da variável controlada e do sinal de controle em função do tempo. A obtenção destes dados é outra dificuldade encontrada, principalmente se os sinais tiverem rápida variação. Portanto, a utilização de um programa computacional para aquisição e exibição destes dados é útil para validação do projeto.

Este trabalho tem por objetivo criar um controlador PID genérico pré-programado e um programa supervisor para auxiliar nos processos de modelagem da planta, configuração do PID e monitoramento dos valores de referência, da variável controlada e manipulada. O controlador deve ser capaz de ser aplicado em diferentes plantas, desde que suas limitações físicas sejam observadas e o sinal do sensor seja condicionado de forma adequada. O sistema deve ser simples e ser construído com hardware popular, de forma a ser acessível a estudantes

e entusiastas. Ao final do trabalho é feito um estudo de caso com um conversor CC-CC do tipo *buck* para avaliar o funcionamento do sistema como um todo.

METODOLOGIA

Conforme o objetivo deste trabalho, foi criado um PID pré-programado em um microcontrolador e um programa supervisorio que é executado em um computador pessoal. Fisicamente, o microcontrolador é ligado a um sensor para receber dados da variável controlada, a um atuador, que receberá o sinal de controle e ao computador por meio de um cabo USB. O sistema criado possui três modos de operação. O modo "Modelagem", o modo "Controle" e o modo "Espera". O modo "Modelagem" é utilizado para fazer a modelagem da planta por meio da resposta ao degrau. O sistema opera da seguinte forma. O usuário digita no sistema supervisorio o valor do período de amostragem, o tempo total de aquisição de dados e a amplitude do sinal degrau. Clicando no devido botão, estes valores são enviados ao microcontrolador, que aplica um sinal em degrau na planta e lê o valor da variável de interesse. Os dados da variável de interesse são enviados ao programa supervisorio, que cria um gráfico de amplitude da variável em função do tempo e salva os dados em um arquivo de texto.

O modo "Controle" é o modo em que o algoritmo do PID é executado. O sistema opera da seguinte forma. O usuário digita no sistema supervisorio os valores dos parâmetros do PID, ganho proporcional, ganho integral, ganho derivativo, período de amostragem e valor de referência. Clicando nos botões "Redefinir", os valores dos parâmetros são enviados para o microcontrolador. Clicando no botão "Controle", o microcontrolador lê o sinal do sensor, calcula o sinal de controle por meio do algoritmo do PID e envia o sinal de controle para o atuador por meio de modulação PWM. Os valores da referência, do sinal de controle e variável controlada são enviados para o programa supervisorio, que gera um gráfico das amplitudes em função do tempo. O modo "Espera" serve para pausar a execução dos outros modos de operação.

Este trabalho foi desenvolvido com hardware popular e de baixo custo, de forma a facilitar a reprodução do mesmo por parte de outros estudantes. Escolheu-se utilizar um microcontrolador ATmega328 em uma placa de desenvolvimento Arduino Nano.

O programa supervisorio é capaz de estabelecer comunicação com o Arduino. Possui interface gráfica com caixas de texto, em que o usuário possa digitar os parâmetros necessário

para configurar o PID. Botões para enviar os valores dos parâmetros para o microcontrolador, e para iniciar os modos de operação. Uma área para geração de gráficos. Bem como salvar os dados adquiridos no modo "Modelagem" em um arquivo de texto.

Revisão Teórica

Um sistema de controle é um conjunto de elementos que interagem de forma que a saída do sistema possa seguir um sinal de referência (CHEN, 1993). No contexto de sistema de controle pode-se dizer que o sistema é o conjunto formado por uma planta que será controlada, por sensores, por controladores e atuadores.

A planta é um equipamento a ser controlado (OGATA, 2010). Um equipamento pode ser definido como um conjunto de elementos (peças) funcionando de maneira ordenada e pode ser visto como um subsistema. Sensores são dispositivos usados para medir alguma grandeza física como tensão ou velocidade. Atuadores são utilizados para interferir no ambiente como uma resistência que atua aquecendo um forno elétrico ou a água de um chuveiro.

Existem diversas configurações para sistemas de controle, como o controle em malha aberta, em que o sinal desejado aciona um atuador que agirá na planta definindo o valor da saída. A configuração em malha fechada, em que o controlador está posicionado no ramo de realimentação também pode ser empregada. Neste trabalho é adotada a configuração de controle realimentado negativamente com o controlador em série com a planta a ser controlada. Esta configuração é amplamente descrita em livros sobre teoria de controle como em (CHEN, 1993), (OGATA, 2010) e (FRANKLIN, 2009).

Para projetar um controlador, primeiramente é necessário conhecer a planta a ser controlada, seu comportamento e definir de maneira clara como ela deve se comportar em conjunto com o controlador. Para o projeto são utilizados modelos que representam a planta real.

Um modelo matemático pode ser criado pelo método da caixa preta, em que os elementos que compõem a planta são desconhecidos. Neste caso pode-se aplicar um sinal na entrada do sistema e analisar o sinal na saída. Uma mesma planta pode ter mais de um modelo, cada modelo com determinadas características de simplicidade e precisão (ÅSTRÖM, 1995).

Os modelos criados, em geral, possuem certa divergência da planta real. Entretanto, em alguns casos, é possível criar modelos aceitáveis recorrendo à funções de transferência de primeira ou segunda ordem. O aumento da complexidade do modelo adotado pode garantir maior precisão, conseqüentemente a análise da planta e o projeto do controlador tornam-se mais complexos. Cabe ao projetista saber balancear a precisão e a complexidade do modelo adotado (OGATA, 2010).

Segundo Åström (1995) o controlador proporcional integral derivativo (PID), ou variações deste, é o mais utilizado. Podendo ser implementado de diversas formas de acordo com a necessidade. A formulação matemática de um controlador PID ideal pode ser dada pela equação (1).

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (1)$$

em que K_p é a constante proporcional, T_i é o tempo integrativo, T_d é o tempo derivativo e $e(t)$ é o sinal de erro que é a diferença entre o sinal de referência e a saída da planta. Alternativamente pode-se usar a seguinte forma.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2)$$

Sendo $K_i = K_p/T_i$ a contante integral e $K_d = K_p \times T_d$ a constante derivativa. A função de transferência é:

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d \times s \quad (3)$$

A eletrônica digital se baseia em sinais lógicos que não precisam ser tão precisos quanto os analógicos, desde que permaneçam numa determinada faixa de valores. A popularização e barateamento de computadores digitais proporcionou sua aplicação em diversas áreas. Controladores podem ser implementados por meio da programação de um microprocessador ou microcontrolador.

A desvantagem dos sistemas digitais é que muitas grandezas de interesse em sistemas de controle são analógicas. Para criar uma interface entre o mundo analógico e o digital o sinal analógico precisa ser discretizado, ou seja, transformado em um sinal discreto. Um sinal analógico é contínuo no tempo, em um intervalo de tempo finito existem infinitos valores. Um sinal discreto possui valores apenas em instantes bem definidos. Após a amostragem, o sinal passa por um conversor analógico para digital (ADC) onde os valores são codificados em representação binária (LATHI, 2007).

Um microcontrolador é um circuito integrado que possui uma unidade lógico aritmética, uma memória de programa, memória RAM e alguns periféricos que podem variar de acordo com a versão do microcontrolador. Dentre os periféricos podem ser citados o ADC, transmissor e receptor de comunicação serial, timers, geradores de sinal PWM, dentre outros periféricos (HUANG, 2005). Alguns autores descrevem um microcontrolador como sendo um computador em um único chip.

Neste trabalho foi utilizado o microcontrolador ATmega328 da família AVR da Atmel. A escolha foi feita baseado na facilidade de encontra-lo no mercado integrado a placa de desenvolvimento Arduino Nano, no baixo custo e por atender as necessidades do projeto.

Sistemas de supervisão e aquisição de dados do inglês *Supervisory Control and Data Acquisition System* (SCADA) são sistemas que supervisionam determinada planta ou conjunto de plantas, podendo interferir no valor de referência do controlador caso necessário, ou ainda emitindo alertas quando determinado valor alcança níveis críticos. Utilizando sistemas supervisórios é possível tomar conhecimento de falhas, ou prevê-las dependendo das características da planta (JURIZATO, 2002).

Existem vários fornecedores de softwares que facilitam o desenvolvimento de um sistemas SCADA. Alguns são comerciais a exemplo da Siemens com o WinCCe da Elipse com o Elipse E3 (JURIZATO, 2002). Existem também projetos grátis de código aberto como o ScadaBR. O ScadaBR começou a ser desenvolvido em 2006 por um conjunto de empresas e a Universidade Federal de Santa Catarina. Foi baseado no *software* Mango, tendo o objetivo de ser um sistema SCADA totalmente em português e voltado para pequenas empresas (SCADABR, 2017).

Em algumas aplicações pode ser mais conveniente desenvolver um sistema supervisório dedicado e otimizado para o caso em questão. Neste trabalho um sistema supervisório foi desenvolvido utilizando a linguagem computacional Python e a biblioteca para interface gráfica Tkinter.

RESULTADOS E DISCUSSÃO

O protótipo do sistema supervisorio criado pode ser visto na figura 1. Na parte inferior esquerda há três botões para determinar o modo de operação. O botão “espera” para a operação do controlador. O botão “modelagem” ativa o modo modelagem, em que é aplicado um degrau na entrada da planta e são lidos os dados em sua saída. É necessário que seja dado um intervalo e tempo no qual a modelagem irá ocorrer na janela de texto do campo envio de dados. O tempo é dado em milissegundos, portanto se for digitado o valor 1000, a modelagem ocorrerá por um segundo.

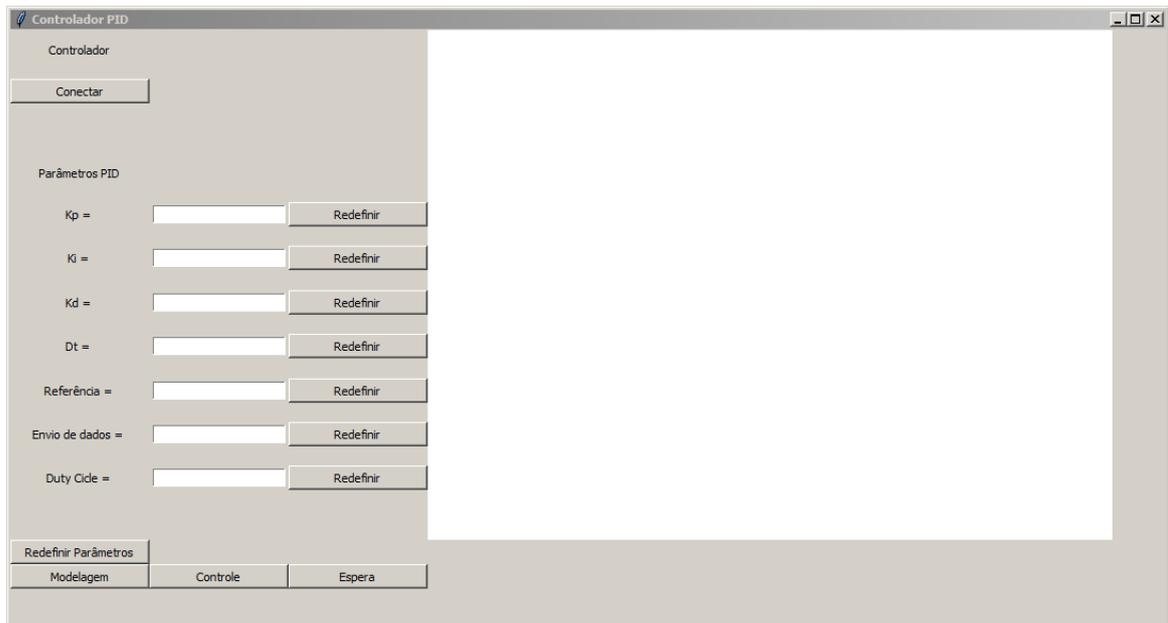


Figura 1: Interface gráfica do programa supervisorio.

Fonte: Autores (2019).

Para avaliar o funcionamento do protótipo, foi feito um estudo de caso com um conversor CC-CC do tipo *buck*. Um conversor *buck* é um circuito de eletrônica de potência que converte uma tensão de entrada em uma tensão de saída menor ou igual à tensão de entrada. A conversão é feita através do fechamento e abertura de uma chave eletrônica e posterior filtragem do sinal em um filtro passa baixas.

A modelagem foi feita aplicando um degrau com Duty Cicle de 20 %, resultando no gráfico da figura 2. Note que, a tensão de operação do microcontrolador é 5 V, logo o Duty

Cicle de 20 % equivale a 1 V na saída do microcontrolador. Pode-se, então, afirmar que foi aplicado um degrau unitário.

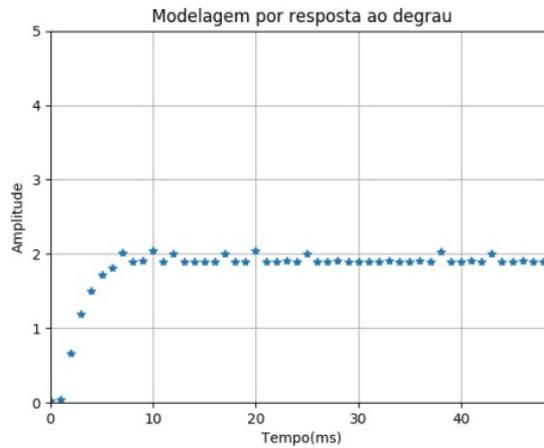


Figura 2: Resposta ao degrau do conversor buck.
Fonte: Autores (2019).

Utilizando o método de Ziegler-Nichols foi projetado um PID com os seguintes parâmetros $K_p = 5,1$, $K_i = 2,55$ e $K_d = 2,55$ e tempo de amostragem de 1 ms. Um PI foi projetado com os seguintes parâmetros $K_p = 2,925$, $K_i = 0,8775$. A resposta do sistema pode ser vista nas figura 3. O sinal de controle em ambos os casos apresentou comportamento bastante oscilatório, lembrando o sinal de um controlador on-off. O sinal de controle foi melhorado por meio de ajustes nos parâmetros do controlador.

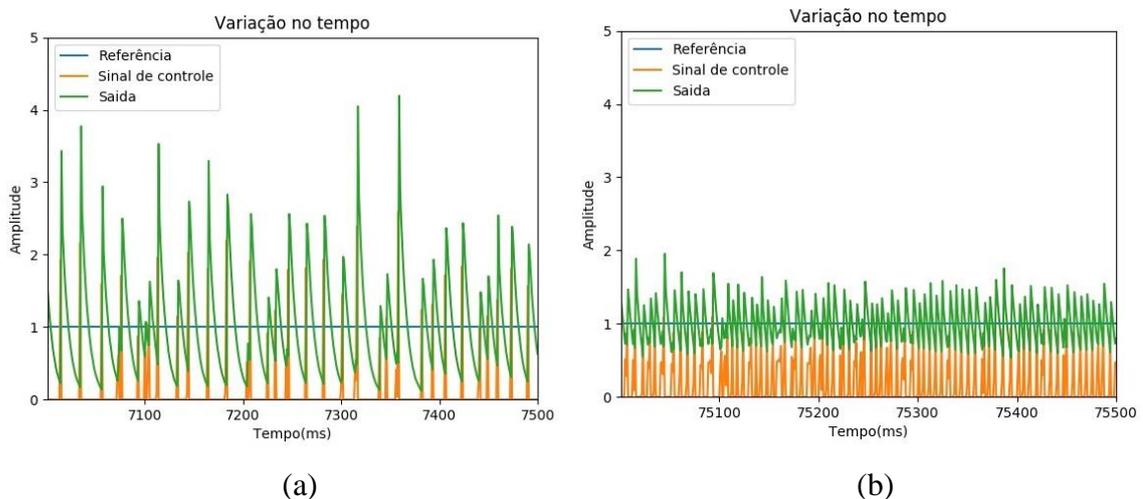


Figura 3: (a) Resposta obtida com PID projetado por Ziegler Nichols, (b) Resposta obtida com PI projetado por Ziegler Nichols.
Fonte: Autores (2019).

O controlador PI teve o desempenho um pouco melhor que o PID, levando-se em consideração a amplitude da oscilação da saída, que no caso do sistema com controlador PI não excedeu 2 V, enquanto que no sistema com controlador PID a saída excedeu 4 V. Portanto, o controlador PI foi escolhido para passar por um ajuste fino. Foram feitos alguns ajustes nos parâmetros do controlador PI até obter uma resposta mais satisfatória. Os parâmetros finais são $K_p = 0,5$ e $K_i = 0,1$. O comportamento do sistema pode ser visto nas figuras 4. Foram feitas variações do valor de referência para verificar o comportamento do sinal de saída e de controle. O sinal de saída foi capaz de seguir o sinal de referência apesar da oscilação em torno dele.

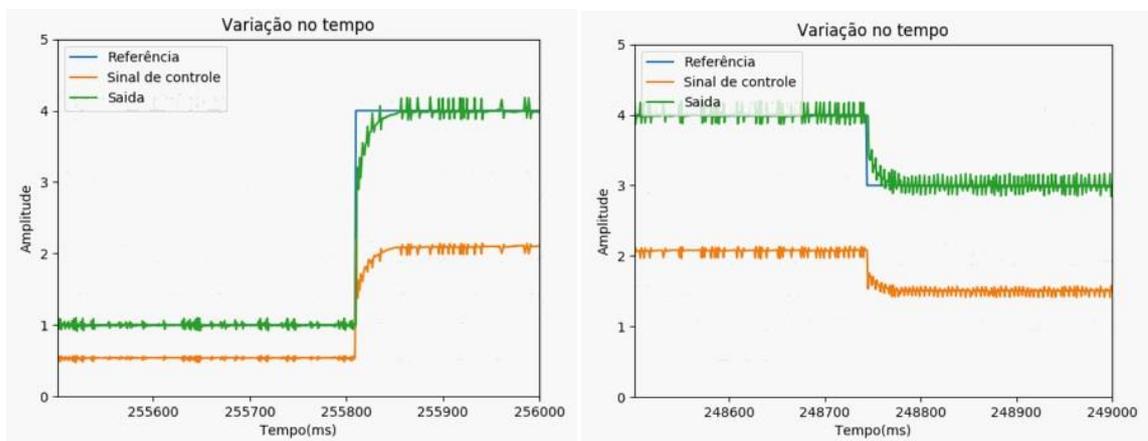
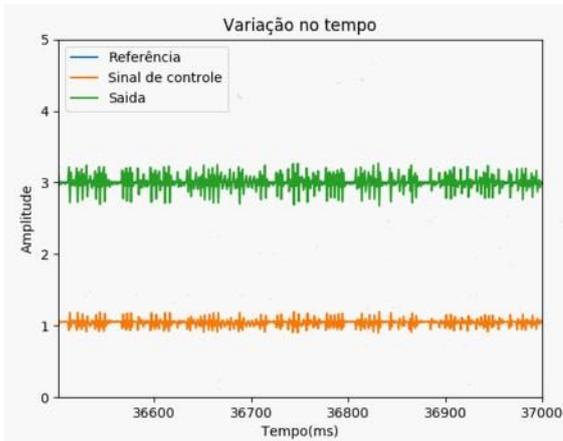
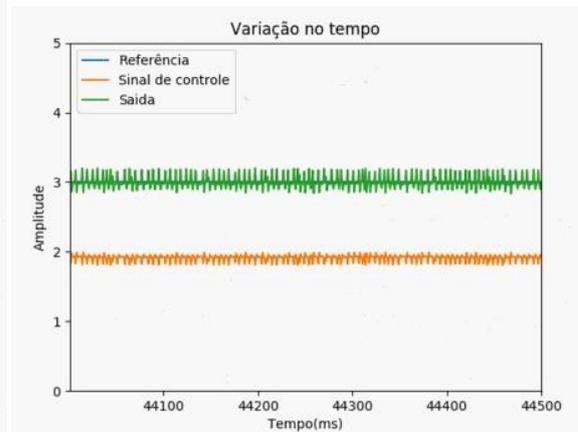


Figura 4: Resposta do sistema com controlador PI.
Fonte: Autores (2019).

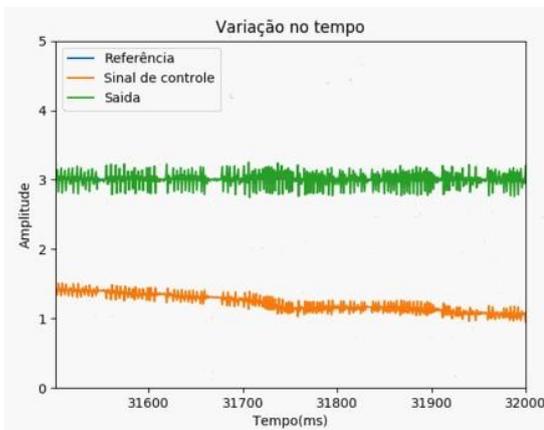
Para avaliar a capacidade do controlador de reagir a uma perturbação, foram feitas alterações do valor da tensão de alimentação do conversor V_s . Foi aplicado o valor $V_s = 15$ V, resultando na diminuição do sinal de controle 5(a). Foi aplicado o valor $V_s = 9$ V, resultando no aumento do sinal de controle 5(b). As figuras 5(c) e 5(d) mostram a variação do sinal de controle para o caso em que a tensão V_s aumentou e diminuiu respectivamente. Ao aplicar uma tensão de 8,5 V ou inferior, o sistema perdeu a estabilidade, como pode ser visto na figura 5(e). O controlador foi capaz de operar mesmo com variações de V_s entre 8,5 V e 15 V. Perdendo a estabilidade para tensões menores que 8,5 V. Não foi aplicado tensão maior que 15 V para evitar possíveis danos ao circuito.



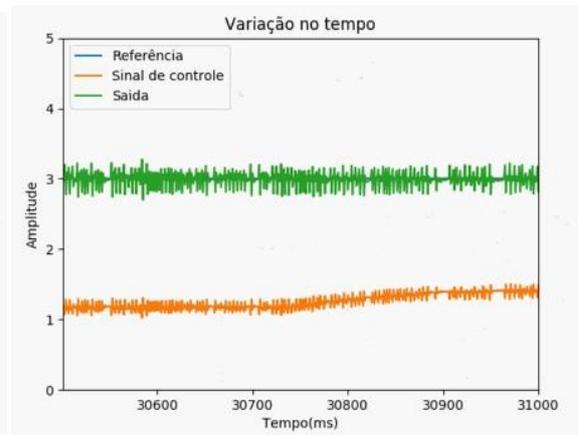
(a)



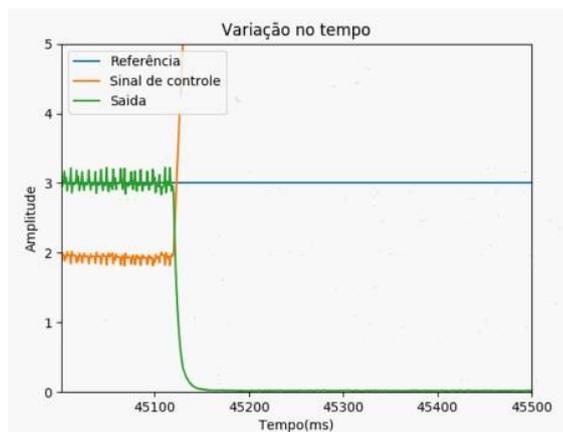
(b)



(b)



(d)



(e)

Figura 5: Resposta do sistema com controlador PI em caso de perturbação. (a) $V_s = 15$ V, (b) $V_s = 9$ V, (c) V_s aumentando, (d) V_s diminuindo e (e) $V_s = 8$ V.

Fonte: Autores (2019).

CONSIDERAÇÕES FINAIS

O desempenho do sistema desenvolvido atendeu às expectativas. Mesmo utilizando-se um microcontrolador de 8 bits com clock de 16 MHz, foi possível realizar operações em ponto flutuante, conversão de variáveis do tipo float em string e vice-versa. Para a temporização foi possível usar um período de amostragem mínimo de 1 ms.

A utilização da linguagem Python no desenvolvimento do sistema supervisorio facilitou o trabalho de programação por ser uma linguagem de fácil aprendizado. A biblioteca Tkinter foi escolhida principalmente por já vir pré-instalada com o interpretador Python. A principal dificuldade no desenvolvimento da interface gráfica foi alocar os vários elementos de maneira a facilitar o entendimento de um usuário iniciante. Por elementos entende-se botões, caixas de texto, dentre outros.

O controlador PI atendeu às expectativas. Sendo capaz de seguir o valor de referência estabelecido pelo operador. Mesmo para valores de V_s entre 9 V e 15 V, o controlador foi capaz de manter o valor da variável controlada. O sistema perdeu a estabilidade para V_s menor que 8,5 V. Como a modelagem foi feita para $V_s = 10$ V, a faixa de estabilidade para V_s entre 8,5 V e 15 V proporciona uma certa segurança para a tensão de saída.

Este trabalho atingiu o objetivo proposto. Que é construir um sistema capaz de auxiliar estudantes e entusiastas em trabalhos de sistemas de controle. Foi utilizado um conversor *buck* como estudo de caso, e com algumas modificações no *firmware* pode ser utilizado com outros sistemas dinâmicos. Tais modificações precisam ser feitas, principalmente, nas partes do código que tratam do sensor e atuador. Visto que foi utilizado um ADC, e sensores com outros tipos de sinais precisam de um tratamento diferente. O mesmo vale para atuadores que não utilizam sinal PWM.

REFERÊNCIAS

ÅSTRÖM, K. J. ;HÄGGLUND, T. **PID controllers: theory, design, and tuning**, 2. ed. [S.l.]: Isa Research Triangle Park, NC, 1995.

CHEN, C. **Analog and digital control system design: transfer-function, state-space, and algebraic methods**. Fort Worth: Saunders College Publishing, 1993.

FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. **Feedback control of dynamic systems**, 6. ed. [S.l.]: Pearson Prentice Hall, 2009.

HUANG, H. **PIC microcontroller: an introduction to software and hardware interfacing.** [S.l.]: Cengage Learning, 2005.

JURIZATO, L.; PEREIRA, P. Sistemas supervisórios. **Network Technologies**, Nova Odessa, v. 1/2, n. 1/2, p. 105-114, 2002.

LATHI, B. P. **Sinais e sistemas lineares.** 2. ed. Porto Alegre: Bookman, 2007

LEMES, A.; SILVEIRA, A.; COELHO, A. Pseudo controlador PID: Projeto e sintonia automática. In XVIII Congresso Brasileiro de Automática–CBA, 2010, Bonito, MS.

MOREIRA, C.; SILVA, F. Automação da discretização de controladores PID e filtros digitais. **Revista Ciência e Tecnologia**, [S.l.], v. 17, n. 31, dez. 2014. ISSN 2236-6733. Disponível em: <<http://www.revista.unisal.br/sj/index.php/123/article/view/358>>. Acesso em: 12 dez. 2017.

OGATA, K. **Engenharia de Controle Moderno.** 5. ed. São Paulo: Pearson Prentice Hall, 2010. SCADABR, 2017. Uma breve história do Projeto. Disponível em: <<http://www.scadabr.com.br/>>, Acesso em: 20 nov. 2017.