19 a 23 de outubro de 2020
Anais do ENEPE
ISSN 1677-6321

Unoeste

ARTIGOS COMPLETOS

ANÁLISE DE SÉRIES TEMPORAIS APLICADA AOS NÚMEROS DE REGISTROS DE ÓBITOS

César Daltoé Berci

Universidade do Oeste Paulisa – UNOESTE, Presidente Prudente, SP. E-mail: cesarberci@unoeste.br

RESUMO

A análise de séries temporais, aplicada como ferramenta de análise computacional de dados, provê uma série de recursos, capazes, entre outras aplicações, de expressar comportamentos dinâmicos de sistemas complexos, através da identificação de modelos e cálculos de previsões. Esse trabalho tem por finalidade aplicar a análise de séries temporais ao número de registros de óbitos, no Brasil, com intuito de obter previsões e identificar, de forma rigorosa e sistemática, a influência da atual pandemia e suas consequências sobre esses números.

Palavras-chave: Registro de óbitos; Modelagem Computacional; ARIMA.

TIME SERIES ANALYSIS APPLIED TO DEATH RECORD NUMBERS

ABSTRACT

Time series analysis, applied as computational data analysis tool, gives many resources capable, among other things, to express dynamical behavior of complex systems through model identification and forecast. This work aims to apply the time series analysis to death record numbers, in Brazil, in order to obtain forecasts and identify, in a systematic and rigorous way, the influence of the actual pandemic e its consequences about this numbers.

Keywords: Death numbers; Computational modeling; ARIMA.

1. INTRODUÇÃO

Dentre as muitas consequências de uma pandemia, como a que presenciamos atualmente, a contagem do número de óbitos atribuídas ao vírus é um dos fatores que afeta diretamente a percepção da população perante a ameaça. Tal fato corrobora diretamente com a análise pretendida, dada sua significância neste contexto.

Dados oficiais de números de infectados e números de óbitos são disponibilizados, diariamente, pelo ministério da saúde, que contabiliza os números totais a partir de informações repassadas diariamente pelas secretárias estaduais de saúde.

Muitas especulações tomam espaço nos veículos de comunicação a respeito desse tema, e diversas matérias são publicadas, contestando a veracidade dos números, umas afirmando que estes são superdimensionados, ao passo que outras os classificam como subdimensionados.

Dada a relevância desse parâmetro em específico, e no impacto que ele tem sobre a percepção da pandemia, uma análise crítica e sistemática, sob a luz dos acontecimentos e com rigor matemático, se faz necessária.

Dos fatos, entende-se que se refletem em dados e dados em números. Esses números podem ser encontrados em meios oficiais, os quais concatenam toda a base de informação nacional, sendo assim, serão aqui tratados como fato e não como estimativa.

À criação de modelos cabe o rigor matemático, na expressão dos dados por sua própria natureza, auxiliando na compreensão dos fenômenos de origem dos dados.

Portanto, pretende-se examinar, de forma minuciosa, o modelo de origem dos dados de óbitos no país e, a partir dele, construir métodos para determinar a dimensão do número de mortos acometidos pelo vírus SARS-CoV-2.

2. DOS DADOS

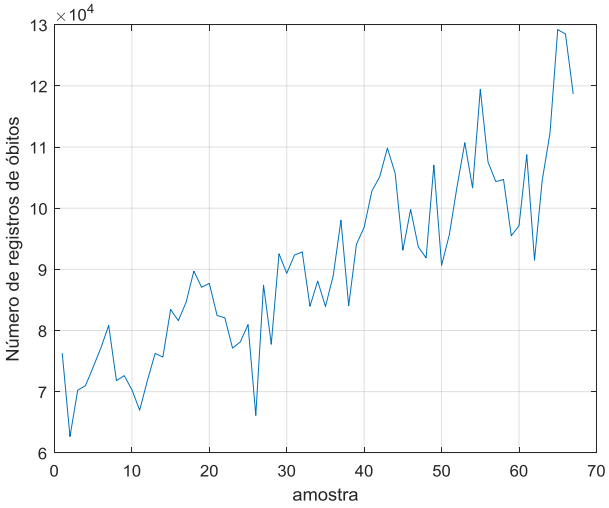
Os dados analisados foram extraídos do registro nacional de óbitos no dia 06 de agosto de 2020, e constam de 67 amostras do número de registro de óbitos, registrados mensalmente no país, tendo início em janeiro de 2015, sendo a última amostra referente a julho de 2020.

Sobre o sistema:

"Publicado em 2018 e mantido pela ARPEN Brasil (Associação Nacional dos Registradores de Pessoas Naturais), o portal de Transparência do Registro Civil é um site de livre acesso, desenvolvido para disponibilizar ao cidadão informações e dados estatísticos sobre nascimentos, casamentos e óbitos, entre outros conteúdos relacionados."

Fonte: https://transparencia.registrocivil.org.br/

Figura 1. Número de casos de óbitos registrados mensalmente, a partir de janeiro de 2015



Fonte: Autor.

3. SÉRIES TEMPORAIS

Neste trabalho, é adotado o conceito de séries temporais estabelecido nas áreas da estatística, econometria, matemática aplicada e processamento de sinais, que as descrevem como sendo coleções de amostras ou observações, indexadas ao tempo, ou ainda, sequenciais.

Uma possível abordagem dessas séries, aqui empregada, estabelecida no campo da econometria e inicialmente estudada por Trygve Haavelmo (SCIENCES, 2003), propõe interpretar os dados como sendo realizações de um processo estocástico.

Neste contexto, o presente trabalho tem por objetivo avaliar as propriedades estatísticas do processo que originou os dados, através das observações disponíveis.

4. AVALIAÇÃO DOS DADOS

As análises aqui descritas foram realizadas com o auxílio do software Matlab.

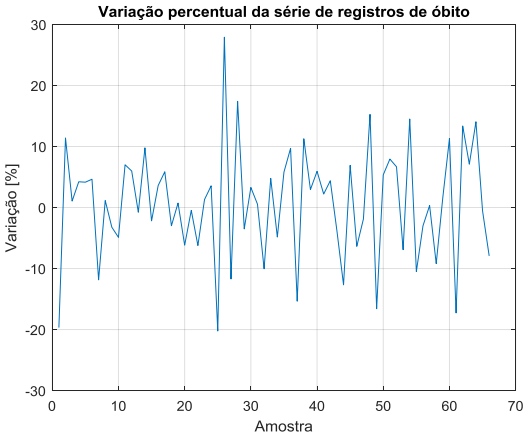
Consideremos incialmente, a primeira diferença entre os logaritmos naturais da série, que, conforme (Tsay, 2002), representa aproximadamente a variação percentual da amostra subsequente.

$$r_i = \log(x_i) - \log(x_{i-1})$$

Uma primeira análise, ainda superficial, pode ser feita através da observação direta da série r, o que se dá através da

Figura **2**

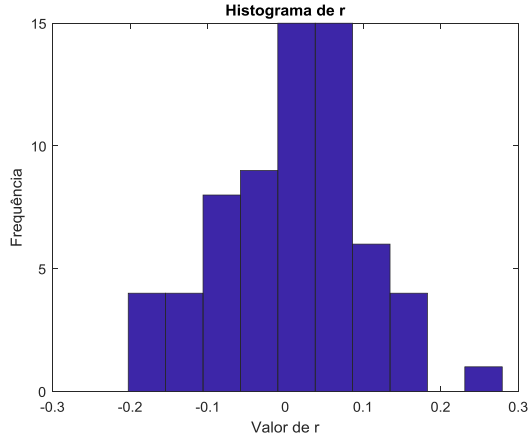
Figura 2. Variações nos registros de óbito



Fonte: Autor.

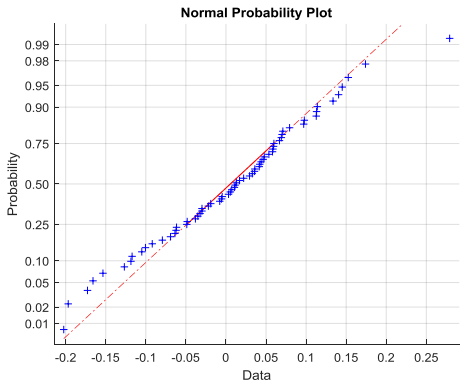
A simples observação do gráfico revela a não existência de uma variação visível nas amostras finais da série, sendo essa, aparentemente, aleatória. Essa conclusão é confirmada analisando a distribuição de frequências de r.

Figura 3. Histograma de r



Fonte: Autor.

Figura 4. Curva de normalidade de r



Fonte: Autor.

Os gráficos da Figura 3 e

Figura **4**, mostram que não há evidências visuais suficientes para rejeitar a hipótese de normalidade na distribuição de r, que apresenta apenas um outlier (27,91% na amostra 26, fevereiro de 2017).

Na tentativa de impor um teste mais rigoroso, além da inspeção visual dos dados, será aplicado uma avaliação que visa identificar se, dado um nível de significância, é possível considerar que os dados foram obtidos de uma distribuição normal. Tal suposição encontra amparo no Teorema do Limite Central.

Aplicando-se o teste de Kolmogorov-Smirnov (Massey, 1951) (Miller, 1956) (Marsaglia, 2003), à série r, obtém-se os seguintes resultados:

$$\rho = 0{,}6349; k = 0{,}0894; c = 0{,}1644$$

Onde ρ é o p-valor, k é a estatística do teste e c é o valor crítico.

Desses valores, conclui-se que, novamente, não há evidências suficientes para rejeitar a hipótese de normalidade dos dados, considerando um nível de significância de 95%.

De tal sorte que, as variações observadas nos números de registros de óbito podem ser consideradas, para fins de análise matemática, como sendo fruto de uma variável aleatória com distribuição normal. Não tendo assim elementos que tornem evidente a ação de um agente externo ao processo nos últimos meses, quando avaliadas as variações.

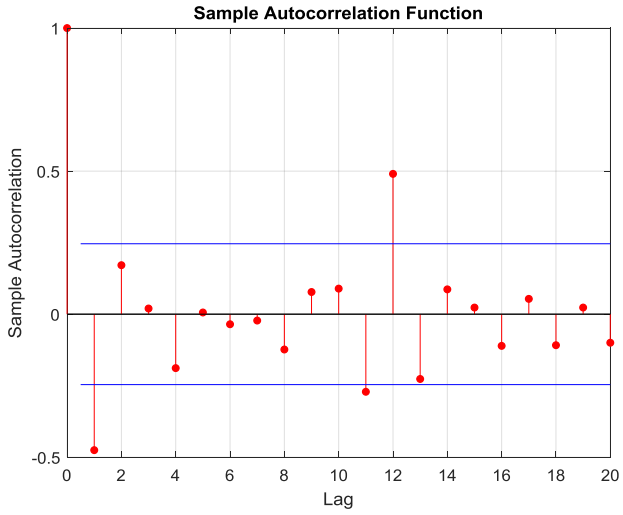
5. MODELO ARIMA (p,d,q)

Como a simples análise das propriedades estatísticas dos dados não revela alterações sensíveis, a tarefa de exame dos dados passa então a construção de modelos, tornando mais criteriosa a análise.

Um modelo, comumente aplicado à avaliação de séries temporais, caso em tela, é o modelo ARIMA: Autorregressivo de médias móveis integrador. A construção desse modelo é realizada, analisando os dados de janeiro de 2015 a dezembro de 2019, ou seja, o período de tempo onde, sabidamente, não há influência do vírus SARS-CoV-2. Após identificado, o modelo é então utilizado para realizar previsões para o período entre janeiro e julho de 2020. Dessa forma, tem-se a estimativa do número de registros de óbitos, sem influência da pandemia.

Inicialmente, a fim de determinar a ordem do modelo a ser estimado, avaliemos as funções de autocorrelação e autocorrelação parcial das variações (Hamilton, 1994) (Box, 1994).

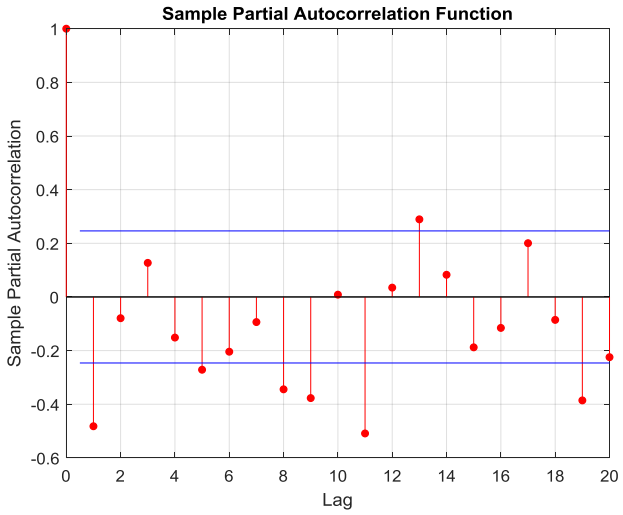
Figura 5. Função de autocorrelação das variações



Fonte: Autor.

A Figura 5 mostra que existe uma contribuição significativa no 12° atraso da série, logo a ordem escolhida para o modelo de médias móveis é 12, $q = 12$.

Figura 6. Autocorrelação parcial de $\boldsymbol{r}$



Fonte: Autor.

A análise da Figura 6 mostra que não existe uma definição clara da ordem do modelo auto regressivo, com base na correlação parcial, portanto adotou-se o valor 2, para reduzir a complexidade numérica do modelo, adequando-o à quantidade de dados disponíveis.

Completando a definição dos parâmetros, é adotado $d = 1$. Tal escolha tem por objetivo, eliminar a tendência de crescimento linear do número de registros.

Portanto, considerando a série de número de registros de óbito X, foi estimado o seguinte modelo (Box, 1994):

$$X_t = ARIMA(2,1,12)$$

$$\left(1 - \sum_{i=1}^{2} \phi L^i\right)(1 - \mathrm{L})X_t = \left(1 + \sum_{j=1}^{12} \theta_j L^j\right)\epsilon_t$$

Onde:

L é o operador de atraso (back-shift), ϕ_i são os parâmetros do modelo autorregressivo, θ_i são os parâmetros do modelo de médias móveis e ϵ_t é um ruído branco:

$$\epsilon_t \approx N(0, \sigma)$$

σ é o desvio padrão do erro.

O modelo foi estimado utilizando o método da máxima verossimilhança (Myung, 2003), resultando nos dados dispostos nas Tabela 1 e 2.

Tabela 1. Modelo ARIMA(2,1,12), parâmetros

Parâmetro	Valor	RMSE
θ_1	$-0{,}81872$	0,49086
θ_2	$-0{,}040972$	0,45502
ϕ_1	0,10017	0,46246
ϕ_2	$-0{,}26633$	0,33509
ϕ_3	$-0{,}042459$	0,28221
ϕ_4	$-0{,}49824$	0,30569
ϕ_5	$-0{,}28796$	0,46896
ϕ_6	$-0{,}30221$	0,35133
ϕ_7	0,088626	0,42502
ϕ_8	$-0{,}085268$	0,35876
ϕ_9	0,26915	0,34723
ϕ_{10}	$-0{,}14197$	0,41057
ϕ_{11}	$-0{,}17483$	0,32725
ϕ_{12}	0,76108	0,35314
σ	$5{,}66E7$	0,0080644

Fonte: Autor.

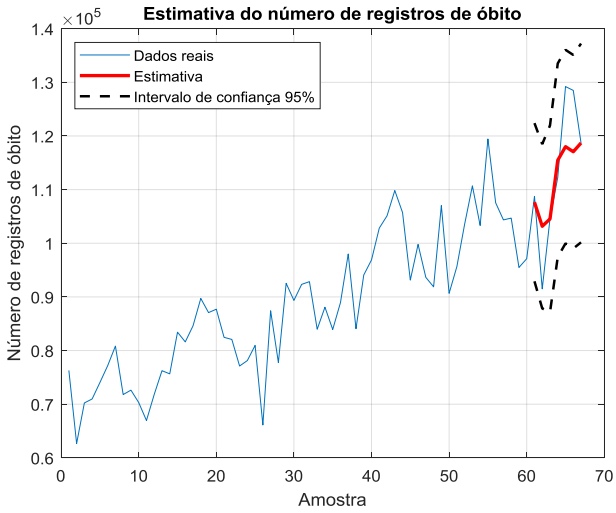
Tabela 2. Modelo ARIMA(2,1,12), estatísticas dos parâmetros

Parâmetro	T-Estatística	ρ-valor
θ_1	$-16{,}679$	0,095331
θ_2	$-0{,}090044$	0,92825
ϕ_1	0,2166	0,82852
ϕ_2	$-0{,}79479$	0,42674
ϕ_3	$-0{,}15045$	0,88041
ϕ_4	$-16{,}299$	0,10313
ϕ_5	$-0{,}61403$	0,53919
ϕ_6	$-0{,}8602$	0,38968
ϕ_7	0,20852	0,83482
ϕ_8	$-0{,}23767$	0,81213
ϕ_9	0,77512	0,43827
ϕ_{10}	$-0{,}34578$	0,7295
ϕ_{11}	$-0{,}53423$	0,59318
ϕ_{12}	21,552	0,031147
σ	$7{,}02E10$	0

Fonte: Autor.

Aplicando o método de Monte Carlo para construção de previsões futuras, sete passos à frente, com intervalo de confiança de 95%, é obtido o gráfico mostrado na Figura 7.

Figura 7. Números estimados de registros de óbitos para 2020



Fonte: Autor.

A previsão do modelo, contrastada com os números realizados, demonstra que, claramente, o número de registros de óbitos no ano de 2020 é plenamente descrito pelos dados entre 2015 e 2019, donde conclui-se que, com base na análise das previsões, não há evidências da influência do vírus SARS-CoV-2 no número de registros de óbito observados em 2020.

6. CONSIDERAÇÕES FINAIS

As consequências de uma pandemia global, com base no que nos conta a história, podem ser observadas em diversas áreas, sendo uma delas, e talvez a mais relevante do ponto de vista do impacto sobre as populações, os óbitos por ela causados, lembrados como sua maior marca.

O presente estudo faz uma tentativa de mensurar, de forma clara, consistente, e sistemática, esse efeito sobre a população brasileira.

Os resultados aqui obtidos, porém demonstram não haver evidências de um aumento no número de registros de óbitos, pois as análises aqui desenvolvidas demonstram que os dados, a priori , são suficientes para explicar o número de registros de óbitos em 2020, sem que exista um agente externo atuando sobre esse modelo.

REFERÊNCIAS

Box, G. E. (1994). *Time Series Analysis: Forecasting and Control 3rd.* Englewood Cliffs, NJ: Prentice Hal.
Hamilton, J. D. (1994). *Time Series Analysis.* Princeton, NJ: Princeton University Press.

Marsaglia, G. W. (2003). Evaluating Kolmogorov's Distribution. *Journal of Statistical Software. Vol. 8, Issue 18,*. https://doi.org/10.18637/jss.v008.i18

Massey, F. J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association Vol. 46, No. 253*, 68–78. https://doi.org/10.1080/01621459.1951.10500769

Miller, L. H. (1956). Table of Percentage Points of Kolmogorov Statistics. *Journal of the American Statistical Association. Vol. 51, No. 273*, 111–121. https://doi.org/10.1080/01621459.1956.10501314

Myung, I. J. (2003). Tutorial on Maximum Likelihood Estimation. *Journal of Mathematical Psychology*, 90-100. https://doi.org/10.1016/S0022-2496(02)00028-7

SCIENCES, R. S. (2003). Time-series Econometrics: Cointegration and Autoregressive Conditional Heteroskedasticity. *Advanced information on the Bank of Sweden Prize in Economic Sciences in Memory of Alfred Nobel.* Stockholm, Sweden.

Tsay, R. S. (2002). *Analysis of Financial Time Series.* Willey. https://doi.org/10.1002/0471264105

AUTOMAÇÃO DE DOSAGEM DE INIBIDOR EM PLANTIO DE ALGODÃO

Renan Bonillo Agostini, César Daltoé Berci, Thiago Rodrigues, Henrique Claro Xavier, Bruno Parpinelli Bonfim

Universidade do Oeste Paulista – UNOESTE, Presidente Prudente, SP. E-mail: renan.bonillo1@gmail.com

RESUMO

O Brasil é um grande produtor de fibra de algodão, que é usado em larga escala, por isso é de elevada importância, realizar o controle de seu crescimento através da aplicação de substâncias inibidoras. O volume do agente de controle aplicado à planta é relacionado à altura do algodoeiro. Neste artigo, é apresentado a construção e o funcionamento de um mecanismo de medição de distâncias, utilizando o sensor ultrassônico HY-SRF05 e um dispositivo de leitura angular, acelerômetro ADXL345. Através dos testes realizados em laboratório, foi possível alcançar uma precisão de $0{,}0001m$ até uma distância de $0{,}5m$ para o sensor ultrassônico. Já para o acelerômetro o desvio padrão foi de apenas $\mp 2°$. Este sistema de medição será utilizado para a automatização e controle do volume de inibidores aplicados às plantações de algodão. Desta maneira, proporcionando uma alta precisão na aplicação destas substâncias, o que pode promover a produtividade e rentabilidade do manejo do algodão.

Palavras-chave: Sensor ultrassônico; acelerômetro; automatização.

AUTOMATION OF INHIBITOR DOSING IN COTTON PLANTING

ABSTRACT – Brazil is a major producer of cotton fiber that is used on a large scale, so it is of great importance to control its growth through the application of inhibitory substances. The volume of the control agent applied to the plant is related to the height of the cotton. This article presents the construction and operation of a distance measurement mechanism using the HY-SR05 ultrasonic sensor and an angle reading device, ADXL345 accelerometer. Through laboratory tests, it was possible to achieve an accuracy of 0.0001m up to a distance of 0.5m for the ultrasonic sensor. For the accelerometer, the standard deviation was only ∓ 2 °. This measurement system will be used for the automation and control of the volume of inhibitors applied to cotton plantations. Thus, providing a high precision in the application of these substances, which can promote the productivity and profitability of cotton management.

Keywords: Ultrasonic sensor; accelerometer; automation.

1. INTRODUÇÃO

O Brasil é um grande produtor de fibra de algodão sendo o quinto colocado no ranking mundial, ficando atrás apenas da China, Índia, Estados Unidos e Paquistão. (FAO, 2012). Por isso seu desenvolvimento é de grande importância para o país. Entretanto, sua produção é bem complexa, pois seu hábito de crescimento, originalmente do tipo perene, foi alterado geneticamente através da biotecnologia, a fim de proporcionar elevada produtividade.

Uma consequência com aspectos negativos dessa mudança é que, em algumas determinadas condições, a planta prioriza o crescimento vegetativo, assim resultando em baixa fixação floral e abortamento prematuro de frutos. (ECHER; ROSOLEM; WERLE, 2013).

Ajustar o crescimento vegetativo e reprodutivo do algodão é um grande desafio, no entanto se faz necessário algum mecanismo que realize tal tarefa.

Através de estudos agrícolas, obteve-se uma equação de ajuste do crescimento do algodoeiro em relação à aplicação de substâncias, inibidoras do crescimento. Exatamente neste ponto é nítida a

necessidade de um sistema que ofereça alta precisão quanto à quantidade de inibidor a ser disperso no algodoeiro (ECHER; ROSOLEM; WERLE, 2013).

Dessa forma, este projeto tem por objetivo apresentar o desenvolvimento de um sensor de distância e um acelerômetro. Ambos serão utilizados para o sensoriamento correto da altura da planta, o que será necessário para a automatização e controle do volume de inibidor que será aspergido.

2. METODOLOGIA

Para o desenvolvimento do projeto foi utilizado o sensor ultrassônico HY-SR05 e acelerômetro ADXL345. O projeto foi elaborado em duas etapas:

- Na primeira, foi realizada a programação de um microcontrolador PIC16F1825 considerando as características do sensor ultrassônico adotado e a programação do acelerômetro pela plataforma Arduino UNO;

- Na segunda etapa, foi realizada a construção do circuito com o sensor ultrassônico e demais componentes para testes em laboratório;

- Na terceira etapa, foram feitos os testes com o sensor acelerômetro ADXL345;

- Por fim, na última etapa, foram realizados os ajustes finos de ambos os sensores, afim de proporcionar maior precisão.

2.1 SENSOR ULTRASSÔNICO HY-SR05

O sensor ultrassônico HY-SR05 habilita o microcontrolador a calcular distâncias através da emissão de sinais ultrassônicos, e da leitura do sinal de retorno (reflexo/eco). A distância entre o sensor e o objeto que refletiu o sinal é calculada com base no tempo, entre o envio e leitura do eco. Assim, a atuação do sensor ultrassônico é essencial para o processo do cálculo da dosagem dos reguladores inibidores, quando aplicado ao algodoeiro.

O sensor é constituído por um circuito eletrônico, onde contém um receptor e um emissor acoplados, além de cinco pinos (VCC, Out, Trigger, Echo e GND) como é apresentado na Figura 1.

Figura 1. Sensor Ultrassônico HY-SRF05.

Fonte: (FILIPEFLOP, 2019).

Suas características elétricas e precisão são apresentadas na Tabela 1.

Tabela 8. Dados sensor.

Tensão de trabalho	DC 5V
Corrente de trabalho	15mA
Frequência de trabalho	40Hz
Faixa máxima	4,5m
Faixa mínima	2cm

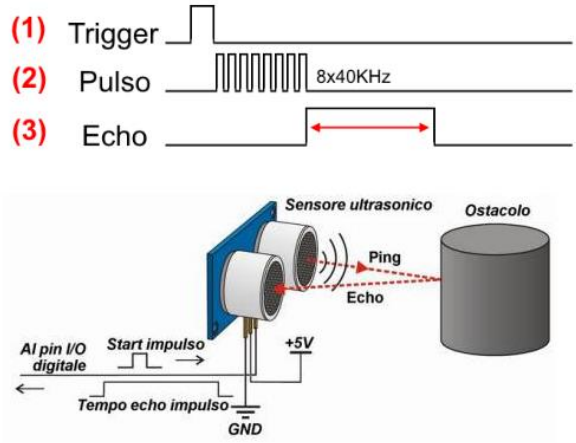
Ângulo de medição	15 graus

Fonte: (ELECFREAKS, 2019).

Há dois modos de operação para este sensor, o primeiro utiliza o pino Out como Trigger e Echo acoplados ao mesmo pino, a fim de economizar um pino no processador. No entanto, neste projeto foi utilizado o segundo modo, que é descrito logo a baixo.

Para a medição o Trigger deve ser colocado em nível alto (high) e receber um pulso de 5V por pelo menos 10µs, proporcionando 8 pulsos ultrassônicos de 40kHz como é mostrado na Figura 2.

Figura 2. Funcionamento do Sensor Ultrassônico HY-SRF05.



Fonte: (MARINESTORE, 2019).

Quando os pulsos são refletidos, o Echo ficará em nível alto e sofrerá uma pausa (delay) de processamento proporcional à distância, que é demonstrada pela seguinte fórmula (NAKATANIL; GUIMARÃES; NETO, 2013):

$$D = \frac{Tv}{2} \qquad (1)$$

onde (D) é a distância [m], (T) o tempo em nível alto [s] e (v) a velocidade do som [m/s], considerada 340m/s.

2.2 MICROCONTROLADOR

O microcontrolador é responsável pelo controle da operação do sensor ultrassônico e cálculo da distância medida através do eco de ultrassom. Para essa tarefa foi utilizado o PIC16F1825, fabricado pela MICROCHIP, pois ele tem uma certa simplicidade e versatilidade na programação, tem um baixo custo em relação aos demais e possui alta velocidade de processamento.

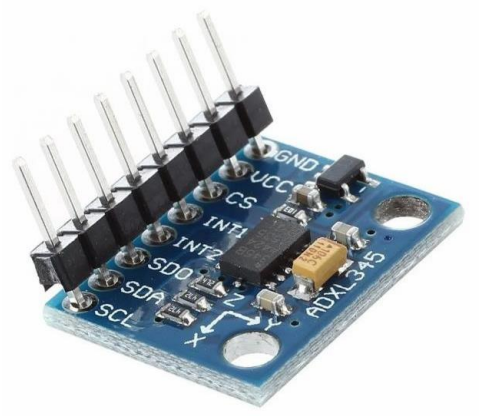
Para elaboração do código foi utilizado o software MPLAB X IDE, desenvolvido pela própria MICROCHIP. Ele oferece um plugin de implementação de drivers para os módulos internos do microcontrolador, denominado MpLab Code Configurator (MCC).

O Trigger, como dito antes, é acionado por pelo menos 10 s, o que produz o sinal ultrassônico. Após a desativação do trigger, o pino Echo vai para nível alto, habilitando a contagem de tempo, a partir da emissão da onda ultrassônica. Quando a contagem do sinal refletido termina, isso significa que a onda colidiu ao objeto e retornou ao módulo receptor, e justamente esse tempo é armazenado. Assim, através de (1), é possível obter a distância percorrida pela onda.

2.3 ACELERÔMETRO ADXL345

O acelerômetro ADXL345 mede a aceleração dinâmica, resultante do movimento e a aceleração estática, por exemplo, a gravidade.

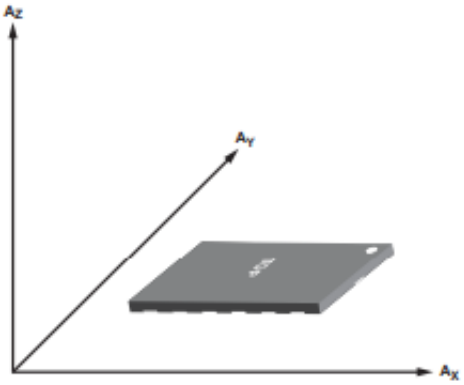
Figura 3. Acelerômetro ADXL345.



Fonte: (Alguém, 2020).

A leitura da gravidade é realizada pela medida da deflexão da estrutura do acelerômetro, a qual utiliza capacitores diferenciais. A aceleração desvia o feixe e desequilibra o capacitor, assim resultando em uma saída, cuja amplitude é proporcional à aceleração.

Figura 4. Representação dos ângulos.



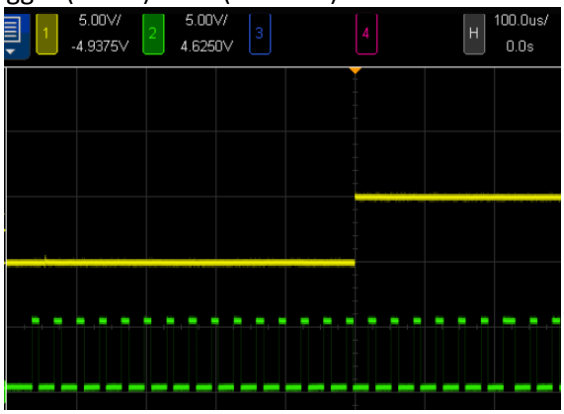
Fonte: (Alguém, 2020).

A comunicação serial utilizada foi a I2C, sendo necessário o uso de 2 pinos SCL e SDA além de habilitar os resistores pull-up em ambos. O módulo contém 6 registradores que possuem os respectivos valores da aceleração nos eixos x, y e z e cada eixo com 2 registradores, sendo necessário a junção dos dois a fim de obter o valor da aceleração no eixo. Para melhor precisão na medição cada eixo contém um registrador de calibração.

3. RESULTADOS

Em laboratório, foram realizados os testes com osciloscópio para o sensor ultrassônico e testes em bancada para o acelerômetro. Como mostrado na Figura 5, é possível visualizar as leituras do pino de Trigger e Echo do sensor ultrassônico.

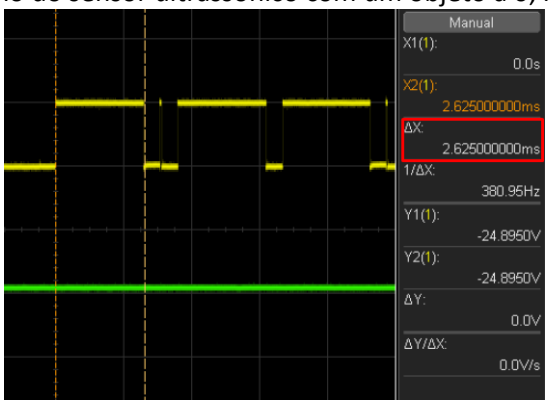
Figura 5. Leitura do pino de Trigger (verde) Echo (amarelo) do sensor ultrassônico.



Fonte: (Autores, 2020).

Neste primeiro momento, através da leitura do Trigger e Echo, é possível observar o funcionamento do sensor ultrassônico, ou seja, a emissão da onda de Echo, quando acionado o Trigger, por pelo menos um intervalo de $10us$. Realizando um teste de leitura, com um objeto a uma distância de $0{,}45m$ do sensor ultrassônico, foi possível extrair o seguinte intervalo de tempo gasto pela onda para percorrer o trajeto, com o auxílio do osciloscópio:

Figura 6. Leitura do pino de Echo do sensor ultrassônico com um objeto a 0,45m de distância.

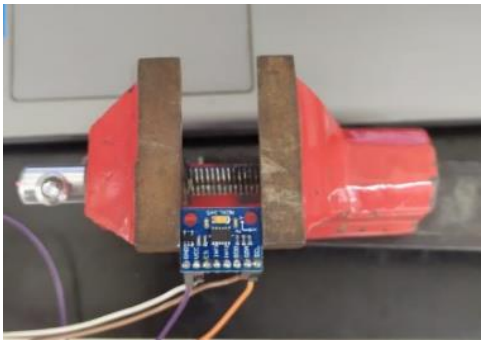


Fonte: (Autores, 2020).

O valor de tempo da onda ultrassônica foi $2{,}6250ms$. Utilizando (1), foi calculado uma distância de $0{,}4501m$ entre o módulo emissor e o objeto.

Para o acelerômetro foram realizados testes em bancada. Na Figura a baixo, são demonstradas as inclinações e orientações lidas pelo acelerômetro através da plataforma de desenvolvimento Arduino UNO.

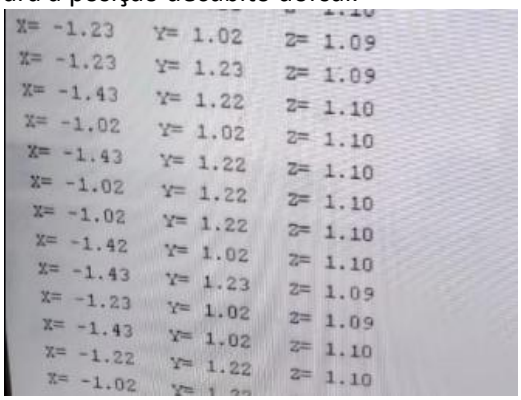
Figura 7. Sensor acelerômetro em decúbito dorsal.



Fonte: (Autores, 2020).

Na figura acima, o acelerômetro foi posicionado para cima, e a saída dos resultados obtidos foram os seguintes:

Figura 8. Leitura acelerômetro para a posição decúbito dorsal.



Fonte: (Autores, 2020).

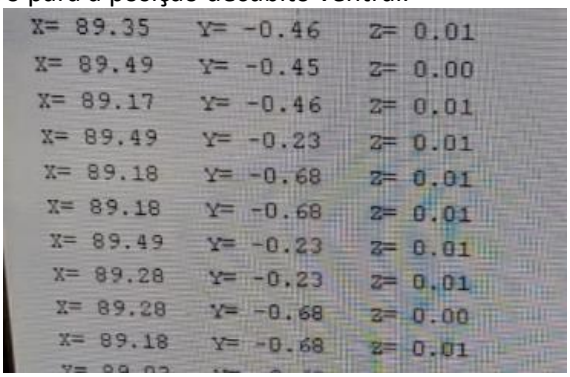
Como visto, as medidas em graus, estão de acordo com o posicionamento do acelerômetro. Ou seja, todas estabilizadas em 1°.

Figura 9. Sensor acelerômetro em decúbito ventral.

Fonte: (Autores, 2020).

Invertendo a posição em 90°, como mostrado na Figura 9, os novos resultados obtidos foram:

Figura 10. Leitura acelerômetro para a posição decúbito ventral.



Fonte: (Autores, 2020).

Já nessa posição, novamente o acelerômetro foi capaz de atualizar os novos ângulos da nova posição, assim ficando com 90° no eixo x.

4. DISCUSSÃO

Para a confirmação da medida lida pelo sensor ultrassônico, foi utilizado uma trena de medição, onde a distância observada foi de $0{,}45m$, ou seja, um erro de apenas $0{,}0001m$.

Após realizado a calibragem do acelerômetro, o menor desvio padrão ajustado foi de aproximadamente $\mp 2°$.

Todos os testes mencionados acima, foram realizados no Laboratório de Instrumentação e Eletro-Eletrônica (IEE) da UNOESTE.

CONCLUSÃO

Através dos testes realizados, constatou-se que o sistema de sensoriamento, composto pelo sensor ultrassônico e acelerômetro, oferece uma precisão satisfatória para sua aplicação. Ou seja, o sistema pode ser utilizado para a automação e controle de aplicação dos inibidores na plantação algodoeira brasileira, auxiliando os produtores no desenvolvimento e qualidade da produção.

REFERÊNCIAS

ECHER, F.R.; ROSOLEM, C.A.; WERLE, R. **Estimativa da dose de regulador a ser aplicada no algodoeiro em função da condição de crescimento.** Instituto Mato-Grossense do Algodão, Cuiabá - MT,n. 01, p. 1-4, jan. 2013.

ELECFREAKS. **Sensor Ultrasonic Ranging Module HY - SRF05,** Disponível em: <www.Elecfreaks.com>. Acesso em: 10 ago. 2019.

FAO. FAOSTAT: **food and agricultural commodities production.** Disponível em: <http://faostat.fao.org/site/339/default.aspx>. Acesso em: 10 ago. 2019.

NAKATANIL, A. M; GUIMARÃES, A. V; NETO, V. M. CIMEC. **Medição com sensor ultrassônico HY-SRF05.** Universidade Tecnológica Federal do Paraná – UTFPR, 11 ago. 2013.

FILIPEFLOP. Sensor de Distância Ultrassônico HC-SR04. Disponível em: < https://www.filipeflop.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>. Acesso em: 10 ago. 2019.

MARINOSTORE. **Sensor Ultrassônico HY-SRF05**. Disponível em: < https://www.marinostore.com/sensores/sensor-ultrassonico-hc-sr04/>. Acesso em: 10 ago. 2019.

IMPLEMENTAÇÃO DE UM ALGORITMO GENETICO EM UM SISTEMA DE CONTROLE

Hugo Gomes Silva, João Victor Pereira da Silva, Bruno Parpinelli Bonfim, Rafael Bratifich, Marcelo Marques da Silva

Universidade do Oeste Paulista – UNOESTE, Presidente Prudente, SP. E-mail: hgs98@outlook.com

RESUMO

Através de um controlador PID pode-se alterar as características da resposta de um sistema, entretanto, a parametrização do controlador torna-se um processo complexo no qual deve-se considerar as características do sistema e da resposta para encontrar os valores de Kp, Ki e Kd do controlador. Atualmente há vários métodos de parametrização do controlador PID, um dos mais conhecidos e aplicados que simplifica o processo de ajuste de seus parâmetros é o método de Ziegler-Nichols. Entretanto, o método de Ziegler-Nichols apresenta um ajuste genérico no qual o ajuste fino do processo é realizado manualmente. Diante deste cenário, propõem-se elaborar um algoritmo genético (AG) a fim de encontrar ajustes factíveis para um controlador PID aplicado a uma planta de 3ª ordem. Para classificar os indivíduos da população do AG considerou-se a minimização do erro em regime estacionário e do sobressinal do sistema ao aplicar-se os parâmetros do PID encontrados no AG.

Palavras-chave: Algoritmos Genéticos, Ziegler-Nichols, Otimização.

IMPLEMENTATION OF A GENETIC ALGORITHM IN A CONTROL SYSTEM

ABSTRACT

Through a PID controller it is possible to change the characteristics of the response of a system, however, the parameterization of the controller becomes a complex process in which one must consider the characteristics of the system and the response to find the values of Kp, Ki and Kd of the controller. Currently, there are several methods of parameterization of the PID controller, one of the best known and applied that simplifies the process of adjusting its parameters is the Ziegler-Nichols method. However, the Ziegler-Nichols method presents a generic adjustment in which the fine-tuning of the process is performed manually. Given this scenario, it is proposed to develop a genetic algorithm (AG) in order to find feasible adjustments for a PID controller applied to a 3rd order plant. In order to classify the individuals in the GA population, it was considered the minimization of the error in steady-state and the system override when applying the PID parameters found in the AG.

Keywords: Genetic Algorithms, Ziegler-Nichols, Optimization

1. INTRODUÇÃO

Os sistemas de controle têm por objetivo alterar parâmetros de interesse em um processo a fim de ajustá-lo ou otimizá-lo. Para tal tarefa utiliza-se um controlador, geralmente um controlador PID - proporcional, integral e derivativo - para atuar na resposta do sistema alterando seu comportamento inicial e habilitando-o a responder conforme as configurações de processo desejadas. Entretanto, o ajuste dos parâmetros do controlador é realizado através das características identificadas do processo, sua função de transferência e a resposta desejada, assim, o ajuste desses parâmetros no controlador torna-se uma tarefa complexa a ser realizada (COSTA, 2019). Dessa forma, existem diversas metodologias aplicadas a encontrar esses parâmetros de ajuste desde as mais complexas, como o ajuste do controlador PID pelo método do lugar das raízes, a metodologias mais simples, como o método de Ziegler-Nichols. Todavia, os métodos mais simples, a nível de parametrização do controlador, apresentam a desvantagem de um ajuste genérico

no qual o ajuste fino do processo é realizado manualmente pelo operador durante a parametrização do controlador (OGATA, 2011).

Diante dessa problemática, podemos modelar esse problema de ajuste dos parâmetros por meio de um problema de otimização para maximizar ou minimizar objetivos descritos como uma função custo (otimização) do problema, tais como: o menor erro em regime estacionário, o máximo sobressinal permitido e o maior tempo de assentamento permitido para o sistema.

Assim, a maximização ou minimização da função custo torna o problema central a ser solucionado. Entretanto, algoritmos genéticos (AG) estão sendo empregados com grande eficiência em problemas de otimização de alto grau de complexidade. A metodologia do AG foi desenvolvida utilizando a teoria de evolução de Darwin, na qual "o ambiente, por meio de seleção natural, determina a importância da característica do indivíduo ou de suas variações, e os organismos mais bem adaptados a esse ambiente têm maiores chances de sobrevivência, deixando um número maior de descendentes. Os organismos mais bem adaptados são, portanto, escolhidos pelo ambiente" (SILVA; PIGNATA, 2014).

O AG busca, portanto, respostas otimizadas para o problema avaliado, sendo que cada indivíduo do AG representa uma resposta em potencial para o problema e através de processos de seleção, cruzamento e mutação presentes no algoritmo – os melhores indivíduos, ou seja, as respostas mais assertivas são selecionadas para que, ao final de vários processos de seleção durante as gerações, o melhor indivíduo seja encontrado representando a possível resposta para o problema a ser otimizado.

Assim, neste trabalho, propõe-se a aplicação do algoritmo genético em um sistema de controle, a fim de ajustar os parâmetros do controlador e comparar a eficiência do controle encontrado em relação ao ajuste proposto pelo método de Ziegler-Nichols para o mesmo processo ajustado no AG.

2. METODOLOGIA

A fim de melhor apresentar alguns conceitos de controle, o algoritmo genético e a metodologia de Ziegler-Nichols, a presente seção foi fragmentada com a seguinte estrutura:

- Na primeira parte, são apresentados alguns conceitos de sistema de controle utilizados para o desenvolvimento do algoritmo.

- Na segunda parte, apresenta-se as principais características do AG e a introdução da função avaliativa (função custo), denominada no AG por função *fitness* que examinará quais são os indivíduos mais adaptados ao meio - problema proposto.

- Na terceira parte, será abordado a metodologia de Ziegler-Nichols utilizada em um controle PID.

2.1 SISTEMA DE CONTROLE

O controle automático é um componente importante e intrínseco de sistemas contínuos e operações industriais que envolvam o controle de elementos como temperatura, pressão, umidade, viscosidade, vazão, etc (OGATA, 2010).

O controle automático é empregado em processos com a finalidade de ajustar parâmetros da resposta do sistema. Um dos controles comumente empregados é o controle PID, devido a sua aplicabilidade geral à maioria dos sistemas de controle e sua atuação proporcional (P), integrativa (I) e derivativa (D), que possibilitam ao sistema reduzir seu tempo de acomodação através da ação do proporcional, reduzir o erro em regime permanente através da ação do integrativo e modificar a posição dos polos dominantes do sistema a fim de implementar as especificações desejadas ao componente transitório da resposta (COSTA, 2019; NISE, 2012).

A função de transferência do controlador PID, no domínio da frequência, é dada por (OGATA, 2010; NISE, 2012):

$$Gc(s) = K_p + \frac{K_i}{s} + K_d s \qquad (1)$$

na qual Kp é o ganho proporcional, Ki o ganho integrativo e Kd o ganho derivativo. Assim, o problema do ajuste dos parâmetros consiste em encontrar os valores desses elementos considerando as características da planta e a resposta que se deseja implantar no sistema.

Dessa forma, nessa proposta empregou-se um controlador PID para controlar uma planta de 3ª ordem. Para ajustar as características desse controlador utilizou-se o algoritmo genético, enquanto, para a comparação dos parâmetros encontrados no AG, foi utilizado o método de Ziegler-Nichols, para encontrar um ajuste ao sistema avaliado. A planta aqui apresentada possui apenas fins didáticos para a demonstração do algoritmo, porém o AG proposto ainda está sendo aprimorado, mas este é capaz de ser aplicado em qualquer sistema PID. A função de transferência do sistema avaliado é:

$$G(s) = \frac{1}{S(S+1)(S+5)} \qquad (2)$$

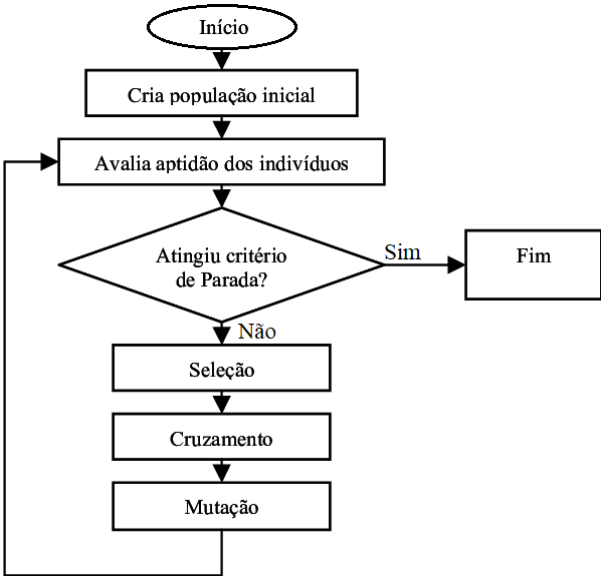
2.2 ALGORITMO GENÉTICO

O algoritmo genético (AG) é uma técnica muito conhecida quanto se trata de otimização de problemas. O AG inspira-se na teoria evolutiva de Charles Darwin. Assim, na concepção do AG é possível observar vários elementos da área biológica presentes na descrição do algoritmo, tais como:

- Cromossomo: um vetor de dados que corresponde a uma possível solução para o problema. Os cromossomos são formados por genes e representam os indivíduos presentes na população;
- Gene: representa características do problema avaliado;
- Fenótipo: representa a característica intrínseca que o indivíduo possui em seu cromossomo frente ao problema avaliado, ou seja, sua adaptabilidade em relação ao meio que seja avaliado;
- População: conjunto de indivíduos representados por cromossomos em uma geração;
- Função de avaliação ou *fitness*: função que avalia o fenótipo de cada indivíduo e classifica-o conforme sua característica frente ao problema analisado.

A aplicação do AG dá-se através do fluxograma apresentado na Figura 1, no qual verifica-se que primeiramente são gerados os indivíduos com suas características, que representam elementos do problema avaliado.

Figura 1. Fluxograma AG



Fonte: (Autores, 2020).

Como deseja-se otimizar o PID, cada cromossomo - que representa um indivíduo da população - será composto por três genes que representam os possíveis valores de K_p, K_i e o K_d do ajuste do controlador PID.

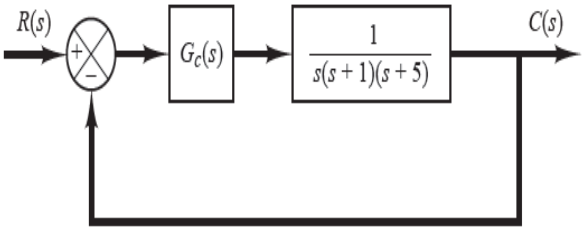
Os valores desses genes são atribuídos aleatoriamente para cada indivíduo. Ao final do processo de atribuição é formada a população, ou seja, um conjunto de possíveis soluções para o controlador através do AG.

Após esse processo, a população passa por uma avaliação realizada através da função *fitness*, na qual verifica-se o fenótipo de cada indivíduo através de seus cromossomos e, a partir desses fenótipos, é possível verificar a qualidade do indivíduo como possível resposta ao problema.

A função *fitness* consiste em simular a resposta do sistema à aplicação de um degrau unitário e, ao fim do processo, armazenar o erro em regime permanente do sistema e o sobressinal.

Para isso, aplicou-se as respostas de Kp, Ki e Kd de cada indivíduo a um controlador PID presente na planta de 3ª ordem, em malha fechada com realimentação unitária, que foi submetida ao degrau unitário como apresentado na Figura 2. Após a aplicação do sinal, armazenou-se o erro estacionário em regime permanente do sistema e seu máximo sobressinal. O erro em regime permanente e o sobressinal do sistema possibilitou classificar os indivíduos da população.

Figura 2. Diagrama de blocos do sistema



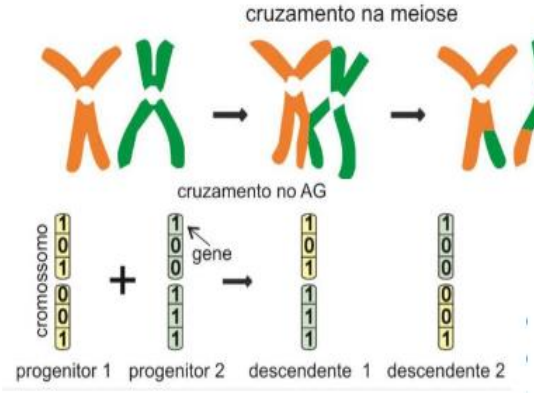
Fonte: (Autores, 2020).

Uma vez classificados os indivíduos através da função *fitness* é realizada a aplicação dos operadores genéticos:

- Seleção: após a avaliação, os indivíduos que apresentarem os menores erros e sobressinais, ou seja, os mais adaptados no sistema, prosseguem ao próximo estágio; os demais elementos serão descartados.

- Cruzamento: depois da seleção, uma porcentagem de 70% da população realiza a troca de suas informações genéticas. O processo consiste em selecionar dois indivíduos sorteados aleatoriamente na população para realizarem a troca de informações genéticas gerando descendentes, esse processo se assemelha como apresentado na Figura 3, originando a nova população ou geração. A porcentagem a ser herdada de cada progenitor, pelo novo individuo é selecionada de forma arbitraria dentro dos parâmetros definidos.

Figura 3. A exemplo ilustrativo é apresentado o operador de cruzamento entre dois individuo de uma população representados por cromossomos com informações binárias



Fonte: (BRATICH, 2018).

- Mutação: nesse processo, uma pequena parcela da população - estipulou-se um valor máximo de 3% - sofrerá uma alteração em um de seus genes que também será sorteado aleatoriamente. A mutação permite ao AG ampliar seu espaço de busca, rompendo sua estagnação ao manter-se próxima de uma solução local que é gerada através da seleção e do cruzamento.

Após o processo de mutação, a nova população é avaliada e o algoritmo prossegue seu ciclo até o momento no qual atinge o critério de parada adotado, que foi um número máximo de gerações. Como o número de indivíduos é reduzido durante a seleção após o processo de mutação, antes de iniciar-se um novo ciclo novos indivíduos são gerados aleatoriamente, a fim de se reestabelecer o número de indivíduos iniciais.

2.3 MÉTODO DE ZIEGLER-NICHOLS

este projeto, também serão ajustadas as constantes do controlador conforme as regras de Ziegler-Nichols. Em conformidade com Ogata (2010), "as regras de Ziegler-Nichols, [...] são úteis quando os modelos matemáticos da planta são desconhecidos". Contudo, também é possível utilizar o mesmo com plantas já conhecidas.

Na teoria de Ziegler-Nichols o ajuste da equação do PID dá-se por:

$$Gc(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

$$= K_p + \frac{K_i}{s} + K_d s \qquad (3)$$

no qual os valores paras Kp, Ti, Td são encontrados através de uma tabela de sintonia apresentada na Tabela 1.

Na referida tabela, o ajuste é realizado através do ganho crítico K_{cr} que produz uma resposta harmônica do sistema em malha fechada a aplicação do impulso ao sistema e P_{cr}, denominado período crítico, que é o período da resposta harmônica (senoidal) encontrada com o ganho crítico aplicado ao sistema em malha fechada. O resultado encontra-se condizente como foi proposto pelo livro referenciado, sendo posteriormente simulado no *software.*

Tabela 1. Regra de sintonia de Ziegler-Nichols baseada no ganho crítico Kcr e no período crítico Pcr (segundo método).

Tipo de controlador	Kp	Ti	Td
P	0,5Kcr	∞	0
PI	0,45Kcr	$\frac{1}{1,2}$Pcr	0
PID	0,6Kcr	0,5Pcr	0,125Pcr

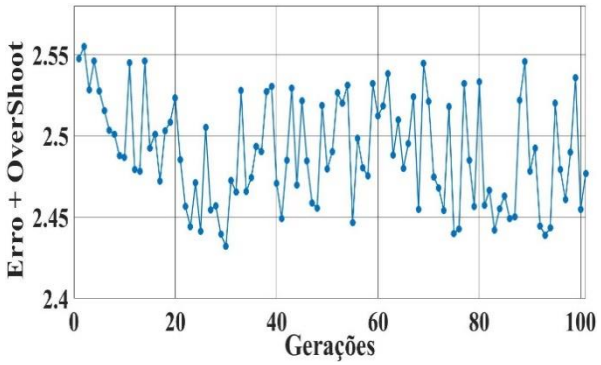
Fonte: (OGATA, 2010).

3. RESULTADOS E DISCUSSÃO

O algoritmo genético foi implementado no *software* MATLAB. Os gráficos a seguir representam o erro em regime permanente adicionado ao sobressinal do melhor indivíduo selecionado como mais apto em cada geração durante a aplicação do AG. Para a aplicação do AG considerou-se populações com 200 e 500 indivíduos, cruzamento em 70% da população, mutação em 3% e critério de parada de 100 gerações.

Na Figura 4 é exposto o erro adicionado ao sobressinal do melhor indivíduo de cada geração para a população de 200 indivíduos.

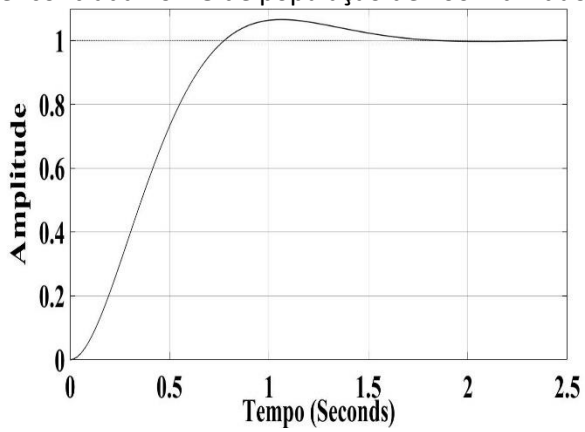
Figura 4. População de 200 indivíduos.



Fonte: (Autores, 2020).

Verifica-se que o erro adicionado ao sobressinal ficou entre 2,43 a 2,56. Após as 100 gerações, foi aplicado o valor de K_p, K_d e K_i do melhor indivíduo ao controlador para simular a resposta ao degrau unitário do sistema.

Figura 5. Resposta ao degrau encontrada no AG de população de 200 indivíduos.

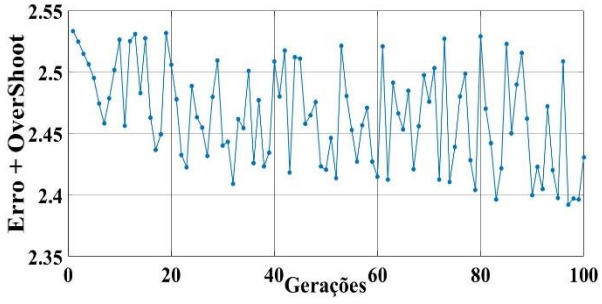


Fonte: (Autores, 2020).

Conforme a Figura 5, é possível verificar um sobressinal inferior a 20%, um tempo de assentamento próximo de 1,5 segundos e um sistema sem erro em regime estacionário.

A Figura 6 apresenta o erro adicionado ao sobressinal do melhor indivíduo de cada geração para a população de 500 indivíduos.

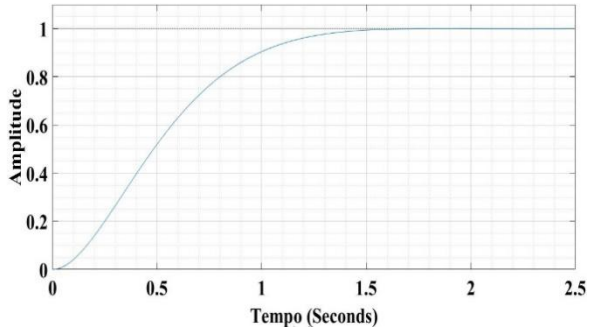
Figura 6. População 500 indivíduos.



Fonte: (Autores, 2020).

Foi possível conferir que o erro adicionado ao sobressinal ficou variando próximo de 2,38 a 2,54. Após as 100 gerações aplicou-se o valor de K_p, K_d e K_i do melhor indivíduo ao controlador, para simular a resposta ao degrau unitário do sistema.

Figura 7. Resposta ao degrau encontrada no AG de população de 500 indivíduos.

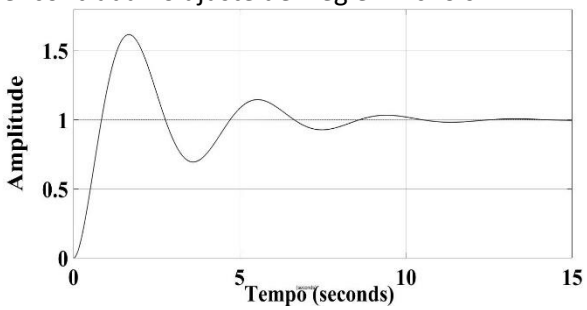


Fonte: (Autores, 2020).

Na Figura 7, não há um sobressinal e um tempo de assentamento próximo de 1,5 segundos e um sistema sem erro em regime estacionário.

Para comparar esses ajustes de PID encontrados pelo AG, foi calculado o ajuste do sistema através do método de Ziegler-Nichols e, posteriormente, aplicado o valor de K_p, K_d e K_i encontrados através da Tabela 1 no controlador para simular a resposta ao degrau unitário do sistema.

Figura 8. Resposta ao degrau encontrada no ajuste de Ziegler-Nichols.



Fonte: (Autores, 2020).

Na Figura 8, apurou-se um sobressinal de 60%, um tempo de assentamento próximo de 10 segundos.

O algoritmo genético, em ambas as situações, apresentou um ajuste com menores sobressinais e tempo de assentamento do sistema em relação ao ajuste encontrado pelo método de Ziegler-Nichols.

A aplicação do AG resultou em 2 ajustes para o controlador, nos quais o ajuste para a população de 500 indivíduos apresentou-se melhor que o ajuste proposto da população de 200 indivíduos considerando-se o critério de sobressinal.

O AG proposto tem por finalidade encontrar os valores de Kp, Kd e Ki para o controlador PID. Dessa forma, verifica-se que tal objetivo foi alcançado, pois, nos ajustes encontrados o sistema apresenta valores aceitáveis de operação e quando se compara aos valores encontrados pelo método de Ziegler-Nichols, observa-se que o ajuste do AG apresenta-se superior para a planta avaliada.

A critério de testes iniciais a avaliação proposta é ampla e considera somente o erro em regime estacionário e ao valor do sobressinal a fim de classificar os indivíduos. Dessa forma, a fim de aumentar a seletividade em relação aos fenótipos de cada indivíduo faz-se necessário um aprimoramento da função fitness com objetivo de impor novos parâmetros de seleção como tempo de assentamento e definir valores de sobressinal que devem ser encontrados. Junto a esses ajustes da função fitness têm-se a possibilidade de executar mecanismos elitistas durante o cruzamento, nos quais os indivíduos mais adaptados segundo a

função fitness terão probabilidades maiores de serem selecionados no processo de cruzamento, repassando suas informações às próximas gerações, visto que, nas Figuras 4 e 5, o melhor indivíduo encontrado na última geração não é o mais qualificado apresentado durante o processo das gerações, ou seja, sua característica genética não apresentou-se dominante e desapareceu ao longo das gerações.

CONCLUSÃO

O ajuste dos parâmetros do controlador PID é realizado através das características identificadas do processo e a resposta desejada, assim, o ajuste desses parâmetros no controlador torna-se uma tarefa complexa a ser realizada. Há diversas metodologias para o ajuste dos parâmetros do PID, considerando-se que, nos métodos mais simples de sintonia, a vantagem da facilidade dos ajustes promove uma configuração genética do PID, no qual o ajuste fino do processo é realizado manualmente pelo operador. Dessa forma, propôs-se modelar o problema de ajuste de PID como um problema de máximo e mínimo no qual foi aplicado um algoritmo genético para encontrar um ajuste de PID, que minimiza o erro em regime permanente e o sobressinal do sistema.

Ao executar o AG com 200 e 500 indivíduos durante 100 gerações, mantendo-se o cruzamento em 70% e a mutação em 3% para ambas as execuções, encontrou-se um ajuste de PID no qual o sobressinal é inferior a 20%, o tempo de assentamento máximo encontrado é de 3 segundos e não há erro de regime estacionário para o sistema. Utilizando o método de Ziegler-Nichols para calcular o valor de Kp, Kd e Ki juste de PID, no qual o sobressinal é inferior a 60%, o tempo de assentamento máximo encontrado é de 8 segundos e com um mínimo erro de regime estacionário para o sistema.

Dessa forma, foi verificado que o AG apresentou melhores ajuste para o PID em comparação ao método de Ziegler-Nichols, entretanto, faz-se necessário ajustar a função *fitness* e implementar um cruzamento elitista a fim priorizar os indivíduos melhor classificados para o problema, assim, demonstrando que o algoritmo implementado é uma opção viável para o ajuste de controladores PID.

AGRADECIMENTOS

Ao departamento de Engenharia Elétrica da Universidade do Oeste Paulista – Campus II, pelo apoio no desenvolvimento deste projeto.

REFERÊNCIAS

BERCI, C. D. **Observadores Inteligentes de Estado**: Propostas.2008. Dissertação (Mestrado em Engenharia Elétrica) -Universidade Estadual de Campinas, Campinas, 2008. Disponível em: <http://journal.unoeste.br/index.php/ce/article/view/3279>. Acesso em: 10 de agosto de 2020

BRATIFICH, R. **Projeto de ilhamento controlado em redes de energia elétrica**. 2018. Trabalho de conclusão (Bacharelado em Engenharia Elétrica) –Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2018.

CAVALCANTE, Zedequias Vieira. SILVA, Mauro Luis Siqueira da. **A importância da revolução industrial no mundo da Tecnologia**. Encontro Internacional de Produção Científica. VII EPCC, 2011. Disponível em: <http://www.cesumar.br/prppge/pesquisa/epcc2011/anais/zedequias_vieira_cavalcante2.pdf>. Acesso em: 08 de agosto de 2020

COSTA, L. P. **Sintonia de um Controlador de Nível com Otimização por Enxame de Partículas**. Trabalho de conclusão (Bacharelado em Engenharia Elétrica) -Universidade do Oeste Paulista, Presidente Prudente, São Paulo – SP, 2019.

DALTOÉ BERCI, C., SOUZA DA SILVA, N., BRATIFICH, R., CARLOS CAMACHO NEVES, R. (2019). **ALGORITMOS GENÉTICOS APLICADOS NA CRIAÇÃO DE AGENDAS DE HORÁRIO SEMANAL DE AULAS**. Colloquium Exactarum. ISSN: 2178-8332, 11(4), 1-9. https://doi.org/10.5747/ce.2019.v11.n4.e292

GRIFFITHS, A.J.F. **Introdução a Genética**. 9 ed. Rio de Janeiro, Guanabara Koogan, 2009.

NISE, N. S. **Engenharia de Sistemas de Controle**. 6 ed. Rio de Janeiro: LTC, 2012.

OGATA, K. **Engenharia de controle moderno**; tradutora Heloísa Coimbra de Souza; revisor técnico Eduardo Aoun Tannuri. -- 5. ed. -- São Paulo: Pearson Prentice Hall, 2010.

SILVA, RICARDO FERNANDES DA. PIGNATA, MARIA IZABEL BARNEZ. **Charles Darwin e a Teoria da Evolução**. CEPAE/UFG,2014. Disponível em: <https://files.cercomp.ufg.br/weby/up/80/o/TCEM2014-Biologia-RicardoFernandesSilva.pdf>
Acesso em: 08 de agosto de 2020

IMPLEMENTAÇÃO DE UM SISTEMA DE ARQUIVOS SIMPLIFICADO PARA APLICAÇÕES EMBARCADAS

CÉSAR DALTOÉ BERCI, JOÃO PUCCI NETO, CARLA PLANTIER MESSAGE

Universidade do Oeste Paulista – UNOESTE, Presidente Prudente, SP. E-mail: cesarberci@unoeste.br

RESUMO

Em um era totalmente tecnológica, os dispositivos móveis, bem como eletroeletrônicos de uma maneira geral, possuem tecnologias que denominamos embarcadas, ou seja, embutidas no próprio dispositivo, porém, por meio de placas, processadores e memórias de tamanhos compactos no sentido físico e lógico. Daí surge um dos problemas enfrentados por desenvolvedores de aplicações embarcadas e um grande desafio para pesquisadores: aprimorar o processamento e armazenamento dos dados. Neste projeto a proposta foi desenvolver um sistema de arquivos que torne o armazenamento dos dados mais simplificado e de menor tamanho (espaço de armazenamento) na memória principal (EEPROM) de equipamentos portáteis, fazendo que com isso diminua o custo do investimento, bem como melhore o desempenho do processamento do dispositivo.

Palavras-chave: Aplicações, Embarcada, Tecnologia, Sistema de arquivos.

IMPLEMENTATION OF A SIMPLIFIED FILE SYSTEM FOR EMBARKED APPLICATIONS

ABSTRACT

In a totally technological era, mobile devices, as well as electronics in general, have technologies that we call embedded, that is, embedded in the device itself, however, by means of plates, processors and memories of compact sizes in the physical and logical sense . Hence one of the problems faced by developers of embedded applications and a major challenge for researchers: improving data processing and storage. In this project, the proposal was to develop a file system that makes the storage of data more simplified and smaller (storage space) in the main memory (EEPROM) of portable equipment, thereby reducing the investment cost, as well as improving the processing performance of the device.

Keywords: Applications, Embedded, Technology, File system.

1. INTRODUÇÃO

Sistemas embarcados podem utilizar memórias EEPROM (*Electrically Erasable Programmable Read-only Memory*) de pequena capacidade para armazenamento de parâmetros e eventos. Tomemos como exemplo os seguintes equipamentos hipotéticos e a aplicação que fazem da memória:

I. Osciloscópio digital: Utiliza memória para armazenar parâmetros de usuário e de calibração de fábrica;
II. Inversor de frequência: Além dos parâmetros de fábrica e de usuário, armazena log de eventos;
III. Softstarters: Armazenam parâmetros e logs de eventos;
IV. Medidor de qualidade de energia: Armazena parâmetros e amostragem de dados.

Todos esses equipamentos utilizam memória de persistência para operar, portanto, necessitam estabelecer alguma forma consistente de acesso à essa memória (3GPP TR, 2011).

Dada a baixa complexidade dos dados, bem como do sistema, geralmente, em sistemas embarcados microprocessados, utilizam-se apenas setores predefinidos onde são salvos dados binários brutos.

Essa abordagem, porém, traz consigo diversas dificuldades e problemas, relacionados principalmente ao baixo nível de abstração utilizado, o que dificulta a arquitetura do software que acessa a memória.

Este trabalho propõe um sistema de arquivos simplificado, para utilização em sistemas microprocessados, sem sistema operacional, que implementa as seguintes funcionalidades:

I. Gerenciamento da memória de persistência;
II. Acesso ao hardware através de uma interface de software (essa camada de abstração visa a aplicação do sistema a qualquer arquitetura de hardware);
III. Criação de arquivos;
IV. Gerenciamento e manutenção dos metadados dos arquivos;
V. Salvar dados binários e cadeias de caracteres nos arquivos;
VI. Ler conteúdo dos arquivos;
VII. Apagar arquivos da memória.

O restante deste artigo está organizado como segue. Na Seção 2 são apresentados os componentes do projeto e um detalhamento de cada item. Na Seção 3 é descrito como o sistema proposto foi implementado, feitas comparações e discussões sobre sua eficácia e os resultados apresentados. Por fim, na Seção 4, são apresentadas as considerações finais deste trabalho.

2. COMPONENTES UTILIZADOS NO PROJETO

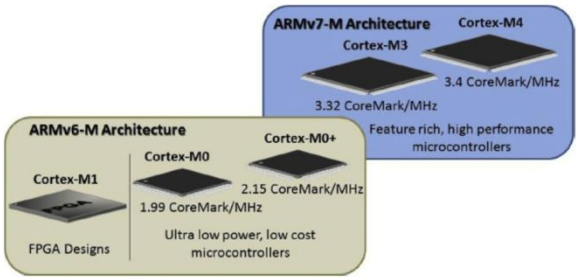
Para o projeto foi utilizado *hardware* próprio, sendo uma placa de equipamento utilizada para prover interface com operadores através de um teclado e um display LCD 128x64. O CPU que equipa essa placa é MK20DN512VLK10, com núcleo ARM Cortex M4, fabricado pela NXP. O *clock* de processamento dessa CPU pode chegar até 100 MHz. A programação e debug da CPU é feita através de uma ferramenta de debug de baixo custo, chamada USBDM, comumente utilizada para gravação de microcontroladores ARM CortxM e Frescale Kintis. O Cortex-M4 utiliza arquitetura de 32 bits, caminho dos dados e as interfaces de barramento possuem 32 bits de largura e registradores internos com banco de registradores. Chamada de *Thumb ISA,* a arquitetura do conjunto de instruções (ISA) da família Cortex-M é baseada na tecnologia *Thumb*-2, que tem suporte para uma junção de instruções de 16 e 32 bits. Esse processador possui: projeto de pipeline de três estágios; arquitetura de barramento *Harvard* com espaço de memória unificado, no qual instruções e dados usam o mesmo espaço de endereçamento; possui endereçamento de 32 bits, suportando até 4 gigabytes de espaço em memória; interfaces de barramento no chip que permite operações de barramento em *pipeline* para maior produtividade com base na tecnologia ARM AMBA (*Advanced Microcontroller Bus Architecture*); um controlador de interrupção chamado NVIC (*Nested Vectored Interrupt Controller*) que tolera até 240 requisições de interrupção e que dependendo da implementação real do dispositivo possui de 8 a 256 níveis de prioridade de interrupção; possui suporte para alguns recursos para implementação do Sistema Operacional, como o ponteiro de pilha sombreado e o temporizador do sistema (*system tick timer*); suporte ao modo de suspensão e outros recursos para baixo consumo de energia; suporte opcional para uma unidade de proteção de memória para fornecer recursos de proteção de memória, como controle de permissão de acesso ou memória programável; suporte para acessos de dados de bits em duas regiões específicas da memória usando um recurso chamado *Bit Band*; conta também com a opção de ser utilizado em projetos de processador único ou multiprocessador. Conectado à CPU há uma memória EEPROM de 64 KB (Macronix MX25L8035EM2I-10G), um Led RGB e um CI de Phy Ethernet (Microchip LAN8720A). A placa ainda possui quatro conectores para acesso aos pinos do CPU, sendo eles, dois conectores micro USB. O conector micro USB J4 (*Target*) é a interface para a comunicação UART do microcontrolador (UART0 - garante a comunicação serial via USB entre o microcontrolador e o mundo externo). O conector micro USB J5 (*Link*) é utilizado para programar e debugar o microcontrolador (LIMA, 2020). A placa também possui um conector dedicado para *debug* utilizando o USBDM.

2.1 PROCESSADORES ARM CORTEX

A ARM foi criada em 1990 como **A**dvanced **R**ISC **M**achines Ltd., uma junção de risco entre a Apple *Computer*, Acorn *Computer Group* e VLSI *Technology*. Em 1991, a ARM começou com a família de processadores ARM6 e o VLSI se tornou o licenciado inicial. Em seguida, outras empresas como a Texas *Instruments*, NEC, Sharp e ST *Microelectronics*, licenciaram os designs dos processadores ARM, estendendo

os aplicativos dos processadores ARM para discos rígidos de computadores, telefones celulares, assistentes digitais pessoais, sistemas de entretenimento doméstico e outros produtos de consumo. A cada ano esses parceiros vendem bilhões de processadores ARM. Diferente de algumas empresas de semicondutores, a ARM não fabrica processadores ou vende os chips diretamente. Ela licencia os designs dos processadores para parceiros de negócios, incluindo a maioria das principais empresas de semicondutores do mundo. Esses parceiros criam seus processadores, microcontroladores e soluções de sistema no chip com base nos projetos de processadores ARM de baixo custo e economia de energia. A esse modelo de negócios é denominado de licenciamento de propriedade intelectual (IP). A ARM também licencia IPs em nível de sistema e vários IPs de *software* (DA SILVA, 2020). A ARM desenvolveu uma base sólida de ferramentas de desenvolvimento, *hardware* e produtos de *software* para dar suporte a esses produtos e também permitir que os parceiros desenvolvam seus próprios produtos. Os processadores Cortex-M4 são produtos da família de processadores ARM Cortex-M. Toda a família de processadores Cortex-M é mostrada na Figura 1. Os processadores Cortex-M3 e Cortex-M4 são baseados na arquitetura ARMv7-M. Os dois são processadores de alto desempenho projetados para microcontroladores.

Figura 1. Família de processadores Cortex-M



Fonte: Adaptado de Embarcados (2020)

Os processadores Cortex-M4 são amplamente utilizados em projetos especializados, como *System on Chips* (SoC) e *Application Specific Standard Products* (ASSP) e em outros produtos modernos de microcontroladores. Os processadores ARM Cortex-M são considerados em geral, como processadores RISC (*Reduced Instruction Set Computing*). Porém, certas características dos processadores Cortex-M4, como o rico conjunto de instruções e tamanhos mistos de instruções, se parecem mais com o CISC (*Complex Instruction Set Computing*). Os conjuntos de instruções da maioria dos processadores RISC também estão se tornando mais complexos conforme as tecnologias de processadores avançam, fato é que essa fronteira tradicional entre a definição de processador RISC e CISC não pode mais ser aplicada (DA SILVA, 2020). Na figura 2 é ilustrado um chip do processador Cortex-M4.

Figura 2. CPU Cortex-M4

Fonte: Própria

2.2 Memórias EEPROM

A memória Intel 2816 foi construída pela Intel em 1978 com os princípios da tecnologia EPROM, porém, com uma fina camada de óxido para que o chip pudesse excluir seus dados sem a necessidade de

uma fonte UV. Anos depois, ex funcionários da Intel formaram a Seeq Tecnologia, que utilizava um dispositivo de bombas de cargas para fornecer as tensões necessárias para a programação da então EEPROM. A EEPROM (*Electrically Erasable Programmable Read-only Memory*) é uma memória do tipo ROM (*read-only memory*) que pode ser programável, reprogramável e apagada várias vezes utilizando uma tensão maior do que a normalmente utilizada nas suas operações (OLIVEIRA, 2018). Diferente das memorias EPROM, as EEPROM's não precisam ser removidas da placa para serem modificadas. O processo de apaga-la e reprograma-la são feitos em todo o seu conteúdo armazenado e não em partes. Esses dois processos possuem limitação no número de vezes que podem ser executados, variando entre cem mil a 1 milhão de vezes, esgotando em seguida e chegando ao fim a vida útil da mesma devido à contínua deterioração interna do chip, causada durante o processo de apagamento. Esse processo de apagamento dela ocorre byte por byte, tornando a versátil, porém, com isso fica mais lenta no processo. Atualmente as memórias EEPROM's podem ser encontradas em celulares, computadores e outros aparelhos eletrônicos. Na figura 3 é ilustrada um chip de memória EEPROM.

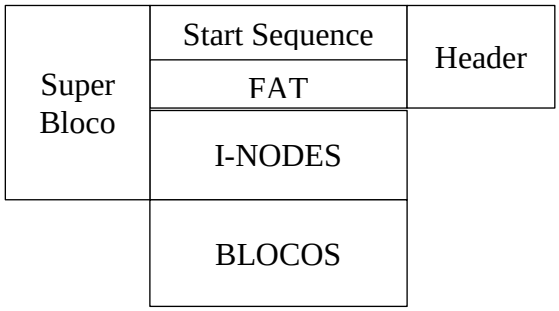
Figura 3. Chip de memória EEPROM

Fonte: Própria

3. IMPLEMENTAÇÃO DO SISTEMA

O sistema implementado é baseado em uma tabela de alocação de arquivos (FAT) e na persistência de metadados em estruturas chamadas I-NODES. O armazenamento dos dados é feito nos blocos, que se interligam de através de uma lista encadeada, eliminando limitações de tamanho do arquivo em virtude dos limites das estruturas I-NODE e do FAT. A memória é dividia em blocos de 32 bytes. Os dados dos arquivos são armazenados em estruturas chamadas I-NODE (nós).
A memória é organizada conforme ilustrado na Figura 4.

Figura 4. Divisão da memória



Fonte: Própria

Onde:

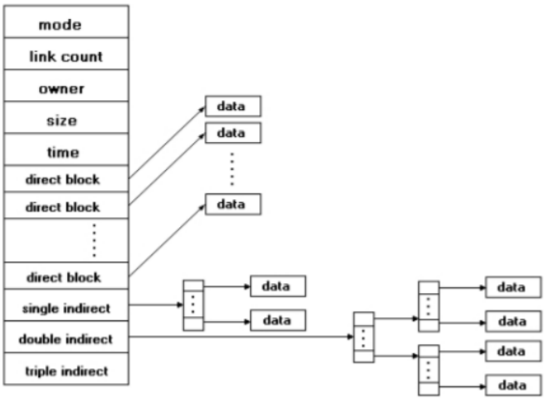
I. *Start Sequence*: Sequência de caracteres utilizada para checar o status da memória. Esse código é utilizado para validar a inicialização e estado do sistema de arquivos;

II. FAT: Tabela de alocação de arquivos, responsável por manter as informações básicas da memória e dos arquivos;
III. I-NODES: Nós do sistema, armazenam os metadados dos arquivos, ou seja, cada arquivo tem um I-NODE associado a ele;
IV. BLOCOS: Blocos de memória para persistência de dados.

3.1 I-Nodes

Um inode é uma estrutura que armazena todos as informações e os dados de um arquivo. Segundo SILBERSCHATZ (2000), o i-node (nó) contém os identificadores dos usuários e do grupo ao qual o arquivo pertence, o tamanho do arquivo em bytes, datas de último acesso e última modificação, o número de *hard links* ao arquivo e o tipo de arquivo. Também possui as informações se é um diretório, um *link* simbólico, um arquivo simples ou outros. Um exemplo de um inode é apresentado na Figura 5, na qual há a ilustração das características descritas anteriormente. Para armazenar os dados do arquivo os i-nodes possuem ponteiros de blocos de discos que apontam para blocos divididos em quatros tipos: _ blocos diretos; _ blocos indiretos; _ blocos indiretos duplos; _ blocos indiretos triplos. Os i-nodes armazenam os metadados dos arquivos, no qual apenas o primeiro bloco de informação é armazenado e os demais blocos utilizados são apontados pelo final do bloco anterior.

Figura 5. Exemplo de um i-node



3.2 File Allocation Table

A FAT (*file allocation table*) é uma tabela que é criada e permanece alocada na memória principal e que contêm as referências aos *clusters* (conjunto de setores que constitui a menor unidade de alocação capaz de ser endereçada num disco) utilizados para armazenar os dados do início de cada arquivo (TANENBAUM, 2003). Para encontrar um dado dentro do arquivo é necessário seguir o encadeamento que permanece inteiramente na memória, podendo ser seguido sem fazer qualquer referência ao disco, pois toda a área de arquivos é dividida em *clusters*. Os arquivos são alocados em um *cluster* por vez nessa mesma área, porém, não essencialmente em *clusters* adjacentes. Uma tabela FAT de alocação de arquivos é utilizada exclusivamente para encadear todos os *clusters* de um arquivo., na qual, para cada *cluster* na área de arquivos existe uma entrada na tabela, possuindo um ponteiro, que nada mais é do que um endereço de *cluster*. Sendo assim, um arquivo é representado por uma cadeia de entradas na tabela FAT e cada entrada aponta para a próxima entrada na cadeia. O sistema proposto nesse projeto utiliza uma FAT para gerenciar a memória. Essa tabela mantém informações a respeito da quantidade de arquivos, dos nós e blocos utilizados.

3.3 BLOCOS

Um disco rígido pode ser visto como um conjunto de blocos de tamanho fixo (geralmente de 512 ou 4.096 bytes). Por exemplo, os blocos de um disco rígido são normalmente denominados blocos físicos. Como esses blocos são pequenos, seu número em um disco dos modelos atuais pode ser imenso. Em um

disco rígido de 500 GB contém mais de um bilhão de blocos físicos. No entanto, para facilitar a gerência dessa quantidade imensa de blocos físicos e melhorar o desempenho das operações de leitura/escrita, os S.O. geralmente agrupam os blocos físicos em blocos lógicos ou *clusters*, que são grupos de 2 n blocos físicos consecutivos (MAZIERO, 2019). A maioria dos SO executam parte das suas operações e estruturas de dados definidas nos discos baseadas em blocos lógicos, que também determinam a unidade mínima de alocação de arquivos e diretórios: cada arquivo ou diretório ocupa um ou mais blocos lógicos para seu armazenamento. Em cada bloco lógico o número de blocos físicos é fixo e definido pelo SO ao formatar a partição, de acordo com alguns parâmetros, como o tamanho da partição, o sistema de arquivos utilizado e o tamanho das páginas de memória RAM. Blocos lógicos com tamanhos de 4 KB a 64 KBytes são frequentemente usados. Blocos lógicos maiores (32 KB ou 64 KB) induzem a uma menor quantidade de blocos lógicos a gerenciar pelo SO em cada disco e implicam em mais eficiência de entrada/saída, pois mais dados são transferidos em cada operação. Blocos grandes, porém, podem provocar muita fragmentação interna. Tomando como exemplo um arquivo com 200 bytes de dados ocupará um bloco lógico inteiro. Se esse bloco lógico tiver 32 KB (32.768 bytes), serão desperdiçados 32.568 bytes, que serão alocados ao arquivo sem serem utilizados. Com isso, Maziero (2019) cita que blocos lógicos menores podem diminuir a perda de espaço útil por fragmentação interna. Entretanto, utilizar blocos menores implica em ter mais blocos a gerenciar por disco e menos bytes transferidos em cada operação de leitura/escrita, o que gera um impacto não muito bom sobre o desempenho do sistema. Neste projeto, os blocos do sistema implementado possuem um tamanho fixo de 32 bytes, consideravelmente menor do que o tamanho do bloco em um sistema FAT32, por exemplo, cujo bloco tem geralmente 512k bytes. Essa escolha de tamanho de bloco possibilita a implementação do sistema em memórias de tamanho reduzido, comumente utilizadas em projetos embarcados.

3.4 IMPLEMENTAÇÃO EM UM SISTEMA COM MEMÓRIA DE 64K

Definições:

I. Nome de arquivos: Contém 4 caracteres; Dados:

Tamanho Total da memória	64000	bytes
	2000	blocks
Tamanho do FAT	308	bytes
Tamanho do INODE	9	bytes
Tamanho do BLOCO	32	bytes
Tamanho do Start Sequence	4	bytes
Tamanho do Header	312	bytes
	9,75	blocks
	10	blocks
Não utilizado	8	bytes
Header + não utilizado	320	bytes
Espaço ocupado por INODES	170	blocks
	5440	bytes
	604	Inodes
Não utilizado	4	bytes
Total de memória não utilizada	12	bytes
Super Bloco	180	blocks
	1440	bytes
Memória de arquivos disponível	1820	blocks
	58240	Bytes
	91	%

O algoritmo ilustrado na Figura 6, representa a estrutura do INODE, descrita em linguagem C.

A tabela de alocação mantém dois vetores, onde, cada bit representa o estado lógico de um bloco ou nó (1 representa espaço utilizado e 0 disponível). Também são mantidas as quantidades de blocos e arquivos armazenados no sistema. O algoritmo de implementação dessa tabela Fat é ilustrado na Figura 7.

Figura 6. Algoritmo – INODE

```
/**
 * Inode
 * This structure defines the
properties of an Inode object.
 * The inode is used to maintain all
information about a
 * file.
 * Only the first block number is kept
in the inode, the rest
 * blocks are pointed in the end of
each block, creating a linked list.
 */
typedef struct INODE {
    uint32_u name;  /*<! File name */
    uint16_u block; /*<! Start block
number*/
    uint16_u size;  /*<! File size in
bytes*/
    uint8 checksum; /*<! Check Sum
integrity verifier */
} Inode;
```

Fonte: Própria

3.5 ARQUIVOS

A estrutura dos arquivos mantém os dados necessários para salvar e recuperar o seu conteúdo da memória. A implementação do mesmo é demonstrada na Figura 8.

3.6 INTERFACE DE ACESSO À MEMÓRIA

Para manter o sistema de arquivos independente do hardware, foi criada uma camada de interface que é representada por meio de um esboço gráfico na Figura 9.

Para a camada de interface *Memory* foi implementado o **Erro! Fonte de referência não encontrada.** ilustrado nas Figuras 10 e 11.

Figura 7. Algoritmo – FAT

```
/**
 * File Allocation Table
 * This structure defines the FAT.
 */
typedef struct FAT {
    uint8 blocks[BLOCK_LOOP_COUNT];
/*<! Block bitmap, 1 = used */
    uint8 inodes[INODE_LOOP_COUNT];
/*<! Inodes bitmap, 1 = used */
    uint16_u used;
/*<! Total number of used blocks */
    uint16_u numberOfFiles;
/*<! Total number of files */
} LocalFat;
```

Fonte: Própria

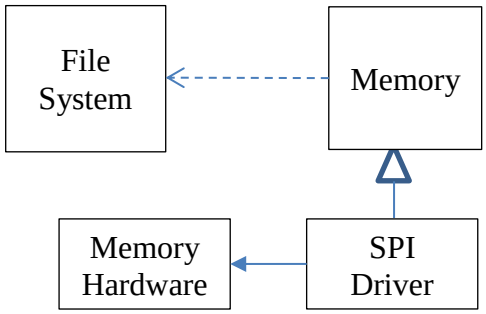
Figura 8. Algoritmo – Estrutura do Arquivo

```
/**
 * File
 */
typedef struct FILE {
    char name[5];           /*<!
Name of the file (only 4 chars) */
    Inode inode;            /*<!
File inode */
    bool isOpen;            /*<!
Boolean, true when file is open */
    int inodeAddress;       /*<! Id
of the file inode */
    bool isNew;             /*<!
Boolean, true if file is new */
    uint16 size;            /*<!
File size in blocks */
    FileDataState dataState;
} File;
```

Fonte: Própria

Os algoritmos ilustrados nas Figuras 10 e 11, definem as funções leitura e gravação na memória, e são divididas em dois tipos, com ou sem bloqueio. A função *save*, por exemplo, toma como parâmetros o endereço onde os dados serão salvos, a quantidade de bytes da transação, um ponteiro para o inicio dos dados e um ponteiro para uma função.

Figura 9. Esboço da camada de Interface



Fonte: Própria

A operação será então realizada de forma assíncrona, enquanto a CPU continua o fluxo normal do programa. Após concluída a gravação dos dados, a função apontada pelo parâmetro passado para a função save será executada, informando o fim do processo.

3.7 PRIMEIRA ABSTRAÇÃO – ACESSO ATRAVÉS DE ARQUIVOS.

Essas funções necessitam de uma referência a uma estrutura do tipo *File* para manipular os dados da memória. Todas as funções são baseadas em call-backs. As funções Abre-Arquivo, Fecha-Arquivo, Apaga-Arquivo, Lê-Arquivo e Escreve-Arquivo foram definidas nos algoritmos ilustrados nas respectivas Figuras 11, 12, 13, 14 e 15.

Figura 10. Algoritmo – Interface *Memory*

```
/**
 * Memory
 *
 * Structure that defines the basic
memory interface.
 * @details This interface should be
implemented by specific hardware
access functions.
 */
typedef struct MEMORY {
    /**
     * Load
     * Load data from memory and
stores in a char array.
     * @details This function uses a
callback function to inform the end of
process and
     * stores all data in an external
array. The load operation is
assynchronous and do not
     * block the system.
     * @param address Address of
memory to be loaded
     * @param length Total amount of
bytes to be loaded
     * @param dest Pointer to char
array where data will be stored
     * @param callback Callback
function, will inform the and of
process
     * @return status Status of the
memory
     */
    MemoryStatus (*load)(uint16
address, uint16 length, char * dest,
void * callback);
    /**
     * LoadBlocking
     * Load data from memory and
stores in a char array.
     * @details This function blocks
the system while retrieves memory data
and
     * stores all data in a external
array.
     * @param address Address of
memory to be loaded
     * @param length Total amount of
bytes to be loaded
     * @param dest Pointer to char
array where data will be stored
     * @return status Status of the
memory
     */
    MemoryStatus
(*loadBlocking)(uint16 address, uint16
length, char * dest);
    /**
     * Save
     * Save data in memory
     * @details This function gets
data from an array and stores in
memory using an asynchronous process.
     * After the end of the process a
callback function will be called.
     * @param address Address of
memory where data will be saved
     * @param length Total amount of
bytes to be loaded
     * @param dest Pointer to char
array where data will be save
```

Figura 11. Continuação do Algoritmo – Interface *Memory*

```
     */
    MemoryStatus (*save)(uint16 address, uint16
length, char * values, void * callback);
    /**
     * SaveBlocking
     * Save data in memory
     * @details This function gets data from an
array and stores in memory using an asynchronous
process.
     * This functions blocks the software until
all data are being save.
     * @param address Address of memory where
data will be saved
     * @param length Total amount of bytes to be
loaded
     * @param dest Pointer to char array where
data will be save
     * @return status Status of the memory
     */
    MemoryStatus (*saveBlocking)(uint16 address,
uint16 length, char * values);
    /**
     * satus
     * This function returns the status of
memory.
     * @return status Status of the memory
     */
    MemoryStatus (*status)(void);
    /**
     * tasks
     * Asynchronous state control.
     * @details This function should be called
periodically to maintain the internal
     * memory state control.
     */
    void (*tasks)(void);
} Memory;
```

Fonte: Própria (Fig. 9 e 10)

Todas as funções não bloqueiam o fluxo do programa, sendo que a função apontada pelo parâmetro void * callback, será executada sempre ao final do processo, indicando a sua conclusão. Mesmo em caso de erro nos processos, a função de call-back será executada, ficando a cargo do usuário checar o estado do sistema para validar a operação, procedimento similar ao sistema de acesso aos arquivos nativo da própria linguagem C.

Figura 11. Algoritmo – Função Abre-Arquivo

```
/**
 * fileOpen
 * Look for a file in file system
and open it.
 * @param file Pointer to a file
object that should be open
 * @param callback Callback
function that runs after end of
the process
 * @returns result Error state
```

Fonte: Própria

Figura 12. Algoritmo – Função Fecha-Arquivo

```
/**
 * fileClose
 * Close a file, also saves meta data
and FAT.
 * @param file Pointer to a file
object that should be close
 * @param callback Callback function
that runs after end of the process
 * @returns result Error state return
 */
int fileClose(File * file, void *
callback);
```

Fonte: Própria

Figura 13. Algoritmo – Função Apaga-Arquivo

```
/**
 * fileDelete
 * Delete a file, also saves meta
data and FAT. A file should be open
in order
 * to be deleted.
 * @param file Pointer to a file
object that should be close
 * @param callback Callback function
that runs after end of the process
 * @returns result Error state return
 */
int fileDelete(File * file, void *
callback);
```

Fonte: Própria

Figura 14. Algoritmo – Função Lê-Arquivo

```
/**
 * fileRead
 * Read the content of a file. A file
should be open in order
 * to be read.
 * @param file Pointer to a file
object that should be read
 * @param callback Callback function
that runs after end of the process
 * @param buffer Char array to store
file data
 * @returns result Error state return
 */
int fileRead(File * file, char *
buffer, void * callback);
```

Fonte: Própria

Figura 15. Algoritmo – Função Escreve-Arquivo

```
/**
 * fileWrite
 * Write data in a file. A file
should be open in order
 * to be read.
 * @param file Pointer to a file
object that should be write
 * @param callback Callback function
that runs after end of the process
 * @param buffer Char array to with
data to be write
 * @param length Number of bytes to
be write
 * @returns result Error state return
 */
int fileWrite(File * file, char *
buffer, uint16 length, void *
callback);
```

Fonte: Própria

3.8 SEGUNDO NÍVEL – ACESSO ATRAVÉS DOS NOMES DE ARQUIVOS

Com a intensão de prover uma API mais amigável ao usuário do sistema, são implementadas WRAPPER FUNCTIONS, que combinam as funções, anteriormente discutidas, para criar métodos diretos de leitura e escrita de arquivos.

Os algoritmos das funções Leitura-Direta, Escrita-Direta e Libera-Memória foram implementados e demonstrados respectivamente nas Figuras 16, 17 e 18.

Figura 16. Algoritmo - Função Leitura-Direta

```
/**
 * File Read Content
 * Read the contents of a file and
stores in the output buffer. This is
a asynchronous function
 * and do not blocks the program.
When the process is done, the
function __fs_free() will
 * return true.
 * @param name File name
 * @param buffer Output buffer
 * @returns result Error state return
 */
int __read(char * name, char *
buffer);
```

Fonte: Própria

Figura 17. Algoritmo - Função Escrita-Direta

```
/**
 * File Write Content
 * Writes the contents of a the input
buffer in a file.
 * If the file whit the specified
name already exists in memory, it
will be overwrite, else
 * a new file will be created and
content save.
 * This is a asynchronous function
 * and do not blocks the program.
When the process is done, the
function __fs_free() will
 * return true.
 * @param name File name
 * @param buffer Output buffer
 * @param length Number of bytes to
be write
 * @returns result Error state return
 */
int __write(char * name, char *
buffer, uint16 length);
```

Figura 18. Algoritmo - Função Memória-Disponível

```
/**
 * File system free
 * @return state True indicates that
the file system is free, than, the
last read or write operation
 * is done, otherwise, the system is
still running the last request.
 */
bool   fs_free(void);
```

Fonte: Própria

4. CONSIDERAÇÕES FINAIS

A implementação do sistema de arquivos proposto, em um dispositivo utilizando um microcontrolador ARM4, possibilitou a manipulação de arquivos de tamanho variável, bem como salvar e recuperar logs de eventos descritos por cadeias de caracteres, também de tamanhos variáveis.

Essas operações seriam um grande desafio caso o acesso a memória fosse implementado de forma direta, através de índices.

Assim, o presente trabalho apresenta uma solução viável para implementação de arquivos em sistemas com extra baixa capacidade de memória e processamento, ampliando a capacidade de software de projetos embarcados.

REFERÊNCIAS

3GPP TR 23.888 V1. 3.10. System improvements for machine-type communications. 2011.

DA SILVA, I. Lopes. Introdução ao microcontrolador ARM Cortex M3. 2020. Disponível em: https://www.embarcados.com.br/introducao-ao-microcontrolador-arm-cortex-m3/. Acesso em: 23 jul. 2020.

LIMA, Thiago (Brasil). Placa de Desenvolvimento NXP LPCXpresso4337. 2020. Disponível em: https://www.embarcados.com.br/lpcxpresso4337/. Acesso em: 07 jul. 2020.

MAZIERO, Carlos A. Sistemas Operacionais: Conceitos e Mecanismos. Curitiba: DINF - UFPR. 2019.

OLIVEIRA, Guilherme, Et Al. Sistemas Microprocessados I. 2018. Disponível em: https://sites.google.com/a/liberato.com.br/sistemas-microprocessados-i/home/memrias---4323/flash-eprom-eeprom. Acesso em: 15 maio 2020.

SILBERSCHATZ, A., Galvin, P., Gagne, G. Sistemas Operacionais: Conceitos e Aplicações; tradução de Adriana Richie. 1a Edição. Elsevier. Rio de Janeiro. 2000.

TANENBAUM, Andrew S. Sistemas Operacionais Modernos. 2 Edi¸c˜ao. Prentice Hall. 2003.

SISTEMA DE CONTROLE DE TEMPERATURA ANALÓGICO

Bruno Parpinelli Bonfim, Thiago Rodrigues, César Daltoé Berci, Rafael Bratifich

Universidade do Oeste Paulista – UNOESTE, Presidente Prudente, SP. E-mail: bruno_parpinelli@hotmail.com

RESUMO

Diversos sistemas e processos utilizam-se de mecanismos de controle a fim manter a harmonia de seu funcionamento, principalmente, diante de perturbações externas que poderiam promover uma situação de instabilidade. Desta forma, os elementos de controle, analógicos ou digitais, fazem-se uma necessidade atual para o bom funcionamento dos setores produtivos auxiliando a linha de produção a manter altas taxas de desempenho ao promover a estabilidade da planta dentro dos parâmetros desejáveis. O controle PID, usualmente, é utilizado para o controle de sistemas, assim, desenvolveu-se um sistema de controle PID analógico para controlar um forno industrial. O controle de temperatura realizar-se-á através de um sensor de temperatura e de resistências que promovem o aquecimento do local e consequentemente a temperatura, foram utilizados circuitos com amplificadores operacionais para aplicar os ganhos proporcional, integral e derivativo na saída da planta. Com o uso de um sistema de potência, o sinal de controle foi capaz de ajustar a temperatura dentro do forno, assim simulando o processo de controle industrial em uma planta.

Palavras-chave: Controle PID; amplificadores operacionais; sinal de controle.

ANALOG TEMPERATURE CONTROL SYSTEM

ABSTRACT

Several systems and processes use control mechanisms in order to maintain the harmony of their functioning, mainly, in the face of external disturbances that could promote a situation of instability. In this way, the control elements, analog or digital, are a current necessity for the good functioning of the productive sectors, helping the production line to maintain high performance rates by promoting the stability of the plant within the desirable parameters. PID control is usually used to control systems, so an analog PID control system has been developed to control an industrial oven. The temperature control will be carried out through a temperature sensor and resistances that provide the heating of the place and consequently the temperature, circuits with operational amplifiers were used to apply the proportional, integral and derivative gains at the plant's output. With the use of a power system, the control signal was able to adjust the temperature inside the oven, thus simulating the industrial control process in a plant.

Keywords: PID control; operational amplifiers; control signal.

1. INTRODUÇÃO

O sistema de controle é essencial em qualquer área da engenharia, como por exemplo: nos processos industriais, projetos de veículos espaciais, sistemas robóticos, e controle de pressão, umidade, temperatura, vazão etc. (OGATA, 2011). No controle têm-se como objetivo principal, uma vez que se identifica as características do sistema, alterar parâmetros de interesse, a fim de otimizar o processo, através do projeto de um controlador que atuará na resposta do sistema permitindo que seu comportamento seja dentro dos parâmetros desejáveis mesmo diante de perturbações que antes poderiam leva-lo a instabilidade.

Dessa forma, propõem-se desenvolver um controlador analógico PID para atuar em um forno industrial ou estufa no qual a temperatura é controlada através do aquecimento elétrico de resistências e monitorada através de sensores analógicos de temperatura. Esse tipo de equipamento é utilizado em

inúmeras aplicações nas quais são necessários um controle e estabilidade da temperatura dentro de uma faixa definida e sua relevância é significativa em vários segmentos, que incluem a indústria de alimentos, insumos hospitalares (esterilização), farmacêutica e o agronegócio.

2. METODOLOGIA

Para o desenvolvimento do projeto, identificação da planta e ajuste do controle, utilizou-se um modelo em escala reduzida, no qual o processo de aquecimento da estuda/forno é realizado através do acionamento de duas lâmpadas DC incandescentes de $12V/50W$ controladas a partir de um circuito amplificador.

A realimentação da planta se dá através de um sensor de temperatura LM35, instalado no interior do equipamento.

Esse sensor produz $10mV$ de tensão elétrica a cada $1°C$ medido no interior do forno. Os sistemas eletrônicos desenvolvidos operam com tensão de $\pm 12VDC$, e existe uma diferença significativa entre as grandezas: tensão do sistema e sinal do sensor. Em virtude deste fato, fez-se necessário realizar uma amplificação de 10 vezes na amplitude do sinal de realimentação, antes do circuito de subtração, alterando assim, intrinsecamente todos os sinais de referência da planta.

O sinal de referência, por sua vez, é criado utilizando-se um circuito ajustável através de potenciômetro linear, disponibilizando ao usuário a opção de ajuste da temperatura interna do forno através dele.

O sinal de erro é calculado através e um circuito diferenciador que realiza a subtração do sinal do sensor (temperatura dentro do forno) com o sinal ajustado no Set Point (sinal de referência), ambos em Volts.

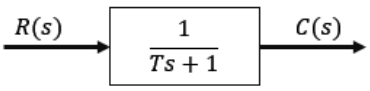
2.1. IDENTIFICAÇÃO DO SISTEMA

Para o projeto do controlador é preciso conhecer a função de transferência (FT) do sistema. Para a construção analítica da FT, aplicou-se um degrau de tensão ao sistema a fim de identificar seu comportamento diante deste sinal e a partir dessas informações estimar sua FT.

Dessa forma, após a construção do sistema físico de isolamento térmico, montagem da alimentação das lâmpada e posicionamento do sensor, foi aplicado um degrau de tensão na alimentação das lâmpadas DC incandescente ligadas em série afim de analisar a resposta de temperatura dentro da estufa com o sensor LM35 por um intervalo de 1317 segundos – intervalos longos permitem que o sistema estabilize-se plenamente após ligado.

É relevante ressaltar que o sistema em questão é de primeira ordem, e pode ser representado em diagrama de blocos, como mostrado na Figura 1. Nele também é considerado que as condições iniciais são nulas. (DOS SANTOS, 2017).

Figura 1. Sistema de primeira ordem.



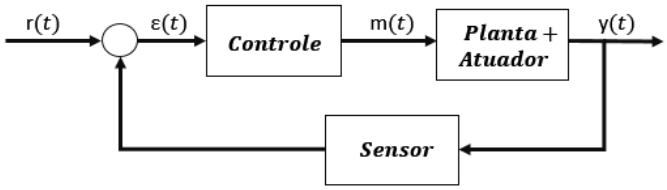
Fonte: (Autores, 2020).

2.2 Controle PID

A partir da FT caracterizada, foi desenvolvido um sistema de controle PID (Proporcional, Integral e Derivativo) para o sistema da estuda, a fim de melhorar a resposta em questão de rapidez e estabilidade (eliminação do erro em regime transitório e permanente). Um sistema de controle dotado de realimentação, como na figura abaixo, compara em cada instante o valor de saída da planta $y(t)$, medida pelo sensor, com o valor desejado de referência $r(t)$ indicado na entrada do sistema. O resultado dessa comparação é o erro atuante $\varepsilon(t)$ que é levado ao controlador, e produz um sinal de controle $m(t)$. Esse

sinal é levado às componentes seguintes do sistema de controle com o intuito de reduzir o desvio da saída em relação ao sinal desejado. (OGATA, 2011).

Figura 2. Sistema controlado.



Fonte: (Autores, 2020).

A atuação do controlador PID, é dividido em proporcional (P), integrativo (I) e derivativo (D). A ação proporcional, é responsável por aumentar proporcionalmente o sinal de erro atuante, o qual é obtido pela subtração da variável de processo (PV), neste caso a saída do sensor, e o sinal de referência (Set Point). Isso faz com que o circuito de malha fechada reaja mais rapidamente sobre o erro, ou seja, menor será o tempo de acomodação, porém o sistema pode apresentar erro em regime estacionário. (COSTA, 2019). Já o efeito integrativo é o de tentar reduzir o erro em regime permanente, aumentando a precisão da resposta. No entanto deve ser bem ajustado, pois pode tornar o sistema mais lento e oscilatório por conta do polo adicionado pelo integrador. (NISE, 2012). Por fim a ação derivativa atua essencialmente sobre a resposta transitória do sistema, adicionando um zero. Para tanto, deve-se modificar a posição dos polos dominantes do sistema de forma a se obter as especificações desejadas relativas ao componente transitório da resposta. (BARROS; ROSSI; SARTOR, 2015).

2.3 MÉTODO DE ZIEGLER-NICHOLS DE MALHA ABERTA

A fim de implementar o projeto do controlador de forma prática e rápida, diversos pesquisadores desenvolveram metodologias para ajuste das constantes do PID. Dentre as diversas metodologias existentes o método de Ziegler-Nichols, amplamente conhecido e difundido em sistemas de controle.

O método de Ziegler-Nichols, foi desenvolvido em 1942, e foi o primeiro trabalho a propor uma sintonia para controladores PID, e com certeza foi a descoberta de uma importante metodologia. A mesma pode ser utilizada em plantas cujo modelo matemático é desconhecido, como o modelo desenvolvido neste projeto que foi estimado através da resposta ao degrau, por exemplo.

O método é de fácil compreensão, porém é necessário um ajuste fino após iniciado os testes físicos no sistema controlado. A equação do PID no domínio do tempo tem a seguinte forma (OGATA, 2011):

$$Gc(s) = Kp\, e(t) + Ki \int e(t)\, dt + Kd \frac{de(t)}{dt} \qquad (1)$$

A função de transferência do controlador PID no domínio da frequência pode ser encontrada através da transformada de Laplace da equação (2).

$$Gc(s) = Kp + \frac{Ki}{S} + KdS \qquad (2)$$

onde Kp é o ganho proporcional, Ki o ganho integrativo e Kd o ganho derivativo.

Há uma outra forma de representar a função de transferência do controlador PID no domínio da frequência, onde é considerado o tempo integrativo e derivativo. Isso é de extrema importância, pois na Figura 3 é visível que os valores encontrados para o PID estão relacionados com o tempo integrativo e derivativo.

$$Gc(s) = Kp(1 + \frac{1}{TiS} + TdS) \qquad (3)$$

A Figura a seguir foi proposta por Ziegler-Nichols, na qual é possível encontrar os valores do ganho proporcional Kp, do tempo integrativo Ti e do tempo derivativo Td. (OGATA, 2011).

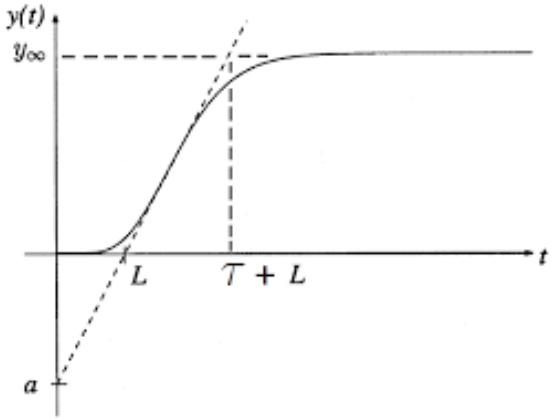
Figura 3. Ajuste PID.

Controlador	Kp	Ti	Td
P	$\frac{T}{L}$	∞	0
PI	$0{,}9\frac{T}{L}$	$\frac{L}{0{,}3}$	0
PID	$1{,}2\frac{T}{L}$	$2L$	$0{,}5L$

Fonte: (Autores, 2020).

O atraso (L) e a constante de tempo (T) são determinadas desenhando-se uma linha tangente no ponto de inflexão da curva com o formato em S e determinando-se a intersecção da linha tangente com o eixo dos tempos e a linha $y(t) = K$, como mostrado na Figura 4.

Figura 4. Curva S do sistema.



Fonte: (VASCONCELLOS, 2017).

Outra forma de encontrar o atraso (L) em um sistema de temperatura, é medir o tempo em que o forno leva para variar a temperatura em $1°C$ após ligado. Sendo assim, foi considerado $L = 5$ segundos, tendo como base a planilha de valores do início da medição extraída do data logger de bancada. No entanto é importante ressaltar que este atraso foi considerado apenas para encontrar os ajustes iniciais dos ganhos através do método de Ziegler-Nichols, uma vez que 5 segundos de atraso é praticamente desprezível com relação aos 1317 segundos de estabilização do sistema. Já a constante de tempo é $T = 119$ segundos, como já encontrado na caracterização da FT.

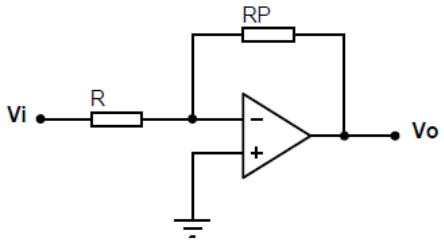
2.4 CIRCUITO DO CONTROLADOR

Analisado as respostas do projeto do controle é necessário realizar a confecção e o ajuste do circuito analógico para a operação de cada módulo do controlador, ou seja, proporcional, integrativo e derivativo. Através da eletrônica dos amplificadores operacionais, é possível construir circuitos com estas características. Cada um deles são demonstrados a seguir.

2.4.1. Circuito proporcional

O circuito proporcional é construído com o uso de amplificadores operacionais. Estes dispositivos possuem alta impedância de entrada, e baixa impedância de saída, e são capazes de realizar um ganho no sinal de entrada.

Figura 5. Circuito proporcional.



Fonte: (Autores, 2020).

Como a impedância de entrada é alta, a corrente (i) na porta inversora do amplificador é nula, assim a corrente que passa em R é praticamente a mesma que passa por RP.

$$i = \frac{Vi-0}{R} \qquad (4)$$

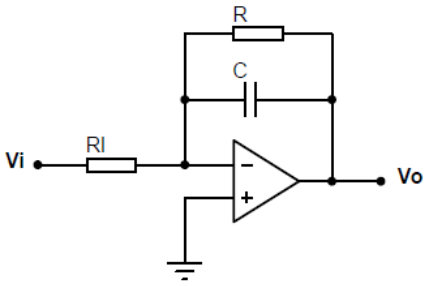
Com o cálculo da tensão de saída do amplificador, é possível encontrar o ganho no sinal de entrada.

$$Vo = -iRP \rightarrow Vo = -\left(\frac{Vi}{R}\right)RP \rightarrow \frac{Vo}{Vi} = -\frac{RP}{R} \qquad (5)$$

2.4.2. Circuito integrativo

Este circuito é responsável por realizar a integral do sinal de tensão de entrada no amplificador operacional. Através dos mesmos cálculos realizados no item anterior, é possível calcular a integral do sinal de entrada.

Figura 6. Circuito integrativo.



Fonte: (Autores, 2020).

A diferença deste circuito com relação ao anterior, é a substituição do resistor de referência (RP) por um capacitor (C) e o acréscimo de mais um resistor R, modificando a equação da tensão de saída Vo.

$$Vo = -\frac{R}{1+RC}\int i(t)\,dt = -\frac{R}{1+RC}\int\left(\frac{Vi(t)}{RI}\right)dt = -\frac{R}{RI+RIRC}\int Vi(t)\,dt \qquad (6)$$

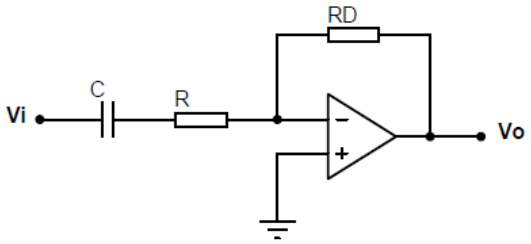
Utilizando mais uma vez a transformada de Laplace, é possível encontrar a função de transferência do circuito integrativo no domínio da frequência.

$$Vo = -\frac{R}{RI+RIRC}\int Vi(t)\,dt \rightarrow Vo = -\frac{R}{RI+RIRCS}Vi \rightarrow \frac{Vo}{Vi} = -\frac{R}{RI+RIRCS} \qquad (7)$$

2.4.3. Circuito derivativo

Por fim a ação derivativa também deve ser implementada no controlador. O circuito que realiza esta função é demonstrado na Figura 7.

Figura 7. Circuito derivativo.



Fonte: (Autores, 2020).

Os cálculos deste circuito são semelhante ao integrativo, a única mudança é com relação a posição do capacitor (C) e do resistor (R).

$$i(t) = \frac{C}{1+RC}\frac{dVi(t)}{dt} \qquad (8)$$

$$Vo = -iRD = -\frac{RDC}{1+RC}\frac{dVi(t)}{dt} \qquad (9)$$

Com a transformada de Laplace, o ganho no domínio da frequência é:

$$Vo = -\frac{RDC}{1+RC}\frac{dVi(t)}{dt} \rightarrow Vo = -\frac{RDCS}{1+RCS}Vi \rightarrow \frac{Vo}{Vi} = -\frac{RDCS}{1+RCS} \qquad (10)$$

2.5 Construção do sistema

Finalizado os estudos das respostas e o projeto dos circuitos e do controlador foi montado todo o sistema utilizando os equipamentos disponibilizados no Laboratório de Eletrônica e Engenharia Elétrica da Universidade do Oeste Paulista. Na figura 8 é possível visualizar todos os circuitos acoplados.

Figura 8. Sistema completo – parte de controle e potência.

Fonte: (Autores, 2020).

A este sistema foram aplicados diversos testes a fim verificar o comportamento do sistema diante a variação no valor de referência e alterações perturbativas na temperatura do forno simuladas através da abertura repentina da tampa. Desta forma pode ser observado o real funcionamento do sistema de controle aplicado à planta.

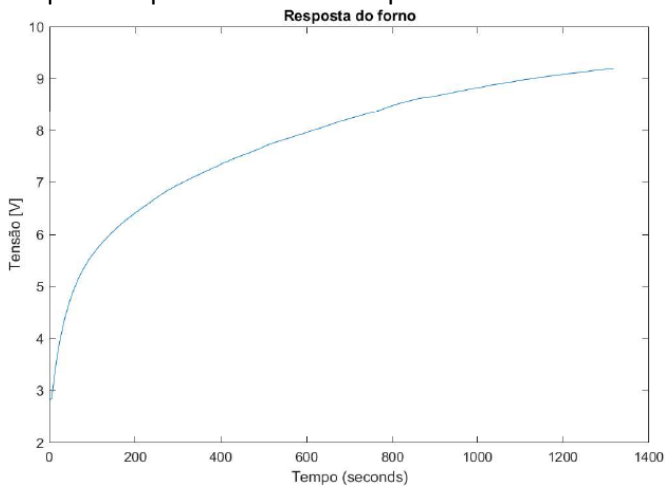
3. RESULTADOS E CONCLUSÃO

Nesta seção serão apresentadas as informações referentes a identificação do sistema, o ajuste do PID, o cálculo das componentes dos circuitos de controle e os dados de teste do controlador.

3.1. IDENTIFICAÇÃO DO SISTEMA

A figura 9 apresenta a resposta do forno quando aplica-se 12V nas lâmpadas, e a unidade de medida da resposta do sistema foi a tensão (Volts) em função do tempo.

Figura 9. Resposta do forno quando aplicado 12V nas lâmpadas DC incandescente 50W.



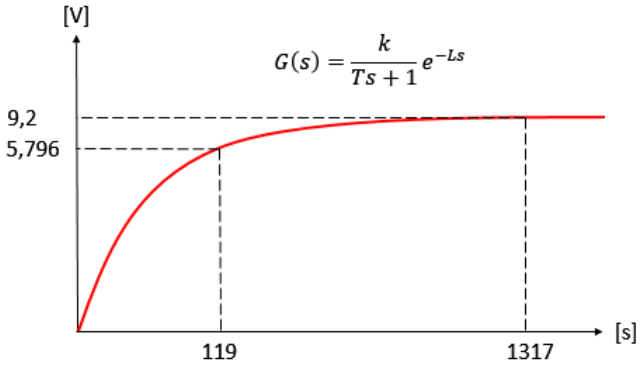
Fonte: (Autores, 2020).

Analisando a figura a cima, é possível verificar que o sistema tende a se estabilizar após os $9{,}2V$ no eixo Tensão $[V]$, com um tempo de 1317 segundos no eixo Tempo $[s]$. Segundo OGATA (2011), a constante de tempo é o tempo necessário para que a resposta ao degrau alcance 63% do seu valor final, ou seja, $5{,}796V$ em 119 segundos. Com estes dados extraídos do ensaio e com o auxílio do software MatLab, é possível estimar a FT do sistema do forno.

$$G(s) = \frac{9{,}2}{119s + 1} \qquad (11)$$

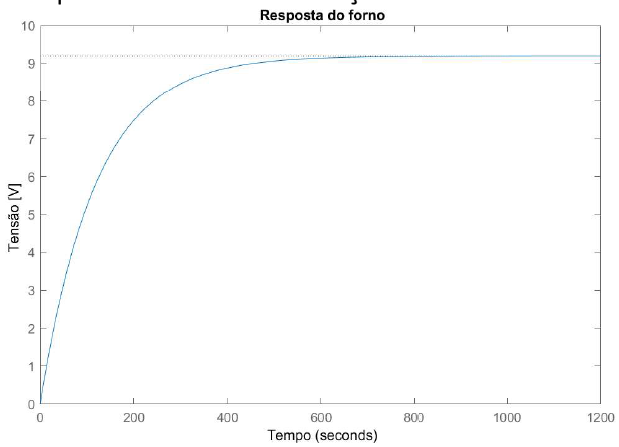
Nota-se que a parte da exponencial da FT do sistema não aparece na equação 11, isso porque na planta do forno não há atraso (L) significativo na resposta. Afim de testar a confiabilidade da FT extraída das simulações no laboratório, foi gerada sua resposta no software MatLab, como é mostrado a baixo. (DOS SANTOS, 2017).

Figura 10. Caracterização do sistema forno.



Fonte: (Autores, 2020).

Figura 11. Resposta do forno quando simulado sua função de transferência.



Fonte: (Autores, 2020).

3.2 CONTROLE PID

Com o auxílio da Figura 3, tem-se os valores de $Kp = 28$; $Ti = 10$ e $Td = 2{,}5$ sendo possível calcular os ganhos relacionando a equação (2) e (3).

$$Kp = 28{,}56 \quad (12)$$

$$Ki = \frac{Kp}{TiS} = 2{,}856 \quad (13)$$

$$Kd = KpTdS = 71{,}4 \quad (14)$$

Com os valores dos ganhos já encontrados, a função de transferência do controlador $Gc(s)$ através da equação (2) é:

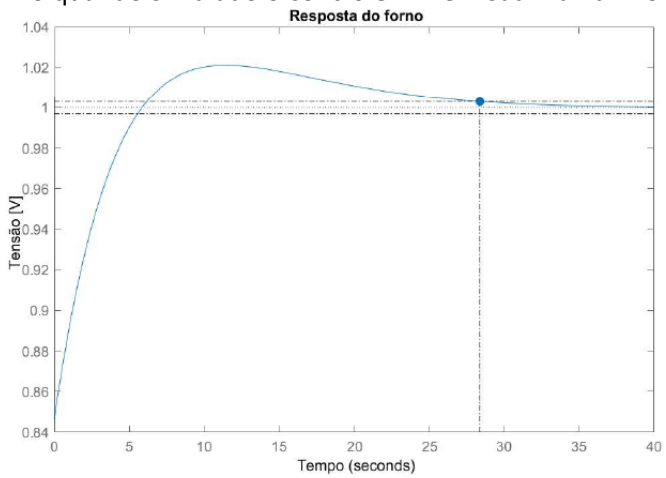
$$Gc(s) = \frac{71{,}4S^2 + 28{,}56S + 2{,}856}{S} \quad (15)$$

Aplicando o controlador ao sistema obtêm-se:

$$FT(s) = \frac{656{,}9S^2 + 262{,}8S + 26{,}28}{775{,}9S^2 + 263{,}8S + 26{,}28} \quad (16)$$

Simulando a aplicação de um degrau a entrada do sistema controlado obtêm:

Figura 12. Resposta do forno quando simulado o controle PID em sua malha – formula (16).



Fonte: (Autores, 2020).

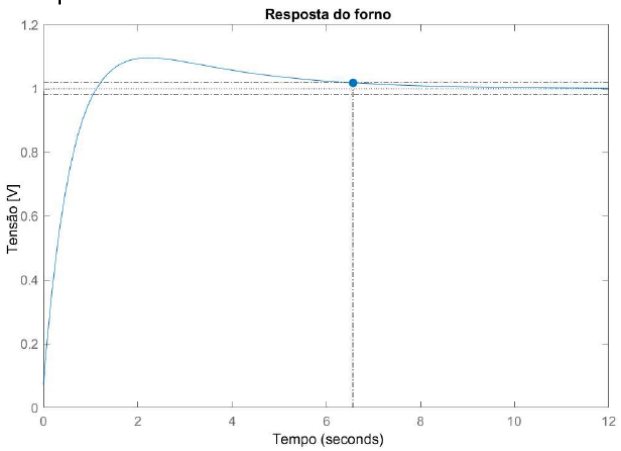
Através da resposta mostrada na Figura 12, o compensador foi capaz de ajustar o tempo de acomodação em 28,4 segundos e o sobressinal em 2,11%. No entanto, como visto, a resposta do sistema de compensação não foi tão rápida, é preciso um ajuste fino nos ganhos. Ajustando os valores em $K_p = 28$; $K_i = 10$ e $K_d = 1$ é encontrado $G_c(s)$ e $FT(s)$.

$$G_c(s) = \frac{1S^2+28S+10}{S} \quad (17)$$

$$FT(s) = \frac{9{,}2S^2+257{,}6S+92}{128{,}2S^2+258{,}6S+92} \quad (18)$$

Simulando a aplicação de um degrau a entrada do sistema controlado obtêm:

Figura 13. Resposta do forno quando simulado o controle PID em sua malha – formula (18).



Fonte: (Autores, 2020).

No gráfico, o novo tempo de acomodação é de 6,56 segundos e o sobressinal em $9{,}64\%$, ou seja, uma resposta muito mais rápida que a anterior. Mesmo que o ganho seja maior, não haverá interferência no acionamento da parte de potência, pois $9{,}64\%$ ainda é considero uma variação pequena do ganho para este tipo de sistema.

3.3 CIRCUITO DO CONTROLADOR

Após encontrado cada componente do controlador, utilizou-se equações dos circuitos proporcional, integrativo e derivativo para calcular seus respectivos elementos e após estimar os elementos basta somar todos os sinais obtidos desses circuitos em um circuito somador. As comparações utilizadas para o ajuste dos componentes dos circuitos em relação aos valores do controlador são:

$$Kp = 28 = \frac{RP}{R} \quad (19)$$

$$Ki = \frac{Kp}{TiS} = 10 = \frac{R}{RI+RIRC} \quad (20)$$

$$Kd = KpTdS = 1 = \frac{RDC}{1+RC} \quad (21)$$

Comparando as equações com os valores de resistores e capacitores comerciais disponíveis, é possível construir uma relação de componentes para o circuito. Na tabela 1, está descrito os valores de cada um dos componentes.

Tabela 1. Componentes.

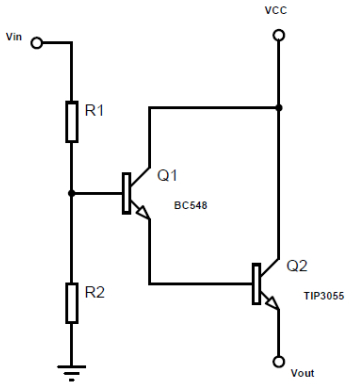
Componente	Valor comercial	Ajuste
RP	500K Ω	280K Ω
R	10K Ω	-
RI	2K Ω	1K Ω
R	10K Ω	-
C	100n F	-
RD	1M Ω	-
R	1K Ω	-
C	1µ F	-

Fonte: (Autores, 2020).

3.3.1 CIRCUITO DE POTÊNCIA

Após realizar o tratamento do sinal de erro do sistema, é necessário acionar as lâmpadas DC incandescentes. Para isso foi construído um sistema de potência capaz de suportar a corrente fornecida da fonte às lâmpadas. Como a temperatura máxima do forno é de $92°C$ $(9{,}2V)$ como mostrado na Figura 9, e mínima de $26°C$ $(2{,}6V)$, tem-se uma janela máxima de $6{,}6V$ na saída do erro no diferenciador. Essa saída ainda será multiplicada pelos ganhos do controlador PID.

Figura 14. Circuito de potência.



Fonte: (Autores, 2020).

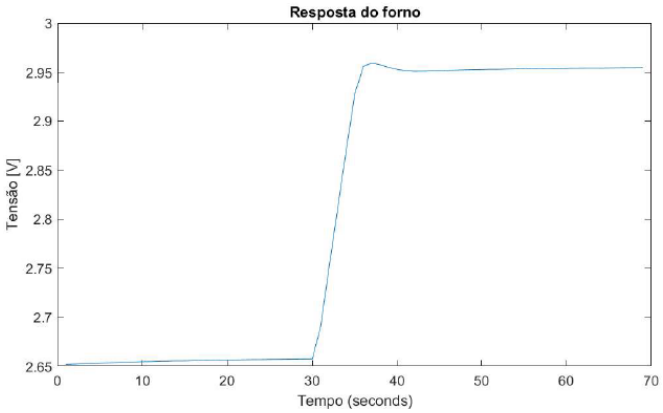
No circuito da Figura 14, foi utilizado um transistor BC548 (o qual tem seu controle de VCE através da corrente de base), a fim de ajustar o acionamento do transistor TIP3055 (suporta maior potência). Definido a "sensibilidade" adequada deste sistema de potência, foi ajustado os valores em $1k\ \Omega$ e $100k\ \Omega$ para os resistores $R1$ e $R2$.

Por fim é importante ressaltar que as lâmpadas DC incandescentes começam a ascender (liberar calor) com uma tensão de aproximadamente $2{,}3V$, o que é de grande valia, pois ocorrendo uma pequena variação no sinal de erro, haverá a correção da temperatura de imediato pelo sistema de potência.

3.4. AVALIAÇÃO DO CONTROLE

O primeiro teste foi realizado com o forno inicialmente em temperatura ambiente $2{,}6V$ $(26°C)$ com tampa fechada, e referência de $3{,}0\ V$ $(30°)$ no controle.

Figura 15. Teste 01 – forno em temperatura ambiente 2,6V (26°C) e referência de 3,0V (30°C) com tampa fechada.

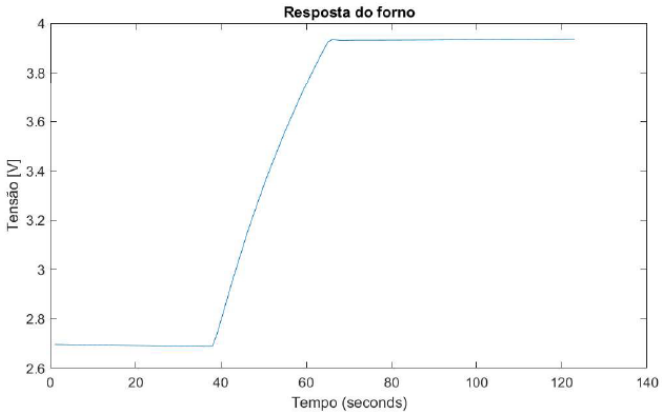


Fonte: (Autores, 2020).

Na figura 15, o sistema levou aproximadamente um tempo de 11 segundos em seu transitório até o valor final. Já com relação ao regime permanente, o valor fixado foi de $2{,}9547V$, ou seja, um erro de $0{,}0453V$. Isso mostra uma variação menor que $0{,}5°C$ na resposta do sistema.

O segundo teste também foi com o forno inicialmente em temperatura ambiente e tampa fechada, no entanto o valor de referência foi aumentado para $4{,}0V$ $(40°C)$.

Figura 16. Teste 02 – forno em temperatura ambiente 2,6V (26°C) e referência de 4,0V (40°C) com tampa fechada.

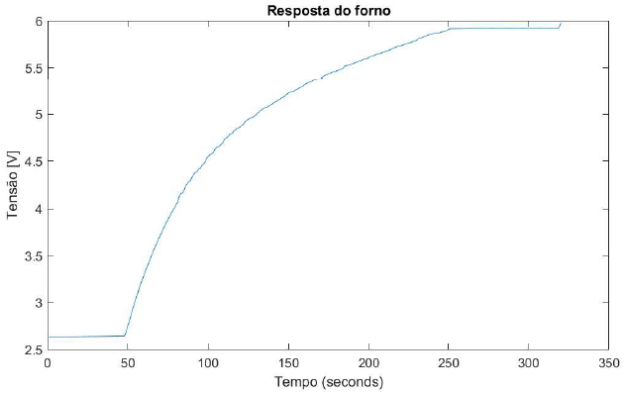


Fonte: (Autores, 2020).

Para este experimento o tempo de transitório foi de aproximadamente 30 segundos, e o valor final em regime permanente de $3{,}9348V$, o que dá um erro de $0{,}0652V$.

Ainda com a tampa fechada e temperatura inicial mantida em $2{,}6V$ $(26°C)$, foi ajustado a referência para $6{,}0V$ $(60°C)$.

Figura 17. Teste 03 – forno em temperatura ambiente 2,6V (26°C) e referência de 6,0V (60°C) com tampa fechada.

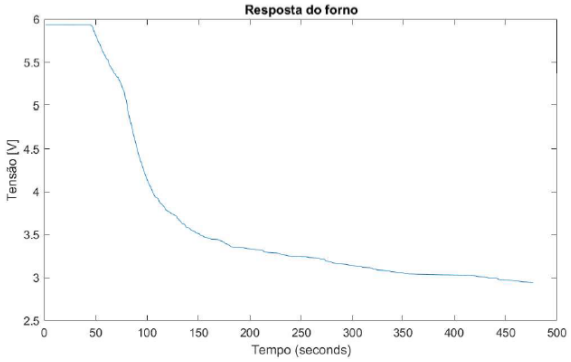


Fonte: (Autores, 2020).

Aqui o tempo de transitório da curva até à estabilização foi de aproximadamente 204 segundos, e o valor de regime permanente foi de $5{,}9233V$, ou seja, um erro de 0,0767.

Após alcançado o valor de $6{,}0V$ $(60°C)$ no forno, foi definido uma referência de $3{,}0V$ $(30°C)$ com a tampa aberta.

Figura 18. Teste 04 – forno em temperatura 6,0V (60°) e referência de 3,0V (30°C) com tampa aberta.

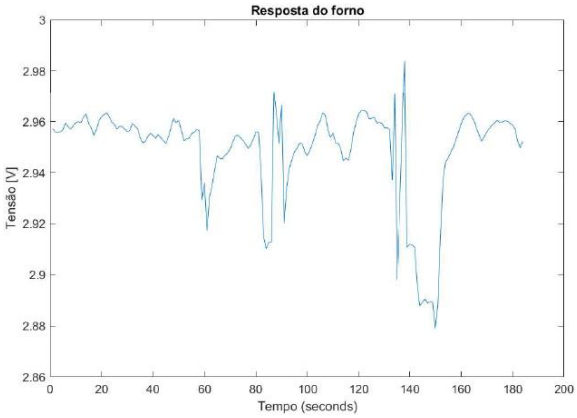


Fonte: (Autores, 2020).

Na figura 18, após passar 432 segundos, o valor no eixo Tensão $[V]$ era de $3{,}0101V$, no entanto decorrido mais 46 segundos o regime permanente se estabilizou em $2{,}9454V$, dando um erro de $0{,}0546V$ da referência.

Ainda com a tampa aberta e valor de referência em $3{,}0V$ $(30°C)$, foi realizado o teste de ventilação forçada. Este experimento diz respeito à velocidade de regime transitório do sistema, ou seja, a estabilização do valor ajustado na referência.

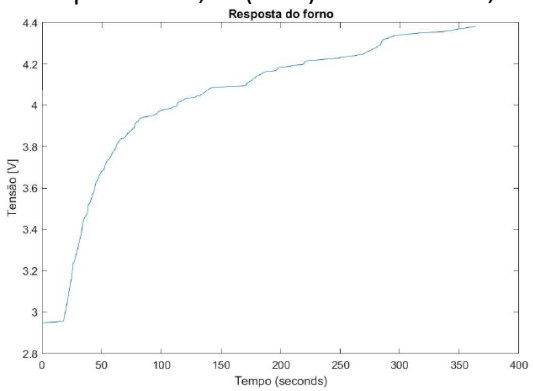
Figura 19. Teste 05 – forno em temperatura 3,0V (30°C), referência 3,0V (30°C) com tampa aberta e ventilação forçada.



Fonte: (Autores, 2020).

Considerando o maior e o menor valor alcançado no eixo Tensão $[V]$, o tempo de resposta para este ajuste é de aproximadamente 12 segundos.

Figura 20. Teste 06 – forno em temperatura 3,0V (30°C) e referência 9,2V (92°C) com tampa aberta.



Fonte: (Autores, 2020).

Finalizando os testes, foi ajustado como referência o valor de $9{,}2V$ $(92°C)$, como apresentado na Figura 20, porém com a tampa aberta. Após percorrido um tempo de 364 segundos, o maior valor alcançado foi de $4{,}3836V$.

CONCLUSÕES

No presente projeto desenvolveu-se um controlador analógico PID a fim de controlar a temperatura de uma estufa/forno através de um modelo simplificado desses elementos através de uma câmara isolante, lâmpada e um sensor de temperatura. Após a identificação do sistema e estimativa de sua função de transferência elaborou-se o ajuste do controlador através da metodologia de Ziegler-Nichols para a implementação física dos circuitos do controlador.

A implantação do circuito de controle possibilitou a realização de testes para verificar o funcionamento e controle do sistema analisado. Com a realização dos testes, foi constatado que não é possível atingir os ganhos apontados na equação (17) para o PID analógico, isso porque o valor máximo da fonte simétrica do laboratório é de $\pm 12VDC$, o que acaba limitando o ganho nos amplificadores operacionais. Desta maneira, foi necessário um ajuste fino nos trimpot's para regular o valor de saída do PID.

Após ajustado os valores de resistência dos ganhos proporcional, integrativo e derivativo em $35k\ \Omega$, $2k\ \Omega$ e $1M\ \Omega$, o circuito de potência foi capaz de acionar as lâmpadas DC incandescentes e aquecer o forno no valor de temperatura definido na referência (Set Point) do controle. Utilizando as formulas (19), (20) e (21), juntamente com os valores das resistências, tem-se que os ganhos no PID ficaram em 3,5; 5 e 1.

Ao final, concluiu-se que o controlador PID exerceu sua função de uma maneira eficaz, com a devida velocidade e pequeno erro em regime permanente.

AGRADECIMENTOS

Ao departamento de Engenharia Elétrica da Universidade do Oeste Paulista – Campus II, pelo apoio no desenvolvimento deste projeto.

REFERÊNCIAS

BARROS, Juliana de SG; ROSSI, Luiz A.; SARTOR, Karina. **Utilização do controlador PID como tecnologia eficiente em sistema de aquecimento de creche de suínos**. Rev. bras. eng. Viola. ambiente., Campina Grande, v. 19, n. 5, pág. 476-480, maio de 2015. Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-43662015000500476&lng=en&nrm=iso>. acesso em 19 de agosto de 2020. https://doi.org/10.1590/1807-1929/agriambi.v19n5p476-480

COSTA, L. P. **Sintonia de um Controlador de Nível com Otimização por Enxame de Partículas.** Presidente Prudente, São Paulo – SP, 2019.

DOS SANTOS, Orlem Lima. **Sistema de Controle de Temperatura para uma Estufa com Monitoramento via Aplicativo**. RCT - Revista de Ciência e Tecnologia, [S.l.], v. 3, n. 4, july 2017. ISSN 24477028. Disponível em: <https://revista.ufrr.br/rct/article/view/4004/2286>. Acesso em: 19 aug. 2020.

NISE, N. S. **Engenharia de Sistemas de Controle**. 6 ed. Rio de Janeiro: LTC, 2012.

OGATA, K. **Engenharia de Controle Moderno.** 5 ed. São Paulo: Pearson Prentice Hall, 2011.

VASCONCELLOS, A. P. M. **Projeto de controladores PI e PID para um forno aquecedor de óleo de uma planta de tratamento de hidrocarbonetos.** Rio de Janeiro, SP. 2017. Projeto de graduação.

RESUMOS DE PESQUISA

Pesquisa (ENAPI)

Comunicação oral on-line

UNIVERSIDADE DO OESTE PAULISTA - UNOESTE
Engenharias
Engenharia Elétrica

CRIAÇÃO DE UM PROTÓTIPO PARA ALTERNÂNCIA DA TEMPERATURA DA ÁGUA AFIM DE REALIZAR A TERAPIA DE CONTRASTE EM ATLETAS

ASAFE MARCONDES PEREIRA
PEDRO VICTOR TONICANTE DA SILVA
THIAGO RODRIGUES
ROBSON CHACON CASTOLDI
EVERTON ALEX CARVALHO ZANUTO

A técnica de contraste significa expor de forma alternada o atleta ao calor e o frio, assim, auxiliando na recuperação do mesmo, buscando a remoção de alguns biomarcadores negativos que interferem na recuperação pós esforço. Com isso, esse trabalho tem como objetivo desenvolver um protótipo maleável e acessível que tenha o controle para alternância de temperaturas, promovendo uma melhor eficácia da terapia. O projeto foi aprovado pela Coordenadoria de Pesquisa, Desenvolvimento e Inovação (CPDI) sob o protocolo de nº 6068 da Universidade do Unoeste Paulista (UNOESTE) campus de Presidente Prudente. O protótipo está sendo desenvolvido em três etapas, sendo a primeira o levantamento bibliográfico, a segunda, análise e simulação através de software PROTEUS e a terceira a parte da montagem tanto do circuito elétrico, quando dos sistemas hidráulico e estrutural. Essas etapas serão realizadas no laboratório de Engenharia Elétrica da UNOESTE. Com a simulação no software PROTEUS pode-se observar efetividade no funcionamento do protótipo, tendo bons indicadores de funcionabilidade e desempenho, porém, deve-se considerar que está sendo desenvolvido em regime ideal e que fora do regime poderá ter alterações nas leituras dos resultados. Até o presente momento, toda parte de prototipagem já foi realizada em simulador CAD, para análise física e topologia em placa de circuito impresso (PCI). Houve também uma simulação para fabricação do mesmo, levando em conta as normas e características para o layout. Para manuseio do equipamento foi criado uma interface (display) de fácil compreensão, manejo rápido, eficaz e sem intercorrências. O produto está sendo desenvolvido o mais automático possível para evitar falhas humanas, assim, sendo necessário apenas o manuseio inicial para programação do aparelho. A programação e instrução de manejo são com base na terapia de contraste (tempo de terapia, temperatura e alternância de temperatura). Assim, foi executado no software PROTEUS a fim de verificar o funcionamento dinâmico do protótipo (planta total), para analisar a resposta e precisão dos sensores, bem como o funcionamento dos motores de bomba, uma vez que os mesmos deverão ter intervalos entre os jatos d'água, por fim, verificou-se a ativação da resistência e pastilha peltier, para saber se o tanque irá esquentar e esfriar, respectivamente. A partir dos resultados obtidos na simulação concluímos que o projeto é viável de execução e irá suprir a proposta de tratamento. Órgão de fomento financiador da pesquisa: UNOESTE