



SISTEMA DIDÁTICO DE CONTROLE USANDO A PLATAFORMA ARDUINO

Paulo Roberto Brero de Campos – brerocampos@gmail.com

Miguel Sovierzoski – miguelaso@utfpr.edu.br

Universidade Tecnológica Federal do Paraná
Avenida Sete de Setembro, 3165
CEP 80230-901 – Curitiba – Paraná

Carlos Alexandre Brero de Campos – carlosbrero@gmail.com

Instituto de Pesos e Medidas do Paraná – IPEM/PR
Rua Estados Unidos, 135
CEP 82.510-050 – Curitiba – Paraná

Resumo: *a teoria de controle, por ser abstrata, é muitas vezes de difícil compreensão por parte dos alunos. Por isso, é muito importante utilizar recursos didáticos que facilitem a compreensão do assunto. Hoje em dia existem diversas plataformas de hardware e software que permitem testar a teoria de controle em sistemas físicos. O Arduino é uma plataforma que pode ser utilizada com pouco conhecimento de software e de hardware. Neste artigo mostra-se a aplicação de um sistema de controle utilizando o Arduino. Mesmo o professor não tendo à sua disposição um kit didático, ele pode usar o Arduino para estudar o funcionamento de um sistema de controle através de uma rede RC emulando um sistema de primeira ordem, como se fosse, por exemplo, um motor DC. Neste artigo uma rede RC é utilizada como planta e são mostrados os resultados para dois tipos de controladores.*

Palavras-chave: *Arduino, Controle de processo, emulação com rede RC, IMC.*

1. INTRODUÇÃO

Existe uma concordância de que as simulações não podem substituir totalmente experimentos reais, (FEISEL & ROSA, 2005), mas a simulação é uma fase importante na formação dos alunos, pois permite que os alunos compreendam o funcionamento do sistema livre de interferências externas.

Em Fisher et al, (FISHER et al., 2000), foi realizado um estudo com ex-alunos que indicaram que o maior aprendizado ocorreu nas aulas de laboratório.

A simulação é importante para o aprendizado, mas a falta de aplicação do conhecimento em plantas reais tem sido uma falha na formação dos alunos. A solução ideal seria integrar os dois laboratórios, simulação e físico, de forma que o aluno pudesse fazer simulações para



entender o processo e projetar o compensador, para em seguida testar o compensador em um processo físico. Contudo, sistemas didáticos utilizados para o ensino de controle não são baratos e nem todas as universidades possuem equipamentos deste tipo.

Hoje em dia existem diversas plataformas de software e hardware que podem ser utilizados para implementar o controle de sistemas físicos. Entre estas, umas das mais fáceis de usar e mais barata é a plataforma Arduino.

Arduino é uma plataforma aberta de projetos de hardware que se tornou mundialmente popular. Ela foi criada em 2005 no Interaction Design Institute Ivrea, Itália, (ARDUINO1, 2017)

O Arduino é baseado na utilização de placas de microcontrolador de baixo custo e oferece um ambiente de desenvolvimento de software. O ambiente utiliza a linguagem C/C++ para a programação.

Outra vantagem da utilização do Arduino é que não é necessário o uso de equipamentos de medição. O próprio software pode capturar os resultados do experimento e desenhar os gráficos de resposta, ou ainda utilizar o Matlab para fazer a captura dos dados e desenhar os gráficos.

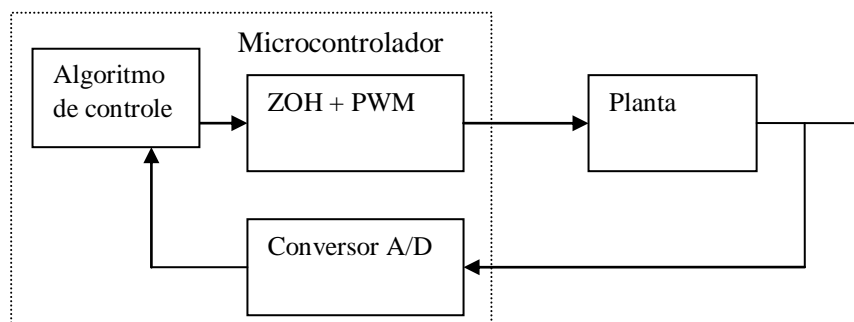
Neste artigo é mostrado um sistema de controle digital com o Arduino, utilizando uma Rede RC como planta.

Este artigo está dividido na seguinte forma: na secção 2 é mostrada a estrutura básica de um sistema de controle digital; na secção 3 faz-se o modelamento dos blocos do Arduino; na secção 4 é mostrado o uso de uma rede RC emulando um motor CC; na secção 5 são mostrados os resultados experimentais e finalmente na secção da conclusão são feitas algumas considerações finais sobre o experimento apresentado neste artigo.

2. ESTRUTURA BÁSICA DE UM SISTEMA DE CONTROLE DIGITAL

Na Figura 1 é mostrada a estrutura básica de um sistema de controle digital, composta por um microcontrolador, um conversor A/D e um PWM. Normalmente os microcontroladores mais simples possuem conversores A/D e circuitos PWM incorporados no próprio circuito integrado.

Figura 1 - Diagrama em blocos do sistema de controle



Fonte: os Autores (2017)

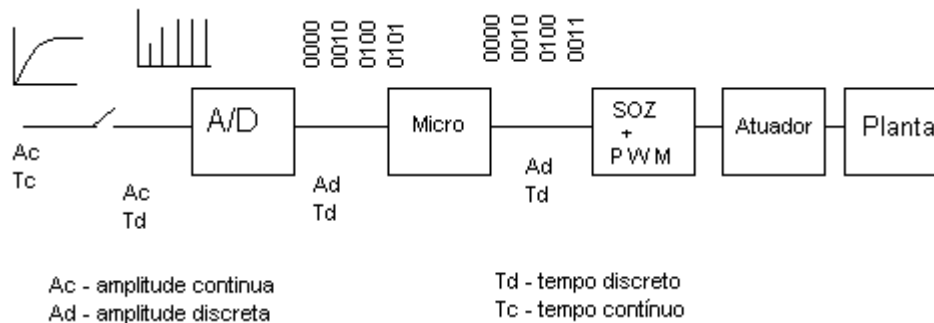
O microcontrolador trabalha sincronizado com um sinal de clock (relógio) interno. As ações acontecem a cada ciclo de clock.

A discretização do sinal contínuo é mostrada na figura 2. O amostrador, indicado no modelo por uma chave, transformar o sinal contínuo em um sinal com amplitude contínua e



tempo discreto, como mostrado na figura 2. O conversor A/D transforma o sinal de amplitude contínua em um sinal de amplitude discreta. O microcontrolador recebe estes sinais e faz o processamento conforme o algoritmo a ser executado. O sinal enviado pelo microcontrolador possui tempo discreto e amplitude discreta. Este sinal é aplicado ao registrador do PWM, atuando como um segurador de ordem zero, mantendo o valor constante durante todo o período de amostragem. O sinal em seguida é aplicado ao atuador (driver) que envia o sinal para a planta.

Figura 2 – Tipos de sinais de um sistema de controle digital



Fonte: os Autores (2017)

Para controlar o processo, o controlador (microcomputador ou microcontrolador) deve:

- 1) Receber as medidas do sistema (conversor A/D).
- 2) Processar estas medidas (algoritmo de controle).
- 3) Enviar os sinais de controle ao atuador, que envia o sinal de saída ao processo.

Na maioria dos sistemas físicos, a planta e o atuador são analógicos e o controlador é digital. Para que o sinal digital possa ser compreendido pelo sistema analógico, é necessário manter esse sinal constante durante o período de amostragem. Para isso é utilizado um registrador. Por ter um registro interno, o PWM é modelado como um segurador de ordem zero, mais um ganho.

O SOZ (segurador de ordem zero) mantém o sinal constante durante o período de amostragem para que ele possa ser reconhecido pelo processo. Devido ao fato dele estar presente na malha de controle, o seu modelo deve ser incluído no modelo do processo. O modelo do SOZ é dado por: $G_{soz} = (1 - e^{-sT})/s$.

Sistemas deste tipo possuem uma parte discreta, o microcontrolador, e uma parte contínua, a planta. A análise do sistema pode ser feita através do equivalente discreto do sistema. Para obter o equivalente discreto do sistema controlado pelo microcontrolador, é necessário calcular a transformada Z do produto das funções de transferência do SOZ e do processo contínuo, como mostrado na equação 1.

$$G(z) = Z \left\{ \frac{1 - e^{-sT}}{s} G(s) \right\} = (1 - z^{-1}) Z \left\{ \frac{G(s)}{s} \right\} \quad (1)$$

3. O ARDUINO UNO

O Arduino Uno é uma placa baseada no microcontrolador ATmega328P. Ele possui 14 pinos de entrada/saída (sendo que 6 podem ser utilizados como saídas de PWM), 6 entradas analógicas atuando como conversores A/D e uma conexão USB. Ele já possui internamente



toda estrutura necessária para fazer qualquer tipo de controle, (ARDUINO2, 2017).

Para programá-lo basta instalar o ambiente Arduino no microcomputador pessoal (PC) e conectar a placa do Arduino a uma entrada USB.

O programa é armazenado em uma EEPROM interna do próprio circuito integrado, de tal forma que uma vez programado ele pode trabalhar sozinho, sem necessidade de estar conectado ao PC.

Para fazer o projeto de controle é necessário modelar cada um dos elementos envolvidos na malha do sistema. Assim, é necessário entender o que representam os blocos PWM e o conversor A/D para a modelagem do sistema.

3.1. MODELAMENTO DO MÓDULO PWM

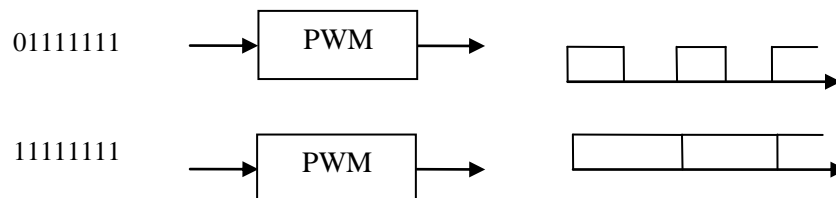
A sigla PWM é o acrônimo de Pulse Width Modulation, que significa Modulação por Largura de Pulso, e é um bloco que fornece um sinal cujo valor médio é proporcional ao valor da contagem aplicada à entrada.

O Aduino possui saídas PWM com frequência de 500 Hz (pinos 3, 9, 10, 11) e com frequência de 1000 Hz (pinos 5 e 6).

O valor máximo que pode ser aplicada na entrada do PWM é igual a 255 (equivalente a 8 bits).

Na figura 3 pode ser visto o resultado na saída do PWM em função do valor aplicado na entrada. Por exemplo, ao aplicar um valor igual a 127 na entrada, a largura de pulso (duty cycle) será igual à 50 por cento do ciclo total. Para um valor de 255, a largura de pulso será de 100 por cento. A tensão média de saída pode ser obtida como: $v_{out} = (\text{duty cycle}/100) \cdot V_{cc}$.

Figura 3 – Característica de funcionamento do PWM



Fonte: os Autores (2017)

Por ter um registro interno mantendo a saída constante, o PWM será modelado como um ganho mais um segurador de ordem zero. Assim o ganho do PWM será igual ao valor médio de saída dividido pelo valor da contagem aplicado no registro do PWM. Isto é, $K = V_o / (\text{valor do registro do PWM})$. Um valor máximo no registro igual a 255 fornece 5 V na saída, então $K = 5/255 = 19,6 \cdot 10^{-3}$.

3.2. MODELO DO CONVERSOR A/D

O conversor A/D possui 10 bits, tendo como resolução $5/1024 = 4,88\text{mV}$. Desta forma, o ganho do A/D é dado por $G_{A/D} = 1024/5$.

Para ajustar o valor lido indicando a tensão na entrada, pode ser usada a relação: $V_{dc} = V_{lido} \cdot 5 / (1024)$



Se esta relação não for utilizada, será necessário acrescentar no modelo o ganho relativo ao A/D.

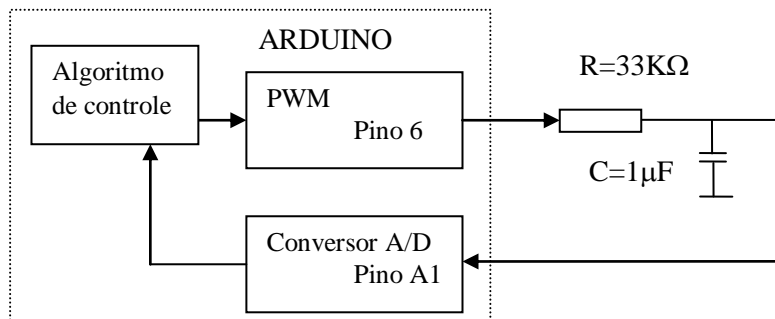
4. USO DE UMA REDE RC EMULANDO UM MOTOR DC

Muitas vezes o professor não dispõe de um modelo físico para testar as estratégias de controle, mas neste caso é possível utilizar uma rede RC emulando a planta física.

A grande vantagem é que este sistema pode ser levado para uma sala de aula teórica e os alunos podem testar sua estratégia de controle apenas alterando o programa do controlador.

Na figura 4 é mostrado o circuito RC acoplado ao Arduino.

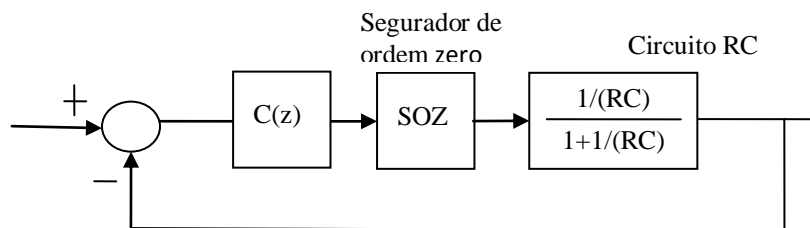
Figura 4 - Diagrama do sistema de controle com uma Rede RC emulando um sistema de primeira ordem.



Fonte: os Autores (2017)

O Modelo do sistema é mostrado na figura 5

Figura 5 - Diagrama em blocos do sistema de controle



Fonte: os Autores (2017)

Neste artigo serão mostradas duas implementações de compensadores para o sistema de controle: uma com compensador proporcional e outra para o IMC (internal model control).

4.1) Compensador proporcional

O compensador proporcional é implementado através de um programa em linguagem C. O programa básico a ser executado é o seguinte:

```
float vin= analogRead(1)* (5.0/1024); //ajusta a tensão lida entre 0 e 5V
e=(ref - vin);
```



```
u=Ke; // ganho proporcional
u=u*255/5; //ajusta tensão enviada para a saída
enviar “u” para saída
```

O comando `analogRead(pino)` é uma função do Arduino que faz a leitura do conversor A/D, sendo o número da porta indicado na própria função (pino).

Para enviar o dado para a porta de saída (no caso o PWM), é usada a função `analogWrite(porta, dado)`, onde o pino de saída do PWM é indicado na porta, e o dado é enviado ao registro do PWM, definindo a largura do pulso.

Mas há uma limitação com relação ao que pode ser enviado para a saída. Como o PWM só aceita tensões positivas, e até o valor de 255, os valores que podem ser enviados estão na faixa entre 0 e 255. Assim é necessário acrescentar as seguintes linhas de código:

```
if (vout>255) { analogWrite(6, 255); vout=255;} if (vout<0) {analogWrite(6, 0);} else
{analogWrite(6, vout);}
```

O equivalente discreto da planta é calculado como $G(z)$, considerando tempo de amostragem $T=1\text{ms}$, como mostrado na equação 2.

$$G(z) = Z \left\{ \frac{1 - e^{-sT}}{s} \frac{G(s)}{1} \right\} = (1 - z^{-1}) Z \left\{ \frac{30,3}{s(s + 30,3)} \right\} = \frac{0,02985}{z - 0,9702} \quad (2)$$

Especificação do compensador proporcional

O compensador proporcional é representado por um ganho: $C(z) = K$.

Para este problema, é especificado que o sistema em malha fechada seja duas vezes mais rápido que o sistema em malha aberta. Isto é: $t_{s5\%(\text{MF})} = t_{s5\%(\text{MA})} / 2$.

Sendo τ a constante de tempo e σ o valor do pólo, o tempo de estabilização é dado por,

$$\text{(OGATA, 1998): } t_{s(5\%)} = 3\tau = \frac{3}{\sigma}$$

Para a rede RC (em malha aberta) $t_{s(\text{MA})}=99\text{ms}$. O valor desejado em malha fechada será $t_{s(\text{MF})}=49,5\text{ms}$.

O valor da constante de tempo em malha fechada será $\tau=t_{s(\text{MF})}/3=49,5/3=16,5\text{ms}$. A localização no plano S será dada por $s = 1/\tau = 1/16,5\text{ms} = -60,6$.

A relação de transformação do plano s para o plano z é dada por: $z = e^{sT}$.

Sendo $T=1\text{ms}$, para $s=-60,6$ o valor de z será: $z = 0,9414$.

A função de transferência em malha fechada é dada por:

O polos em malha fechada é determinado pela equação característica: $1+KG(z)=0$, mostrada na equação 3.

$$z - 0,9702 + k * 0,02985 = 0 \quad (3)$$

Para atender a especificação, a equação característica em malha fechada deverá resultar em um polo, como mostrado na equação 4.

$$z - 0,9414 = 0 \quad (4)$$

Comparando as equações 3 e 4, chega-se ao valor do ganho, $k=0,965$.



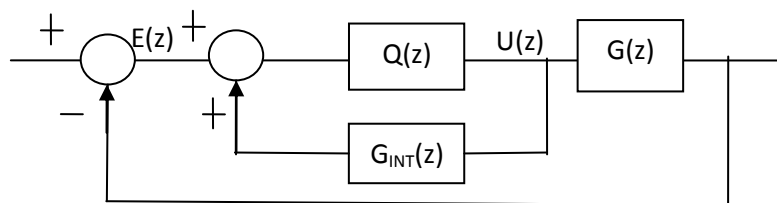
4.2) Controle por modelo interno

As regras de sintonia IMC (*Internal Model Control*) têm se mostrado com características de desempenho e robustez aceitáveis para processos industriais, (CAMPOS & CAMPOS, 2014).

Na estrutura IMC uma cópia da planta é realimentada para a entrada, e é subtraída do sinal de saída (y), como mostrado na Figura 6. Esta configuração é denominada controle por modelo interno, pois um modelo da planta G_{INT} é colocado no laço de realimentação interno, ($G_{INT} = G(z)$), (WOLOVICH, 1994).

Para uma planta de fase mínima, o compensador pode incluir o inverso da planta mais um filtro, de forma que em malha fechada o equivalente resultante contenha apenas o termo do filtro. A estrutura do modelo interno é mostrada na figura 6.

Figura 6 – controle por modelo interno



Fonte: os Autores (2017)

Deseja-se que o compensador tenha uma característica Proporcional-Integral, como mostrado na equação 5, para o período $T=1\text{ms}$.

$$G_{PI}(z) = \frac{Q(z)}{1 - Q(z)G_{INT}(z)} \quad (5)$$

$Q(z)$ é escolhido como sendo o inverso da planta vezes um polo e ganho, como mostrado na equação 6.

$$Q(z) = \frac{1}{G(z)} \frac{K}{z - a} \quad (6)$$

Assim o compensador denominado GPI possui a forma mostrada na equação 7.

$$G_{PI} = \frac{U(z)}{E(z)} = \frac{(z - 0,9702)}{0,0197} \frac{K}{z - a - K} \quad (7)$$

Com visto no item anterior, para $t_{s(MF)}=49,5\text{ms}$, o polo em malha fechada deve estar em $z=0,941$. A função de transferência em malha fechada é dada pela equação 8.

$$G_{MF} = \frac{G_{PI}(z)G(z)}{1 + G_{PI}(z)G(z)} = \frac{K}{z - a} \quad (8)$$

Para ter comportamento integral, na equação G_{PI} impõe-se a condição $a+K=1$ e para atender a resposta no tempo impõe-se $a=0,941$. Assim $K=0,059$.

A equação discreta a ser implementada no programa é mostrada na equação 9.

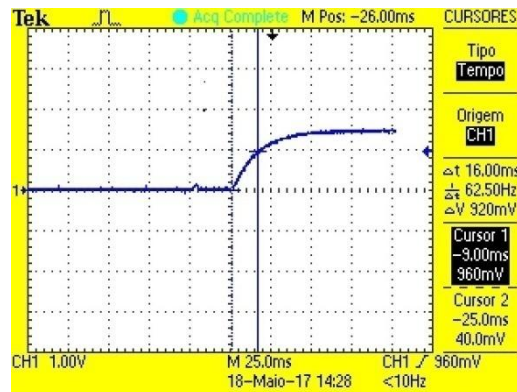
$$u_k = u_{k-1} + 2,994 * e_k - 2,91 * e_{k-1} \quad (9)$$



5. RESULTADOS

A resposta do controlador proporcional é mostrada na figura 7. É possível notar que o tempo de estabilização, $t_s \approx 16,5$ ms, está dentro do especificado. O valor medido da saída resultou em 1,48V, possibilitando calcular o erro em regime prático como sendo igual a 1,52V. O erro em regime obtido pelo cálculo teórico resultou em 1,53V, confirmando a concordância entre teoria e prática.

Figura 7 – Resposta temporal do compensador proporcional.

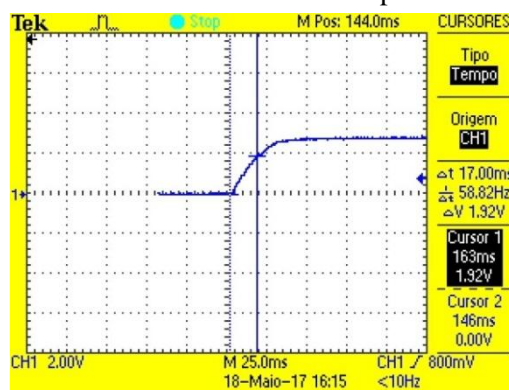


Fonte: os Autores (2017)

As respostas no tempo para o sistema IMC são mostradas nas figuras 8 e 9. Na figura 8, é possível ver que a constante de tempo do sistema compensado está dentro do especificado, aproximadamente $t_s = 16,5$ ms.

Apesar da constante de tempo estar dentro do especificado, o sistema tem uma resposta mais lenta, como mostrado na figura 9, devido ao não cancelamento exato do polo da planta com o zero do compensador, resultando em um resíduo que deixa o sistema mais lento.

Figura 8 – Resposta temporal da estrutura IMC. Figura a: medição da constante de tempo.

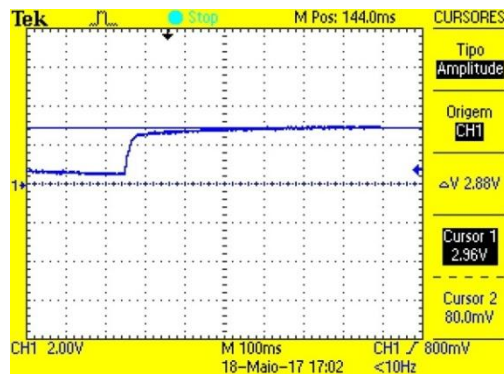


Fonte: os Autores (2017)

Procurando entender o que poderia estar causando a diferença entre o modelo e o sistema real, foi medido o tempo de amostragem através da instrução millis() do Arduino. Apesar de ter sido colocado um delay de 1ms, o atraso real foi de 3ms, diferente do qual foi feito o projeto.



Figura 9 – Resposta temporal da estrutura IMC. Tempo de estabilização mais lento devido ao não cancelamento de polo e zero.



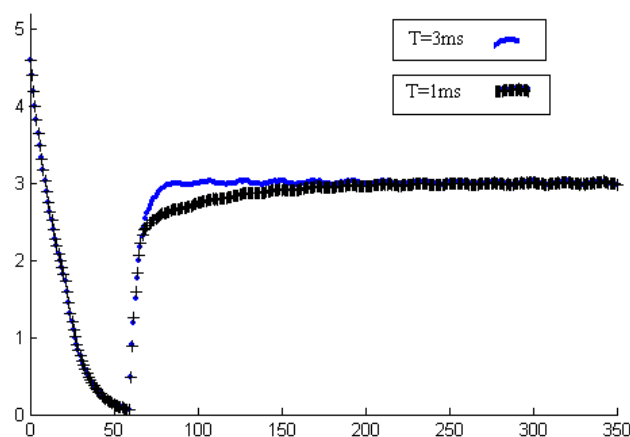
Fonte: os Autores (2017)

O projeto foi feito para $T=3\text{ms}$, resultando no compensador mostrado na equação 10.

$$u_k = u_{k-1} + 1,9131 * e_k - 1,7475 * e_{k-1} \quad (10)$$

Este compensador foi testado e na figura 10 são mostradas as curvas de resposta para o novo compensador (período de amostragem de 3ms) e para o antigo compensador (período de amostragem de 1ms). É possível notar que com o novo projeto para $T=3\text{ms}$ o sistema comportou-se como o previsto. Estas curvas foram obtidas utilizando-se o Matlab para fazer a leitura em tempo real do sinal do Arduino.

Figura 10 – Resposta temporal da estrutura IMC comparando o projeto para dois períodos de amostragem



Fonte: os Autores (2017)

6. CONCLUSÕES

Neste artigo foi utilizada a plataforma Arduino conectada a uma Rede RC como auxílio didático para o ensino de controle digital.

Esta plataforma é bem simples de ser utilizada permitindo que o aluno, mesmo sem conhecimento de hardware de microcontroladores, possa utilizá-la rapidamente.

Neste artigo foram mostradas duas técnicas de projeto de compensadores que podem ser facilmente testadas, auxiliando no entendimento dos conceitos de sistemas de controle: a)



Controle Proporcional e b) Controle por Modelo Interno. Estas técnicas foram implementadas e os resultados práticos obtidos foram condizentes com a teoria.

Foi possível notar que devido ao não conhecimento exato do período de amostragem, mesmo atendendo algumas especificações, o sistema apresentou uma resposta não adequada. Após o projeto ser refeito, utilizando-se o período de amostragem real, o sistema comportou-se segundo as especificações.

7.REFERÊNCIAS BIBLIOGRÁFICAS

Livro

WOLOVICH, W.A. **Automatic Control Systems**, Hartcourt Brace College Publishers, 1994.
OGATA, K. **Engenharia de Controle Moderno**, 3a ed., Rio de Janeiro: Editora Prentice Hall do Brasil, 1998.

Artigo de periódico

FEISEL, L. D. and ROSA, A. J. (2005). **The role of the laboratory in undergraduate engineering education**, Journal of Engineering Education 94(1): 121–130.

FISHER, P. D., FAIRWEATHE, J. S. and HASTON, L. A. (2000). **Establishing learning objectives and assessing outcomes in engineering service courses**, 30th ASEE/IEEE Frontiers in Education Conference.

Trabalhos em eventos

CAMPOS, P.R.B e CAMPOS, C.A.B. **Sintonia de controladores para processos didáticos sem atraso de transporte**, Anais: XLII – Congresso Brasileiro de Ensino de Engenharia. Juiz de Fora: UFJF, 2014. Disponível em <<http://198.136.59.239/~abengeorg/cobenge-2014/Artigos/128518.pdf>>

Internet

ARDUINO1. Disponível em <<https://www.arduino.cc/en/Main/AboutUs>> acesso em 19/05/17

ARDUINO2. Disponível em < <https://www.arduino.cc/en/Main/ArduinoBoardUno>> acesso em 19/05/17

DIDACTIC CONTROL SYSTEM USING ARDUINO PLATFORM

Abstract: *The theory of control, being abstract, is often difficult for students to understand. For this reason, it is very important to use didactic resources that facilitate the understanding of the subject. Nowadays there are several hardware and software platforms that can be used to test the theory of control in physical systems. Arduino is a platform that can be used with little knowledge of software and hardware. This article shows the application of a control system using Arduino. Even if the teacher does not have a didactic model at his disposal, he can use the Arduino to study the operation of a control system through a RC network emulating a first order system, such as a DC motor. In this article an RC network is used as a plant and the results for two types of controllers are shown.*

Key-words: *Arduino, Process control, emulation with RC network, IMC.*

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



Promoção

