



ANÁLISE DOS GANHOS PARA UM SISTEMA DE CONTROLE DIGITAL UTILIZANDO O ARDUINO

Paulo Roberto Brero de Campos – brerocampos@gmail.com

Miguel Sovierzoski – miguelaso@utfpr.edu.br

Universidade Tecnológica Federal do Paraná

Depto. de Eletrônica (DAELN-CT)

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:** para realizar o controle digital de um sistema é necessário conhecer todos os elementos que compõem a malha de controle. Para isso, é preciso identificar os ganhos da malha de controle, tendo vista que os valores da parte digital são definidos tanto pelo programa quanto pela estrutura física. Neste artigo procura-se mostrar como diferentes implementações do programa alteram os ganhos no modelo, levando a respostas diferentes. Para isso, são testadas duas configurações com controle proporcional utilizando o Arduino Uno.*

***Palavras-chave:** Arduino Uno, controle digital, emulação com rede RC.*

1 INTRODUÇÃO

Hoje em dia a teoria de controle digital faz parte do conteúdo das disciplinas obrigatórias dos currículos de diversas engenharias, como, por exemplo, Engenharia Elétrica, Engenharia Eletrônica, Engenharia de Computação, etc.

Em muitos currículos esta disciplina é trabalhada apenas em um enfoque teórico por falta de equipamentos e laboratórios específicos para a implementação de estratégias de controle, pois muitos dos sistemas de controle comerciais possuem um custo proibitivo para diversas faculdades e universidades.

Atualmente existem diversas plataformas de microcontroladores de baixo custo, que são utilizadas tanto para o estudo de microcontroladores quanto para o estudo de sistemas operacionais, mas que podem ser utilizadas para a aplicação dos conceitos de controle digital. Entre essas plataformas pode-se citar o Arduino Uno.

O Arduino Uno é uma plataforma aberta de baixo custo, desenvolvida para ser utilizada por estudantes, independente do seu conhecimento do hardware do microcontrolador. A placa do Arduino Uno é conectada ao PC, através de um cabo USB, sendo programada em



linguagem C/C++, (ARDUINO1, 2017). Devido ao baixo custo, ela pode ser adquirida pelo próprio estudante.

O sistema possui conversores A/D e PWM que fazem a interface com a planta analógica e que devem ser modelados e levados em consideração no momento da definição do programa e da estrutura do modelo.

Conforme se faz a implementação do programa, pode-se obter alterações significativas nos ganhos dos blocos constitutivos do modelo.

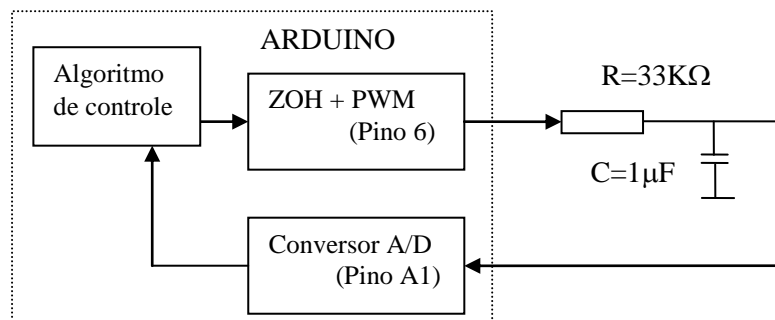
Neste artigo são propostos dois experimentos onde o professor poderá mostrar aos alunos como obter o modelo do sistema a partir do conhecimento da estrutura física dos blocos constituintes do sistema. Como planta foi utilizada uma rede RC (Resistor- Capacitor), devido à sua simplicidade de montagem e facilidade de medição.

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

2 ESTRUTURA BÁSICA DO SISTEMA DE CONTROLE DIGITAL

Na Figura 1 é mostrada a estrutura básica do sistema de controle digital utilizando o Arduino, onde conversores A/D e circuitos PWM estão incorporados no próprio circuito integrado do microcontrolador.

Figura 1 - Diagrama do sistema de controle com uma Rede RC emulando um sistema de primeira ordem.



Fonte: os Autores (2017)

Neste sistema a planta é analógica e o controlador é digital. Para que o sinal digital possa ser compreendido pelo sistema analógico, é necessário mantê-lo constante durante cada período de amostragem.

A porta de saída, normalmente sendo composta por um registrador, pode ser modelada como um SOZ (segurador de ordem zero). O SOZ mantém o sinal constante durante o período de amostragem para que ele possa ser enviado ao processo. O registrador está acoplado ao PWM. Assim, o PWM será modelado como um SOZ e um ganho. O modelo do SOZ é dado por: $G_{soz} = (1 - e^{-sT})/s$, (CASTRUCCI e SALES, 1990).

A análise do sistema pode ser feita obtendo o equivalente discreto do sistema. Para isso é necessário calcular a transformada Z do produto do G_{soz} pela função de transferência do processo contínuo, como mostrado na equação 1:



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

3 O ARDUINO UNO

O Arduino Uno é um sistema baseado 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).

A placa é programada usando uma interface USB e o programa é armazenado em uma EEPROM interna ao próprio circuito integrado, de tal forma que uma vez programado ele pode trabalhar sozinho.

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

3.1 Obtenção do modelo do módulo PWM

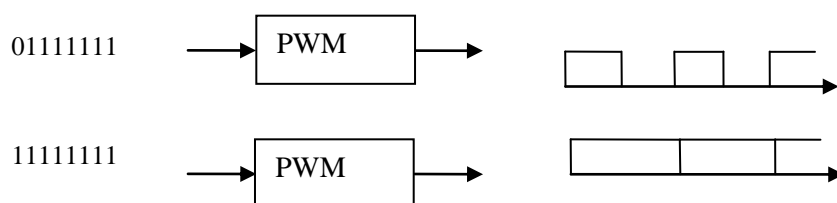
A sigla PWM é o acrônimo de Pulse Width Modulation, que significa Modulação por Largura de Pulso, e é um bloco que fornece um sinal cujo valor médio é proporcional ao valor da contagem aplicada à entrada.

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

A contagem máxima que pode ser aplicada na entrada do PWM é igual a 255 (equivalente a 8 bits). Este valor é armazenado em um registrador, que é modelado como um segurador de ordem zero.

Na figura 2 pode ser visto o resultado na saída do PWM em função do valor aplicado na entrada. Por exemplo, ao aplicar uma contagem de 127 na entrada, a largura de pulso (duty cycle) será aproximadamente igual a 50 por cento do ciclo total. A tensão média de saída será dada por: $V_o = (\text{dutycycle}/100) * V_{cc}$.

Figura 2 – Característica de funcionamento do PWM



Fonte: os Autores (2017)

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 registrador igual a 255, fornece 5 V na saída, resultando em um ganho $K = 5/255$.



3.2 Modelo do conversor A/D

A resolução de um conversor A/D é dado por (tensão máxima de saída)/(2ⁿ), onde n é o número de bits do conversor, (MEASUREMENT COMPUTING, 2017). O Arduino Uno possui um conversor A/D de 10 bits, isto é, n=10, e 5V de tensão máxima, assim, a resolução será igual a 4,88mV.

O ganho do A/D é dado por G_{A/D}=1024/5, sendo o inverso da sua resolução.

Para ajustar o valor lido em termos de tensão pode ser utilizada a relação: V_{dc}=V_{lido}*5/(1024).

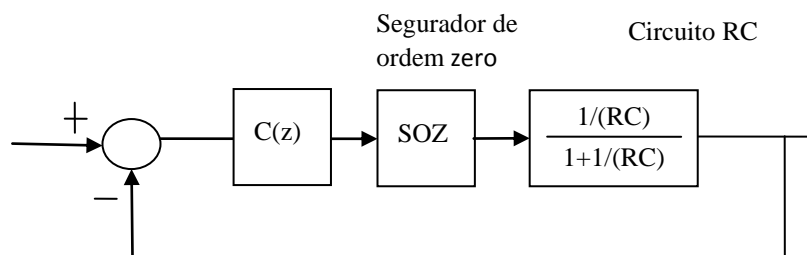
Se esta relação não for utilizada, será necessário acrescentar no modelo o ganho relativo ao A/D.

4 TESTE DOS GANHOS EM UMA REDE RC EMULANDO UM SISTEMA FÍSICO

O objetivo é utilizar um sistema que seja de baixo custo e fácil montagem. Por isso, como planta será utilizada uma Rede RC (Resistor-Capacitor), emulando um sistema físico, como, por exemplo, um motor CC.

O Modelo do sistema é mostrado na figura 3

Figura 3 - Diagrama em blocos do sistema de controle



Fonte: os Autores (2017)

O equivalente discreto da planta com o SOZ, denominado G(z), considerando tempo de amostragem T=3ms, é mostrado na equação 2:

$$G(z) = Z \left\{ \frac{1 - e^{-sT}}{s} \frac{G(s)}{1} \right\} = (1 - z^{-1}) Z \left\{ \frac{30}{s(s + 30)} \right\} = \frac{0,08607}{z - 0,9139} \quad (2)$$

O comando analogRead(Entrada) é uma função do Arduino que faz a leitura do conversor A/D, sendo que a definição de qual conversor é usado está indicado na própria função (Entrada).

Para enviar o dado para o PWM, é usada a função analogWrite(Porta, dado), onde o endereço do PWM usado é indicado na própria função (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 programa para limitar a tensão aplicada ao PWM:

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



Para este sistema serão realizados dois experimentos, um sem ajuste dos sinais lido/escrito e outro com ajuste.

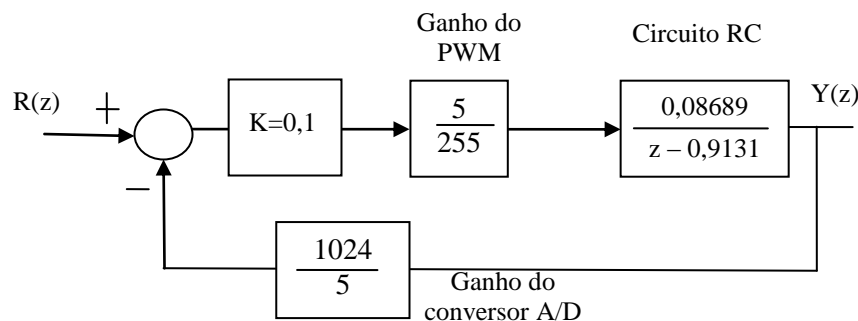
4.1 Leitura e escrita dos sinais sem qualquer ajuste

Para este experimento o compensador proporcional é implementado através de um programa em linguagem C, sem nenhum ajuste nos valores lido/escrito. O programa básico a ser executado é o seguinte:

```
float vin = analogRead(1); //leitura da saída da planta
e = (ref - vin);
u = Ke; // para este experimento K = 0,1
if (vout>255) {analogWrite(6, 255); vout=255;} if (vout<0) {analogWrite(6, 0);} else
{analogWrite(6, vout);} // envia “u” para saída, limitando entre 0 e 255
```

O modelo do sistema usando esse programa é mostrado na figura 4.

Figura 4 - Diagrama em blocos da primeira configuração sem ajuste nos valores lido/escrito.



Fonte: os Autores (2017)

Para esse sistema o ganho será definido como $K=0,1$. O valor da referência deverá estar entre 0 a 1024.

A função de transferência em malha fechada é mostrada na equação 3.

$$\frac{Y(z)}{R(z)} = \frac{0,003375}{z - 0,8782} \quad (3)$$

Cálculo do erro em regime

O cálculo da constante de erro de posição (K_p) é mostrada na equação 4.

$$K_p = \lim_{z \rightarrow 1} 0,1 \frac{5}{255} \frac{1024}{5} \frac{0,08607}{z - 0,9139} = 0,4014 \quad (4)$$

O erro em regime pode ser obtido a partir de K_p , para um valor de referência $R=512$, como mostrado na equação 5.

$$erro = \frac{R}{1 + K_p} = 365,34 \quad (5)$$



Como não é possível medir o erro com o osciloscópio, será feita a medição da tensão de saída.

Pela função de transferência mostrada na figura 4, é possível calcular o valor da tensão de saída, obtendo-se $y=0,7164$.

Cálculo do tempo de estabilização em malha fechada

O tempo de estabilização no plano s é dado pela equação 6, (OGATA, 1998):

$$t_{s(5\%)} = 3\tau = \frac{3}{\sigma} \quad (6)$$

Onde, τ é a constante de tempo e σ é o valor do polo em malha fechada.

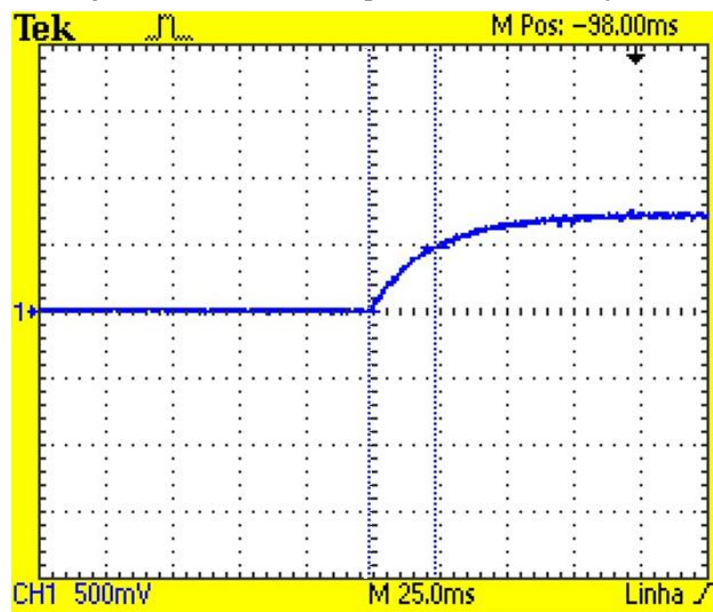
A relação de transformação do plano s para o plano z é dada por $z = e^{sT}$. O polo em malha fechada está localizado em $z=0,8782$, que equivale no plano s ao valor de $s=-43,29$. A partir do valor do polo no plano s , obtém-se $t_s=69,3\text{ms}$ e $\tau=23,1\text{ms}$.

Medidas práticas

As medições foram efetuadas utilizando o programa do Arduino mostrado no início da seção 4.1. A forma de onda do sinal de saída é mostrada na figura 5. O valor da tensão de saída em regime permanente medida foi igual a $0,72\text{V}$.

A constante de tempo medida foi de $\tau=26\text{ms}$ e o tempo de estabilização foi de $t_{s(5\%)}=78\text{ms}$.

Figura 5 - forma de onda para o sistema sem ajuste.



Fonte: os Autores (2017)

É possível comparar os valores teóricos com os valores medidos através da tabela 1, onde pode-se observar que os valores calculados e teóricos ficaram próximos.



Tabela 1 - comparação dos valores calculados e medidos.

	Valor calculado	Valor medido	Erro
Tensão de saída	0,7164 V	0,72 V	0,5 %
Constante de tempo	23,1 ms	25 ms	8,2 %

Fonte: os Autores (2017)

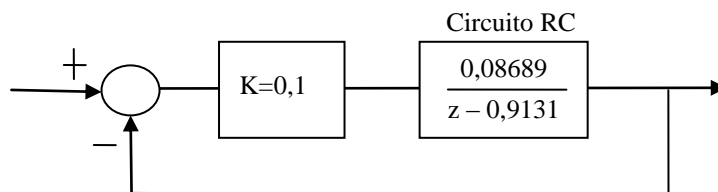
4.2 Ajuste nos valores lidos e enviados pelo microcontrolador

O compensador proporcional foi implementado através do seguinte programa, no qual são feitos ajustes nos sinais lidos e escritos:

```
float vin= analogRead(1)* (5.0/1024); // ajusta-se o valor lido entre 0 e 5
e = (ref - vin);
u = Ke; //K=0,1
u = u*255/5; //ajusta-se o sinal que será enviado para a saída
if (vout>255) { analogWrite(6, 255); vout=255;} if (vout<0) {analogWrite(6, 0);} else
{analogWrite(6, vout);} // envia “u” para saída
```

Como os ganhos são ajustados no programa, os ganhos dos blocos A/D e PWM não necessitam ser colocados no modelo. Na figura 6 é mostrado o diagrama em blocos para essa configuração.

Figura 6 - Diagrama em blocos da segunda configuração.



Fonte: os Autores (2017)

Será utilizado o mesmo valor de ganho definido para o exemplo do item 4.1, isto é, $K=0,1$. O valor da referência deverá estar entre 0 a 5.

A função de transferência em malha fechada é mostrada na equação 7.

$$\frac{Y(z)}{R(z)} = \frac{0,008607}{z - 0,90529} \quad (7)$$

Cálculo do erro em regime

A forma de calcular a constante de erro de posição (K_p) é mostrada na equação 8.

$$K_p = \lim_{z \rightarrow 1} 0,1 \frac{0,08607}{z - 0,9139} = 0,1 \quad (8)$$

O erro em regime pode ser obtido a partir de K_p , para $R=5$, como mostrado na equação 9.

$$erro = \frac{R}{1 + K_p} = 4,545 \quad (9)$$



Como não é possível medir o erro com o osciloscópio, será feita a medição da tensão de saída.

Pela função de transferência mostrada na figura 6, é possível calcular o valor da tensão de saída, obtendo-se $y=0,455$.

Cálculo do tempo de estabilização em malha fechada

O tempo de estabilização é calculado usando a equação 6, onde τ é a constante de tempo e σ é o valor do polo em malha fechada.

A relação de transformação do plano s para o plano z é dada por $z = e^{sT}$. O polo em malha fechada está localizado em $z=0,90529$, que equivale no plano s ao valor de $s=-33,17$. A partir do valor no plano s é possível obter $t_{s(5\%)}=90\text{ms}$ e $\tau=30\text{ms}$.

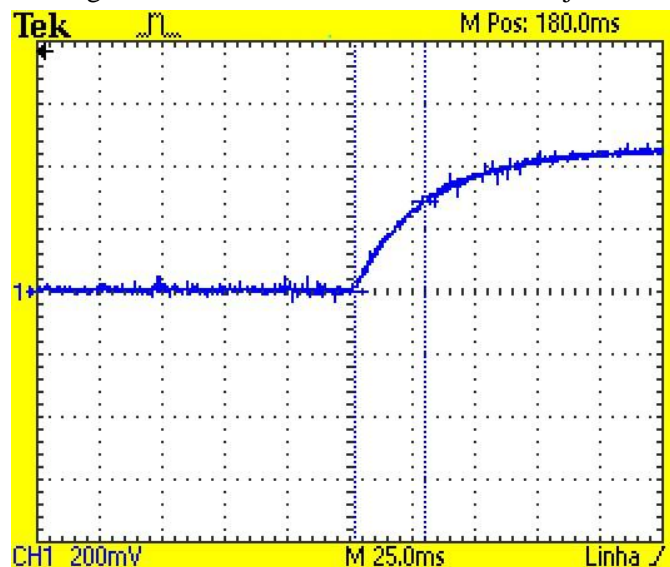
Medições

As medições foram efetuadas utilizando o programa do Arduino mostrado no início da seção 4.2.

A forma de onda do sinal de saída é mostrada na figura 7. O valor medido da tensão de saída em regime permanente foi de $0,456\text{V}$.

A constante de tempo medida através da figura 7 resultou em $\tau=28\text{ms}$ e o tempo de estabilização de $t_{s(5\%)}=84\text{ms}$.

Figura 7 - forma de onda do sistema com ajuste.



Fonte: os Autores (2017)

É possível comparar os valores teóricos com os valores medidos através da tabela 2, onde é possível observar que os valores calculados e teóricos ficaram próximos.

Tabela 2 - comparação dos valores calculados e medidos.

	Valor calculado	Valor medido	Erro
Tensão de saída	0,455 V	0,456 V	0,22 %
Constante de tempo	30 ms	28 ms	6,7 %

Fonte: os Autores (2017)



5 CONCLUSÕES

Neste artigo foi mostrado como obter o modelo do Arduino Uno atuando em uma malha de controle, utilizando-se um conversor A/D e um PWM como interfaces com o mundo analógico.

O conversor A/D foi modelado como um ganho.

O PWM foi modelado como um Segurador de Ordem Zero e um ganho.

Foi mostrado como calcular os ganhos destes blocos, que dependem do número de bits que os compõem.

O compensador proporcional é realizado pelo programa. A forma como o programa é implementado irá definir a necessidade ou não de se incluir os ganhos dos blocos A/D e PWM no modelo.

Foi mostrado como analisar a forma final do modelo, dependendo do programa que é executado.

As medições foram comparadas com os valores calculados e, através do cálculo do erro entre os valores teóricos e práticos, pode-se verificar que eles ficaram próximos.

REFERÊNCIAS BIBLIOGRÁFICAS

Livro

OGATA, K. **Engenharia de Controle Moderno**, 3a ed., Rio de Janeiro: Editora Prentice Hall do Brasil, 1998.

CASTRUCCI, P. e SALES, R.M. **Controle Digital**, Vol. 3, São Paulo: Editora Edgard Blucher Ltda, 1990.

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

MEASUREMENT COMPUTING. **Analog to Digital Conversion**. Disponível em <<https://www.mccdaq.com/PDFs/specs/Analog-to-Digital.pdf>> acesso em 19/05/17

ANALYSIS OF THE GAINS FOR A DIGITAL CONTROL SYSTEM USING ARDUINO

Abstract: *To implement a digital control system is necessary to know all the elements that are in the control loop. One problem faced in the implementation of digital control is how to identify the systems gains. The values of the digital part are defined by the program and by physical configuration. In this paper is shown that the implementation of the program changes the gains in the model, consequently leading to different responses. The Arduino Uno is used to shown it, and two configurations with proportional control are tested.*

Key-words: *Arduino Uno, digital control, emulation with RC network.*

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



Promoção

