



Pró-Reitoria de Pesquisa  
e Pós-Graduação



12º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2021



Câmpus  
Presidente Epitácio

## ENSINO E IMPLEMENTAÇÃO DE SISTEMAS DE CONTROLE APLICADO A PLANTAS DE BAIXO CUSTO.

Graduanda em Bacharelado em Engenharia de Controle e Automação, Bolsista Projeto de Extensão. IFSP, Campus Guarulhos, carla.beatriz@aluno.ifsp.edu.br.

<sup>2</sup> Mestre em Engenharia da Informação e Docente – IFSP, Campus Guarulhos, rogerio.dantas@ifsp.edu.br.

<sup>3</sup> Mestre em Automação e Controle de Processos e Docente – IFSP, Campus Guarulhos, caio.jacob@ifsp.edu.br.

Área de conhecimento (Tabela CNPq): 3.00.00.00-9 Engenharias

**RESUMO:** Neste trabalho buscamos relatar a montagem do projeto de sistema de controle com circuito RCRCR. O projeto iniciou com a montagem do circuito utilizando componentes passivos como resistores e capacitores e o hardware Arduino Uno IDE peça fundamental do sistema em questão. Após a montagem do circuito, foi necessário a programação do Arduino no programa Matlab e em seguida, a análise da planta através de blocos de ferramenta do Simulink com a entrada e saída de dados programados na montagem do sistema. Os dados foram simulados pelas plantas de projeto utilizando métodos da curva de reação do processo de Ziegler-Nichols em malha aberta e da Autossintonia em malha fechada com aplicação dos PWMs. Aplicando PWM com degrau de 30% o ganho entre a relação do sinal de saída sobre o de entrada foi aproximadamente 0,60 valor esse também identificado no PWM com degrau 50%, porém o valor do tempo de subida teórico encontrado foi de 0,38s diferente do degrau de 50% que foi 0,237s.

**PALAVRAS-CHAVE:** Circuito RCRCR. Matlab. Plantas em Malha Fechada. Plantas em Malha Aberta. Arduino Uno IDE.

**ABSTRACT:** In this work we seek to report the assembly of the design of a control system with RCRCR circuit. The project began with the assembly of the circuit using passive components such as resistors and capacitors and the Arduino Uno IDE hardware, a fundamental part of the system in question. After the assembly of the circuit, it was necessary to program the Arduino in the Matlab program and then to analyze the plant through Simulink tool blocks with the input and output of data programmed in the assembly of the system. The data were simulated by the project plants using Ziegler-Nichols process reaction curve methods in open loop and Autotuning in closed loop with application of PWMs. Applying PWM with a step of 30%, the gain between the ratio of the output signal 0,60 value, which was also identified in the PWM with step 50%, but the theoretical rise time found was 0,38s different from the 50% step which was 0,237s.

**KEYWORDS:** RCRCR Circuit. Matlab. Closed Mesh Plants. Open Mesh Plants. Arduino Uno.

## INTRODUÇÃO

Os sistemas de controle possuem uma grande importância dentro da sociedade. Desde as épocas mais remotas, os gregos utilizavam o meio para descrever o tempo gotejamento de água através de um recipiente. Atualmente, muitas aplicabilidades estão sendo implementadas com o intuito de facilitar, agilizar e aprimorar os processos industriais (NISE, 2012).

A princípio, vamos começar definindo o que é sistema? Um sistema nada mais é que um estudo específico de qualquer matéria retirada ou não da natureza, na qual delimitamos e definimos suas grandezas de interesse para um determinado fim. Controlar sistema requer em identificar essas grandezas que entram, efetuam e atuam no processo de modo a manipulá-las para obter uma saída com o melhor requisito de desempenho (MATSUURA e AFONSO, 2016).

Entretanto, existem dois tipos de estruturas para controlar o sistema, sendo elas: malha fechada e malha aberta. Na primeira malha, a planta de projeto necessita de uma realimentação para corrigir os principais erros eventuais a fim de se estabilizar e chegar no resultado desejado, de acordo com uma

referencia pré-determinada. Já a segunda malha dispõe de uma referência inicial, mas não tem o controle do resultado final, isto é, não sabemos se a resposta chegará com exatidão no referencial (NISE, 2012).

Vamos estudá-las em maiores detalhes durante o desenvolvimento da pesquisa e simulá-las pelo programa do Matlab e Simulink criando uma programação no sinal da planta do circuito RCRCR e a montagem modelo físico no Arduino para que assim, possamos fazer as comparações necessárias.

### MATERIAL E MÉTODOS

Os materiais utilizados na construção do modelo físico do projeto foram: Resistor 1 – 100 Ohms; Resistor 2 – 10k Ohms; Resistor 3 – 10k Ohms; Capacitor 1 – 100 nFarad; Capacitor 2 – 100 uFarad; Jumpers; Protoboard; Arduino Uno.

Durante a montagem do circuito (figura 1), os resistores e capacitores foram utilizados como componentes passivos. Os jumpers foram manuseados na protoboard para a interligação de cada componente e assim que finalizamos o circuito, conectamos-o ao Arduino para a realimentação de entrada ( $V_i$ ) de 5V e após esse procedimento, vinculamos a placa do microcontrolador ao computador para a manipulação dos dados por intermédio da planta de projeto criada no programa Simulink para chegarmos as respostas do sistema ( $V_o$ ).

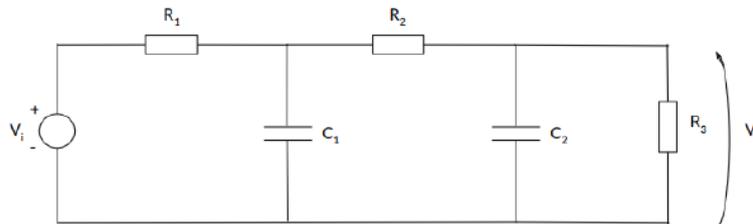


FIGURA 1 – Modelo físico do circuito RCRCR (Autor, 2021).

Para está finalidade, o circuito RCRCR foi resolvido pelo método de Thevenin para chegarmos na função transferencia (FT), modelo matemático relacionado à equação  $V_i/V_o$ . Mas antes disso, foi necessário a princípio, declarar todas as variáveis dentro do Matlab e habilitar o código do plano (s), Domínio de Laplace, para facilitar a manipulação do cálculo dentro do programa. No Matlab, também programamos os intervalos de acionamento dos PWM's (Modulações por largura de pulso) com degrau de 30% e 50%.

### Método de Ziegler-Nichols

Um dos métodos aplicados para o controle da planta em sistema de malha aberta foi de Ziegler-Nichols. Está técnica consiste em encontrar a FT (equação 1) para chegar na resposta do sistema por meio da curva de reação, de acordo com PWM aplicado. Desta forma, a FT encontrada, apresentará um resultado de primeira ordem com certo atraso ou não de transporte do sinal (figura 2), isso em regime permanente. O procedimento presume que a resposta não varie muito o sinal conforme o degrau aplicado, a partir do tempo de início e para isso, optamos por aplicar a regra de sintonia do método junto com o tipo de controlador proporcional integral PI de modo a controlar os parâmetros adequadamente, de acordo com a tabela 1.

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1}$$

EQUAÇÃO 1 - Fórmula da função transferência (Jacob, 2021).



FIGURA 2 - Resposta planta em malha aberta, entrada em degrau (Jacob, 2021).

Tipo de controlador	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5L$

TABELA 1 - Regra de sintonia de Ziegler-Nichols para os tipos de controladores (Jacob, 2021).

### Método de Autossintonia

Outro método aplicado a planta, agora em sistema de malha fechada, e feito na própria ferramenta do Simulink foi de Autossintonia. Para isso, inserimos o bloco do controlador proporcional integral (PI) na malha, modificando o tópico *Automated Tunning* no item *Form Parallel* para o *ideal* para sintonizar a curva do gráfico da resposta do modelo com o bloco da resposta de ganho do controlador PI por meio do ajuste do tempo (ajuste entre lento à rápido) e do comportamento transitório (ajuste de agressivo à robusto) para que a saída seja mais rápida e apresente pouco ou principalmente nenhum sobressinal.

### RESULTADOS E DISCUSSÕES

Será apresentados os resultados obtidos no nosso estudo em 4 momentos.

O primeiro momento foi programado os PWM's por meio do diagrama de bloco *sinhal* em malha aberta (figura 3). Para isso, aplicamos o PWM com 30% na metade do sinal de entrada de 5V no sistema, equivalente ao valor de 1,5V de entrada. Logo depois, calculamos a média das tensões de saída chegando no resultado correspondente a 0,90V e no ganho, obtendo um valor de aproximadamente 0,60. Em seguida, calculamos também o tempo de subida teórico do sinal, correspondendo a 63,2% da amplitude máxima de alcance, encontrando um valor equivalente a 0,5688V. A partir disso, traçamos uma linha vertical no gráfico da resposta diante da tensão aproximada de 0,5816, obtendo o tempo de 0,38s.

Na outra metade do PWM aplicado, com degrau de 50%, fizemos o mesmo procedimento na entrada do sinal, obtendo um valor de 2,5V de entrada. A tensão de saída encontrada foi de 1,5V com ganho de 0,60. O resultado teórico do tempo de subida foi equivalente a 0,237s e a tensão mais próxima resolvida através do cálculo da amplitude máxima foi aproximadamente 0,9680V.

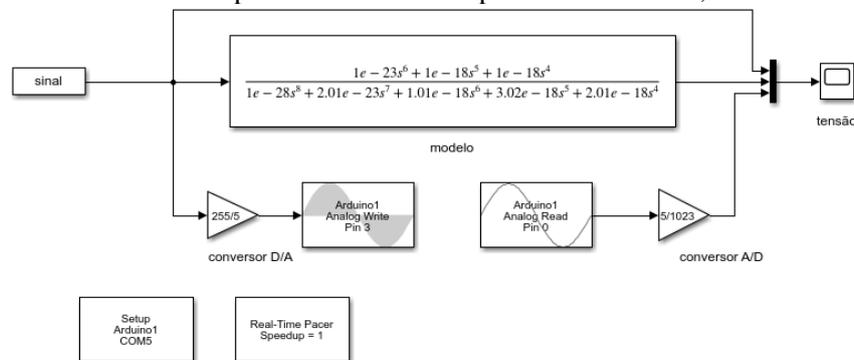


FIGURA 3 - Diagrama de blocos no Simulink – Sistema em malha aberta RCRCR (Autor, 2021).

O segundo momento, a FT da mesma planta física (Figura 1) e programada (Figura 3) foi substituída pela fórmula do método de Ziegler-Nichols junto com os resultados apresentados e o cálculo com os dados  $K = (0,9/1,5)V = 0,6$  (Ganho);  $T_s = 0,388s$  (Tempo de Subida);  $C(s)/U(s) = 0,60/(0,388s + 1)$  (FT) permitiu a descoberta da nova FT e a resposta do sistema do modelo físico e programado.

Em seguida, ensaiamos uma simulação para encontrar a curva de reação do processo de Ziegler-Nichols na planta em malha aberta. Para isso, inserimos o bloco chamado *step* e aplicamos na entrada do sistema com degrau de 2V. Deste modo, geramos um sinal de saída e traçamos uma reta tangente no ponto de inflexão da curva do sinal, encontrando os valores de ganho (K), do atraso de transporte (L) e tempo de subida (T).

Os resultados obtidos pelo método foram aplicados na fórmula FT. Note que será necessário modificar está FT no programa do Matlab para que a ferramenta do Simulink junto a planta em malha aberta pudessem reconhecê-la e para isso, utilizamos o comando *pade()* no programa. Abaixo será apresentada a FT calculada e a FT substituída.

• **Função Transferência**

$$C(s)/U(s) = 0,475 \cdot e^{-0,06s} / (0,20s + 1);$$

• **Função Transferência Substituída**

$$FT = \frac{-0.475 s^3 + 95 s^2 - 7917 s + 2.639e05}{0.2 s^4 + 41 s^3 + 3533 s^2 + 1.278e05 s + 5.556e05}$$

-----

$$0.2 s^4 + 41 s^3 + 3533 s^2 + 1.278e05 s + 5.556e05$$

E para fim de melhorar a resposta do sinal, introduzimos o controlador PI e fechamos a malha da planta (figura 4). Em seguida, calculamos os valores de Kp e Ti (tabela 1) e alteramos esses valores no bloco PID Controller, sendo que o valor de Ti encontrado necessita ser reescrito pelo inverso por ser uma fórmula própria do bloco controlador, chegando no sinal de saída do sistema.

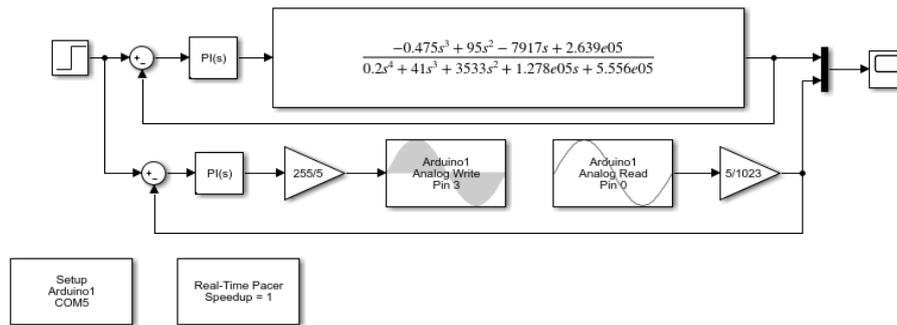


FIGURA 4 – Planta de projeto em malha fechada com função transferência substituída (Autor, 2021).

No quarto e último momento foi realizado o método da Autossintonia no circuito RCRCR em sistema de malha fechada para sintonizar a resposta do sinal do modelo com o bloco de ganho do controlador PI, como é representado na figura 5. Assim, com o bloco PID Controller apresentado na ferramenta do Simulink, conseguimos ajustar o tempo e o comportamento transitório da curva de resposta, visando adquirir uma saída mais rápida e buscando reduzir pelo menos 20% do sobressinal do circuito.

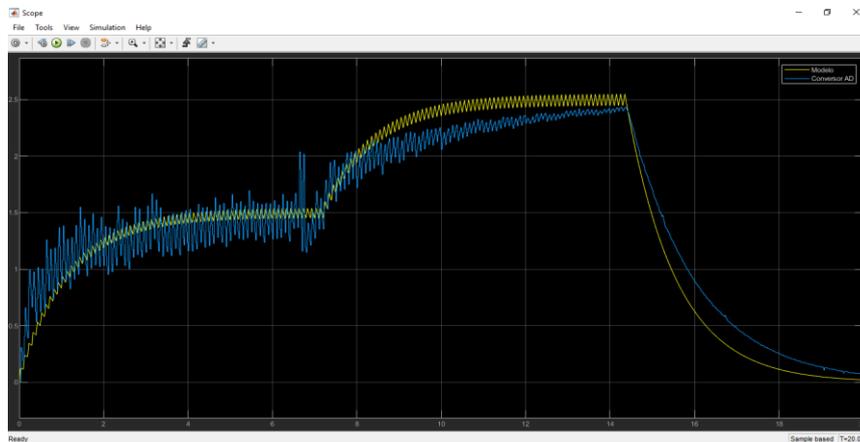


FIGURA 5 - Resposta do sistema em malha fechada (Autor, 2021).

**CONCLUSÕES**

No primeiro momento, é possível concluir que em sistema de malha aberta, a planta do modelo físico e programado não foi capaz de corrigir o erro do controle em regime permanente. Porém, nota-se que o sistema apresentou em sua resposta, um comportamento estável de primeira ordem.

Assim, fechamos a malha no segundo momento e aplicamos o método de Autossintonia no controlador por meio da ferramenta do Simulink, identificando que o mesmo conseguiu corrigir a resposta do sistema, mas uma pequena dispersão no sinal do PWM de 50% foi identificada.

Na terceira etapa, buscamos obter a FT a partir dos valores encontrado no ponto de inflexão da curva junto a reta tangente para analisar e sintonizar a resposta do sinal em malha aberta, utilizando o método da curva de reação do processo Ziegler-Nichols e percebemos que a resposta não sintonizou com a aplicação feita com degrau de 2V por meio do bloco *step*.

Desta forma, fechamos novamente a malha e com a utilização da tabela de sintonia para o controlador proporcional e integral concluímos que somente assim, o sistema controlado foi capaz corrigir e sintonizar seu sinal de saída, demonstrando estável com um pequeno sobressinal.

### **AGRADECIMENTOS**

Superado os desafios iniciais do projeto, agradecemos ao Departamento de Projeto de Extensão do IFSP campus Guarulhos pela oportunidade do estudo em questão e ao 12º Congresso de Iniciação Científica e Tecnológica do IFSP campus Cubatão pelo espaço para apresentarmos a reposta de nossa pesquisa.

### **REFERÊNCIAS**

DORF, Richard C. Sistemas de controle modernos. 8.º ed. Rio de Janeiro: LTC, 2001. 659 p. ISBN 0201308649.

JACOB, Caio C. Controle de Processo II – CPRA7. Aula 6 - 9. 33 slides. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo. Campus Guarulhos. 2020. Disponível em: <<https://ifspedubr.sharepoint.com/sites/CPRA6/Documentos%20Compartilhados/06%20-%20Sintonia%20de%20controladores%20PID/Aula%2006%20%20Sintonia%20de%20Controladores%20PID.pdf>>. Acesso: 29 dez 2020.

NISE, Norman S. Engenharia de sistemas de controle. 3.º ed. Rio de Janeiro: LTC, c2002. 695, 27-38 p. ISBN 8521613016.

MATSUURA, J. P. AFONSO, R. J. M. Introdução ao Controle de Sistemas. Instituto Tecnológico de Aeronáutica. 2016. In: Plataforma Cousera “<https://www.coursera.org/learn/control/home/welcome>”. Acesso em: 26/11/2020.

OGATA, Katsuhiko. Engenharia de controle moderno. 4.º ed. São Paulo: Prentice-Hall do Brasil, 2003. 788 p. ISBN 8587918230.

PINTO, Jan E. M. G. Aplicação Prática do Método de Sintonia de Controladores PID Utilizando o Método do Relé com Histerese. 2014. 21-23 p. Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte. Rio Grande do Norte. 2014. PpgEE:M425. Disponível em: <[https://repositorio.ufrn.br/jspui/bitstream/123456789/15507/1/JanEMGP\\_DISSERT.pdf](https://repositorio.ufrn.br/jspui/bitstream/123456789/15507/1/JanEMGP_DISSERT.pdf)>. Acesso em: 29 dez 2020.