

FELIPE FERNANDES FRANÇA

Controlador PID para controle de temperatura de uma carga resistiva AC

Felipe Fernandes França

Controlador PID para controle de temperatura de uma carga resistiva AC

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

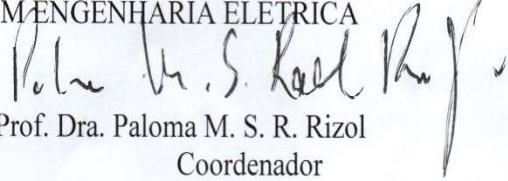
Orientador: Prof. Dr. Francisco Antonio Lotufo

F814c	<p>França, Felipe Fernandes Controlador PID para controle de temperatura de uma carga resistiva AC/ Felipe Fernandes França– Guaratinguetá, 2018. 62 f : il. Bibliografia: f. 59-60</p>
	<p>Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2018. Orientador: Prof. Dr. Francisco Antonio Lotufo</p>
	<p>1. Microcontroladores 2. Controladores PID 3. Reles eletricos I. Título</p>
	CDU 621.381


FELIPE FERNANDES FRANÇA

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE
“ENGENHEIRO ELETRICISTA”

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE
GRADUAÇÃO EM ENGENHARIA ELÉTRICA


Prof. Dra. Paloma M. S. R. Rizol
Coordenador

BANCA EXAMINADORA:


Prof. Dr. Francisco Antonio Lotufo
Orientador/UNESP-FEG


Prof. Dr. José Feliciano Adami
UNESP-FEG


Prof. Dr. Fernando Ribeiro Filadelfo
UNESP-FEG

Novembro 2018

AGRADECIMENTOS

Em primeiro lugar agradeço aos meus pais pela educação concedida a mim e papel fundamental na minha formação acadêmica.

ao meu orientador, Prof. Dr. Francisco Antonio Lotufo por toda sua ajuda e disponibilidade sempre quando necessário.

aos demais professores e funcionários da Faculdade de Engenharia de Guaratinguetá, que contribuíram de alguma forma para a realização deste estudo.

aos gênios do passado, que dedicaram sua vida em prol da ciência e progresso da humanidade.

“A ciência é, portanto, uma perversão de si mesma, a menos que tenha como fim último, melhorar a humanidade.”

Nikola Tesla

RESUMO

Este trabalho de conclusão de curso consiste em um estudo e desenvolvimento de um sistema no qual seja possível controlar a temperatura, automaticamente, de uma carga em corrente alternada. O sistema é composto por um sensor de temperatura, um resistor cerâmico, um circuito externo de potência e um sistema de monitoramento, composto basicamente pelo microcontrolador Arduino. Para a sintonia do controlador PID e obtenção de seus parâmetros foram utilizados os métodos de Ziegler-Nichols, Cohen-Coon e do Relé. Posteriormente é feita uma comparação de desempenho qualitativa e quantitativa com relação a resposta ao degrau, com e sem distúrbio externo, para cada método utilizado.

PALAVRAS-CHAVE: Controlador PID. Ziegler-Nichols. Cohen-Coon. Relé.

ABSTRACT

This course completion work consists of a study and development of a system in which it is possible to automatically control the temperature of a load on alternating current. The system consists of a temperature sensor, a ceramic resistor, an external power circuit and a monitoring system, composed basically of an Arduino microcontroller. The Ziegler-Nichols, Cohen-Coon and the Relay methods were used for tuning the PID controller and obtaining its parameters. Subsequently a comparison of the qualitative and quantitative performance with respect to the step response, with and without external disturbance, is made for each method used.

KEYWORDS: PID Controller. Ziegler-Nichols. Cohen-Coon. Relay.

LISTA DE FIGURAS

Figura 1 – Governador centrífugo de James Watt.....	14
Figura 2 – Diagrama de blocos de um controlador proporcional.....	17
Figura 3 – Resposta de um sistema com controle proporcional à uma entrada degrau para diversos valores de ganho proporcional K_p	18
Figura 4 – Diagrama de blocos de um controlador PI.....	19
Figura 5 – Resposta de um sistema com controlador PI à uma entrada degrau para diversos valores de tempo integral T_i	19
Figura 6 – Diagrama de blocos de um controlador PD.....	21
Figura 7 – Resposta de um sistema com controlador PD à uma entrada degrau para diversos valores de tempo derivativo T_d	21
Figura 8 – Diagrama de blocos de um controlador PID.....	22
Figura 9 – Resposta típica de um sistema com controlador PID à uma entrada degrau para alguns valores de parâmetros K_p , K_i e K_d	23
Figura 10 – Curva de reação.....	24
Figura 11 – Método da sensibilidade limite.....	26
Figura 12 – Método do decaimento de $\frac{1}{4}$ de onda.....	27
Figura 13 – Método de Cohen-Coon Malha Aberta.....	28
Figura 14 – Diagrama de blocos do método do relé.....	30
Figura 15 – Resposta do sistema à aplicação do relé.....	30
Figura 16 – Automatização do método do relé.....	31
Figura 17 – Critérios de desempenho.....	32
Figura 18 – Circuito detector de passagem por zero.....	33
Figura 19 – Sinal pulsado de saída do detector de passagem por zero.....	34
Figura 20 – Sensor de temperatura DS18B20.....	35
Figura 21 – Carga resistiva acoplada ao sensor.....	36
Figura 22 – Disparo do TRIAC.....	37
Figura 23 – Circuito de potência.....	37
Figura 24 – Circuito Geral.....	39
Figura 25 – Fluxograma básico do programa.....	40
Figura 26 – Esquemático do distúrbio.....	41
Figura 27 – Leitura de temperatura no monitor serial do Arduino.....	42

Figura 28 – Comportamento em malha aberta do sistema para o método de Ziegler-Nichols Malha Aberta	43
Figura 29 – Resposta do sistema para $K_p = 52.68$, $K_i = 2.64$ e $K_d = 264$	44
Figura 30 – Resposta do sistema, com distúrbios, para $K_p = 52.68$, $K_i = 2.64$ e $K_d = 264$	44
Figura 31 – Resposta do sistema para $K_p = 200$, $K_i = 0$ e $K_d = 0$	45
Figura 32 – Resposta do sistema para $K_p = 120$, $K_i = 0.873$ e $K_d = 4125$	46
Figura 33 – Resposta do sistema, com distúrbios, para $K_p = 120$, $K_i = 0.873$ e $K_d = 4125$	47
Figura 34 – Comportamento em malha aberta do sistema para o método de Cohen-Coon Malha Aberta	48
Figura 35 – Resposta do sistema para $K_p = 87.6$, $K_i = 3.55$ e $K_d = 322.44$	49
Figura 36 – Resposta do sistema, com distúrbios, para $K_p = 87.6$, $K_i = 3.55$ e $K_d = 322.44$	49
Figura 37 – Resposta do sistema ao relé.....	50
Figura 38 – Resposta do sistema para $K_p = 0.764$, $K_i = 0.0041$ e $K_d = 35.335$	51
Figura 39 – Resposta do sistema, com distúrbio, para $K_p = 0.764$, $K_i = 0.004$ e $K_d = 35.33$	52
Figura 40 – Resposta para todos métodos apresentados.....	53
Figura 41 – Critérios de desempenho para a resposta ao distúrbio	55

LISTA DE TABELAS

Tabela 1 – Efeito dos modos, individualmente, em um controle tipo PID.....	23
Tabela 2 – Parâmetros do PID para o método de Ziegler-Nichols Malha Aberta.....	25
Tabela 3 – Parâmetros do PID para o método da sensibilidade limite.....	26
Tabela 4 – Parâmetros do PID para o método do decaimento de $\frac{1}{4}$ de onda.....	27
Tabela 5 – Parâmetros do PID para o método de Cohen-Coon Malha Aberta.....	29
Tabela 6 – Parâmetros do PID para o método do Relé.....	31
Tabela 7 – Características básicas do acoplador óptico 4N25.....	34
Tabela 8 – Características básicas da ponte de diodo W08M.....	35
Tabela 9 – Características básicas do acoplador óptico MOC3011.....	38
Tabela 10 – Características básicas do TRIAC BTA16.....	38
Tabela 11 – Características básicas do Arduino Mega 2560.....	39
Tabela 12 – Níveis de distúrbio aplicado.....	41
Tabela 13 – Valores calculados de K_p , K_i e K_d para o método de Ziegler-Nichols Malha Aberta.....	43
Tabela 14 – Valores calculados de K_p , K_i e K_d para o método de Ziegler-Nichols Malha Fechada com decaimento de $\frac{1}{4}$ de onda.....	46
Tabela 15 – Valores calculados de K_p , K_i e K_d para o método de Cohen-Coon Malha Aberta.....	48
Tabela 16 – Valores calculados de K_p , K_i e K_d para o método do Relé.....	50
Tabela 17 – Valores de K_p , K_i e K_d para todos os métodos.....	52
Tabela 18 – Critérios de desempenho para todos os métodos.....	54
Tabela 19 – Critérios de desempenho ao distúrbio leve.....	56
Tabela 20 – Critérios de desempenho ao distúrbio moderado.....	56
Tabela 21 – Critérios de desempenho ao distúrbio forte.....	57

LISTA DE ABREVIATURAS E SIGLAS

AC	<i>Alternating Current</i>
DC	<i>Direct Current</i>
SP	<i>Set Point</i>
PV	<i>Process Value</i>
MV	<i>Manipulated Value</i>
M	<i>Overshoot</i>
M%	<i>Overshoot percentual</i>
P	Proporcional
PI	Proporcional-Integral
PD	Proporcional-Derivativo
PID	Proporcional-Integral-Derivativo
CO	<i>Controller Output</i>
PWM	<i>Pulse Width Modulation</i>
TRIAC	<i>Triode for Alternating Current</i>
Z-N	Ziegler-Nichols
C-C	Cohen-Coon

LISTA DE SÍMBOLOS

t	Tempo
T	Temperatura
$e(t)$	Sinal de erro
$u(t)$	Sinal de saída do controlador
$y(t)$	Sinal de saída do processo
K_p	Ganho proporcional
K_i	Ganho integral
K_d	Ganho derivativo
T_i	Tempo integral
T_d	Tempo derivativo
T_U	Período de oscilação
t_0	Tempo de duplicação
t_S	Tempo de subida
t_A	Tempo de assentamento
$T_{MÍN}$	Temperatura mínima durante distúrbio
$T_{MÁX}$	Temperatura máxima após retirada do distúrbio
t_{REC}	Tempo de recuperação ao distúrbio

SUMÁRIO

1	INTRODUÇÃO	14
1.1	HISTÓRIA	14
1.2	MOTIVAÇÃO.....	15
1.3	ORGANIZAÇÃO.....	16
2	CONTROLADOR PID	17
2.1	AÇÃO PROPORCIONAL (P)	17
2.2	AÇÃO PROPORCIONAL-INTEGRAL (PI).....	18
2.3	AÇÃO PROPORCIONAL-DERIVATIVA (PD)	20
2.4	AÇÃO PROPORCIONAL-INTEGRA-DERIVATIVA (PID)	21
3	MÉTODOS DE SINTONIA	24
3.1	ZIEGLER-NICHOLS MALHA ABERTA	24
3.2	ZIEGLER-NICHOLS MALHA FECHADA	25
3.2.1	Sensibilidade limite.....	25
3.2.2	Decaimento de $\frac{1}{4}$ de onda	26
3.3	COHEN-COON MALHA ABERTA	28
3.4	RELÉ	29
3.5	CRITÉRIOS DE DESEMPENHO	32
4	FUNCIONAMENTO	33
4.1	CIRCUITO ELETRÔNICO	33
4.1.1	Detector de passagem por zero.....	33
4.1.2	Sensor de temperatura	35
4.1.3	Carga	36
4.1.4	Circuito de potência	36
4.1.5	Microcontrolador.....	38
4.1.6	Circuito geral	38
4.2	PROGRAMA	39
5	APLICAÇÃO E RESULTADOS	41
5.1	CONSIDERAÇÕES INICIAIS PARA ANÁLISE	41
5.2	RESULTADOS DE ZIEGLER-NICHOLS MALHA ABERTA.....	42
5.2.1	Resposta do método Z-N malha aberta	42
5.2.2	Resposta ao distúrbio do método Z-N malha aberta.....	44
5.3	RESULTADOS DE ZIEGLER-NICHOLS MALHA FECHADA	45

5.3.1	Resposta do método z-n malha fechada.....	45
5.3.2	Resposta ao distúrbio do método Z-N malha fechada	46
5.4	Resultados de Cohen-Coon malha aberta	47
5.4.1	Resposta do método C-C malha aberta	47
5.4.2	Resposta ao distúrbio do método C-C malha aberta	49
5.5	RESULTADOS DO MÉTODO DO RELÉ	50
5.5.1	Resposta do método do Relé	50
5.5.2	Resposta ao distúrbio do método do Relé.....	51
5.6	DESEMPENHO	52
5.6.1	Comparação de resposta	52
5.6.2	Comparação de resposta ao distúrbio	54
6	CONCLUSÕES E TRABALHOS FUTUROS.....	58
	REFERÊNCIAS	59
	BIBLIOGRAFIA CONSULTADA.....	60
	APÊNDICE A – PROGRAMA COMENTADO.....	61

1 INTRODUÇÃO

1.1 HISTÓRIA

A necessidade de se controlar sistemas e processos industriais nos acompanha desde o surgimento da revolução industrial. O controle manual foi a primeira forma de controle utilizada em larga escala, no qual, depende de um operador humano que conheça o processo e possua habilidade na atuação deste. Com o crescente avanço na tecnologia e automatização nos processos industriais, essa tarefa foi ficando cada vez mais árdua, devido à complexidade para se atuar no sistema. Esta situação toda gerou uma demanda para solucionar o problema, que rapidamente atingiu a mente de grandes nomes do mundo acadêmico e inventores da época.

Em 1788, James Watt criou seu primeiro governador centrífugo, conhecido em inglês como *flyball governor*, que foi um dispositivo mecânico regulador que era capaz de controlar a velocidade de um motor a vapor através da regulação da quantidade de vapor admitido nos cilindros, a fim de manter a velocidade constante para diversos tipos de cargas aplicadas ao eixo, dentro de um certo limite operacional, utilizando apenas o conceito de controle proporcional (P). Na Figura 1 é mostrado o governador de Watt acoplado a um motor a vapor, exposto em um museu, localizado em Londres.

Figura 1 – Governador centrífugo de James Watt



Fonte: Junge (2008).

O controle proporcional, por si só, não é capaz de zerar o erro em regime permanente para muitos processos e sistemas, podendo inclusive causar instabilidade para valores de ganho proporcional muito alto. Por volta de 1930, engenheiros de controle descobriram que o erro em regime permanente poderia ser zerado ao se integrar o mesmo e posteriormente adicionar o resultado desta operação ao termo proporcional, hoje este tipo de controlador é conhecido como Proporcional-Integral (PI). Entretanto, a adição do termo integral ao sistema, pode gerar novos problemas, como o aumento do sobressinal máximo, *overshoot*, e também aumento da oscilação, *hunting*, (SANCHES, 2017).

Em 1940, foi lançado pela *Taylor Instrument Companies*, o primeiro controlador Proporcional-Integral-Derivativo (PID) pneumático. A ação derivativa era chamada em inglês de *pre-act*, e veio para corrigir os problemas gerados pelo controlador PI, especificamente, atuando para diminuir o *overshoot* e o *hunting* do sinal controlado. Contudo, nesta época, os controladores PID eram ainda pouco populares, devido as dificuldades de sintonização e pelo fato de que a maioria dos controladores eram pneumáticos, possuindo assim inúmeras limitações na capacidade de processamento. A popularidade dos controladores PID só aumentou em 1942, com a criação de um método simplificado de sintonia, que foi apresentado através de um artigo escrito por Ziegler e Nichols com o título de “*Optimum settings for automatic controllers*”, onde, de forma relativamente simples, era proposto uma metodologia para se obter os parâmetros do controlador, (DILLENBURG, 2016).

O começo da era do transistor, juntamente com o novo método de sintonia de Ziegler e Nichols, foi decretado o fim dos controladores pneumáticos, e assim possível a criação de controladores PID eletrônicos automáticos com capacidade de processamento elevada, ideais para sistemas e processos complexos, (DILLENBURG, 2016).

Atualmente, existem inúmeros métodos de sintonia e diferentes tipos de tecnologia para implementação de controladores PID, onde, em sua grande maioria, estão integrados internamente aos controladores lógicos programáveis e são capazes de realizar a sintonia de forma automática e segura, por exemplo, utilizando o método do Relé, que será apresentado mais adiante neste trabalho.

1.2 MOTIVAÇÃO

Tendo em vista a importância histórica e o uso abundante dos controladores PID na indústria, busca-se elaborar um estudo teórico e prático sobre os mesmos, voltado para o controle de temperatura de uma carga resistiva em corrente alternada.

O objetivo principal do controlador PID é manter a carga numa determinada temperatura, mesmo sujeito a distúrbios externos de diferentes intensidades, dentro de um certo limite operacional, tudo isto utilizando a placa de desenvolvimento Arduino Mega 2560.

1.3 ORGANIZAÇÃO

Além deste capítulo, o trabalho é apresentado da seguinte forma:

O capítulo 2, “Controlador PID”, apresenta uma fundamentação teórica sobre este tipo de controlador.

O capítulo 3, “Métodos de Sintonia”, descreve quatro métodos de sintonia utilizados para determinar os parâmetros do controlador.

O capítulo 4, “Funcionamento”, apresenta o funcionamento do circuito eletrônico e uma explicação sobre o programa utilizado no Arduino.

No capítulo 5, “Aplicação e Resultados”, são apresentados os resultados para cada método de sintonia e posteriormente é feita uma comparação de desempenho entre eles.

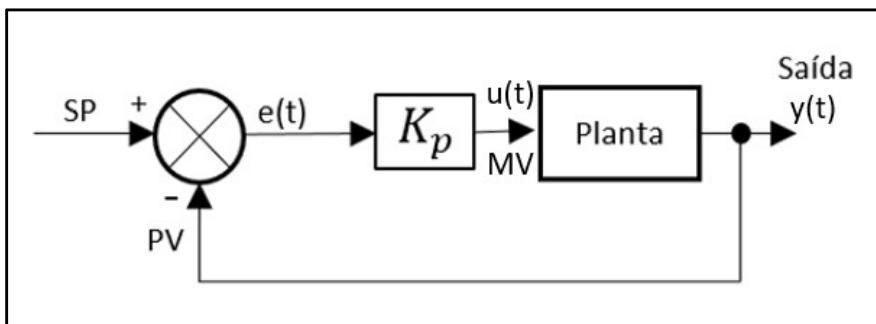
O capítulo 6, “Conclusões e Trabalhos Futuros”, evidencia as conclusões finais a respeito deste trabalho e propostas de trabalhos futuros.

2 CONTROLADOR PID

2.1 AÇÃO PROPORCIONAL (P)

O controlador com ação puramente proporcional, é aquele cujo a saída, $y(t)$, é diretamente proporcional ao sinal de entrada, *set point* (SP), multiplicado por uma constante. De maneira fácil, pode se entender o controlador proporcional como um amplificador com ganho ajustável. Um exemplo de controlador proporcional é apresentado na Figura 2.

Figura 2 – Diagrama de blocos de um controlador proporcional



Fonte: Próprio autor.

A relação entre o sinal de saída do controlador, $u(t)$, e o sinal de erro, $e(t)$, é dado pela equação (1), onde K_p é o ganho proporcional.

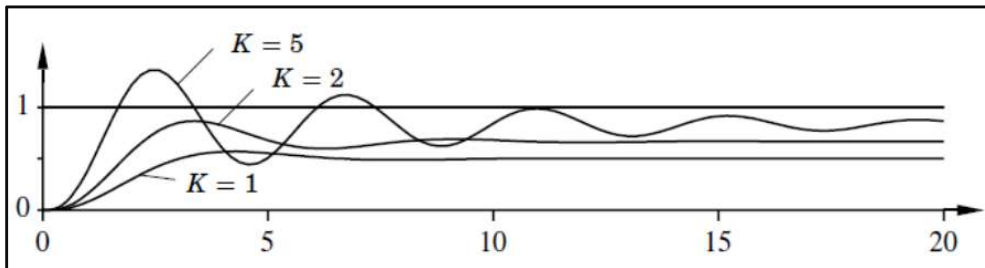
$$u(t) = K_p e(t) \quad (1)$$

A principal vantagem do modo proporcional está no fato de que ele fornece um ajuste rápido da variável manipulada (MV), tornando mais rápida a dinâmica do processo, (PINTO, 2014).

A grande desvantagem é que o aumento demasiado de K_p , pode conduzir o sistema à oscilação, e assim o processo deixa de ser estável, (SANCHES, 2017).

Na Figura 3, é apresentado a resposta de um sistema com controle puramente proporcional à uma entrada degrau, para diversos valores de K_p . Podemos notar que o aumento do ganho proporcional, realmente, contribui para que o sistema oscile indesejadamente.

Figura 3 – Resposta de um sistema com controle proporcional à uma entrada degrau para diversos valores de ganho proporcional K_p



Fonte: Åström e Hägglund (1995).

2.2 AÇÃO PROPORCIONAL-INTEGRAL (PI)

A ação integral é responsável por eliminar o erro em regime permanente, porém, nunca é utilizada sozinha. É sempre indicado usar um controlador proporcional-integral (PI), que mantém as propriedades quanto ao erro de um controlador integral puro e adiciona as vantagens do controlador proporcional a este, (GALDINO, 2012). A componente integral adiciona um pólo na origem da função de transferência do controlador, eliminando o erro se o sistema for estável em malha fechada, (SANCHES, 2017).

A relação entre o sinal de saída do controlador proporcional-integral, $u(t)$, e o sinal de erro, $e(t)$, é dado pela equação (2), onde T_i é o tempo integral ou tempo de *reset*, dado em segundos ou minutos.

$$u(t) = \frac{K_p}{T_i} \int e(t) dt \quad (2)$$

O ganho integral K_i é definido como:

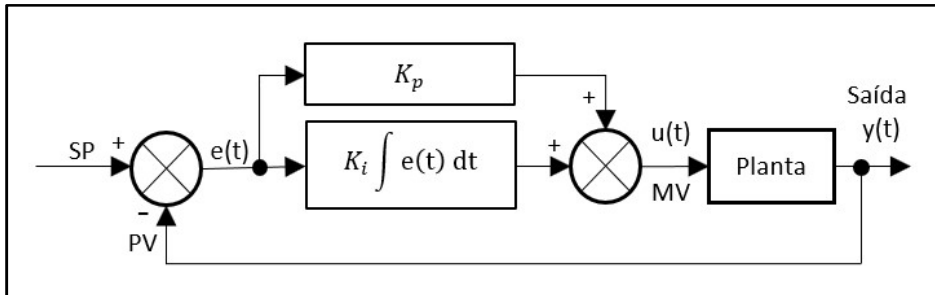
$$K_i = \frac{K_p}{T_i} \quad (3)$$

Portanto, a equação de saída deste tipo de controlador também é dada por:

$$u(t) = K_i \int e(t) dt \quad (4)$$

Na Figura 4 é apresentado um modelo básico de controlador PI.

Figura 4 – Diagrama de blocos de um controlador PI

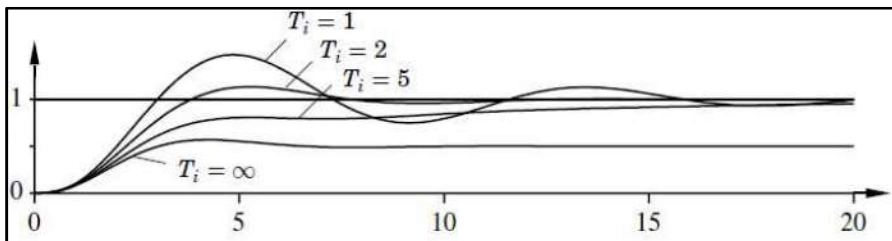


Fonte: Próprio autor.

A grande vantagem do controlador PI é o fato de eliminar o erro em regime permanente. Para valor de ganho integral K_i elevado, o *overshoot* do sinal de saída aumentará consideravelmente, o tempo que leva para chegar ao regime permanente também aumentará e a estabilidade relativa do sistema será degradada. Por isto, sempre quando utilizado a ação integral em um controlador, é necessário que o ajuste de sintonia dos parâmetros seja a melhor possível para evitar problemas indesejados, (SANCHES, 2017).

Na Figura 5, tem-se a resposta de um sistema com controlador PI à uma entrada degrau, para diversos valores de T_i . Pode-se notar que a diminuição do tempo integral e consequentemente um aumento do ganho integral K_i , de fato aumenta o *overshoot* e o tempo de estabelecimento.

Figura 5 - Resposta de um sistema com controlador PI à uma entrada degrau para diversos valores de tempo integral T_i



Fonte: Åström e Hägglund (1995).

2.3 AÇÃO PROPORCIONAL-DERIVATIVA (PD)

“A ação derivativa nunca é utilizada sozinha, pois se o erro fosse constante, o valor do controle seria nulo. Então, sempre é usada em conjunto com um controlador proporcional que resulta no controlador PD”, (OGATA, 2003, pg. 234).

Portanto, em concordância à citação acima, temos que a ação derivativa é usada somente em conjunto com a ação proporcional, formando assim o controlador proporcional-derivativo (PD). A função principal deste tipo de controlador é atuar no sentido de diminuir as variações do sinal de saída, reagindo de forma antecipada ao comportamento futuro do sinal de erro com base na sua taxa de variação, desta forma melhorando o desempenho do processo durante possíveis distúrbios transitórios, (MENEGHETTI, 2007).

A relação entre o sinal de saída do controlador proporcional-derivativo, $u(t)$, e o sinal de erro, $e(t)$, é dado pela equação (5), onde T_d é o tempo derivativo e representa o avanço introduzido pela ação derivativa, é dado em segundos ou minutos.

$$u(t) = K_p T_d \frac{de(t)}{dt} \quad (5)$$

O ganho derivativo K_d é definido como:

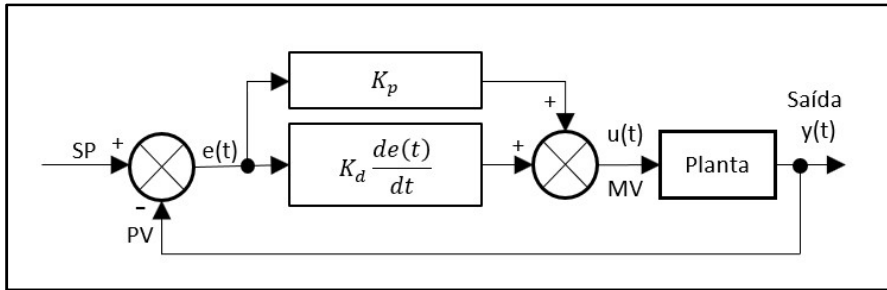
$$K_d = K_p T_d \quad (6)$$

Portanto, a equação de saída deste tipo de controlador também é dada por:

$$u(t) = K_d \frac{de(t)}{dt} \quad (7)$$

Na Figura 6, é apresentado a topologia de um controlador do tipo PD.

Figura 6 - Diagrama de blocos de um controlador PD

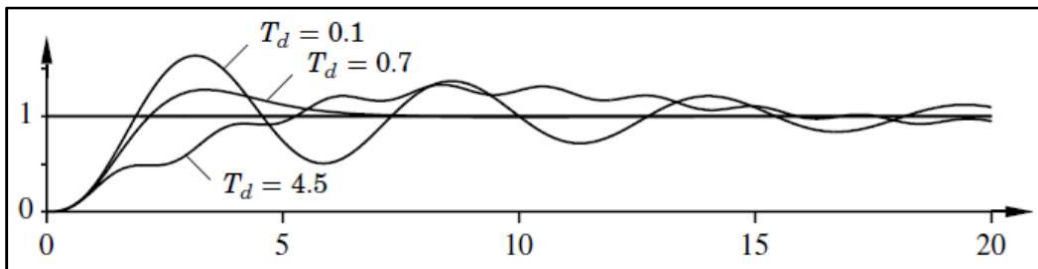


Fonte: Próprio autor.

Pode-se então citar como grande vantagem do controlador PD a capacidade de se antecipar ao erro futuro, realizando assim uma ação corretiva mais cedo. “A ação derivativa tem a desvantagem de amplificar os sinais de ruído, o que pode causar um efeito de saturação nos atuadores do sistema.” (MENEGHETTI, 2007, pg. 22)

Na Figura 7, é mostrada a resposta de um sistema com controlador PD à uma entrada degrau, para diversos valores de T_d . É possível perceber que com o aumento do tempo derivativo, acontece, de fato, uma diminuição na oscilação do sinal de saída.

Figura 7 - Resposta de um sistema com controlador PD à uma entrada degrau para diversos valores de tempo derivativo T_d



Fonte: Åström e Hägglund (1995).

2.4 AÇÃO PROPORCIONAL-INTEGRAL-DERIVATIVA (PID)

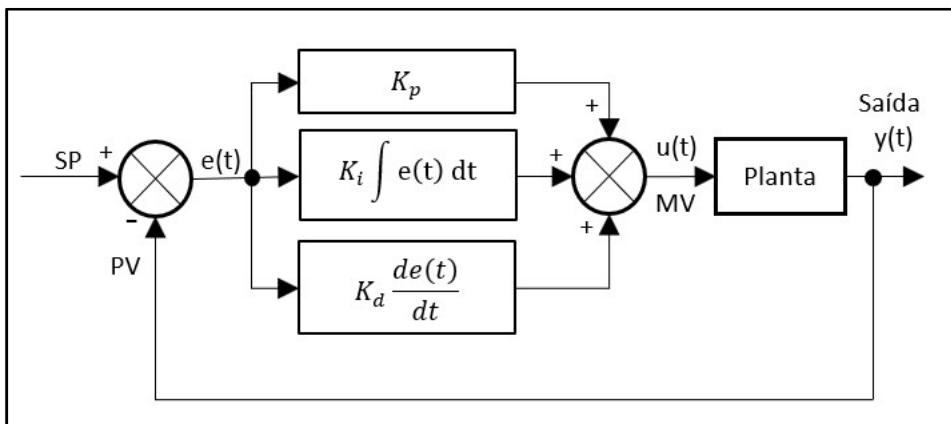
A ação proporcional-integral-derivativa nasce da junção das vantagens e desvantagens dos controladores PI e PD em um único dispositivo, o controlador PID. De acordo com os itens anteriores pode-se inferir que o modo proporcional fornece um ajuste rápido da varável manipulada (MV), o modo integral elimina o erro em regime permanente e o modo derivativo reduz a tendência a oscilações.

A relação entre o sinal de saída do controlador PID, $u(t)$, e o sinal de erro, $e(t)$, é dado basicamente pela soma das equações dos controladores PI e PD, resultando na equação (8)

$$u(t) = K_p + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (8)$$

Na Figura 8 é apresentado um modelo padrão de representação de um controlador PID em blocos.

Figura 8 – Diagrama de blocos de um controlador PID

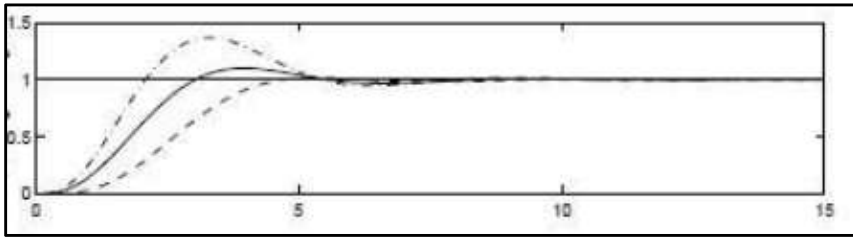


Fonte: Próprio autor.

Uma boa escolha dos valores dos parâmetros K_p , K_i e K_d , é de grande importância para um ótimo desempenho do controlador. Diversos métodos de sintonia podem ser usados para determinar esses parâmetros, porém, é sempre necessário a análise dos resultados obtidos por parte dos profissionais responsáveis pelo controle, pois cada processo é único e pode precisar de ajustes manuais devido a diferentes condições operacionais. Por exemplo, se o processo for extremamente sensível e não for tolerável a nenhum *overshoot*, o termo integral deve ser evitado e o ganho proporcional reduzido a fim de não causar problemas maiores. “Nem sempre, no entanto, faz-se necessário ou é viável o uso das três ações de controle para controle de variáveis em processos. Pois cada qual possui suas limitações perante determinadas situações”, (OGATA, 2003, pg. 557).

A Figura 9 mostra a resposta típica de um sistema com controlador PID.

Figura 9 - Resposta típica de um sistema com controlador PID à uma entrada degrau para alguns valores de parâmetros K_p , K_i e K_d



Fonte: Åström e Hägglund (1995).

As curvas de resposta ao degrau contidas na Figura 9 representam situações com diferentes escolhas de valores para os parâmetros do controlador PID, apresentando assim diferentes características de desempenho, como valor de *overshoot* e tempo de assentamento.

A Tabela 1, resume as ações de cada modo, individualmente, na resposta total da saída de um processo controlado por um PID.

Tabela 1 – Efeito dos modos, individualmente, em um controle tipo PID

Modo	Tempo de Subida	<i>Overshoot</i>	Erro	Tempo de Estabilização	Estabilidade
Proporcional	Diminui	Aumenta	Diminui	-	Degrada
Integral	Diminui	Aumenta	Diminui	Aumenta	Degrada
Derivativo	-	Diminui	-	Diminui	Aumenta

Fonte: Adaptado de Saraiva (2011).

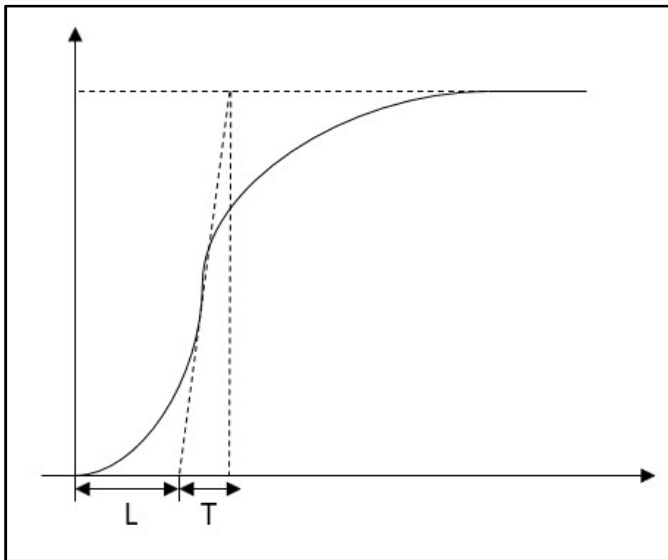
3 MÉTODOS DE SINTONIA

A seguir são apresentados quatro métodos para sintonia de controladores PID e suas respectivas vantagens e desvantagens, bem como, o procedimento técnico para cálculos dos parâmetros K_p , K_i e K_d .

3.1 ZIEGLER-NICHOLS MALHA ABERTA

O método de Ziegler-Nichols em malha aberta, também conhecido como curva de reação, é o método mais simples para obtenção dos parâmetros de um controlador PID, necessitando apenas da resposta ao degrau do sistema em malha aberta. É indicado que a resposta tenha o formato de um S, para situações onde a resposta foge muito deste formato, este método não é recomendado, (SARAIVA, 2011).

Figura 10 – Curva de reação



Fonte: Próprio autor.

A Figura 10 apresenta uma curva de reação em malha aberta, é necessário então traçar uma reta tangente no ponto de inflexão da curva. O tempo T é a constante de tempo e o tempo L é o atraso na resposta, ambos dados em segundos ou minutos. Dispondo dos valores de T e L e da Tabela 2, é possível então calcular as constantes K_p , K_i e K_d , (SARAIVA, 2011).

Tabela 2 – Parâmetros do PID para o método de Ziegler-Nichols Malha Aberta

Ziegler-Nichols Malha Aberta			
Controlador	K_p	K_i	K_d
P	$\frac{T}{L}$	-	-
PI	$0.9 \frac{T}{L}$	$0.3 \frac{K_p}{L}$	-
PID	$1.2 \frac{T}{L}$	$\frac{K_p}{2L}$	$K_p \frac{L}{2}$

Fonte: Próprio autor.

A grande desvantagem deste método é que para sistemas mais complexos onde a curva de reação não acompanhe o formato em S, não apresenta uma boa escolha de parâmetros. Mesmo assim, para curvas próximas de um S, é possível traçar uma reta tangente aproximada e assim obter os parâmetros de sintonia, de modo satisfatório.

É importante ressaltar que o método de Ziegler-Nichols é um método empírico, todavia, segundo Lotufo (2018), existem estudos recentes que comprovam a robustez deste, resultando em um bom método para escolha dos parâmetros do controlador PID.

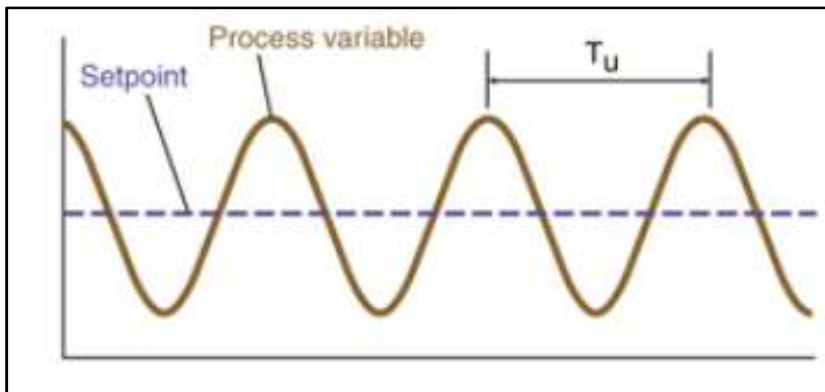
3.2 ZIEGLER-NICHOLS MALHA FECHADA

Existem dois métodos para determinação dos parâmetros do controlador PID por Ziegler-Nichols em malha fechada e que serão apresentados nos itens a seguir.

3.2.1 Sensibilidade limite

Este método consiste em zerar os ganhos integral e derivativo, e aumentar desde zero o valor do ganho proporcional até um valor onde se obtenha na saída uma oscilação sustentada e simétrica em torno do *set point*, ou seja, levando o sistema a uma estabilidade marginal. A Figura 11 demonstra esta situação, onde o ganho proporcional está neste patamar limite, beirando a instabilidade.

Figura 11 – Método da sensibilidade limite



Fonte: VanDoren (2009).

O ganho proporcional responsável por levar o sistema a essa oscilação é chamado de ganho crítico e é representado por K_u , o período de oscilação é chamado de período crítico e é representado por T_u . Disposto destas duas informações apenas, se é capaz de calcular os parâmetros do PID segundo a Tabela 3.

Tabela 3 – Parâmetros do PID para o método da sensibilidade limite

Ziegler-Nichols Malha Fechada (Sensibilidade Limite)			
Controlador	K_p	K_i	K_d
P	$\frac{K_u}{2}$	-	-
PI	$\frac{K_u}{2.2}$	$1.2 \frac{K_p}{T_u}$	-
PID	$0.6 K_u$	$2 \frac{K_p}{T_u}$	$\frac{K_p T_u}{8}$

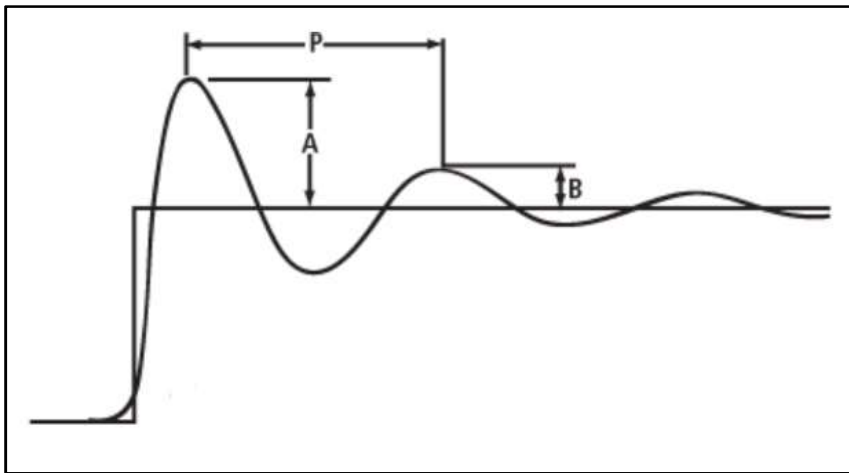
Fonte: Próprio autor.

A grande desvantagem em utilizar este método, é o fato de termos que levar o sistema a oscilação, o que para alguns processos pode ser extremamente desastroso, como processos envolvendo robôs. Outro fator importante ressaltar é que achar o ganho crítico pode ser muito demorado, pois é um processo de tentativa e erro, limitando a utilização deste método para processos extremamente lentos.

3.2.2 Decaimento de $\frac{1}{4}$ de onda

Neste método, tem-se uma variação do apresentado anteriormente, onde não se precisa levar o sistema a uma oscilação sustentada, precisando apenas que ocorra um decaimento de $\frac{1}{4}$ de onda entre o primeiro pico e o segundo pico de sinal, dado pela relação B/A , conforme mostrado na Figura 12.

Figura 12 – Método do decaimento de $\frac{1}{4}$ de onda



Fonte: Wade (2005).

O período P mostrado na Figura 12 é representado por T_u , que é o período de amortecimento, e o ganho proporcional que leva o sistema a esta condição é chamado de K_u . A relação de B/A deve ser de aproximadamente 0,25 para que a determinação dos parâmetros ocorra de forma satisfatória. Portanto, dispondo de K_u e T_u , é possível calcular as constantes do PID segundo a tabela 4.

Tabela 4 – Parâmetros do PID para o método do decaimento de $\frac{1}{4}$ de onda

Ziegler-Nichols Malha Fechada (Decaimento de $\frac{1}{4}$ de onda)			
Controlador	K_p	K_i	K_d
P	$\frac{K_u}{2}$	-	-
PI	$\frac{K_u}{2.2}$	$1.2 \frac{K_p}{T_u}$	-
PID	$0.6 K_u$	$2 \frac{K_p}{T_u}$	$\frac{K_p T_u}{8}$

Fonte: Próprio autor

Uma desvantagem deste método é que, às vezes, não é fácil conseguir esse decaimento de $\frac{1}{4}$ de onda, o que torna o método inviável para muitos casos. Além de ser, também, um processo tedioso de tentativa e erro, similar ao anterior, (SARAIVA, 2011).

3.3 COHEN-COON MALHA ABERTA

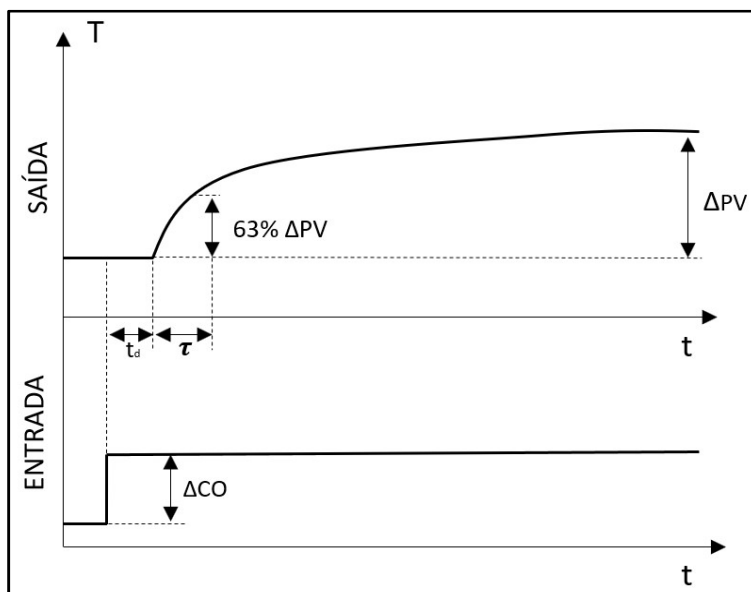
O método heurístico criado por Cohen e Coon (1953) é uma alternativa ao método de Ziegler-Nichols, necessitando, também, apenas do comportamento do sistema em malha aberta mediante uma entrada em degrau. Dispondo da curva de reação do sistema, é preciso apenas determinar o tempo morto, a constante de tempo e o ganho do processo.

O tempo morto, *dead time*, é o tempo em que a saída demora a reagir a um estímulo na entrada, representado por t_d . A constante de tempo, τ , é o intervalo em que a saída demora a atingir 63% do valor alvo. O ganho do processo pode ser determinado como a variação da saída, *process value*, pela variação da entrada, *controler output*, e é representado por g_p , conforme a equação (9).

$$g_p = \frac{\Delta PV}{\Delta CO} \quad (9)$$

A Figura 13, mostra graficamente como determinar esses parâmetros.

Figura 13 – Método de Cohen-Coon Malha Aberta



Fonte: Adaptado de Smuts (2011).

Com os valores de t_d , τ e o ganho do processo g_p , pode-se utilizar da tabela 5 para determinar as constantes K_p , K_i e K_d .

Tabela 5 - Parâmetros do PID para o método de Cohen-Coon malha aberta

Cohen-Coon Malha Aberta			
Controlador	K_p	K_i	K_d
P	$\frac{1.03}{g_p} \left(\frac{\tau}{t_d} + 0.34 \right)$	-	-
PI	$\frac{0.9}{g_p} \left(\frac{\tau}{t_d} + 0.092 \right)$	$K_p / \left(3.33 t_d \frac{\tau + 0.092 t_d}{\tau + 2.22 t_d} \right)$	-
PID	$\frac{1.35}{g_p} \left(\frac{\tau}{t_d} + 0.185 \right)$	$K_p / \left(2.50 t_d \frac{\tau + 0.185 t_d}{\tau + 0.611 t_d} \right)$	$K_p 0.37 t_d \left(\frac{\tau}{\tau + 0.185 t_d} \right)$

Fonte: Adaptado de Smuts (2011).

A grande vantagem deste método é a facilidade na obtenção do comportamento em malha aberta do sistema e conseqüentemente na determinação dos parâmetros do controlador.

Segundo Lotufo (2018), por ser um método bastante similar ao de Ziegler-Nichols apresenta resultados com boa robustez.

É um método que apresenta bons resultados para sistemas com tempo morto mais elevado, se mostrando uma ótima alternativa a outros métodos mais complexos e com possibilidade de autodestruição com ganhos proporcionais muito altos, (SMUTS, 2011).

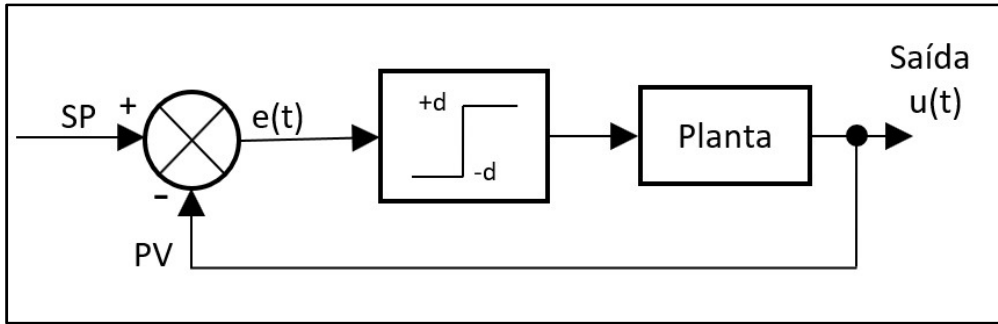
3.4 RELÉ

O método do relé foi apresentado primeiramente por Åström e Hägglund (1984) e se apresenta como um método simples de sintonizar controladores PID, que evita o longo processo de tentativa e erro requerido por outros métodos, e, além disso, não é preciso estressar o sistema levando-o a níveis perigosos de ganho proporcional.

O processo de obtenção dos parâmetros do controlador PID consiste em colocar um relé no sistema em malha fechada, com amplitude limitada, a fim de provocar uma oscilação no sistema com amplitude e período bem definidos.

A Figura 14 demonstra o diagrama de blocos básico para aplicação do método do relé no sistema.

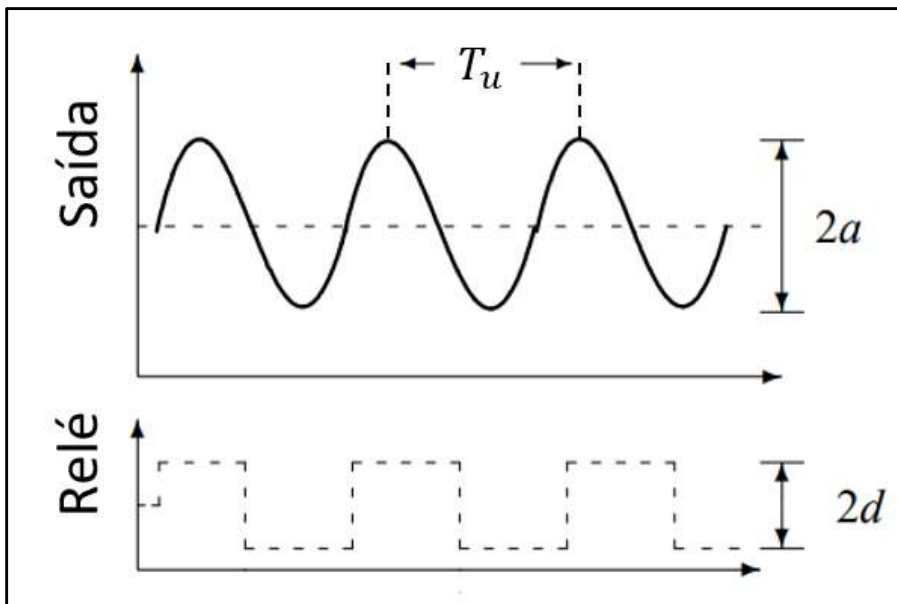
Figura 14 – Diagrama de blocos do método do relé



Fonte: Próprio autor.

Após a aplicação do relé ao sistema, pode ser observada na saída, uma oscilação de amplitude a e período T_u . A amplitude do relé é representada por d , conforme a Figura 15.

Figura 15 – Resposta do sistema à aplicação do relé



Fonte: Wilson (2005).

O cálculo do ganho crítico K_u , depende apenas da amplitude do relé e também da amplitude de saída, dado pela equação (10).

$$K_u = \frac{4d}{\pi a} \quad (10)$$

Após o cálculo do ganho crítico, pode-se utilizar da Tabela 6 para determinar os parâmetros do controlador PID.

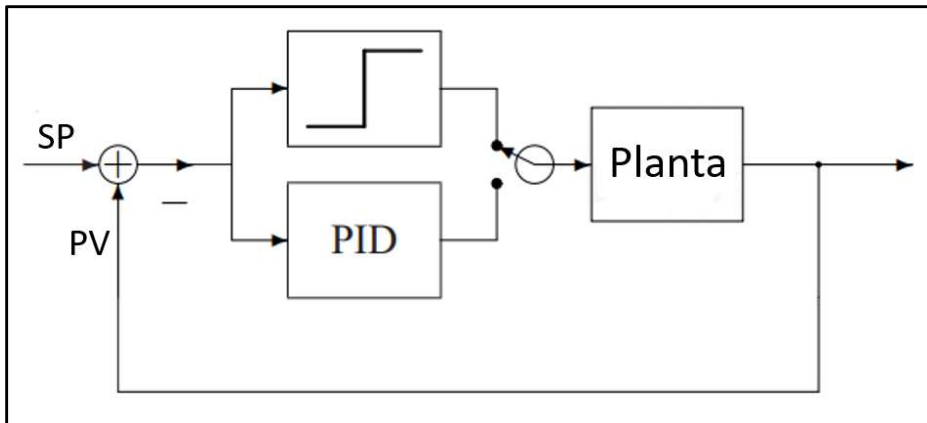
Tabela 6 - Parâmetros do PID para o método do Relé

Método Relé Malha Fechada			
Controlador	K_p	K_i	K_d
P	$\frac{K_u}{2}$	-	-
PI	$\frac{K_u}{2.2}$	$1.2 \frac{K_p}{T_u}$	-
PID	$0.6 K_u$	$2 \frac{K_p}{T_u}$	$\frac{K_p T_u}{8}$

Fonte: Próprio autor.

A grande utilização do método do relé nos dias atuais é nos chamados auto sintonizadores, *autotuners*, por ser um processo facilmente automatizado e sem grande necessidade de processamento lógico. A automatização é feita introduzindo o relé durante um tempo limitado, necessário apenas para determinação dos parâmetros do controlador, e após isso ele é desligado e o controlador PID assume o controle do sistema. A Figura 16 demonstra esse processo.

Figura 16 – Automatização do método do relé

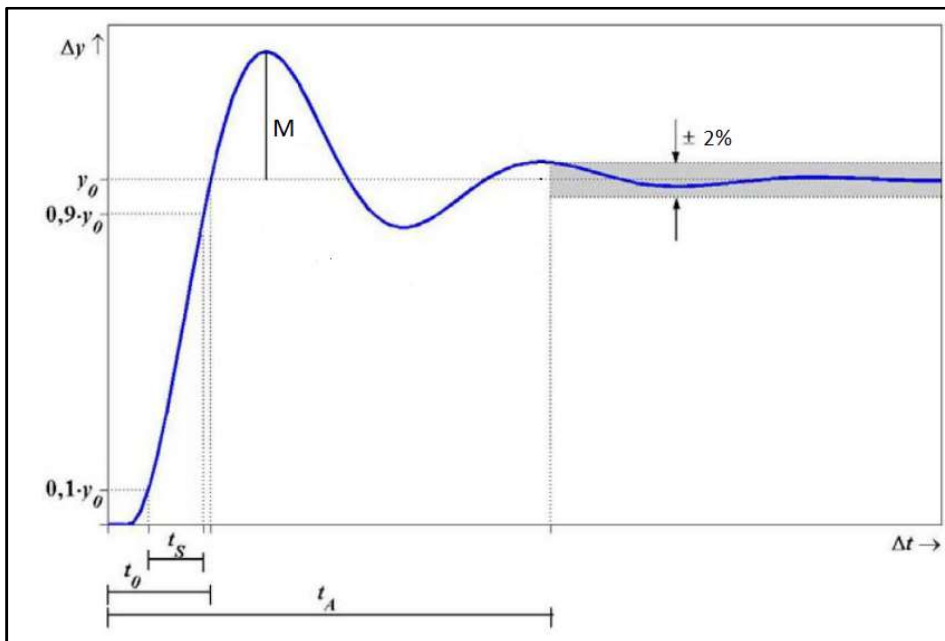


Fonte: Wilson (2005).

3.5 CRITÉRIOS DE DESEMPENHO

Para efeito de comparação entre os métodos na análise de resultados, no capítulo 5, são utilizados alguns critérios de desempenho para este propósito. A Figura 17 ilustra os critérios que irão ser utilizados.

Figura 17 – Critérios de desempenho



Fonte: Adaptado de Saraiva (2011).

O **tempo de duplicação**, t_0 , é o tempo necessário para que ocorra pela primeira vez erro nulo.

O **tempo de subida**, t_s , é o tempo requerido que o sistema leva para passar de 10% a 90% do valor desejado de *set point*, representado por y_0 .

O **tempo de assentamento**, t_A , é o tempo total até que se obtenha na saída um erro menor ou igual a 2%.

O **overshoot**, M , é o valor máximo de pico da curva de resposta medido a partir da resposta desejada do sistema.

O **overshoot percentual**, $M\%$, é a relação percentual entre o *overshoot* M e o valor de resposta desejada, y_0 , dado pela equação (11), (OGATA, 2003).

$$M\% = \frac{M}{y_0} * 100 \quad (11)$$

4 FUNCIONAMENTO

O funcionamento do protótipo é dividido em duas partes principais, o funcionamento do circuito eletrônico e do programa utilizado no Arduino.

4.1 CIRCUITO ELETRÔNICO

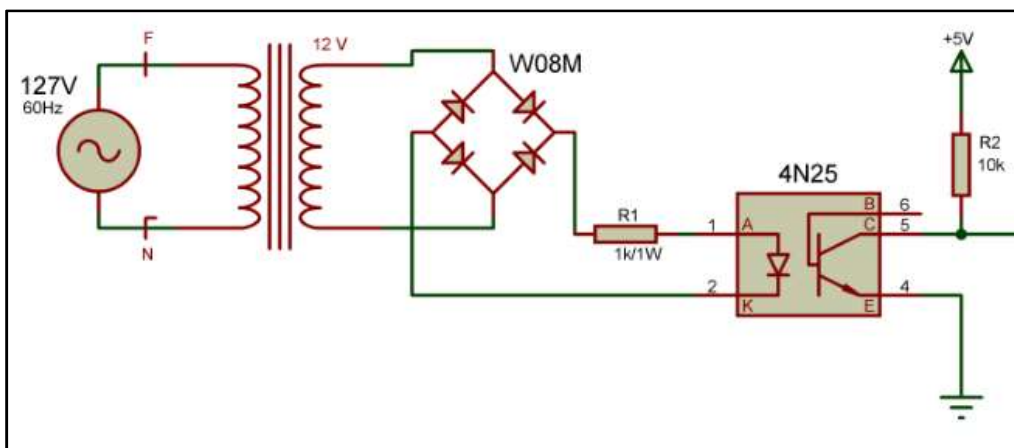
O circuito eletrônico é subdividido em algumas partes para melhor explicação do funcionamento.

4.1.1 Detector de passagem por zero

A função deste circuito é basicamente gerar um pulso positivo de +5V cada vez que a tensão alternada da rede cruzar o ponto de tensão zero. Esses pulsos gerados são utilizados como referência para o microcontrolador gerar o *delay* de disparo do dispositivo eletrônico de potência responsável pelo controle da carga, que é discutido posteriormente.

O circuito projetado é mostrado na Figura 18.

Figura 18 – Circuito detector de passagem por zero



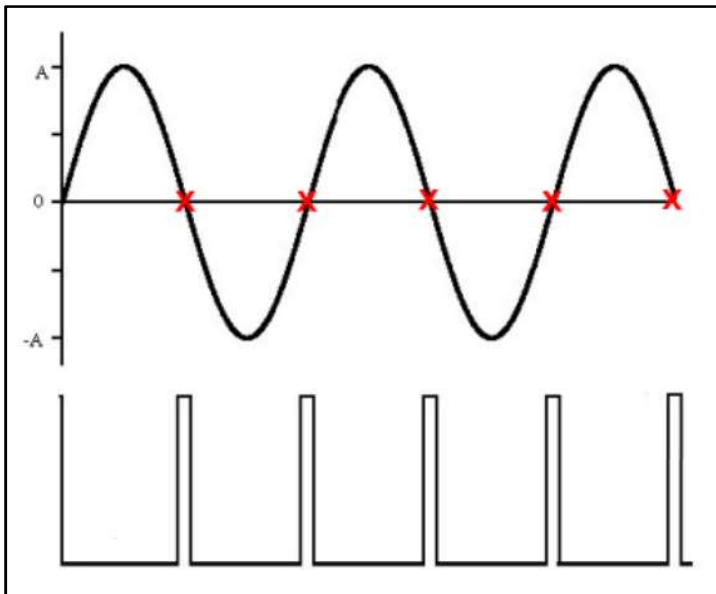
Fonte: Próprio autor.

A tensão da rede de 127 Volts e 60 Hertz é aplicada em um transformador abaixador, com saída de 12V, que então é retificada por uma ponte de diodos, W08M, a fim de eliminar a parte negativa do sinal. Este sinal pulsante, já retificado, é aplicado através de um resistor limitador de corrente de $1\text{ k}\Omega$ em um acoplador óptico, circuito integrado 4N25. Este

acoplador óptico tem como função principal isolar totalmente a tensão da rede do sinal pulsado que queremos enviar ao microcontrolador.

Cada vez que o sinal da rede cruza o valor nulo de tensão o transistor interno do acoplador é cortado, e o nível lógico de +5V é garantido no coletor pelo resistor de 10 kΩ. A Figura 19 demonstra a relação entre o sinal de entrada da rede e o sinal pulsado gerado no pino 5 do circuito integrado 4N25.

Figura 19 – Sinal pulsado de saída do detector de passagem por zero



Fonte: Loflin (2012).

Tabela 7 – Características básicas do acoplador óptico 4N25

4N25		
Entrada	Máxima Tensão Reversa	5 V
	Corrente Máxima Direta	60 mA
	Máxima Corrente de Surto	3 A
	Potência de Dissipação	100 mW
Saída	Tensão Coletor-Emissor Máxima	70 V
	Tensão Base-Emissor Máxima	7 V
	Máxima Corrente de Coletor	50 mA
	Potência de Dissipação	150 mW

Fonte: Próprio autor.

Tabela 8 – Características básicas da ponte de diodo W08M

W08M	
Corrente Direta Máxima	1.5 A
Corrente Reversa Máxima	10 μ A
Tensão de Queda por Diodo	1.0 V
Máxima Tensão Reversa de Pico	800 V

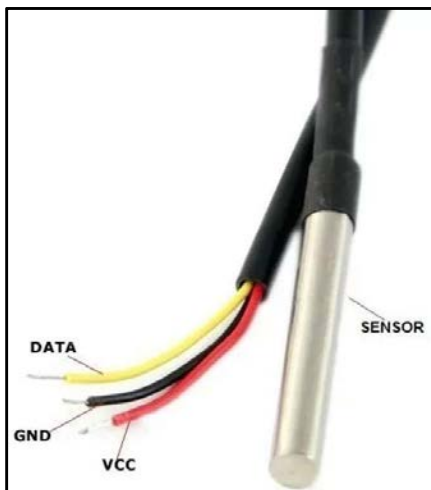
Fonte: Próprio autor.

4.1.2 Sensor de temperatura

O sensor de temperatura é o elemento responsável pela aquisição dos dados de temperatura a cada intervalo de tempo pré-determinado.

O sensor escolhido para o uso neste projeto foi o DS18B20 fabricado pela *Dallas Instruments*, possui uma precisão de $\pm 0,5$ °C e pode medir valores numa faixa entre -10 °C e 85°C. Funciona com a tecnologia de *1-Wire*, ou seja, requiere apenas uma linha de comunicação com o microcontrolador. A figura 20 ilustra uma fotografia do sensor utilizado.

Figura 20 – Sensor de temperatura DS18B20



Fonte: Johnson (2016).

O fio vermelho, **VCC**, é a alimentação positiva do sensor, que pode variar entre 3.3V e 5.0 V.

O fio preto, **GND**, é a tensão de referência 0V ou *ground*.

O fio amarelo, **DATA**, é o pino de saída do sensor responsável por enviar as leituras de temperatura ao microcontrolador.

4.1.3 Carga

A carga é o elemento físico onde se deseja controlar a temperatura, é puramente resistiva e formada unicamente por um resistor cerâmico com valor de $3,6\text{ k}\Omega$ e potência máxima de dissipação de 15 W .

Figura 21 – Carga resistiva acoplada ao sensor



Fonte: Próprio autor.

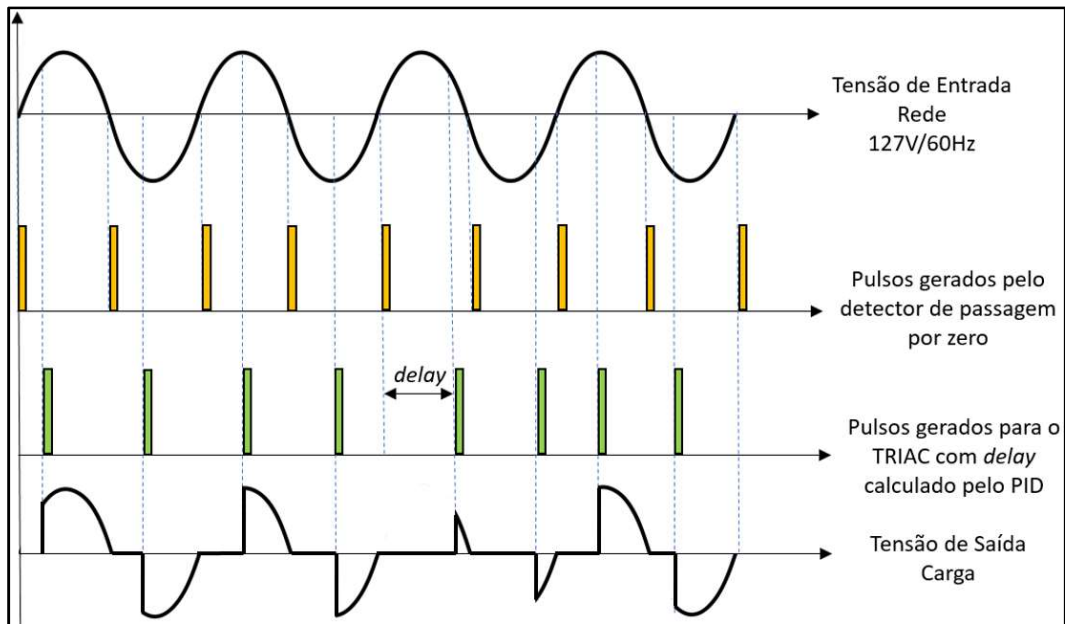
4.1.4 Circuito de potência

Para realizar o controle de potência numa carga em corrente alternada, é necessário variar o valor da tensão que é aplicada a esta, variando assim a potência. O valor da tensão é alterado através da variação do ângulo de disparo sobre o TRIAC.

Quanto maior o *delay* aplicado, menor será a tensão eficaz aplicada a carga. Com essa variação de ângulo de disparo é possível controlar a potência sobre a carga, e consequentemente a temperatura sobre a mesma, visto que é uma carga puramente resistiva.

Após o Arduino receber o pulso do circuito detector de passagem por zero, o algoritmo do PID calcula o *delay* necessário para levar ou manter a carga no *set point* desejado, e assim manda um pulso para o acoplador óptico MOC3011, que é responsável pelo disparo do TRIAC de potência BTA16. A figura 22 mostra as ações, de cada etapa citada acima, de forma gráfica.

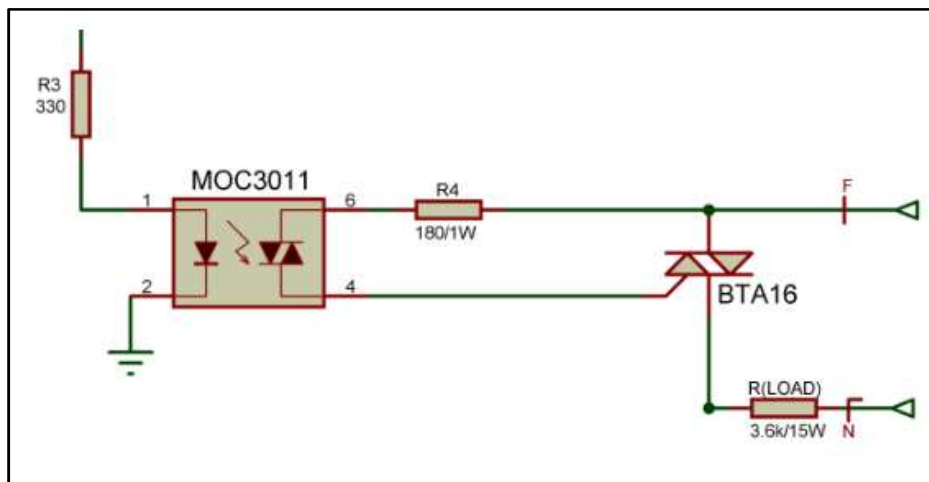
Figura 22 – Disparo do TRIAC



Fonte: Próprio autor.

O circuito de potência é ilustrado na figura 23.

Figura 23 – Circuito de potência



Fonte: Próprio autor.

Tabela 9 – Características básicas do acoplador óptico MOC3011

MOC3011		
Entrada	Máxima Corrente Direta	60 mA
	Potência de Dissipação	100 mW
Saída	Corrente de Surto Máxima	1 A
	Potência de Dissipação	300 mW

Fonte: Próprio autor.

Tabela 10 – Características básicas do TRIAC BTA16

BTA16	
Máxima Corrente	16 A
Máxima Corrente de Pico	170 A
Tensão de Bloqueio	800 V

Fonte: Próprio autor.

4.1.5 Microcontrolador

O microcontrolador escolhido foi o Arduino Mega 2560 devido a sua facilidade na programação e preço acessível.

O dispositivo possui 54 pinos digitais, nos quais 15 podem ser configurados como saída PWM. Cada pino pode fornecer ou receber um valor máximo de corrente de 40 mA.

A alimentação da placa pode ser feita através da comunicação USB com o computador, ou através de fonte externa de +7 a +12 V.

O *software* de interface de programação do Arduino possui um monitor serial que permite que dados de texto sejam enviados da placa para o controlador, ou vice-versa. A Tabela 11 resume as principais características do dispositivo.

Tabela 11 – Características básicas do Arduino Mega 2560

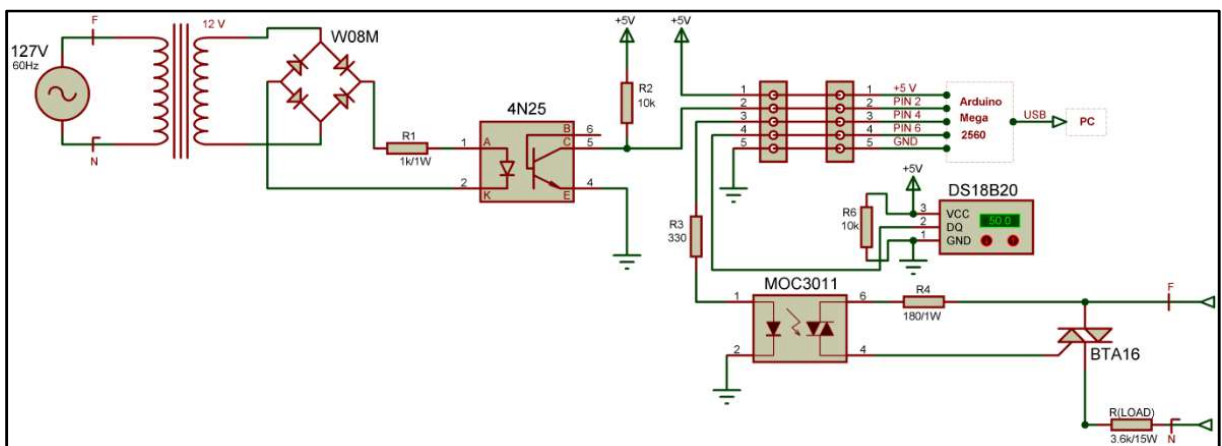
Arduino Mega 2560	
Chip	ATMEGA 2560
Tensão de Operação	5 V
Tensão de Alimentação (Recomendada)	7-12 V
Tensão de Alimentação (Limite)	6-20 V
Entradas e Saídas digitais	54 (15 PWM)
Entradas Analógicas	16
Memória Flash	256 <u>kB</u>
Memória SRAM	8 <u>kB</u>
EEPROM	4 <u>kB</u>
Clock	16 MHz
Dimensões	101.6 x 53.4 mm
Peso	150 g

Fonte: Próprio autor.

4.1.6 Circuito geral

O circuito geral contém todos os elementos e circuitos parciais apresentados anteriormente e é apresentado na Figura 24.

Figura 24 – Circuito Geral



Fonte: Próprio autor.

4.2 PROGRAMA

O programa completo e comentado pode ser visto no Apêndice A deste trabalho. Neste capítulo é descrito, apenas, o funcionamento modular do programa.

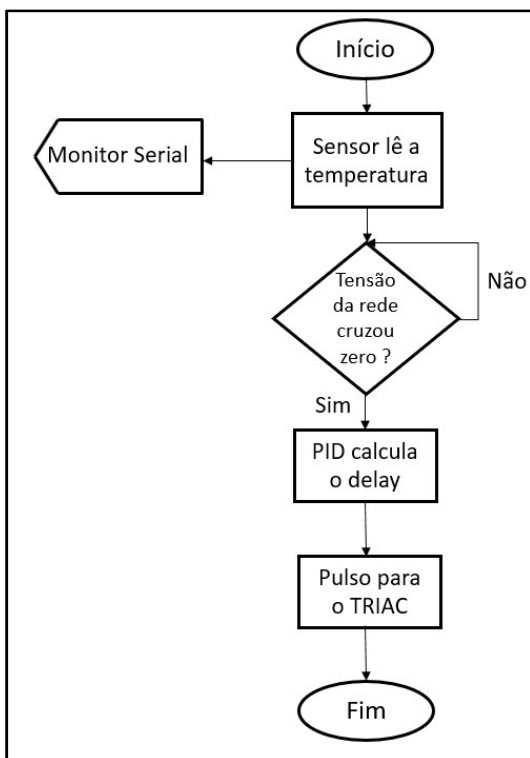
Como visto anteriormente, a cada vez que a tensão da rede atinge o valor nulo, um pulso é gerado na saída do detector de passagem por zero. Este pulso é recebido pelo Arduino e é tratado como uma interrupção no programa. A interrupção é um evento externo que faz o processador parar a execução do programa principal e desviar para a execução de uma chamada rotina de interrupção, evitando assim que o microcontrolador tenha que varrer constantemente o programa para saber se a tensão da rede cruzou ou não o valor zero.

No programa a leitura de temperatura é feita de forma constante, com amostras intervaladas por 1 segundo e então enviadas ao monitor serial da interface do Arduino.

O cálculo do *delay* necessário para levar a carga ao valor de temperatura desejado é calculado constantemente, a cada 500 ms, pelo algoritmo da biblioteca PID utilizada. Como a frequência da rede é 60 hertz, o período de meio ciclo é de aproximadamente 8,33 ms, portanto o *delay* poderá variar entre zero e 8333 us.

A Figura 25 demonstra o fluxograma básico que norteia o programa.

Figura 25 – Fluxograma do programa



Fonte: Próprio autor.

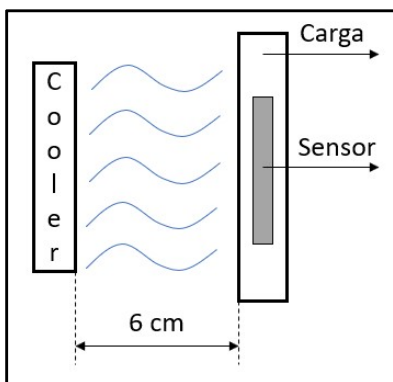
5 APLICAÇÃO E RESULTADOS

5.1 CONSIDERAÇÕES INICIAIS PARA ANÁLISE

A temperatura da carga pode variar entre a temperatura aproximada de 25°C, até a temperatura máxima de 90°C, que é alcançada quando a carga possui tensão máxima de 127 volts. Tendo isto em vista, a temperatura desejada de trabalho, *set point*, foi adotada como 50°C, que é um valor intermediário entre a máxima e a mínima temperatura.

Para se testar o desempenho do controlador, usando um distúrbio externo, foi utilizado um *cooler* de tensão nominal 12 Volts posicionado cerca de 6 cm distante da face lateral do sensor de temperatura como o elemento físico causador do distúrbio, conforme o esquema mostrado na Figura 26.

Figura 26 – Esquemático do distúrbio



Fonte: Próprio autor.

Ao variar a tensão aplicada ao cooler é possível obter níveis diferentes de distúrbio, ou seja, diferentes tensões resultando em velocidades diferentes, conseqüentemente intensidades diferentes de distúrbio, como pode ser visto na Tabela 12.

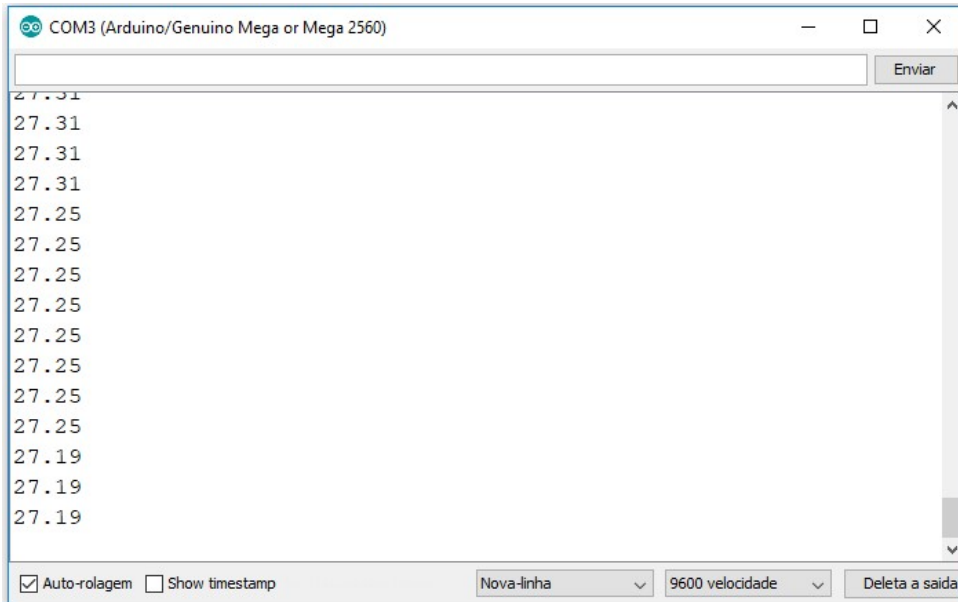
Tabela 12 – Níveis de distúrbio aplicado

Cooler	
Tensão [V]	Intensidade do distúrbio
5	Leve
9	Moderado
12	Forte

Fonte: Próprio autor.

As leituras de temperatura foram realizadas com intervalo fixo de 1 segundo e enviadas ao monitor serial do Arduino em forma de coluna. Posteriormente estes dados foram importados por um programa responsável em traçar os gráficos utilizados neste trabalho.

Figura 27 – Leitura de temperatura no monitor serial do Arduino



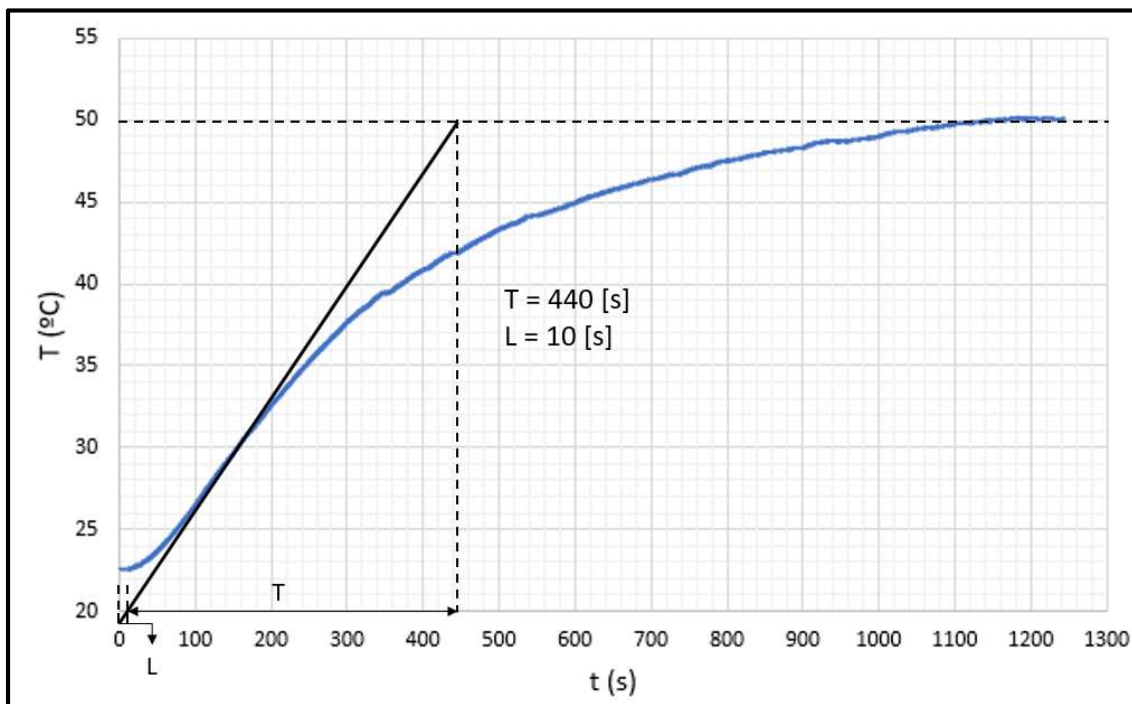
Fonte: Próprio autor.

5.2 RESULTADOS DE ZIEGLER-NICHOLS MALHA ABERTA

5.2.1 Resposta do método Z-N malha aberta

Para a determinação dos parâmetros do controlador PID por este método, primeiramente precisa-se obter a resposta do sistema ao degrau em malha aberta. Foi preciso achar então o *delay* aplicado ao TRIAC que levasse a carga a atingir a temperatura desejada de 50°C, o valor encontrado foi de aproximadamente 4200 microssegundos. Ao aplicar este *delay*, a curva encontrada, já com a reta tangente traçada no ponto de inflexão, é representada na Figura 28.

Figura 28 – Comportamento em malha aberta do sistema para o método de Ziegler-Nichols
Malha Aberta



Fonte: Próprio autor.

Os únicos dois parâmetros necessários para o cálculo de K_p , K_i e K_d , podem ser extraídos graficamente da Figura 28. O termo T possui valor de, aproximadamente, 440 segundos e o termo L , possui valor de 10 segundos. Dispondo destes valores e de acordo com a Tabela 2 é possível calcular os parâmetros de sintonia do PID. O resultado destes cálculos da origem aos dados da Tabela 13, que apresenta os valores finais que buscamos.

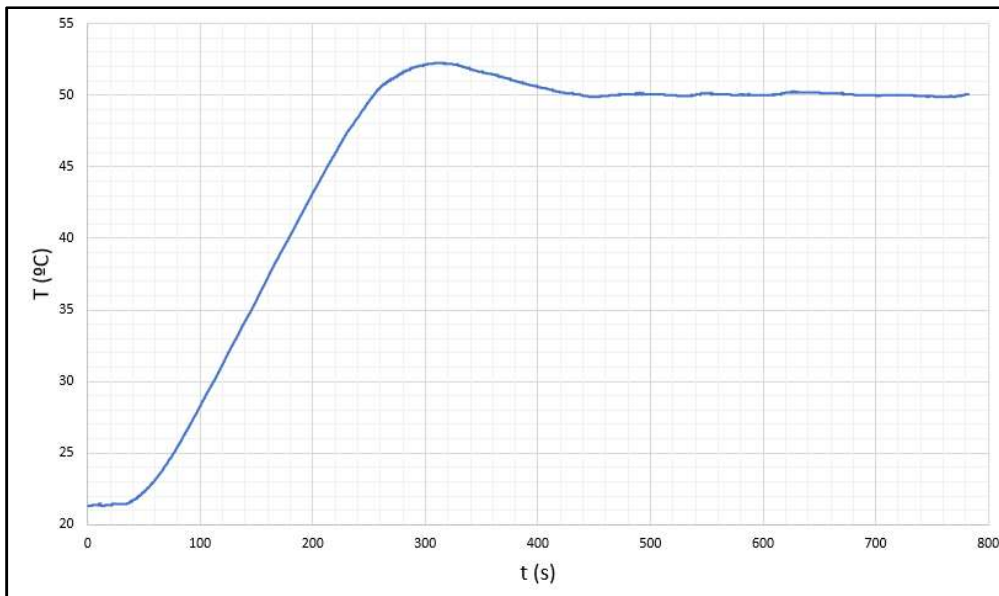
Tabela 13 – Valores calculados de K_p , K_i e K_d para o método de Ziegler-Nichols Malha Aberta

Parâmetro	Valor
K_p	52.68
K_i	2.64
K_d	264

Fonte: Próprio autor.

Agora é possível obter a curva de resposta ao degrau com o controlador PID sintonizado de acordo com a Tabela 8, originando a curva exposta na Figura 29.

Figura 29 – Resposta do sistema para $K_p = 52.68$, $K_i = 2.64$ e $K_d = 264$

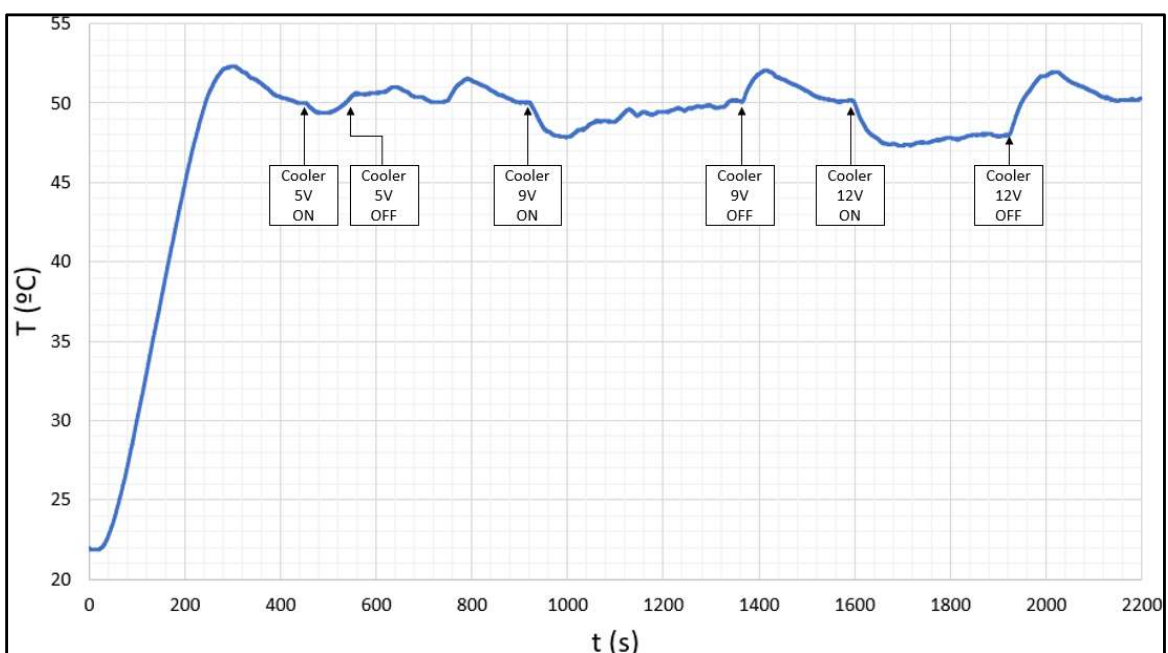


Fonte: Próprio autor.

5.2.2 Resposta ao distúrbio do método Z-N malha aberta

Em um outro experimento foi introduzido o distúrbio externo ao sistema, nas suas três diferentes intensidades e em momentos distintos, de modo a não interferir no comportamento um do outro. A Figura 30 ilustra esta situação.

Figura 30 – Resposta do sistema, com distúrbios, para $K_p = 52.68$, $K_i = 2.64$ e $K_d = 264$



Fonte: Próprio autor.

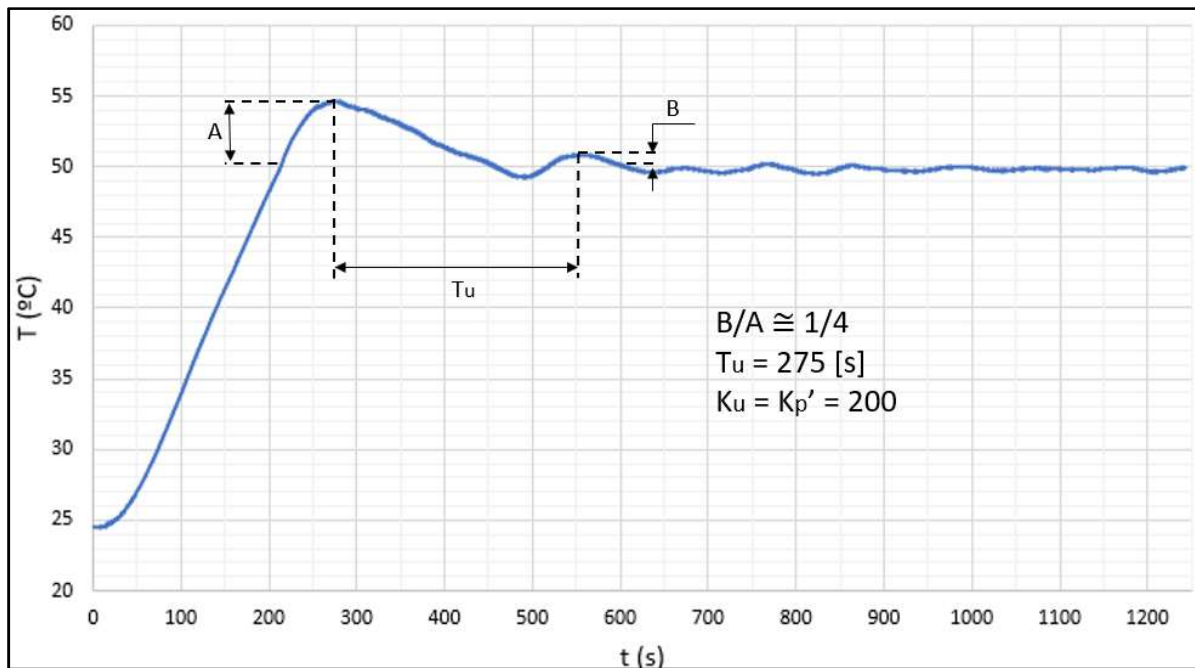
5.3 RESULTADOS DE ZIEGLER-NICHOLS MALHA FECHADA

Através do método da sensibilidade limite não foi possível encontrar um valor de ganho proporcional que levasse o sistema a uma oscilação sustentada, por esta razão foi adotado o método de Ziegler-Nichols em malha fechada com decaimento de $\frac{1}{4}$ de onda.

5.3.1 Resposta do método Z-N malha fechada

Para determinação das constantes do controlador PID usando este método, primeiramente é preciso encontrar o valor de ganho proporcional que leve o sistema a um decaimento, entre os dois primeiros picos, com razão de amortecimento de $\frac{1}{4}$. Após exaustivas tentativas, com o valor de ganho proporcional K_p de 200, observa-se esta situação, que é demonstrada na Figura 31.

Figura 31 – Resposta do sistema para $K_p = 200$, $K_i = 0$ e $K_d = 0$



Fonte: Próprio autor.

O parâmetro principal retirado do gráfico é o período observado entre o primeiro e o segundo pico máximo, com amortecimento de $\frac{1}{4}$, possuindo o valor de aproximadamente 275 segundos. Após esta etapa, é possível calcular os parâmetros do controlador PID através da Tabela 4. Os resultados obtidos destes cálculos são mostrados na Tabela 14.

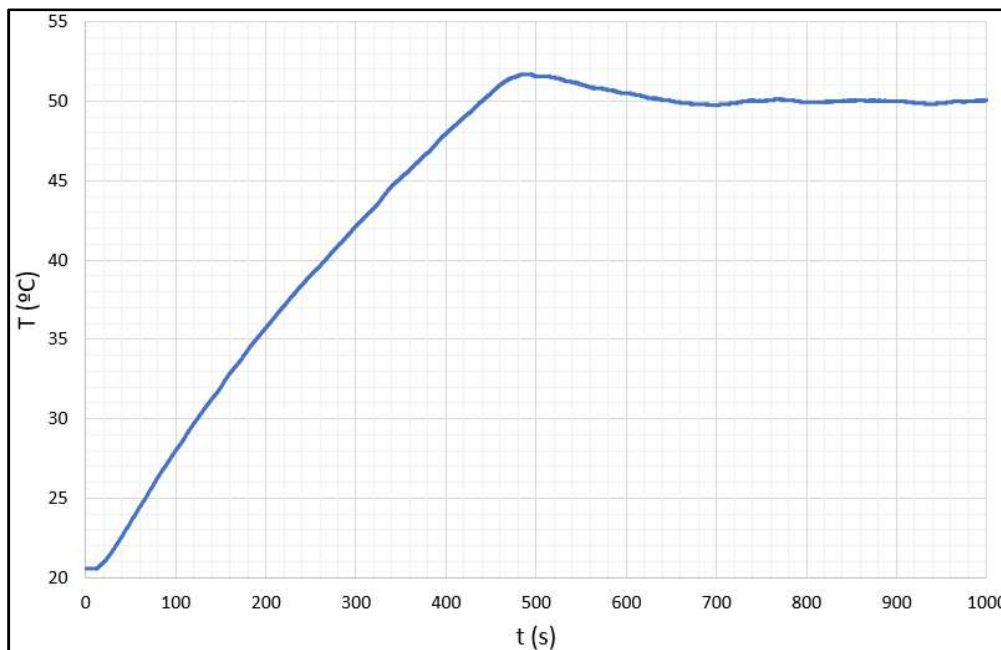
Tabela 14 - Valores calculados de K_p , K_i e K_d para o método de Ziegler-Nichols Malha Fechada com decaimento de $\frac{1}{4}$ de onda

Parâmetro	Valor
K_p	120
K_i	0.873
K_d	4125

Fonte: Próprio autor.

Já em posse dos valores de K_p , K_i e K_d , foram introduzidos ao programa e foi obtida a curva de resposta conforme a Figura 32.

Figura 32 – Resposta do sistema para $K_p = 120$, $K_i = 0.873$ e $K_d = 4125$

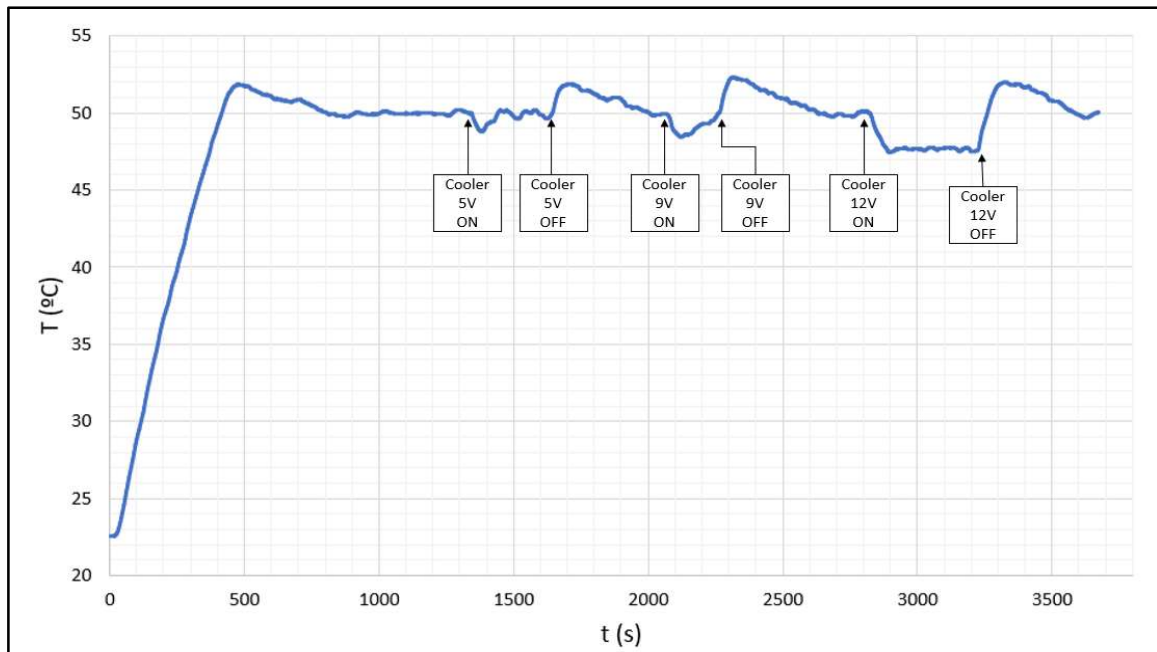


Fonte: Próprio autor.

5.3.2 Resposta ao distúrbio do método Z-N malha fechada

A resposta aos mesmos distúrbios impostos ao método anterior, pode ser vista na Figura 33.

Figura 33 - Resposta do sistema, com distúrbios, para $K_p = 120$, $K_i = 0.873$ e $K_d = 4125$



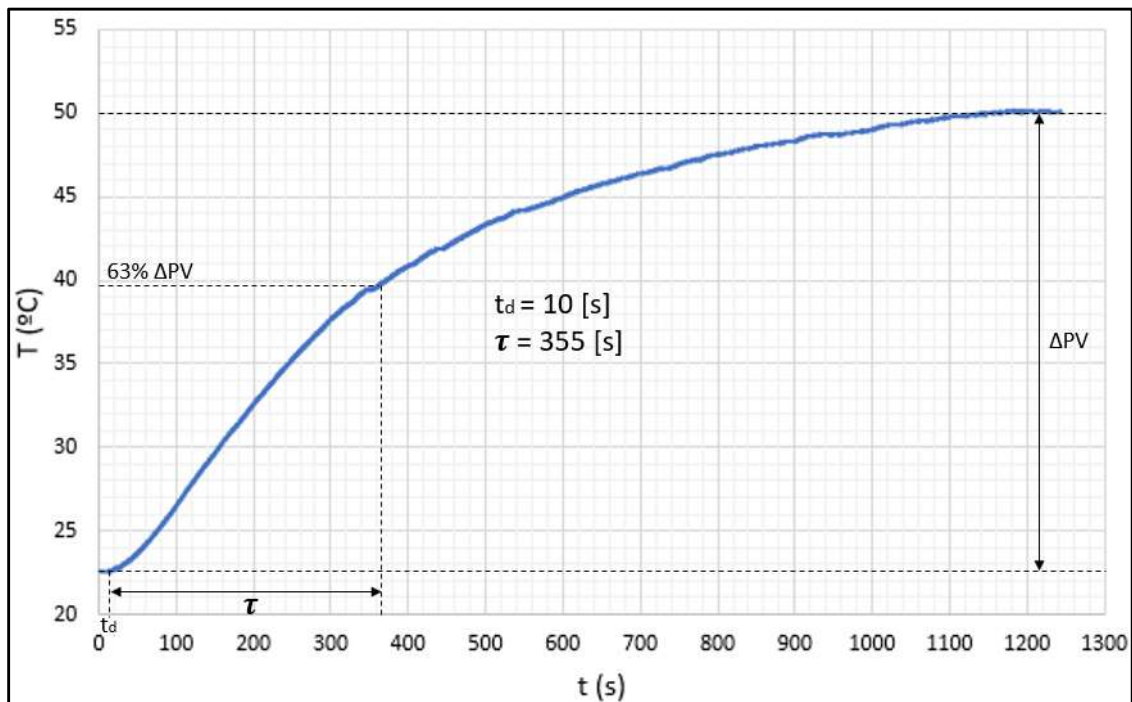
Fonte: Próprio autor.

5.4 RESULTADOS DE COHEN-COON MALHA ABERTA

5.4.1 Resposta do método C-C malha aberta

Neste método também é utilizado a curva de resposta ao degrau em malha aberta do sistema, onde é preciso determinar basicamente o *dead time*, t_d , e a constante de tempo τ . A determinação destes tempos foi feita graficamente conforme a Figura 34.

Figura 34 – Comportamento em malha aberta do sistema para o método de Cohen-Coon Malha Aberta



Fonte: Próprio autor.

Considerando a entrada, ΔCO , com o mesmo valor do *set point* de 50°C , é possível calcular o ganho do processo segundo a equação (9) e posteriormente obter os parâmetros do controlador PID segundo a Tabela 5. O resultado destes cálculos são mostrados na Tabela 15.

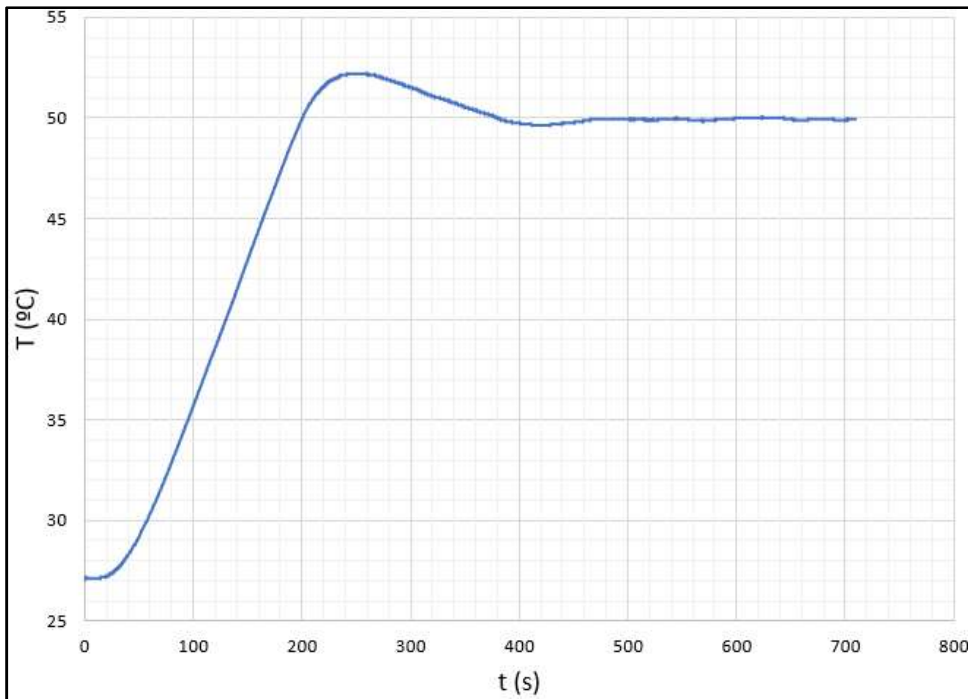
Tabela 15 - Valores calculados de K_p , K_i e K_d para o método de Cohen-Coon Malha Aberta

Parâmetro	Valor
K_p	87.6
K_i	3.55
K_d	322.44

Fonte: Próprio autor

Os valores dos parâmetros do PID são inseridos no programa do Arduino e então obtemos a seguinte resposta do sistema, conforme a Figura 35.

Figura 35 - Resposta do sistema para $K_p = 87.6$, $K_i = 3.55$ e $K_d = 322.44$

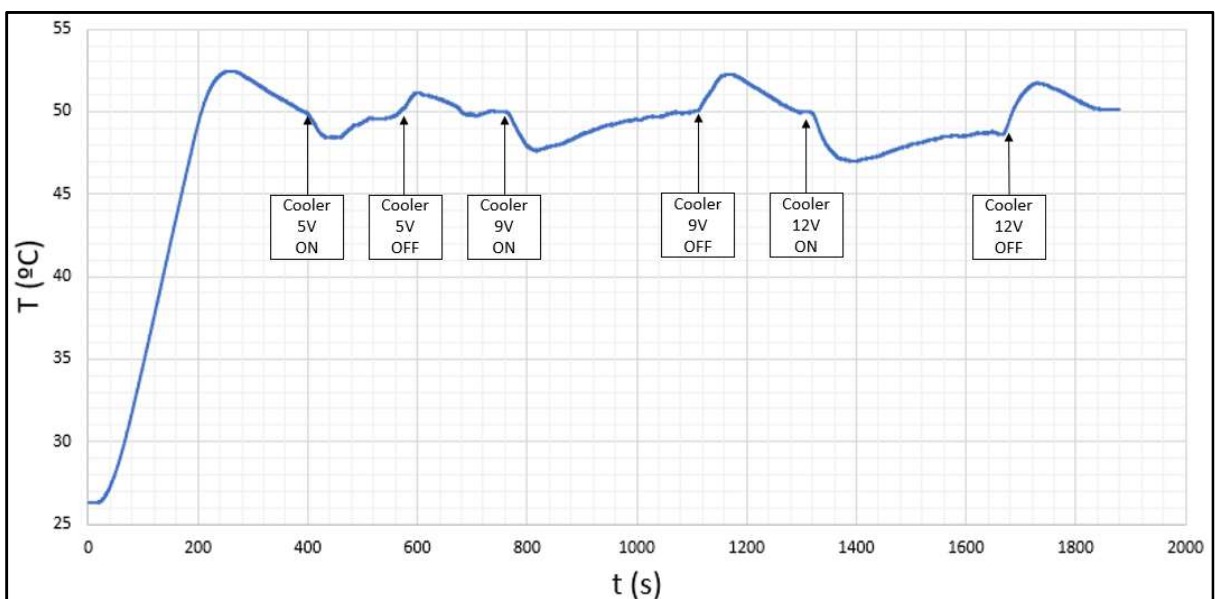


Fonte: Próprio autor

5.4.2 Resposta ao distúrbio do método C-C malha aberta

O comportamento do sistema mediante distúrbio externo pode ser visto na figura 36.

Figura 36 - Resposta do sistema, com distúrbios, para $K_p = 87.6$, $K_i = 3.55$ e $K_d = 322.44$



Fonte: Próprio autor.

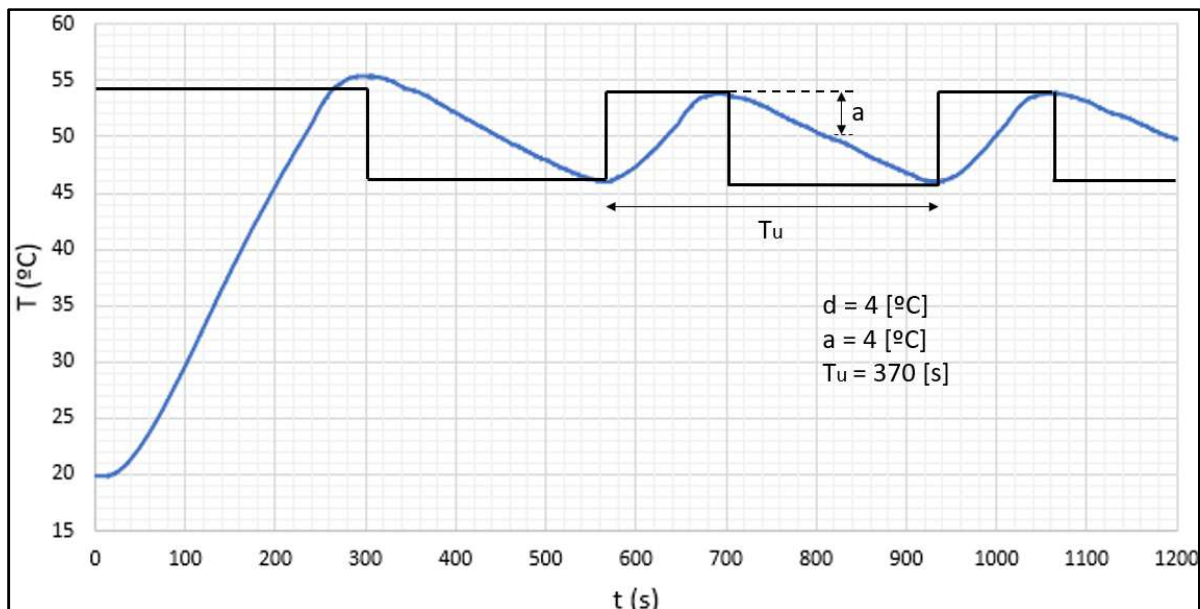
5.5 RESULTADOS DO MÉTODO DO RELÉ

5.5.1 Resposta do método do Relé

Para determinação dos parâmetros do PID pelo método do Relé, precisa-se aplicar um relé no sistema em malha fechada, com amplitude fixa. Deve ser observado na saída uma oscilação do sinal em torno do *set point* de 50°C.

O relé aplicado possui amplitude, d , de 4°C e a oscilação observada possui amplitude, a , de 4°C e período, T_u , de 370 segundos aproximadamente, conforme pode ser visto na Figura 37.

Figura 37 – Resposta do sistema ao relé



Fonte: Próprio autor.

Utilizando a equação (10) e a Tabela 6 é possível calcular os valores de K_p , K_i e K_d , originando assim a Tabela 16.

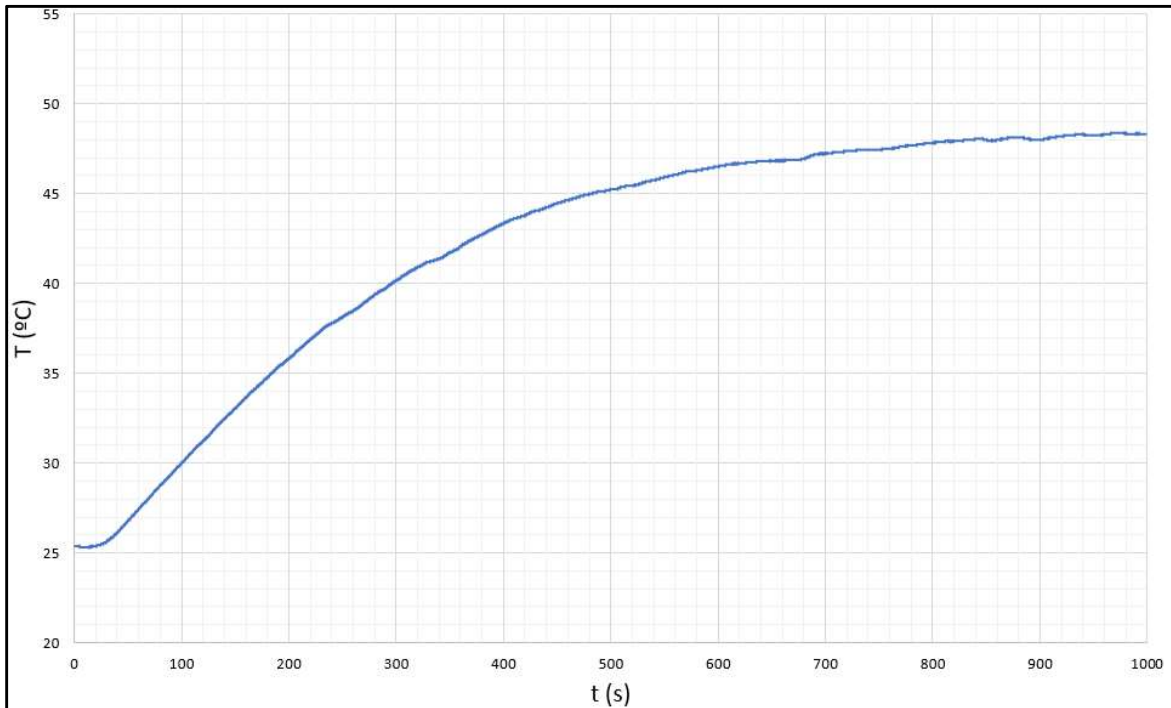
Tabela 16 - Valores calculados de K_p , K_i e K_d para o método do Relé

Parâmetro	Valor
K_p	0.764
K_i	0.0041
K_d	35.335

Fonte: Próprio autor

Configurando as constantes do PID no programa com os valores da Tabela 16, foi obtida a curva de resposta contida na Figura 38.

Figura 38 - Resposta do sistema para $K_p = 0.764$, $K_i = 0.0041$ e $K_d = 35.335$

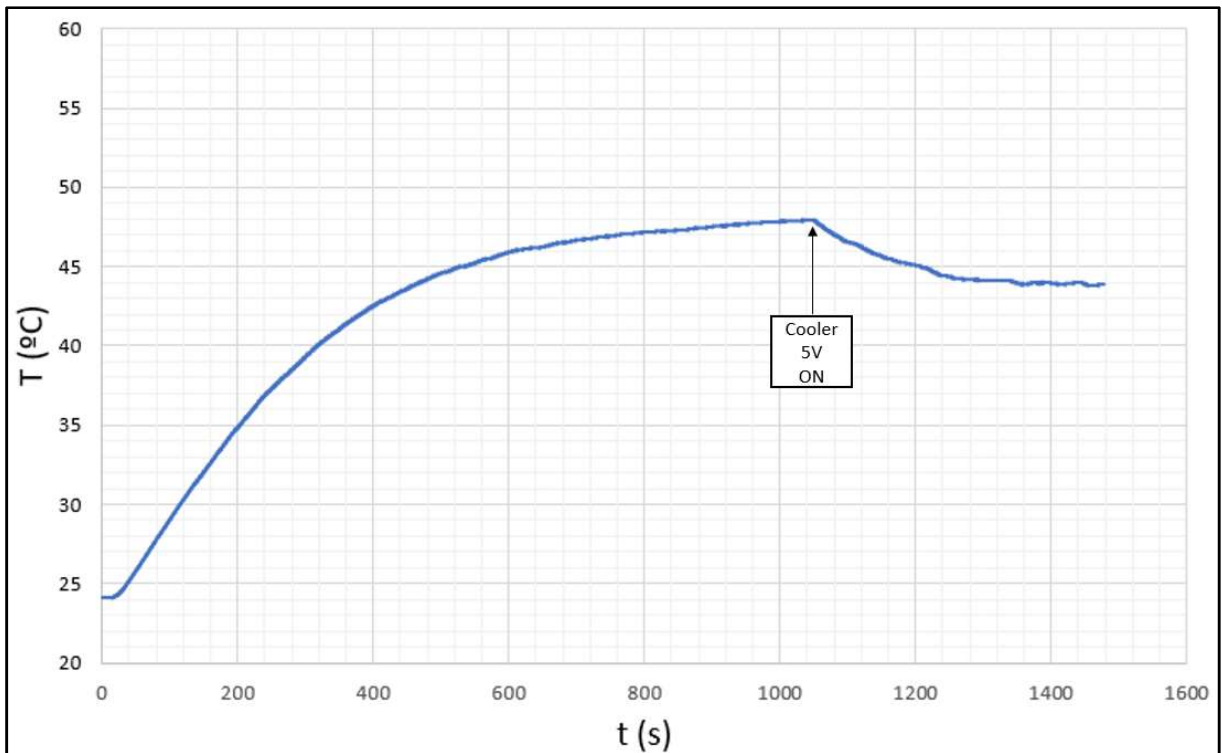


Fonte: Próprio autor

5.5.2 Resposta ao distúrbio do método do Relé

Devido ao baixo ganho proporcional e integral obtidos, a resposta ao distúrbio não foi satisfatória, ou seja, o sistema não conseguiu se recuperar, nem ao menos, ao menor distúrbio aplicado. A temperatura se estabilizou em torno de 44°C, conforme ilustrado na Figura 39.

Figura 39 - Resposta do sistema, com distúrbio, para $K_p = 4.77$, $K_i = 0.025$ e $K_d = 220.61$



Fonte: Próprio autor

5.6 DESEMPENHO

5.6.1 Comparação de resposta

A Tabela 17 resume os ganhos proporcional, integral e derivativo calculados anteriormente para cada método utilizado.

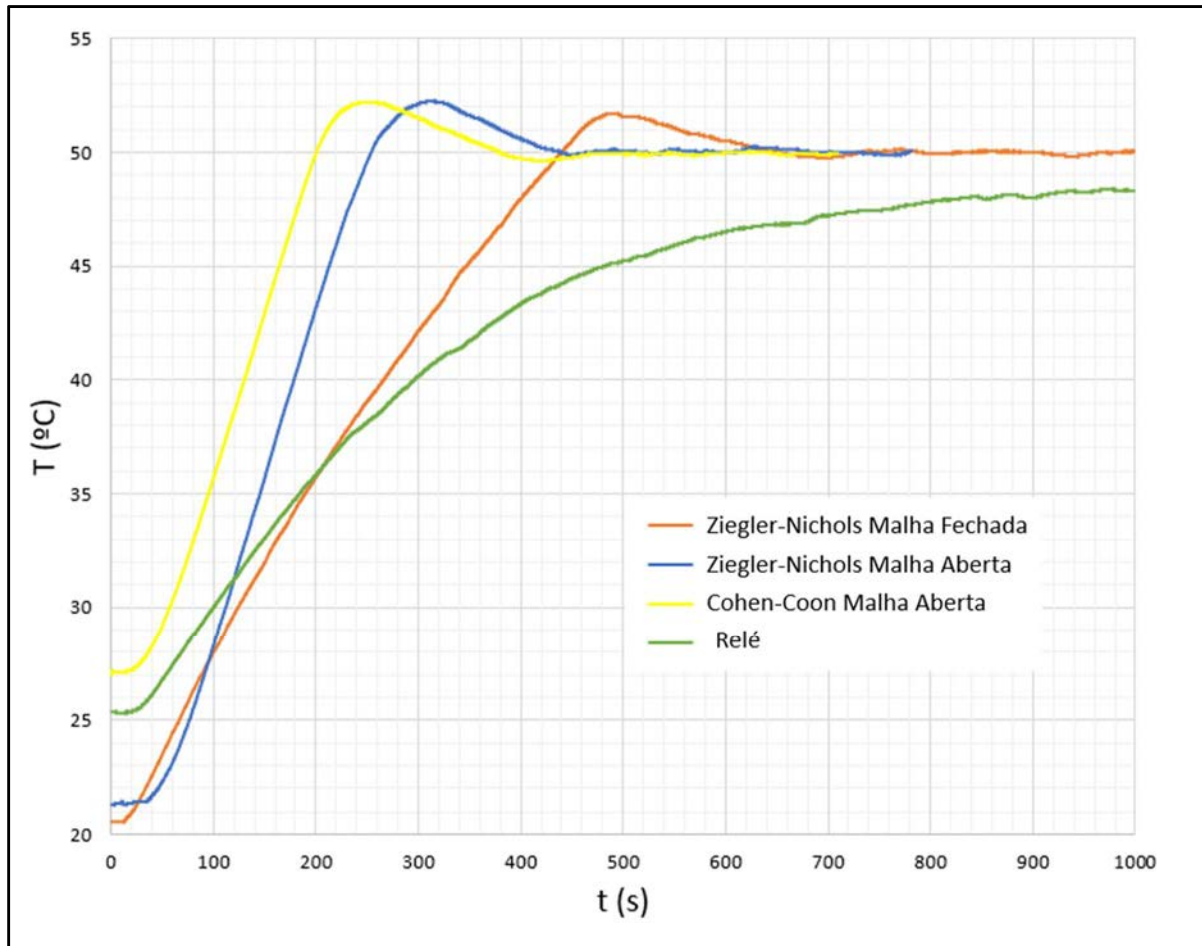
Tabela 17 – Valores de K_p , K_i e K_d para todos os métodos

Controlador (PID)	K_p	K_i	K_d
Ziegler-Nichols Malha Aberta	52.8	2.64	264
Ziegler-Nichols Malha Fechada (Decaimento de ¼)	120	0.873	4125
Cohen-Coon Malha Aberta	87.6	3.55	322.44
Relé Malha Fechada	0.764	0.0041	35.335

Fonte: Próprio autor

A Figura 40 reúne em um mesmo gráfico a resposta normal encontrada para cada método utilizado, já apresentados anteriormente de forma individual.

Figura 40 – Resposta para todos métodos apresentados



Fonte: Próprio autor

Para efeito de comparação de resultados entre os diferentes métodos apresentados anteriormente foram considerados os critérios de desempenho contidos no capítulo 3.5 deste trabalho. Através dos conceitos teóricos destes critérios e em posse das curvas de resposta contidas na Figura 40, é então possível construir a Tabela 18.

Tabela 18 – Critérios de desempenho para todos os métodos

Método	Tempo de Duplicação t_0 (s)	Tempo de Subida t_s (s)	Tempo de Assentamento t_A (s)	Overshoot M ($^{\circ}C$)	Overshoot Percentual $M\%$
Ziegler-Nichols Malha Aberta	255	160	450	2.25	4.5
Ziegler-Nichols Malha Fechada (Decaimento de $\frac{1}{4}$)	440	330	680	1.63	3.26
Cohen-Coon Malha Aberta	200	130	470	2.19	4.38
Relé	-	-	-	0	0

Fonte: Próprio autor

Pode-se observar que os métodos de Ziegler-Nichols Malha Aberta e Cohen-Coon Malha Aberta possuíram valores próximos para os critérios apresentados, tendo apenas uma diferença significativa no tempo de duplicação.

O método de Ziegler-Nichols Malha Fechada obteve valores bem elevados, principalmente no tempo de assentamento, porém, devido ao menor ganho integral teve um menor *overshoot*.

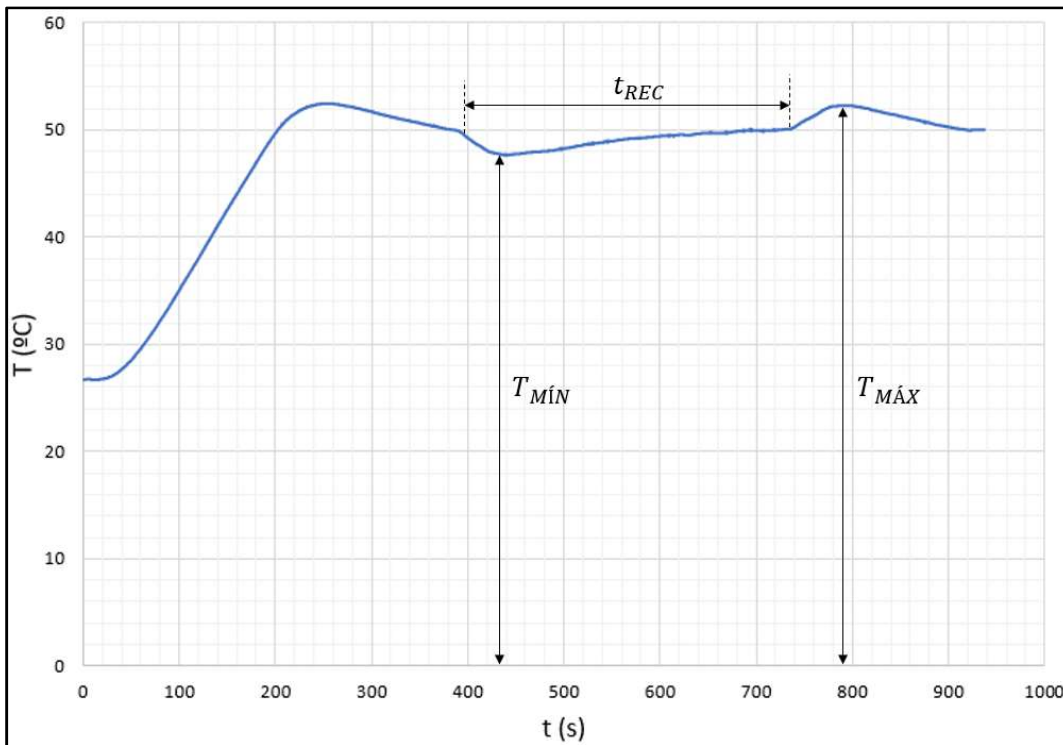
No método do Relé o sistema não atingiu o *set point* desejado de $50^{\circ}C$, por isso não foi calculado o tempo de duplicação e de subida. O tempo de assentamento presente na Tabela 18 para o método do relé, se refere a temperatura máxima na qual o sistema se estabilizou, em torno de $48^{\circ}C$, possuindo então um erro em regime permanente de 4%.

De modo geral o *overshoot* obtido em todos os métodos foi relativamente baixo para processos não críticos, algumas literaturas sugerem que o valor seja menor do que 20% sempre que possível.

5.6.2 Comparação de resposta ao distúrbio

Na comparação da resposta ao distúrbio, alguns critérios de desempenhos adicionais precisam ser apresentados para que a comparação faça sentido. A Figura 41 demonstra os critérios a serem utilizados.

Figura 41 – Critérios de desempenho para resposta ao distúrbio



Fonte: Próprio autor

O **tempo de recuperação**, t_{REC} , é o tempo necessário para que a temperatura retorne, pela primeira vez, ao valor desejado de *set point*.

A **temperatura mínima**, $T_{MÍN}$, é a temperatura mínima obtida durante o distúrbio.

A **temperatura máxima**, $T_{MÁX}$, é a temperatura máxima obtida após o desligamento do distúrbio.

Com a definição destes critérios e através de análise gráfica das Figuras 30, 33, 36 e 39, é possível construir as tabelas a seguir, que demonstram de forma tabelada os valores numéricos obtidos para cada nível de distúrbio.

Ao aplicar o distúrbio leve ao sistema, *cooler* em 5V, é possível construir a Tabela 19.

Tabela 19 – Critérios de desempenho ao distúrbio leve

Método	Temperatura Mínima Durante Distúrbio $T_{MÍN} (°C)$	Tempo de Recuperação do Distúrbio $t_{REC} (s)$	Temperatura Máxima Após Retirada do Distúrbio $T_{MÁX} (°C)$
Ziegler-Nichols Malha Aberta	49.38	105	51.5
Ziegler-Nichols Malha Fechada (Decaimento de ¼)	48.81	110	51.88
Cohen-Coon Malha Aberta	48.44	170	51.06
Relé	44.00	-	-

Fonte: Próprio autor

É possível perceber que, exceto no método do Relé, todos os outros métodos conseguiram se recuperar ao distúrbio leve num tempo maior do que 100 segundos, e não deixando a temperatura cair de forma abrupta no ponto de temperatura mínima durante o distúrbio.

Ao aplicar o distúrbio moderado ao sistema, *cooler* em 9V, é possível construir a Tabela 20.

Tabela 20 – Critérios de desempenho ao distúrbio moderado

Método	Temperatura Mínima Durante Distúrbio $T_{MÍN} (°C)$	Tempo de Recuperação do Distúrbio $t_{REC} (s)$	Temperatura Máxima Após Retirada do Distúrbio $T_{MÁX} (°C)$
Ziegler-Nichols Malha Aberta	47.81	440	52.06
Ziegler-Nichols Malha Fechada (Decaimento de ¼)	48.44	185	52.25
Cohen-Coon Malha Aberta	47.63	310	52.25
Relé	-	-	-

Fonte: Próprio autor

Novamente, todos os métodos, exceto o método do Relé, conseguiram se recuperar ao distúrbio moderado e com valores bem distintos de tempo de recuperação. O destaque é para o método de Ziegler-Nichols Malha Fechada que conseguiu se recuperar muito mais rápido que os demais e apresentou também um menor valor de queda de temperatura durante a presença do distúrbio.

Ao aplicar o distúrbio forte ao sistema, *cooler* em 12V, é possível construir a Tabela 21.

Tabela 21 – Critérios de desempenho ao distúrbio forte

Método	Temperatura Mínima Durante Distúrbio $T_{MÍN}$ ($^{\circ}C$)	Tempo de Recuperação do Distúrbio t_{REC} (s)	Temperatura Máxima Após Retirada do Distúrbio $T_{MÁX}$ ($^{\circ}C$)
Ziegler-Nichols Malha Aberta	47.38	-	51.94
Ziegler-Nichols Malha Fechada (Decaimento de ¼)	47.44	-	52
Cohen-Coon Malha Aberta	47	-	51.69
Relé	-	-	-

Fonte: Próprio autor

No distúrbio forte, nenhum método de sintonia se mostrou eficaz na recuperação do mesmo. Apesar de não conseguirem se recuperar, com exceção do método do relé, a temperatura não apresentou valores distantes do *set point* desejado, mesmo com um distúrbio forte aplicado. A estabilização ocorreu em torno de 47°C e possuindo assim um erro de aproximadamente 6%, o que pode ser considerável aceitável para muitos processos.

6 CONCLUSÕES

Os métodos de Ziegler-Nichols Malha Aberta e Cohen-Coon Malha Aberta apresentaram valores de ganho proporcional, integral e derivativo próximos, conseqüentemente a resposta normal e a resposta ao distúrbio também apresentaram desempenho similar e satisfatório.

No método de Ziegler-Nichols Malha Fechada os valores de ganho proporcional e integral se mostraram substancialmente maiores do que nos demais métodos, isto ocasionou um aumento grande nos tempos de duplicação, subida e assentamento, porém, uma melhor resposta dinâmica ao distúrbio, diminuindo o tempo de recuperação. O valor reduzido de ganho integral possibilitou que o sistema chegasse ao *set point* com um menor valor de overshoot.

Através do método do Relé foram obtidos ganhos proporcional e integral muito abaixo dos demais métodos, por esta razão o sistema não conseguiu atingir o *set point* desejado e sua resposta ao distúrbio não foi satisfatória para nenhum nível de intensidade.

A escolha do método ideal depende, na verdade, do processo ao qual o controlador PID é responsável e de uma análise detalhada por parte dos profissionais da área técnica e de engenharia de controle e automação. Neste trabalho, como o processo não é crítico ou sensível, portanto, não existe restrição ou limitação em nenhum quesito de desempenho. A principal ideia que se deve ter é que um bom método de sintonia é um método simples e que satisfaça as condições operacionais do processo.

Existem dois pontos a ressaltar neste trabalho como possíveis sugestões de trabalhos futuros. O primeiro ponto diz respeito ao fato de não termos conseguido atingir a condição de oscilação sustentada pelo método de Ziegler-Nichols da sensibilidade limite, o que necessitaria mudanças no programa ou no circuito, a fim de atingir esta condição. O segundo ponto a ressaltar é relativo ao ganho proporcional extremamente baixo obtido através do método do Relé, que fez a resposta do sistema não ser satisfatória com ou sem distúrbio externo.

REFERÊNCIAS

- ÅSTRÖM, K. J.; HÄGGLUND, T. **PID controllers: theory, design and tuning**. 2. ed. North Carolina: Instrument Society of America, 1995. 793 p.
- DILLENBURG, M. R. **Indo além do controle PID**. Disponível em: <<https://www.novus.com.br/downloads/Arquivos/indoalemdocontrolepid.pdf>>. Acesso em: 19 ago. 2018.
- GALDINO, J. **Fundamentos de controle**. Disponível em: <http://www3.ifrn.edu.br/~jeangaldino/dokuwiki/lib/exe/fetch.php?media=aula_01.2.pdf>. Acesso em: 20 set. 2018
- LOFLIN, L. **Zero-crossing detectors circuits and applications**. Disponível em: <http://www.bristolwatch.com/ele2/zero_crossing.htm>. Acesso em: 12 set. 2018.
- MENEGHETTI, F. **Sistemas de controle**. Disponível em: <<https://www.dca.ufrn.br/~meneghet/FTP/Controle/scv20071.pdf>>. Acesso em: 12 set. 2018.
- OGATA, K. **Engenharia de controle moderno**. 4. ed. Rio de Janeiro: Pearson, 2003. 788 p.
- PINTO, J. E. **Aplicação prática do método de sintonia de controladores PID utilizando o método do relé com histerese**. 2011. 46 f. Dissertação de Mestrado (Mestrado em Engenharia Elétrica e de Computação) – Universidade Federal do Rio Grande do Norte, Natal, 2014.
- SANCHES, F. M. **Controladores PID**. Disponível em: <<http://fabriciosmf.blogspot.com/2012/06/normal-0-21-false-false-false-pt-br-x.html>>. Acesso em: 02 set. 2018
- SARAIVA, F. A. **Métodos de sintonia em controladores PID**. 2011. 46 f. Trabalho de Graduação (Graduação em Engenharia de Telecomunicações) – Cento Universitário La Salle, Canoas, 2011.
- SMUTS, J. **Cohen-Coon tuning rules**. Disponível em: <<http://blog.opticontrols.com/archives/383>>. Acesso em: 15 ago. 2018
- VANDOREN, V. **Relay method automates PID loop tuning**. Disponível em: <http://d1.amobbs.com/bbs_upload782111/files_36/ourdev_614499E39LAH.pdf>. Acesso em: 11 set. 2018.
- WADE, H. **Trial and error: an organized procedure**. Disponível em: <<https://ww2.isa.org/standardsandpublications/isapublications/intechmagazine/2005/may/control-and-tuning-trial-and-error-an-organized-procedure>>. Acesso em: 04 set. 2018

WILSON, D. I. **Relay-based PID tuning.** Disponível em: <http://www.processcontrolstuff.net/wp-content/uploads/2015/02/relay_autot-2.pdf>. Acesso em: 15 ago. 2018

BIBLIOGRAFIA CONSULTADA

ASSIS, M. **Controle PID de potência em corrente alternada.** Disponível em: <<http://automatobr.blogspot.com/2013/05/controle-de-potencia-em-corrente.html>>. Acesso em: 15 jul. 2018.

BASILIO, J. C.; MATOS, S. R. Design of PI and PID controllers with transient performance specification. **IEEE Transactions on Education**, v. 45, p. 364–370, 2002.

BISHOP, R. H.; DORF, R. C. **Sistemas de controle modernos.** 11. ed. Rio de Janeiro: Addison-Wesley, 1999. 1017 p.

HORNSEY, S. **A review of relay auto-tuning methods for the tuning of PID-type controllers.** Disponível em: <https://warwick.ac.uk/fac/cross_fac/iatl/reinvention/issues/volume5issue2/hornsey/>. Acesso em: 04 set. 2018

LEE, J. H.; CHOI, J. H.; LEE, K. S. **Quantitative PID tuning methods.** Disponível em: <<https://www.cheric.org/files/education/practical/mpc1-29-40.pdf>>. Acesso em: 27 ago. 2018.

MADEIRA, D. **Controlador integral.** Disponível em: <<https://www.embarcados.com.br/controlador-integral/>>. Acesso em: 22 set. 2018.

APÊNDICE A – PROGRAMA COMENTADO

(continua)

```

/*
Programa Arduino Mega 2560
Controle PID para carga em corrente alternada
Felipe F. França
Outubro, 2018
*/

#include <PID_v1.h> // Inclui a biblioteca de controle PID
// Inclui a biblioteca de comunicação 1-Wire
#include <OneWire.h>
// Inclui a biblioteca do sensor DS18B20
#include <DallasTemperature.h>

#define TRIAC 4 // Atribui ao pino 4 o TRIAC
#define kp 4.77 // Definição da variável Kp
#define ki 0.025 // Definição da variável Ki
#define kd 220.61 // Definição da variável Kd
#define Pin 6 // Atribui ao pino 6 o sensor de temperatura

//Declara-se o pino 6 como bus para a comunicação OneWire
OneWire ourWire(Pin);

// Inicia-se 1 instância da biblioteca DallasTemperature
DallasTemperature sensors(&ourWire);

// Declara-se como double SP, PV e MV
double SetPoint, PresentValue, ManipulatedValue;

//Configuração inicial do PID, modo direto
PID pid(&PresentValue, &ManipulatedValue, &SetPoint, kp, ki,
kd, DIRECT);

//Declara-se como volatile a variável power
volatile int power = 100;

void setup() // Rotina de configuração
{
  sensors.begin(); // Inicia-se comunicação com o sensor
  // Inicia-se comunicação serial com velocidade 9600 bps
  Serial.begin(9600);
  // Configuração do pino do TRIAC como saída
  pinMode(TRIAC, OUTPUT);
  // Configuração da interrupção, borda de subida
  attachInterrupt(0, zero_crosss_int, RISING);
  pid.SetOutputLimits(50, 200); // Limitação do output do PID
  pid.SetMode(AUTOMATIC); // Configura-se PID como automático
  SetPoint=50; // Configura-se o setpoint como 50°C
}

```

APÊNDICE A – PROGRAMA COMENTADO

(conclusão)

```
void zero_crosss_int() // Rotina de disparo
{
    // Cálculo do delay de disparo
    int powertime = (32*(256-power));
    // Configura-se o valor do delay com o valor de powertime
    delayMicroseconds(powertime);
    digitalWrite(TRIAC, HIGH); // Disparo do TRIAC
    delayMicroseconds(5); // Delay de 5 microssegundos
    digitalWrite(TRIAC, LOW); // Retira-se o disparo
}

void loop() // Rotina principal
{
    // Requer ao sensor que leia a temperatura
    sensors.requestTemperatures();

    // Variável temp recebe a leitura de temperatura
    float temp = sensors.getTempCByIndex(0);

    // Imprime o valor de temperatura no monitor serial
    Serial.println(temp);

    PresentValue=temp; // PV recebe o valor de temperatura

    pid.Compute(); // Atualização do PID

    // Variável power recebe o valor de MV
    power = ManipulatedValue;
}
```