



- **ENSINO MULTIDISCIPLINAR DE CONTROLE AUTOMÁTICO E MICROPROCESSADORES, ATRAVÉS DE UM SIMULADOR DE MICROCONTROLADOR PARA O MATLAB.**

Carlos Eduardo R. Batista – Carlos.petcontrole@hotmail.com

Letícia H. Moura - letismoura@gmail.com

Luis Claudio G. Lopes – gamboa@leopoldina.cefetmg.br

Marlon J. Carmo - marloncarmo@ieee.org

Centro Federal de Educação Tecnológica de Minas Gerais, Campus III

Rua José Peres, 558 – Centro.

36700-000 – Leopoldina – MG

***Resumo:** A utilização de software de simulação no ensino de Engenharia é indispensável. Devido o grande número de estudantes de engenharia nas faculdades e a falta de estrutura, é praticamente impossível oferecer acessibilidade, a um laboratório real, para todos os alunos. A proposta do artigo é unir a disciplina de microprocessador à disciplina de Controle Automático. O objetivo de vincular o uso de microprocessador ao ensino de Controle, é que esse trabalha com tempo discreto, sendo possível detectar resultados de curvas de forma mais simplificada, diminuindo conseqüentemente os cálculos. Visando essa metodologia de ensino, desenvolveu um ambiente de simulação de microcontroladores para o software de simulação MATLAB. A simulação é executada por meio da leitura das rotinas a ser gravado no microcontrolador de acordo o tempo de atualização a ser utilizado durante a execução. O trabalho tem como meta auxiliar os alunos a absorverem melhor o conhecimento, melhorando o processo de ensino – aprendizagem de forma a contribuir com a multidisciplinaridade no ensino de Microcontroladores, Eletrônica Digital, Controle Digital e Controle Avançado. As vantagens deste simulador são que para testar o código do microcontrolador não é necessário montar o circuito e uma vez funcionando o código pode ser utilizado em um microcontrolador real sem problemas, quebrando a limitação dos alunos de desenvolverem seus projetos somente nos laboratórios de pesquisas.*

***Palavras-chave:** Microcontrolador, Simulação, Controle, Processos Dinâmicos, MATLAB*

1. INTRODUÇÃO

O modelo atual de ensino preza pelos estilos convencionais de aula onde a lição apresentada em sala é, em geral, cobrada de forma sistemática. Embora o conteúdo programático dos cursos busque obter variedade de matérias dentro dos seus respectivos ramos de segmento, ainda pode faltar uma contextualização necessária entre elas, obtida na prática (BELHOT *et al*, 2001). Em escolas de engenharia, um exercício bem difundido é o uso de simuladores para auxiliar o aluno a entender e ver a aplicabilidade das teorias aprendidas (SILVA *et al*, 2010).

Realização:

 **ABENGE**

Organização:



**o ENGENHEIRO
PROFESSOR E O
DESAFIO DE EDUCAR**



A simulação de processos reais não substitui as aulas de laboratório nas quais é possível ver a aplicação da teoria, porém podem suprir um pouco da carência de experiências reais em sua formação. Através das ferramentas de simulação é possível quebrar as primeiras barreiras e unir os conceitos teóricos com as aplicações práticas, formando profissionais no conceito de habilidades e competências (MCDERMOTT *et al*, 2007).

Além do compromisso didático, um simulador auxilia o projetista na realização de seus experimentos quando estes ainda estão em sua fase de teste (CARVALHO *et al*, 2009). Dessa forma, é possível tentar prever o comportamento real de um protótipo que possa vir a ser construído.

O nome MATLAB vem da junção das palavras MATrix LABoratory, que é uma alusão à base operacional do programa, as matrizes. Esse software é muito útil para modelagens, simulações, visualização de gráficos e diversos cálculos matemáticos (GILAT, 2006). O software é amplamente utilizado nos cursos de engenharias por ser uma poderosa ferramenta auxiliar no desenvolvimento de projetos e pesquisas, possuindo funções comuns a diversas áreas, além de apresentar algumas ferramentas extras, os “toolboxes”, que são coleções de programas especiais dedicados para resolução de problemas específicos. Porém, dentro dessas inúmeras possibilidades que o programa oferece ainda não há ferramentas para simulação de microcontroladores. O objetivo desse artigo é mostrar uma ferramenta desenvolvida e testada para simulação em MATLAB com os princípios básicos de um microcontrolador.

Os microcontroladores são dispositivos eletrônicos compactos, que já inclui processador, memória e mecanismos para receber e enviar dados além de temporizadores, geradores de modulação por largura de pulso (PWM), conversor analógico/digital e controladores de interrupção. Esse complexo circuito integrado já possui os componentes necessários para sua aplicação e permite, ao programá-lo controlar as saídas baseadas em entradas referenciais ou um programa interno.

Dentro dos diversos microcontroladores existentes existe uma categoria muito conhecida, o PIC que é produzido pelo Microchip Technology. Há diversos modelos que se diferenciam, principalmente, pelo número de linhas de entrada/saída, quantidade de memória para armazenar dados, forma de alimentação e velocidade de processamento (ZANCO, 2005). Pelo fato do microcontrolador ser um dispositivo eletrônico muito utilizado na área de controle é possível encontrar vários trabalhos desenvolvidos para o auxílio no estudo dos mesmos como kits didáticos (RODRIGUES *et al*, 2007) e laboratório de experimentação remota (FELIPE *et al*, 2010). Unido as disciplinas de Controle automático, controle avançado, microprocessadores e eletrônica digital.

O trabalho mostrará o desenvolvimento de um ambiente de simulação para os Kits de microcontroladores PIC16F628A, PIC16F877A e PIC18F452, no SIMULINK do MATLAB. A partir desse simulador foi traçado como objetivo controlar processos dinâmicos utilizando-se de técnicas a lógica de relé. Com a implementação do microcontrolador espera-se obter como resultado final o controle de um processo dinâmicos de uma forma simples, didáticos servindo como auxílio nas grades curriculares dos cursos técnicos e superiores.

O trabalho foi estruturado em quatro etapas. A seção dois apresenta-se a modelagem do microcontrolador, feita para seu uso no MATLAB. A seção três traz as técnicas de controle de processos dinâmicos utilizando microcontroladores. Na seção quatro apontam-se os resultados e as análises das simulações. Na seção cinco conclui-se este trabalho.



2. O SIMULADOR PIC NO MATLAB

Com o grande uso de microcontroladores, faz-se a necessidade de um simulador confiável para facilitar os testes de um programa a ser passado para um PIC real ou mesmo para o aprendizado em salas de aula. O MATLAB é uma ferramenta extremamente utilizada nos cursos de engenharia, porém ainda não possui um “*toolbox*” que permita trabalhar com microcontroladores. O microcontrolador utilizado foi desenvolvido de uma forma que ele suporte alguns modelos da categoria PIC, nele já estão previamente configurados os modelos *PIC16F628A*, *PIC16F877A*, *PIC16F777* e *PIC18F452*. O bloco do simulador, uma “*s-function*”, é utilizada junto com componentes comuns do SIMULINK como: um multiplexador, um seletor e um conversor digital analógico de 16 bits (*DAC_16*).

O multiplexador representa os pinos do dispositivo eletrônico e possui 40 entradas. Só as configuradas pelo programa interno do microcontrolador são utilizadas, as entradas extras não serão lidas.

A “*s-function*” carrega o arquivo “*m_picsim.c*” compilado (arquivo *.MEX* do MATLAB). Ela é responsável por fazer a ponte entre SIMULINK e a biblioteca *picsim libpicsim.dll*, permitindo que códigos nativos do microcontrolador sejam usados diretamente na simulação sem nenhuma alteração. Esse *picsim* (LOPES, 2011) é um simulador de microcontroladores PICs e seus periféricos. Ainda na *s-function* são passados alguns parâmetros para as rotinas do ‘*m_picsim.c*’ que são: o modelo do PIC a ser utilizado, o arquivo em ‘*.hex*’ que contém o programa que se deseja que o microcontrolador execute e o período de atualização do bloco pois um PIC trabalha com tempo discreto. Esse período de atualização deve ser ajustado de forma que não seja nem tão grande a ponto de deixar falha a simulação e nem tão pequeno a ponto de deixar o tempo total de execução muito longo. A função do *m_picsim* é atualizar os pinos configurados como entrada, executar o programa passado como parâmetro e então ler os valores do simulador e escrever na saída.

O seletor possui a informação na forma vetorial dos pinos de saída, pois não é possível no SIMULINK fazer com que um pino possa representar ora uma entrada, ora uma saída, por isso a *s-function* tem 40 pinos de entrada e 40 pinos de saída para um microcontrolador de 40 pinos.

Por fim, o *DAC_16* faz a conversão digital analógica, já que o microcontrolador não possui esse conversor internamente. Também poderia ter sido utilizado um PWM, mas ele pode deixar a simulação mais lenta por utilizar, normalmente, alta frequência de amostragem.

A grande vantagem desse simulador é que para testar o código do microcontrolador não é necessário montar o circuito e uma vez funcionando o código pode ser utilizado em um microcontrolador real sem problemas.

O diagrama mostrado na Figura 1 representa a idéia do funcionamento do simulador. Cada bloco representa uma função do arquivo ‘*m_picsim.c*’ e, para cada função é mostrada as etapas de execução.

3 CONTROLE DE PROCESSOS DINÂMICOS UTILIZANDO O SIMULADOR PIC

Processos dinâmicos são os processos nos quais ao decorrer do tempo, ou por interferência de agentes internos e/ou externos modificam o andamento do processo.

Escolheu-se para testar o simulador proposto a identificação de um sistema através de um método de relé (WANG, 1997). A identificação de sistemas para conseguir uma função de

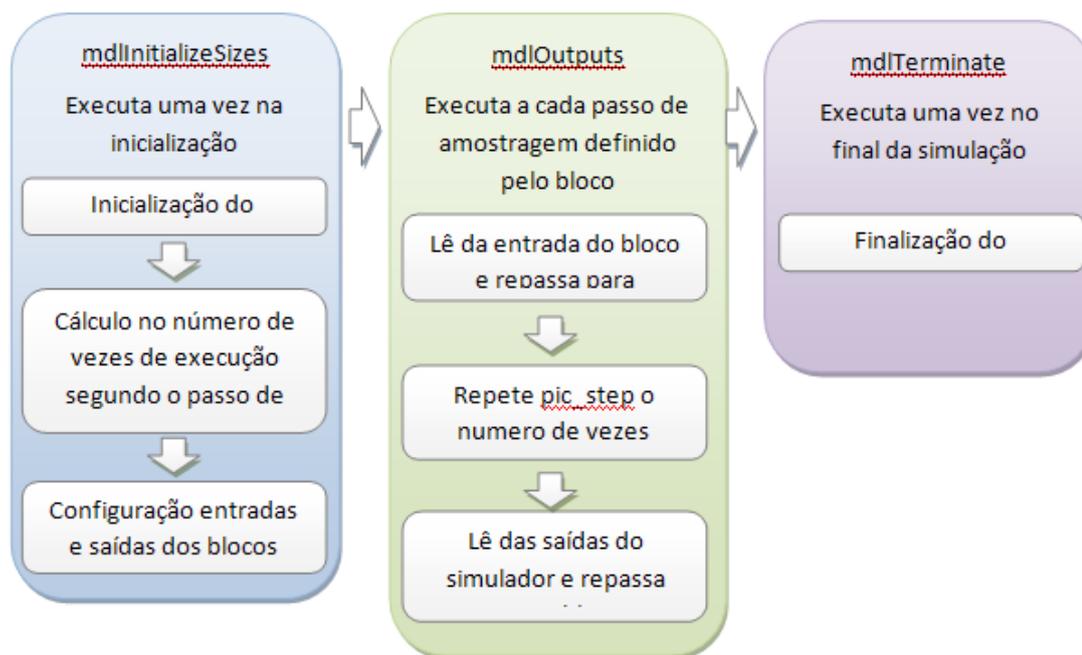


Figura 1 – Funcionamento do simulador

transferência próxima a real é útil quando a modelagem matemática é difícil de ser obtida ou não é satisfatória. Abaixo é apresentada uma introdução sobre o método e os testes realizados.

3.1 Introdução ao método de identificação via relé

O método de identificação via relé vem sendo muito utilizado para um procedimento de sintonia chamado de *autotuning*. O método é normalmente usado em controladores PID, porém atualmente foi estendido para utilização em outros controladores, nos quais precisam de modelos de funções de transferência.

Com a utilização desse método é possível encontrar expressões para o período e amplitude dos ciclos limites com realimentação por relé, que é derivado de um processo de primeira ordem mais tempo morto. As informações do domínio no tempo são obtidas através de expansões da série de Fourier no ponto da resposta em frequência e podem ser identificadas com apenas um único teste com o relé.

A função de transferência aproximada que será encontrada possuirá uma estrutura como a mostrada na “Equação 1”.

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

A “Equação 1” é utilizada como base para a identificação do método de forma que toda nova função de transferência obtida obedeça a essa estrutura. Nela, K representa o ganho estático, T uma constante de tempo e L o atraso de transporte. O ganho estático K pode ser encontrado aplicando-se um degrau ao sistema e verificando o seu valor de estabilização. Apesar da “Equação 1” representar um sistema de primeira ordem, na prática, muitos processos de ordem superiores podem ser aproximados por um sistema de primeira ordem com tempo morto.



O relé deve possuir uma histerese que será útil para evitar chaveamentos freqüentes e indevidos causados por ruídos na planta. O acionamento do relé é dado pelo esquema mostrado na “Figura 2”.

O relé é inserido antes do sistema a ser identificado para a obtenção dos parâmetros K, L e T mostrados na "Equação 1". O conjunto relé mais sistema possui uma realimentação, ou seja, é uma malha fechada e o gráfico dessa malha apresentará uma oscilação sustentável, como a mostrada na “Figura 3” juntamente com o sinal de chaveamento do relé.

Na “Figura 3”, a curva azul representada por y converge para uma oscilação estacionária com período $T_{u1}+T_{u2}$. A oscilação é caracterizada por:

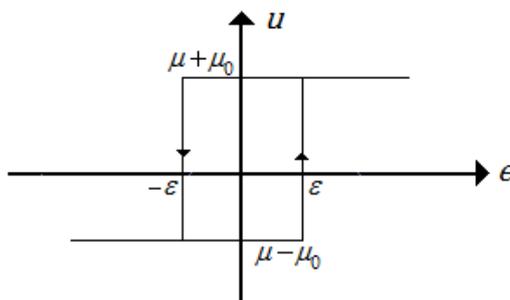


Figura 2 - Desenho esquemático do funcionamento do relé

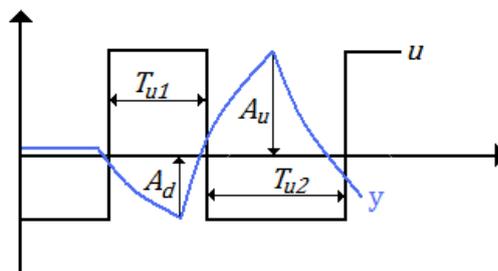


Figura 3 - Sinal de controle do relé (onda quadrada) e ciclo limite apresentado no sistema

$$A_u = \mu_0 + \mu K \frac{1 - e^{-L/T}}{1 + e^{-L/T}} + \epsilon e^{-L/T} \quad (2)$$

$$A_d = \mu_0 + \mu K \frac{1 - e^{-L/T}}{1 + e^{-L/T}} + \epsilon e^{-L/T} \quad (3)$$

$$T_{u1} = T \ln \frac{2\mu K e^{L/T} + \mu_0 K - \mu K + \epsilon}{\mu K + \mu_0 K - \epsilon} \quad (4)$$

e

$$T_{u2} = T \ln \frac{2\mu K e^{L/T} - \mu_0 K - \mu K + \epsilon}{\mu K - \mu_0 K - \epsilon} \quad (5)$$

As “Equações 2” a “Equações 3”, onde ϵ é a histerese, fornecem as expressões para amplitude e período do ciclo limite do processo representado na “Equações 1”. Porém, até esse ponto ainda não se sabe o valor do atraso de transporte e da constante de tempo, faz então

$$\Theta = L/T \quad (6)$$



que é dada pela seguinte fórmula para Θ :

$$\Theta = \ln \frac{\mu_0 + \mu K - \epsilon}{\mu_0 + \mu K - A_u} \quad (7)$$

Dessa forma, tem-se:

$$T = T_{ul} \left(\ln \frac{2\mu K e^{\Theta} + \mu_0 K - \mu K + \epsilon}{\mu K + \mu_0 K - \epsilon} \right)^{-1} \quad (8)$$

Com T e Θ calculados, através da “Equação 6”, tem-se L.

3.2 Identificação de sistemas utilizando o método via relé no MATLAB

A utilização desse método de identificação no MATLAB é feito com auxílio de um “*m file*” e um arquivo do SIMULINK. Caso este possua variáveis sem valores atribuídos, no arquivo texto as variáveis da simulação, podem ser inicializadas com seus valores e então o arquivo do SIMULINK deve ser executado pelo comando sim passando como parâmetro o diretório com o nome do arquivo com seu formato. O sistema a ser identificado pode ser montado conforme o desenho da “Figura 4”. Após o comando sim é feita então a manipulação algébrica para se obter os três parâmetros necessários da “Equação 1”.

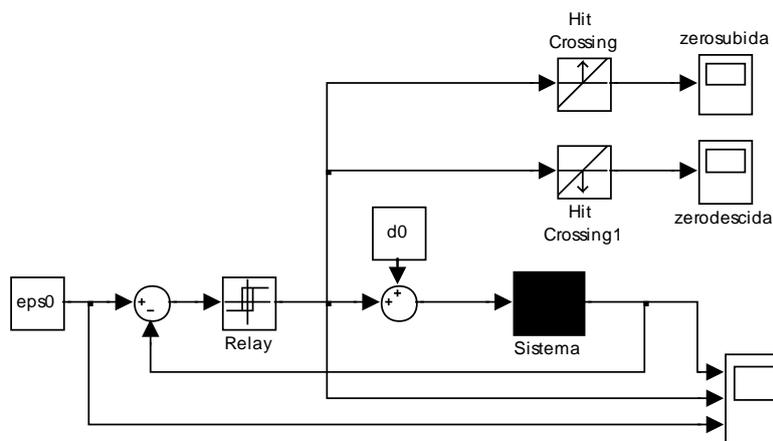


Figura 4 - Modelo do sistema simulado em SIMULINK

Na “Figura 4” pode-se observar um bloco preto, que na verdade é um *subsystem* contendo o sistema a ser identificado, e para efeitos de teste é um sistema desconhecido o qual se deseja levantar uma função de transferência aproximada. Existem também dois blocos “*Hit Crossing*”, que auxiliaram nas contas a serem feitas no *m file*, um identificando os momentos que ocorrem subidas e outro as descidas. Há também um bloco *com* de constante incrementado um valor após a realimentação do sistema, útil para deslocar a curva analisada e, dessa forma, facilitar a compreensão das respostas.



3.3 Identificação de sistemas utilizando o método de relé no simulador de microcontrolador proposto

A identificação de processos através do microcontrolador é feita de forma semelhante à realizada no MATLAB, o código a ser inserido no simulador do microcontrolador precisa possuir todas as funcionalidades dos blocos mostrados na “Figura 4”. Para isso, a onda quadrada da “Figura 3”, que representa o acionamento do relé, necessita ser gerada pelo algoritmo a ser inserido no microcontrolador. Essa onda gerada será enviada através de uma das portas no microcontrolador que estará ligada à entrada do sistema a ser identificado. Essa excitação promoverá uma onda do tipo dente de serra, também mostrada na “Figura 3” pela curva azul. A montagem do microcontrolador ligado ao sistema pode ser vista na “Figura 5”. O sinal de saída do sistema é enviado para o simulador que faz a leitura dos dados e a manipulação para a captura dos valores necessários para identificar o sistema pelo método via relé.

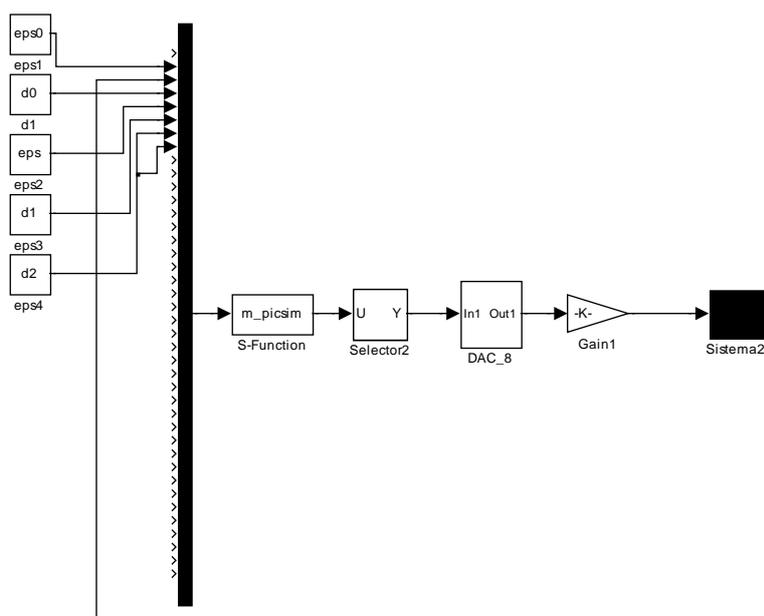


Figura 5 - Modelo do sistema utilizando o simulador do microcontrolador no SIMULINK

Pela “Figura 5” nota-se que a resposta do sistema é enviada novamente para o pino 3 do microcontrolador. Existem outras entradas conectadas aos pinos do sistema, estas são entradas referencias com valores que definem o relé não simétrico, cuja onda será gerada pelo microcontrolador. Em um teste realizado com esse sistema colheu-se a onda do relé e a resposta do sistema à aplicação do acionamento do relé como entrada. Esse gráfico pode ser visto na “Figura 6”.

Com o auxílio da “Figura 6” é possível perceber que as ondas obtidas na simulação são semelhantes às apresentadas para explanação sobre o método de relé na “Figura 3”. Essa semelhança demonstra que não só a onda foi gerada com sucesso, como também o sistema se comportou de forma esperada para que os cálculos dos parâmetros necessários para a nova função de transferência pudessem ser feitos.



3.4 Cálculo do compensador para o sistema utilizando o microcontrolador

Com uma pequena inclusão de código, o mesmo algoritmo utilizado para identificar o sistema pode calcular os ganhos de um compensador PID. Nas simulações realizadas utilizou-se a sintonia por parâmetros de Ziegler e Nichols, um método *First Order Delay Time*, baseado na curva de reação com tempo morto (CARMO, 2006). O sistema identificado pode então ser compensado por estes parâmetros.

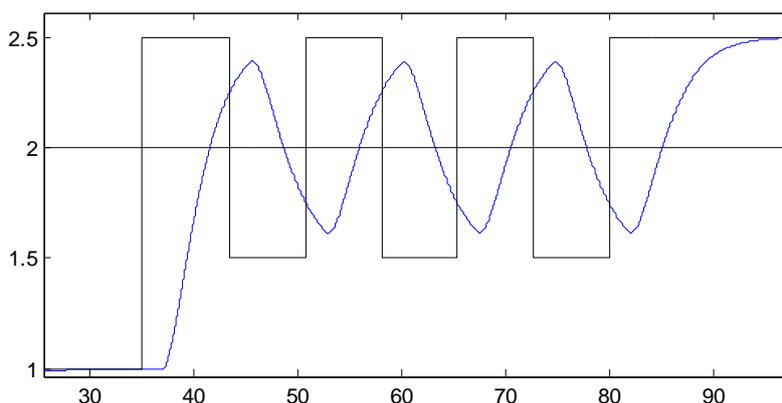


Figura 6 - Ondas quadrada gerada pelo microcontrolador (preta) e resposta do sistema a essa entrada (azul)

Com a função de transferência aproximada já levantada pelo método de relé, tem-se os valores de T , L e K , o sistema pode então ser compensado ao se encontrar os ganhos do controlador, K_p , K_i e K_d , com o auxílio das relações apresentadas na

Tabela 1 que mostra a relação dos parâmetros encontrados com as constantes K_p , T_i e T_d . A partir desses valores os ganhos são obtidos através das relações mostradas na “Equação 9” e “Equação 10”.

$$K_i = \frac{K_p}{T_i} \quad (9)$$

$$K_d = K_p \times T_i \quad (10)$$

Tabela 1 – Ajuste para projeto de compensadores segundo método em malha aberta de Ziegler e Nichols

Controlador	K_p	T_i	T_d
PID	$\frac{1,2T}{L}$	$2L$	$\frac{L}{2}$



Feito os cálculos pode-se utilizar a comunicação serial através do bloco *Serial Receive* do SIMULINK. No algoritmo, envia-se os ganhos calculados pela saída serial e esses podem ser exportados pelos blocos *To Workspace*, assim, após a identificação, pode-se utilizar outro arquivo de simulação do SIMULINK, com o controle implementado, com o bloco padrão do PID ou utilizando o microcontrolador.

4 COMPARAÇÃO DOS RESULTADOS DA SIMULAÇÃO EM MATLAB COM E SEM O MICROCONTROLADOR PIC

A simulação utilizou o microcontrolador montado conforme pode ser visto na “Figura 5”. Escolheu-se o sistema de segunda ordem mais tempo morto da “Equação 11” para os testes. Na mesma simulação as relações vistas na “

Tabela 1” foram utilizadas, dessa forma, juntamente com a nova função de transferência obtida, os parâmetros do controlador PID também foram encontrados.

$$G(s) = \frac{0.25}{(s + 0.5)^2} e^{-2s} \quad (11)$$

Os valores de K_d , K_i e K_p foram inseridos tanto como referência para um controle feito no microcontrolador como em um bloco “*PID Controller*” do SIMULINK. A montagem com as duas formas de controle para simulação está mostrada na “Figura 7”.

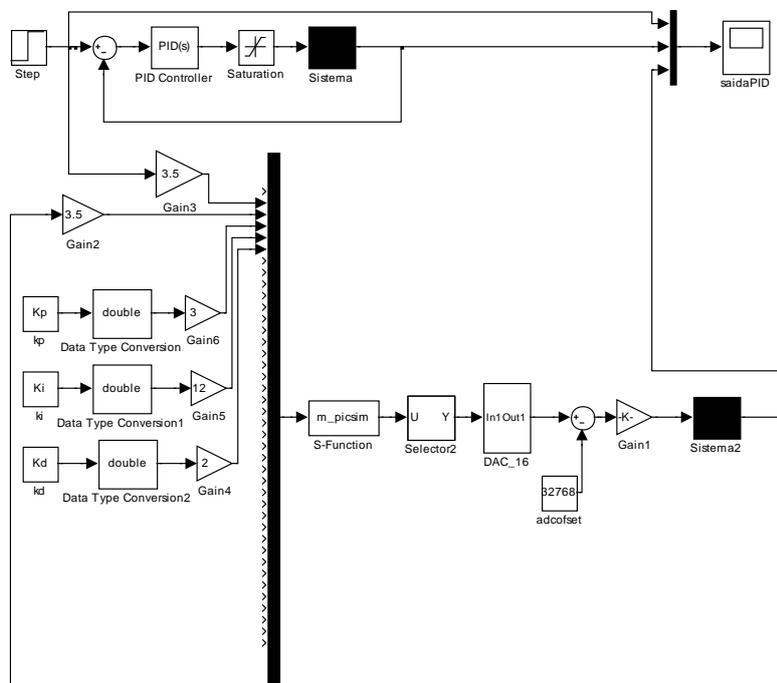


Figura 7 – Montagem da simulação para o controle do sistema

Na Figura 7, foi inserido um bloco de saturação após o “*PID Controller*”, para que o envio do sinal ao sistema fique compatível com a saturação natural dos componentes eletrônicos como o próprio microcontrolador. As curvas de controle estão ambas mostradas no gráfico mostrado na “Figura 8”.



Pela análise do gráfico, nota-se que não há diferença perceptível entre elas e que ambas rastreiam a referência. Esse resultado mostra que o microcontrolador se comportou de forma esperada apresentando um resultado muito preciso. Há uma imensa vantagem em se possuir um simulador como este que permite testar a resposta da aplicação desejada sob o comando do microcontrolador para um arquivo *.hex*, ou seja, já no formato a ser gravado no PIC real pois, uma vez funcionando a simulação, sabe-se que o algoritmo está correto para a prática a que se destina. Resultados satisfatórios, semelhantes aos do MATLAB, passam a confiança necessária para que este simulador seja inserido em salas de aula e forneçam apoio aos projetos de professores e alunos.

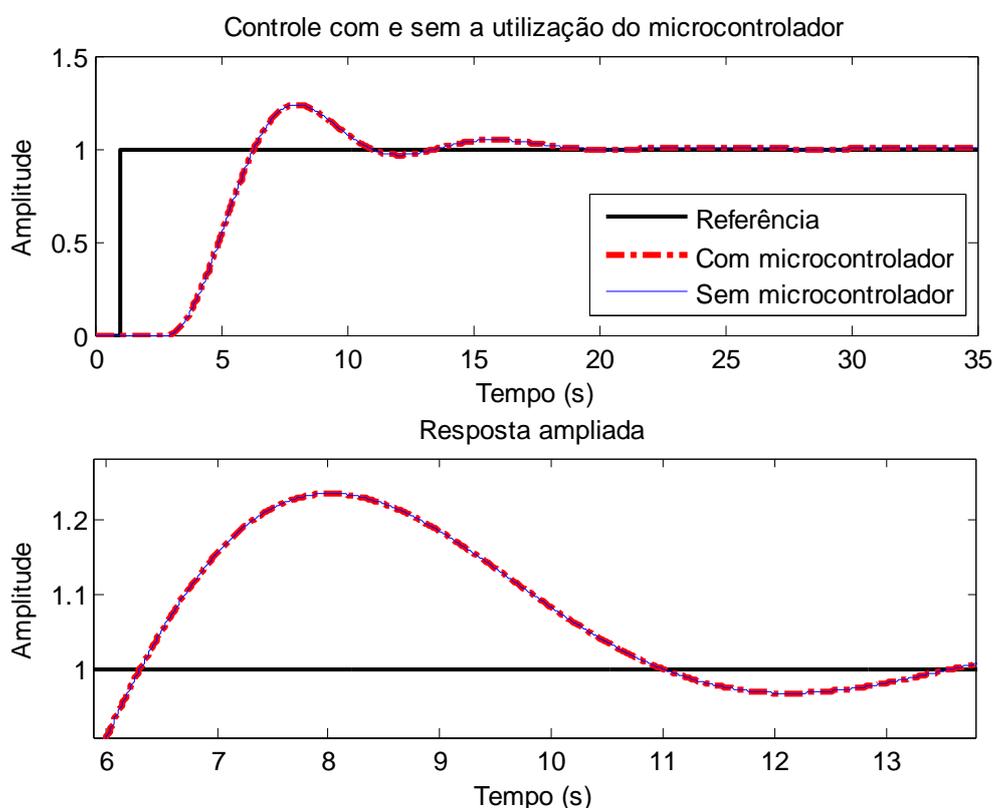


Figura 8 – Controle do sistema pelo *PID Controller* e pelo microcontrolador

1 CONCLUSÃO

A grande vantagem de se utilizar este simulador é poder testar o algoritmo feito para o microcontrolador, já no formato que ele será gravado. Uma vez testado o programa contendo as rotinas a serem inseridas no microcontrolador é possível utilizar o mesmo código em um dispositivo eletrônico real sem a necessidade de se fazer alterações. Poder contar com a previsão da resposta do sistema à aplicação a que se destina fornece segurança para o projeto, podendo evitar inclusive perdas de componentes por comportamento inesperado da saída do microcontrolador.

Os resultados obtidos com o microcontrolador virtual foram satisfatórios, com respostas muito próximas às das simulações sem microcontrolador no MATLAB. Conclui-se então que



o simulador proposto por esse artigo mostrou-se versátil e útil para desenvolvedores e estudantes, podendo ser inserido no ensino de Controle como ferramenta de apoio na formação dos engenheiros nas grades curriculares dos cursos técnicos e superiores

Agradecimentos

Os autores do artigo agradecem ao apoio do MEC/SESu/PET, FAPEMIG, FNDE, CAPES e Fundação CEFET Minas pelo suporte logístico e financeiro à execução deste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

BELHOT, R.V., FIGUEIREDO, R.S., e MALAVE, C.O. O uso da simulação no ensino de Engenharia. *COBENGE*, 2001.

CARMO, M. J. UNIVERSIDADE FEDERAL DE JUIZ DE FORA-UFJF, Ambiente Educacional Multifuncional Integrado para Sintonia e Avaliação do Desempenho de Malhas Industriais de Controle. 2006 Tese (Mestrado).

CARVALHO, A.S., barone, D.A.C., e Zaro, M.A. Uma plataforma tecnológica para o ensino de engenharia de controle. **Revista Renote – Novas Tecnologias na Educação**, v. 7, n. 3, 2009.

FELIPE, A., KELVIN, A., e TEXEIRA, C. Laboratório de experimentação remota com microcontroladores PIC. V CONNEPI, 2010.

GILAT, A., MATLAB® com aplicações em engenharia. Bookman, 2006.

LOPES, L. C. G., PICsim - PIC microcontroller simulator, <http://sourceforge.net/projects/picsim/>, 2011.

MCDERMOTT, K.J , NEDIC Z., NAFALSKI A., MACHHOTKA J. Experiential learning for first year engineering students, In: 10th UICEE, Annual Conference on Engineering Education Bangkok, Thailand, 2007.

RODRIGUES, C.A., MENDONÇA, E.S.B., CHAVES, R.F., e TANOMARU, J. Desenvolvimento de um kit didático para microcontroladores. **Estudos & Pesquisas**, vol. 10, n. 1, 2007.

SILVA, A.N., MARTINS, D.D.S., e NIGRO, I.C. Laboratórios computacionais no ensino de Engenharia de Produção. **Revista INGEPRO – Inovação, Gestão e Produção**, vol. 02, nº 12, 2010.

WANG, Q.G., HANG, C.C., ZOU B. Low-Order Modeling from Relay Feedback. **Industrial & Engineering Chemistry Research**, vol. 36, nº2, 1997.

ZANCO, W.S., Microcontroladores PIC uma abordagem prática e objetiva. Erica, 2005.



MULTIDISCIPLINARY TEACHING OF AUTOMATIC CONTROL MICROPROCESSOR, THROUGH A SIMULATOR FOR MICROCONTROLLER MATLAB,

Abstract: *Use of simulation software in teaching Engineering is essential. Due to the large number of engineering students in colleges and lack of structure, it is virtually impossible to provide accessibility to a real lab for all students. The aim of this paper is to bring discipline Microprocessor to discipline Automatic Control. The goal to link the use of microprocessor the teaching of control, is that this works with discrete time, and can detect curves results in a simplified form, consequently reducing the calculations. Toward that teaching methodology developed a simulation environment for microcontrollers simulation software MATLAB. The simulation is executed by reading the routines to be written to the microcontroller according to the update time to be used during execution. The work aims to help students absorb knowledge better, improving the teaching - learning to contribute to the multidisciplinary approach in the teaching of microcontrollers, Digital Electronics, Digital Control and Advanced Control. The advantages of this simulator is to test the code that the microcontroller is not necessary to mount the circuit and once running the code can be used in a microcontroller with no real problems, breaking the limitation of students to develop their projects only in research laboratories.*

Key-words: *Microcontroller, Simulation, Control, Dynamic Processes, MatLab.*