Departamento de Engenharia Eletrotécnica e de Computadores

Eletrónica de Potência em Acionamentos

2021/2022 – 1º Semestre

Relatório

Implementação do retificador

Professor: Pedro Pereira

Turno: P1

Data: 28/12/2022

Identificação

Nome	Número
Joaquim Lopes	58182
Simão Costa	56810
Matheus Brito	57003
Diogo Delgado	64062

Índice

1- Introdução

Este trabalho tem como objetivos o estudo, dimensionamento e a avaliação experimental de um retificador AC-DC controlado, monofásico em ponte, como representado na figura abaixo

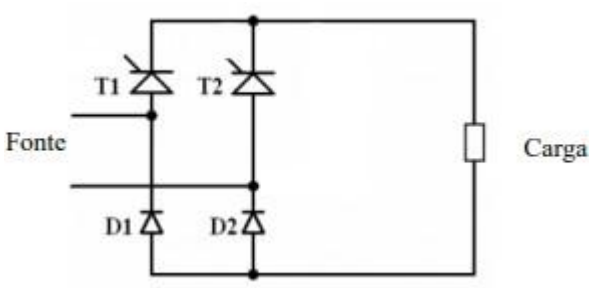


Figura 1 - Ponte retificadora controlada

O retificador será alimentado através de uma fonte alternada sinusoidal de valor eficaz 12V, à frequência de 50 Hz. A carga a considerar deverá ser puramente resistiva de valor 10 Ω.

Neste trabalho devemos desenvolver um controlo do ângulo de disparo de modo a controlar a potência entregue à carga.

2- Desenvolvimento

2.1. - Hardware

O circuito implementado na montagem experimental foi o seguinte:

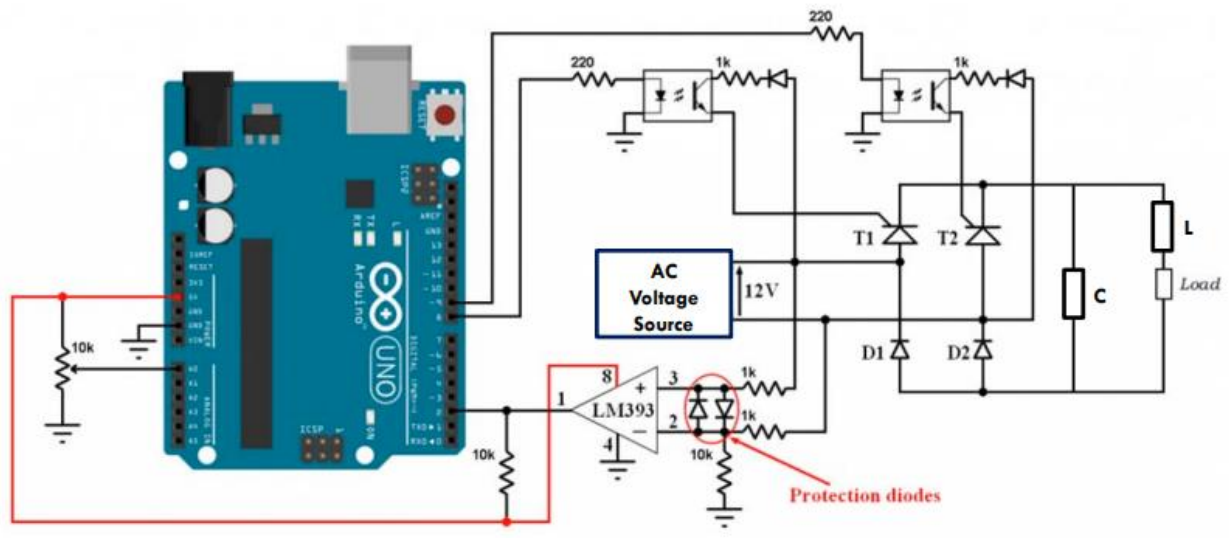
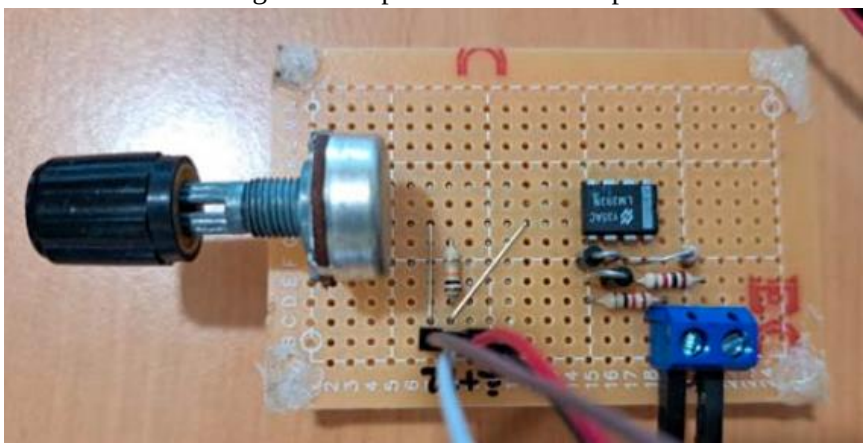


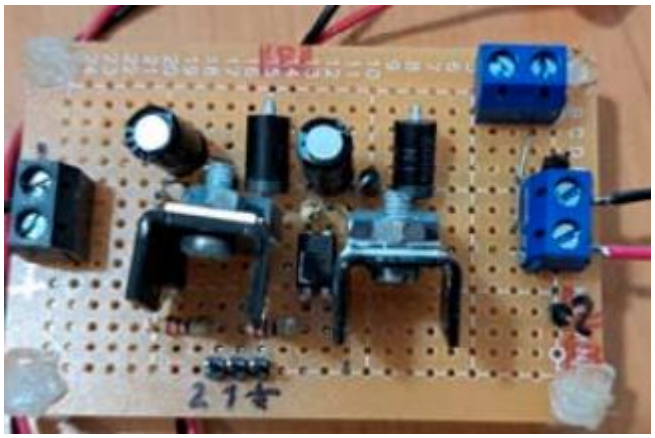
Figura 2 - Montagem experimental

Este é dividido em 4 partes:

-O circuito de controlo do ângulo de disparo através de um potênciometro

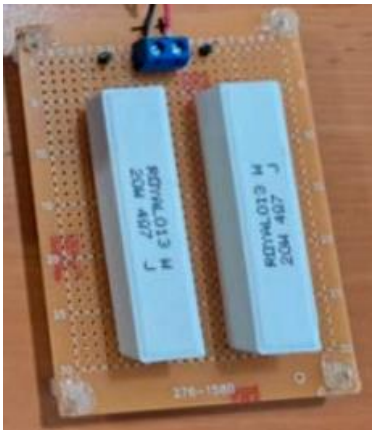


-Ponte Refiticadora



-Circuito de filtragem

-Carga resistiva



2.2. – Software:

Foi preciso desenvolver um código que, tendo em conta a posição do potenciómetro calculava o ângulo correspondente a essa posição, e desse ângulo calcula o tempo a esperar antes de disparar o tirístor correspondente.

Para se realizar esta correspondência entre o valor lido do potenciómetro, realizou-se uma relação linear em que, para quando se lia do potenciómetro o valor máximo “1023”, temos o ângulo de disparo α = 180°, e para o valor mínimo “0”, temos α = 0°, todos os outros valores são obtidos por relação linear.

```
ADCValue = analogRead(pot);
alpha = ADCValue * 180;
alpha = alpha/1023;
VirtualDelay = (alpha * 9500) / 180;
RealDelay = VirtualDelay;

delayMicroseconds(RealDelay);
digitalWrite(scr1_gate,HIGH);
ZC = 0;
```

Figura 3 - Cálculo do delay

O delay é propositadamente maximizado a 9,5ms, que é ligeiramente inferior aos 10ms de meia alternância a 50Hz, esta distorção garante que o controlo não interfira com a próxima alternância da fonte.

A deteção do zero e a escolha do SCR a disparar é feita por uma função que é ativada sempre que deteta mudança no sinal à entrada do pino 2.

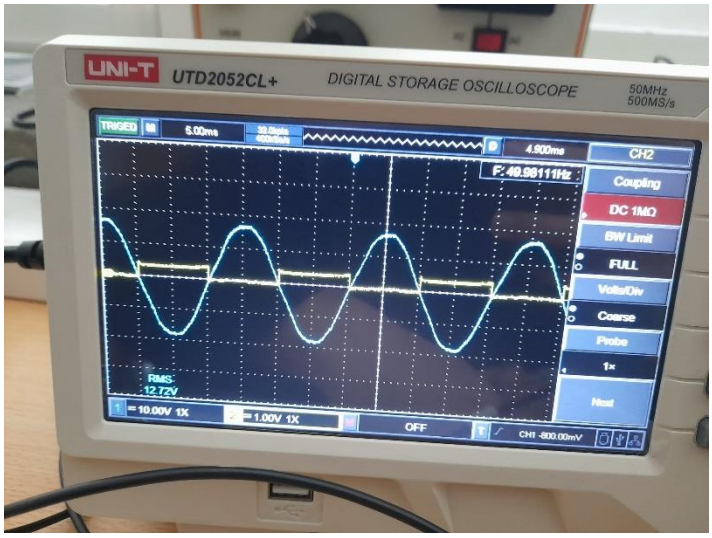


Figura 4 - Sinal da fonte(azul) e sinal no pino 2 (amarelo)

No pino 2 entra a saída de um comparador que fica no nível lógico “1” durante a alternância positiva, e na negativa põe o nível lógico “0”.

```
void ZC_detect() {
  digitalWrite(scr2_gate,LOW);
  digitalWrite(scr1_gate,LOW);
  if(digitalRead(2))
    ZC = 1;
  else
    ZC = 2;
}
```

Figura 5 - Função que lida com a passagem por zero

Esta função é ativada sempre que a saída do comparador muda o seu valor, (sinal amarelo figura 4), o que corresponde a uma passagem por zero da tensão de alimentação.
Em primeiro lugar acaba todos os acionamentos dos tirístores, em segundo, conforme o pino 2 tem na sua entrada o nível lógico 1 ou 0, escolhe qual dos tirístores acionar.

```
void loop() {
  if(ZC == 1){
    ADCValue = analogRead(pot);
    alpha = ADCValue * 180;
    alpha = alpha/1023;
    VirtualDelay = (alpha * 9500) / 180;
    RealDelay = VirtualDelay;

    delayMicroseconds(RealDelay);
    digitalWrite(scr1_gate,HIGH);
    ZC = 0;
  }

  if(ZC == 2){

    ADCValue = analogRead(pot);
    alpha = ADCValue * 180;
    alpha = alpha/1023;
    VirtualDelay = (alpha * 9500) / 180;
    RealDelay = VirtualDelay;

    delayMicroseconds(RealDelay);
    delayMicroseconds(alpha);
    digitalWrite(scr2_gate,HIGH);

    ZC = 0;
  }


}
```

Figura 6 - Função loop

Na função *loop* juntam-se os anteriores passos de determinação do α e da escolha do SCR a disparar.

3- Experimental

Os resultados obtidos da experimentação foram os seguintes:

Tensão à entrada da ponte (azul):

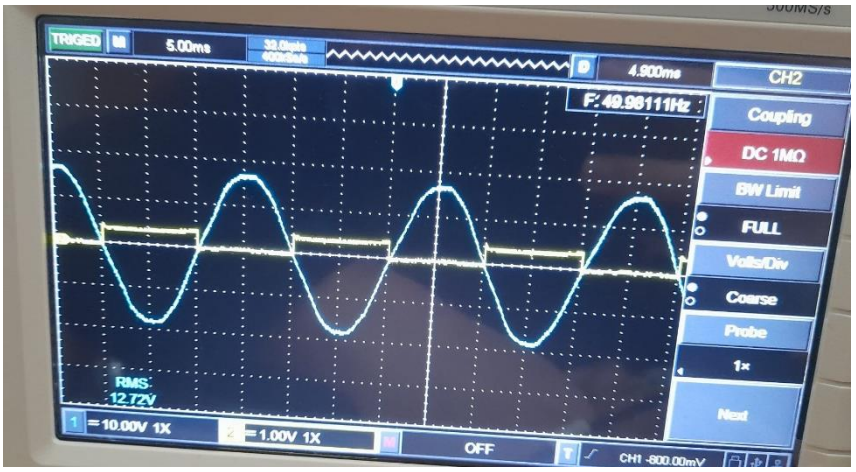


Figura 7 - Fonte 12V e saída comparador

Tensão na carga:

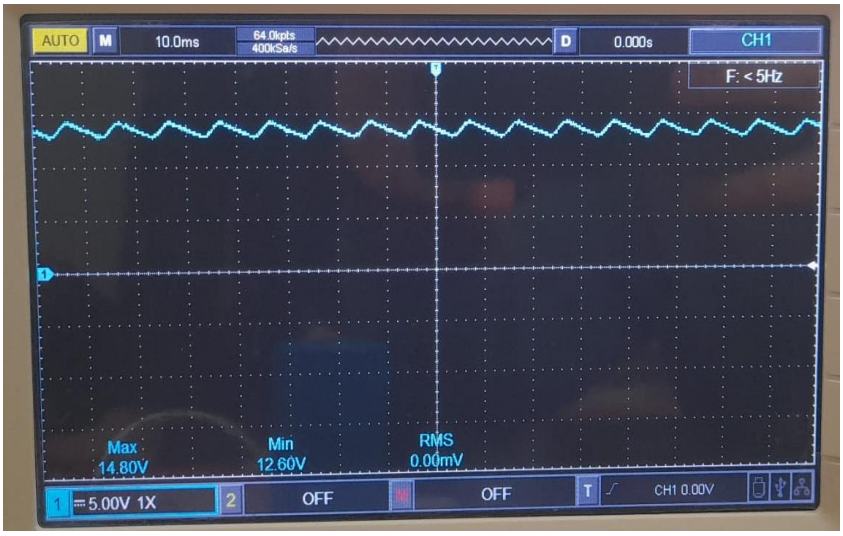


Figura 8 - Tensão na carga para alpha=0

Figura 10 - Tensão na carga para alpha intermédio

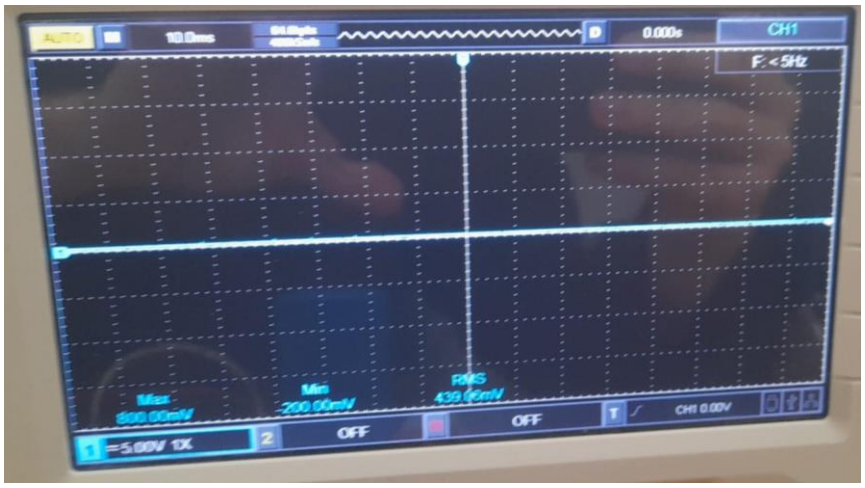


Figura 9 - Tensão na carga para alpha ~180

4- Conclusões

Podemos assim concluir que quanto maior o ângulo de disparo dos tiristores menor será o valor médio de tensão entregue à carga, e nas gamas intermédias de α temos o ripple de tensão maior, e podemos assim com o ângulo de disparo controlar a potência entregue a carga, até que para um ângulo de disparo máximo de 180º a potência entregue a carga é nula, e para o disparo a 0° é entregue a carga toda a potência disponível. Podendo assim este sistema ser usado no controlo de fornos ou aquecedores, embora um simples par de tirístores em antiparalelo seja mais adequado para esta situação
Sendo possível assim variar o valor da tensão na carga, variando o ângulo de disparo dos tirístores, confirmando-se assim o resultado esperado.

5- Anexos

Código Arduino:

```
// Controlled bridge rectifier with Arduino
// EPA 08/12/2022
//*******************************************************
*****************************

//Definir portas Arduino
#define scr1_gate   8 //disparo tiristor 1 no pino
digital 8
#define scr2_gate   10 //disparo tiristor 2 no pino
digital 10
#define pot        A0 //potenciometro pino analogico A0
#define comp        2 //comparador no pino digital 2

//*******************************************************
*****************************
//definição de variáveis
int ZC=0;
float ADCValue;
float alpha;
float VirtualDelay;
int RealDelay;


//*******************************************************
*****************************
//rotina SETUP
void setup(void) {
  pinMode(scr1_gate, OUTPUT); //define pino tiristor 1
como saida
  digitalWrite(scr1_gate, LOW); //coloca pino tiristor 1
a nivel baixo
  pinMode(scr2_gate, OUTPUT); //define pino tiristor 2
como saida
  digitalWrite(scr2_gate, LOW); //coloca pino tiristor 2
```

```
a nivel baixo
  attachInterrupt(digitalPinToInterrupt(comp),
ZC_detect, CHANGE);        // activa interrupt externo
para a função de deteccao de cruzamento por zero

  pinMode(13, OUTPUT); //led da placa, se for necessário
utilizar
  digitalWrite(13, LOW);

  Serial.begin(9600);
}
//*********************************************************
*****************************
//função de detecção cruzamento por zero
void ZC_detect() {
  digitalWrite(scr2_gate,LOW);
  digitalWrite(scr1_gate,LOW);
  if(digitalRead(2))
    ZC = 1;
  else
    ZC = 2;
}

//*********************************************************
*****************************
//rotina loop
void loop() {
  if(ZC == 1){
    ADCValue = analogRead(pot);
    alpha = ADCValue * 180;
    alpha = alpha/1023;
    VirtualDelay = (alpha * 9500) / 180;
    RealDelay = VirtualDelay;

    delayMicroseconds(RealDelay);
    digitalWrite(scr1_gate,HIGH);
    ZC = 0;
  }

  if(ZC == 2){

    ADCValue = analogRead(pot);
    alpha = ADCValue * 180;
    alpha = alpha/1023;
    VirtualDelay = (alpha * 9500) / 180;
    RealDelay = VirtualDelay;

    delayMicroseconds(RealDelay);
    delayMicroseconds(alpha);
    digitalWrite(scr2_gate,HIGH);

    ZC = 0;
```

```
  }



}


/*****************************************************
****************************
 * INFO
 * delayMicroseconds(alpha) - FAZ UM ATRASO DE alpha
MICROSEGUNDOS
 * delay(alpha) - FAZ UM ATRASO DE alpha MILISEGUNDOS
 * analogRead(A0) - Le da porta analógica 10bits entao
lê de 0 a 1024
 * digitalWrite(D1, Estado) - escreve na porta digital
D1 o estado lógico: HIGH ou LOW
 * digitalRead(D1) - Lê o pino digital e retorna HIGH ou
LOW
 * byte variavel = 0; variável tipo byte
 * uint16_t variavel = 0; variavel tipo int
 */
```