

QUARTO RELATÓRIO DE ATIVIDADES PROJETO

Adailton Braga Júnior (201421170), André Luis de Souza Freitas (201511222),
e Beatriz Cristina Reis Cordeiro (201421168)

UNIVERSIDADE FEDERAL DE RONDÔNIA

Projeto - Inversor de tensão monofásico de baixa potência.

Orientador: Ciro José Egoavil Montero.

Curso: Engenharia Elétrica.

Disciplina: Sistema Microprocessados

Período de Atividades: Primeira quinzena de Outubro.

I. METAS DAS ATIVIDADES

Conforme o cronograma apresentado no pré-projeto e nas atividades futuras descritas no terceiro relatório de projeto, as metas deste período de atividades foram:

- **Estudos e projetos de filtros:** nesta etapa, os discentes farão estudos para execução de projetos de filtros.
- **Programação:** nesta etapa, os discentes irão desenvolver a programação do microcontrolador.
- **Montagem e teste de módulos e sensores.** serão feitas as montagens dos módulos necessários para a realização do projeto.

II. METAS ALCANÇADAS E RESULTADOS

Durante a primeira quinzena do mês de Outubro, foram desenvolvidas algumas das atividades propostas. Para alcançar a primeira meta, os discentes buscaram nos diversos meios de acesso à informação assuntos relacionados a filtros para inversores de tensão. Na meta sobre programação, foi desenvolvido um código para criar uma modulação SPWM. A meta de construção de módulos foi alcançada em parte, pois só foi possível realizar a construção da placa para o microcontrolador. A seguir será exposto os resultados alcançados.

• ESTUDOS E PROJETOS DE FILTROS

A topologia de inversor escolhida pelo grupo é a em ponte completa, utilizando quatro chaves semicondutoras de potência (MOSFETS). Para o controle dessas chaves, foi escolhido implementar a modulação SPWM unipolar.

Como mostrado no *Primeiro Relatório de Projeto*, a modulação SPWM unipolar gera múltiplos harmônicos distantes da fundamental quanto maior for

a relação entre a frequência do sinal da portadora e do sinal modulante, enquanto que os múltiplos harmônicos iniciais são cancelados. Dessa forma, é necessário filtrar os componentes harmônicos de maior frequência através de filtros tipo *passa-baixa*.

Os filtros passa-baixa deixam passar sinais com frequências baixas até uma frequência de corte ω_c e rejeita frequências altas. A figura 1 mostra a resposta em frequência desse tipo de filtro considerando o caso ideal.

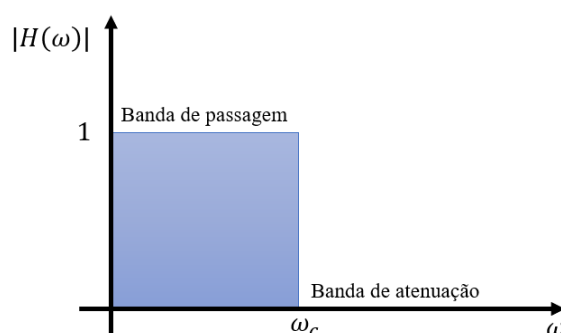


Figura 1: Resposta ideal em frequência de um filtro passa-baixa. Fonte: autor.

Nos filtros reais, o corte não ocorre de forma abrupta como nos filtros ideais. Para os filtros passa-baixa, o comportamento assemelha-se ao mostrado na figura 2.

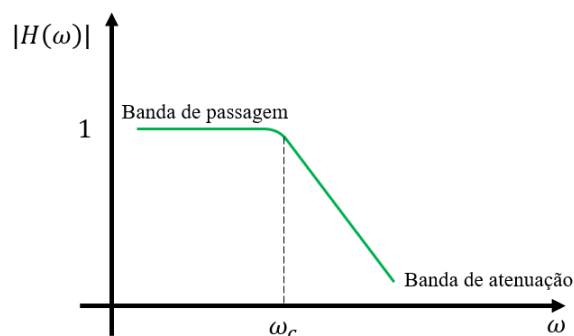


Figura 2: Resposta real em frequência de um filtro passa-baixa genérico. Fonte: autor.

Existem várias derivações de filtros, dentre eles está os filtros passivos. Um circuito é um filtro passivo

se for formado apenas pelos elementos passivos: resistor, indutor e capacitor [1]. Quanto maior for a ordem do filtro, mais preciso será o controle que ele oferece - ficando cada vez mais próximo do comportamento ideal -, como também seu preço e complexidade [2]. A figura 3 mostra um filtro passa-baixa tipo *Butterworth* com várias ordens n .

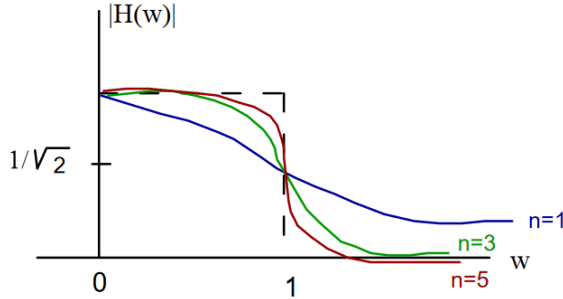


Figura 3: Resposta em frequência de um filtro passa-baixa tipo Butterworth para várias ordens n . Fonte: [3]

A figura 4 mostra um filtro passa-baixa de primeira ordem do tipo RC. Neste tipo de topologia, em altas frequências o capacitor se comporta de tal maneira que curto-circuita estes sinais, por conta de sua impedância da forma $X_C = \frac{1}{sC}$ (domínio s).

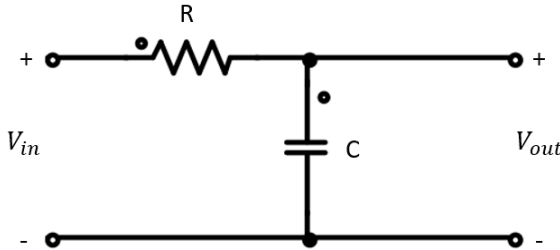


Figura 4: Filtro passa-baixa de primeira ordem tipo passivo RC. Fonte: autor.

A função de transferência $H(s)$ do filtro da figura 4 é dada por:

$$H(s) = \frac{1}{1 + RCs} = \frac{1}{1 + \frac{s}{\omega_c}} \quad (1)$$

Onde a frequência de corte é dada por $\omega_c = \frac{1}{RC}$.

A figura 5 mostra um filtro passa-baixa de primeira ordem do tipo RL. Neste tipo de topologia, o indutor apresenta uma alta impedância para as altas frequências ($X_L = sL$), deixando-as atenuadas. De forma semelhante ao desenvolvimento anterior no domínio s , a frequência de corte desse tipo de filtro é dada por $\omega_c = \frac{R}{L}$, como mostrado em 3.

$$H(s) = \frac{1}{1 + \frac{L}{R}s} = \frac{1}{1 + \frac{s}{\omega_c}} \quad (2)$$

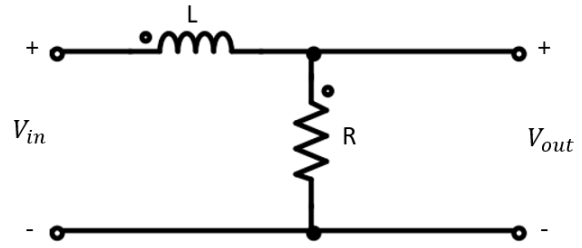


Figura 5: Filtro passa-baixa de primeira ordem tipo passivo RL. Fonte: autor.

A figura 6 mostra um filtro passa-baixa de segunda ordem do tipo LC. O indutor faz oposição a frequências altas, enquanto que o capacitor curto-circuita as frequências altas que passaram pelo indutor. Esse tipo de configuração, com dois ou mais estágios, é bastante utilizada em inversores em forma de onda senoidal pura.

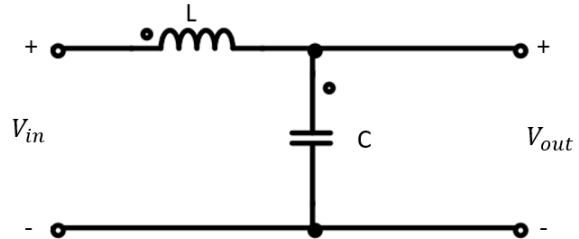


Figura 6: Filtro passa-baixa de segunda ordem tipo passivo LC. Fonte: autor.

Para o filtro da figura 6, a função de transferência $H(s)$ é dada por:

$$H(s) = \frac{\frac{1}{LC}}{s^2 + \frac{1}{LC}} = \frac{\omega_c^2}{s^2 + \omega_c^2} \quad (3)$$

Assim, a frequência de corte é igual a $\omega_c = \frac{1}{\sqrt{LC}}$.

Na construção de filtros, geralmente utiliza-se um capacitor comercial e o indutor deve ser produzido. Quanto maior for a frequência de corte que se deseja filtrar, menor será o indutor e a indutância necessária.

• PROGRAMAÇÃO

A programação desenvolvida para gerar um SPWM unipolar no microcontrolador PIC16F877A é mostrada no Apêndice A. O resultado de testes em laboratório é representado na figura 7.

• MONTAGEM E TESTE DE MÓDULOS E SENSORES

Os módulos montados durante o período de atividades foram:

Módulo microcontrolador PIC16F877A:

Representado na figura 8.



Figura 7: Teste da programação da modulação SPWM unipolar no PIC16F877A

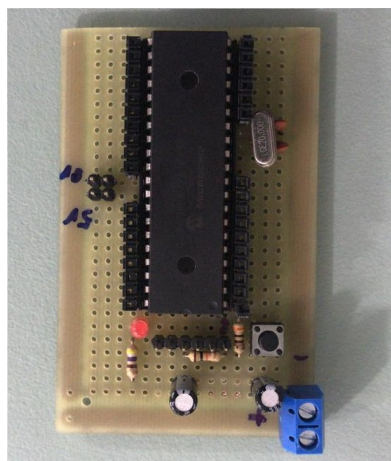


Figura 8: Módulo do microcontrolador PIC16F877A

Módulo das chaves semicondutoras - ponte completa:

Chave semicondutora utilizada: MOSFET - IRFZ44N

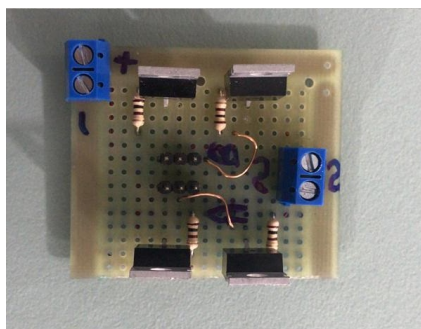


Figura 9: Módulo chaves semicondutoras ponte completa

Módulo dos drivers das chaves semicondutoras:

Driver utilizado: IR2102.

O circuito foi montado conforme orientações do fabricante.

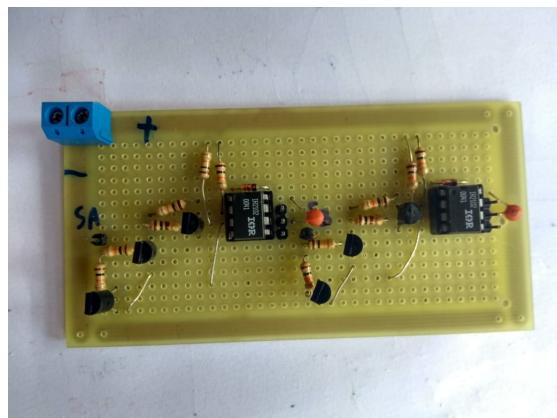


Figura 10: Módulo drivers.

III. TAREFAS DESTINADAS AOS INTEGRANTES

Conforme descrito no cronograma de atividades do pré-projeto, as tarefas destinadas aos integrantes estão expostas a seguir:

Adailton Braga Júnior: 1) Programação. 2) Montagem e testes de módulos e sensores.

André L. de S. Freitas: 1) Montagem e testes de módulos e sensores.

Beatriz C. R. Cordeiro: 1) Estudos e projetos de filtros. 2) Montagem e testes de módulos e sensores.

IV. PRÓXIMAS METAS

Segundo o cronograma de atividades, considerando as atividades já concluídas, as próximas metas serão:

- **Simulação:** nesta etapa, os discentes irão simular o circuito elétrico do projeto através de softwares de simulação.
- **Programação:** nesta etapa, os discentes irão desenvolver a programação do microcontrolador.
- **Montagem e teste de módulos e sensores.** serão feitas as montagens dos módulos necessários para a realização do projeto.

REFERÊNCIAS

- [1] M. N. O. Sadiku and C. K. Alexander, *Fundamentos de Circuitos Elétricos*, 3rd ed. McGraw Hill, 2013.
- [2] M. J. C. BONFIM, *Notas de aula - Circuitos Lineares - Capítulo 5*. UFPR, 2010. [Online]. Available: <http://www.eletrica.ufpr.br/marlio/te054/capitulo5.pdf>
- [3] *Notas de aula - Filtros Lineares Passivos*. Engenharia Elétrica - UFPE. [Online]. Available: <http://www2.ee.ufpe.br/codec/aula%20pcom%20filtros.pdf>



Adailton B. Júnior nasceu em Porto Velho, Rondônia, em 1996. Ele é graduando em Engenharia Elétrica pela Universidade Federal de Rondônia desde julho de 2014.

E-mail: adailtonjn68@gmail.com.

Currículo Lattes disponível em: <http://lattes.cnpq.br/5914795787360469>.



André L. de S. Freitas nasceu em Porto Velho, Rondônia, em 1993. Ele é graduando em Engenharia Elétrica pela Universidade Federal de Rondônia desde fevereiro de 2015.

E-mail: lsf-andre@hotmail.com.

Currículo Lattes disponível em: <http://lattes.cnpq.br/6935331423148390>.



Beatriz C. R. Cordeiro nasceu em Porto Velho, Rondônia, em 1996. Ela é graduanda em Engenharia Elétrica pela Universidade Federal de Rondônia desde julho de 2014.

E-mail: beatriz.cristina.240@gmail.com.

Currículo Lattes disponível em: <http://lattes.cnpq.br/6109459940410639>.

APÊNDICE

APÊNDICE A - SPWM UNIPOLAR

```

// CONFIG
#pragma config FOSC = HS
#pragma config WDTE = OFF
#pragma config PWRTE = ON
#pragma config BOREN = OFF
#pragma config LVP = OFF
#pragma config CPD = OFF
#pragma config WRT = OFF
#pragma config CP = OFF

#include <xc.h>

#define _XTAL_FREQ 20000000

#define TMR0_init_value 10

// PWM duty cycles for each leg of the inverter (20 values)
//static const char pwm1[] = {125,164,198,226,244,250,244,226,198,164,125,86,52,24
//                               ,6,0,6,24,52,86};
//static const char pwm2[] = {125,86,52,24,6,0,6,24,52,86,125,164,198,226,244,250,
//                               244,226,198,164};
// (40 values)
//static const char pwm1[] = {125,145,164,182,198,213,226,236,244,248,250,248,244,
//                               236,226,213,198,182,164,145,125,105,
//                               86,68,52,37,24,14,6,2,0,2,6,14,24,37,
//                               52,68,86,105};
//static const char pwm2[] = {125,105,86,68,52,37,24,14,6,2,0,2,6,14,24,37,52,68,
//                               86,105,125,145,164,182,198,213,226
//                               ,236,244,248,250,248,244,236,226,213
//                               ,198,182,164,145};

// (80 values)
const char pwm1[] = {125,135,145,154,164,173,182,190,198,
                    206,213,220,226,232,236,240,244,247,248,
                    250,250,250,248,247,244,240,236,232,226,
                    220,213,206,198,190,182,173,164,154,145,
                    135,125,115,105,96,86,77,68,60,52,44,37,
                    30,24,18,14,10,6,3,2,0,0,0,2,3,6,10,14,18
                    ,24,30,37,44,52,60,68,77,86,96,105,115};
const char pwm2[] = {125,115,105,96,86,77,68,60,52,44,37,30,
                    24,18,14,10,6,3,2,0,0,0,2,3,6,10,14,18,24,
                    30,37,44,52,60,68,77,86,96,105,115,125,135,
                    ,154,164,173,182,190,198,206,213,220,226,
                    232,236,240,244,247,248,250,250,250,248,247,
                    244,240,236,232,226,220,213,206,198,190,182,
                    173,164,154,145,135};

unsigned char i = 0;
//
void __interrupt(high_priority) ISR(void) {
    if (T0IF) {
        if (i==80) {i = 0;}
        CCPR1L = pwm1[i];
        CCPR2L = pwm2[i];
        ++i;
    }
}

```

```

//
    TOIF = 0;
    TMR0 = TMR0_init_value;
}
}

void main() {

//*****
//***** TMR0 configuration *****
//*****
// Reference PIC16F877A Datasheet, section 5.0, pag. 53
OPTION_REG = 0b10000001; // TMR0 on, Prescaler = 4
GIE        = 1;
TMR0IE     = 1;          // Enables the TMR0 interrupt
TMR0       = TMR0_init_value; // Initial TMR0 value

//*****
//***** PWM (CCP1 and CCP2) configuration *****
//*****
// Reference PIC16F877A Datasheet, section 8.3, pag. 67

PR2        = 199; // Tsw = [(PR2) + 1] * 4 * TOSC * (TMR2 Prescale Value)
           // 99 50 kHz
CCPR1L     = 0;   // Initial PWM
CCPR2L     = 0;
TRISC      &= 0b11111001; // Declare RC1 and RC2 as output
           // 199 00
           // 124 01
T2CON      = 0b00000101; // TMR2 = on, Prescale = 1 (<1:0> 00) 01 = 4prescaler
CCP1CON    = 0b00001100; // Enable PWM output (CCP1)
CCP2CON    = 0b00001100; // Enable PWM output (CCP2)

while(1) {
}
}

```