

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA**

GRUPO DE ROBÓTICA TAURA BOTS

MELHOR MECÂNICA – ROBOCHARQ

TAURA BOTS – AMP OP LINE

Bruno Gabriel Flores Sampaio

Charqueadas, RS, Brasil

2022

CAPÍTULO 1 MOTIVAÇÃO

A equipe de robótica *Taura Bots* da Universidade Federal de Santa Maria esteve presente na competição de robótica *RoboCharq* realizada pelo Instituto Federal Sul Campus Charqueadas na categoria de Robô seguidor de linha analógico contando com os membros: Bruno Gabriel Flores Sampaio.

A equipe *Taura Bots* realizou toda etapa de desenvolvimento do robô, desde sua idealização até a montagem final, sendo um projeto focado na área de eletrônica, realizado como forma de verificação dos conhecimentos de eletrônica aprendidos em sala de aula pela Universidade Federal.

Com o propósito de realizar um carro autônomo com controle inteligente, foram utilizadas lógicas de controle utilizando amplificadores operacionais, os chamados *Amp-Ops*. Os *Amp-Ops* são dispositivos eletrônicos que são capazes de realizar operações matemáticas das mais diversas formas, sendo capazes de realizar operações de soma, subtração, multiplicação, integração e derivação, entre outros. Esses dispositivos são muito empregados na eletrônica analógica, uma vez que o mesmo Circuito Integrado (CI) pode ser empregado de diversas formas para finalidades distintas.

A característica mais importante dos *Amp-Ops* está na elevadíssima impedância de entrada e baixíssima impedância de saída, sendo um componente importantíssimo para aplicações de controle.

Fazendo uso dos conhecimentos de eletrônica básica e analógica, fora proposto a criação de um robô que utilizando *Amp-Ops* que fosse capaz de fazer o seguimento de referência dado pela linha na pista utilizando lógica de controle em malha fechada fazendo uso de um controlador Proporcional, Integral e Derivativo, também chamado de controle PID, muito empregado em sistemas de controle com rastreamento de referência.

CAPÍTULO 2 IDEALIZAÇÃO

O primeiro passo para a realização do robô autônomo, foi a definição dos objetivos que se almejou alcançar, dessa forma fora identificado os requisitos:

- Possuir eletrônica totalmente analógica;
- Possuir um sistema de controle em malha fechada;
- Possuir um sistema de tração diferencial;
- Possuir um controlador inteligente com controle de velocidade.

Com os objetivos em mente, fora necessário então se esboçar as primeiras simulações para avaliar o andamento da eletrônica do robô. Para fazer as simulações foram utilizados os seguintes softwares:

- *Falstad* (Framework simulador de circuitos eletrônicos online e 100% gratuito);

- *Proteus 8* (Software simulador de circuitos eletrônicos proprietário e para prototipagem);
- *EasyEDA* (Framework online para criação de circuitos impressos e prototipagem);
- *LTSpice* (Software para simulações confiáveis de circuitos eletrônicos).

Com o objetivo de criar uma malha de controle retroalimentada para poder fazer a mudança proporcional de velocidade das rodas direita e esquerda, como mostrado na Figura 1, propôs-se:

- Fazer a captação da linha como um sinal de referência e entrada do sistema;
- Fazer o controle da planta utilizando um controlador PID;
- Realizar a variação de velocidade das rodas utilizando um sinal de PWM;

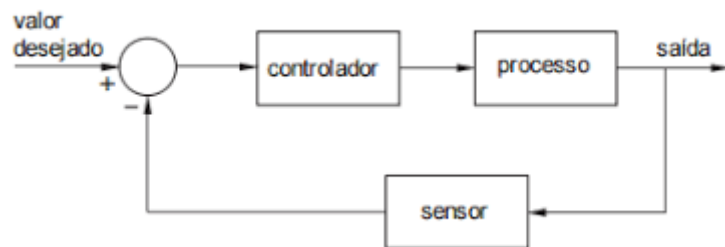


Figura 1: Malha de controle retroalimentada.

2.1 Captação da linha e sinal de referência

O sistema tem como referência que o robô esteja com o seu centro alinhado com a linha. Para realizar a retroalimentação o robô irá possuir 4 sensores de infravermelho que medem sinais de refletância do campo. Quando o sensor passa sobre o campo branco, o sensor reflete mais luz para o foto transistor que possui um sinal de tensão mais elevado, quando o sensor passa sobre a linha preta, ele reflete menos luz e por consequência, gera um sinal com baixa tensão.

Com a possibilidade de se realizar a operação de soma dos sensores onde um par possui um sinal positivo de tensão e outro par com um sinal de tensão negativo, com os pesos como mostrados na Figura 2. Quando a linha estiver no centro do robô, o valor de tensão gerado pela soma dos sensores será 0V.

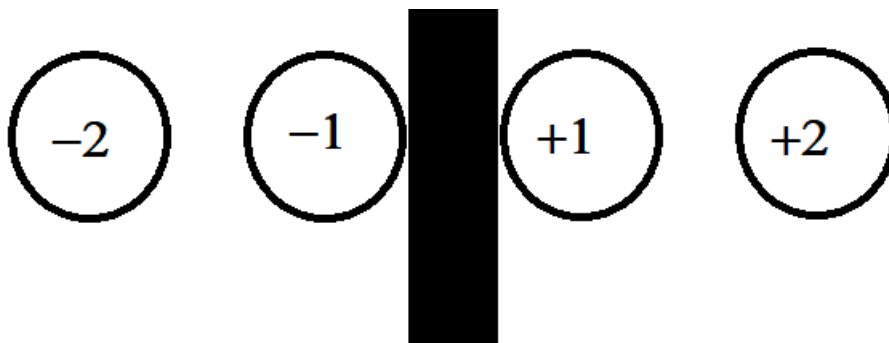


Figura 2: Sensores de infravermelho

Uma vez que a tensão de referência do robô é seu centro, a referência valerá 0V e se a soma dos sensores também valer 0V, então o robô não possui nenhum erro de angulação e o robô deverá andar para frente. Se em algum momento houver uma diferença de tensão positiva ou negativa, então o robô deverá fazer uma correção de trajetória e acelerar mais um dos lados para conseguir voltar à sua referência.

2.2 Controle PID

Para se realizar o controle do sistema como mostrado na Figura 1, foi utilizado um controlador PID que se baseia em pegar o valor do erro gerado pela diferença da referência com a medição dos sensores e aplicar um ganho Proporcional, aplicar um ganho Integrativo que será a soma do erro ao longo do tempo e um ganho Derivativo. Os *Amp-Ops* possuem configurações que são capazes de realizar esses ganhos.

2.3 Diferença de velocidade utilizando PWM

Uma das etapas mais cruciais e difíceis de se realizar, foi o desenvolvimento de um PWM que fosse variável de acordo com o valor do erro. A solução encontrada para o problema foi na criação de uma onda triangular que serviria como onda portadora a qual seria comparada com o sinal de erro gerado pelo controlador, como ilustrado na Figura 3.

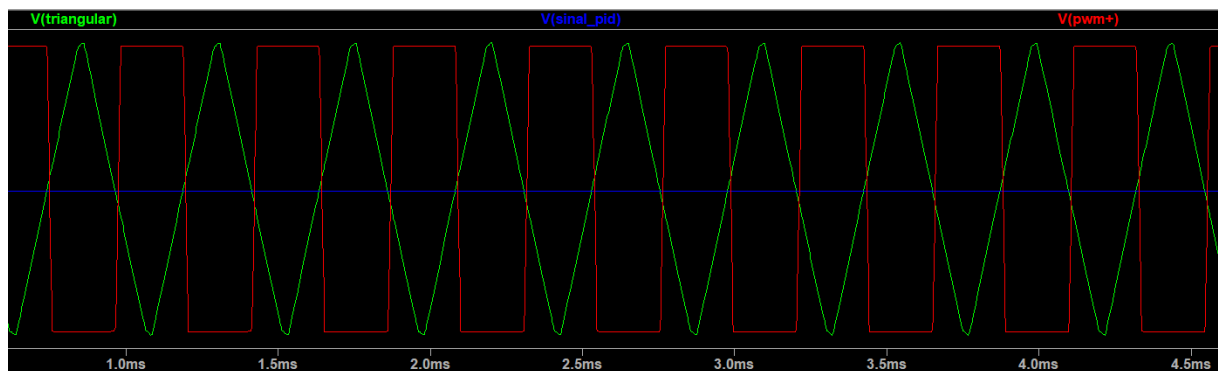


Figura 3: Comparação da onda triangular com o sinal de erro. PWM

A Figura 3 ilustra um *print* da simulação utilizando o *LTSpice* da configuração adotada. Em verde tem-se a onda triangular da portadora e em azul a onda de erro. Fazendo a comparação dos sinais, tem-se uma amplitude positiva de tensão quando a onda verde é maior que a azul e uma amplitude negativa de tensão quando o inverso ocorre. Dessa forma, gera-se um sinal PWM modulado de acordo com a amplitude do sinal de erro.

Para se fazer a diferença de velocidade das rodas, pode-se simplesmente usar um sinal inverso para as rodas, dessa forma, se o erro for 0V, tem-se um sinal PWM de 50% de *duty cycle* e ambas as rodas deveriam ter a mesma velocidade, se o erro for positivo ou negativo, muda-se o *duty cycle* para

mais ou menos que 50%, dessa forma um dos motores irá girar mais rápido que o outro, levando o carrinho ao equilíbrio.

CAPÍTULO 3 DESENVOLVIMENTO

Definido o escopo do projeto, iniciou-se as etapas de simulação dos circuitos eletrônicos do robô utilizando o software *LTSpice*. A Figura 4 ilustra o escopo final do projeto, pós todas as adaptações.

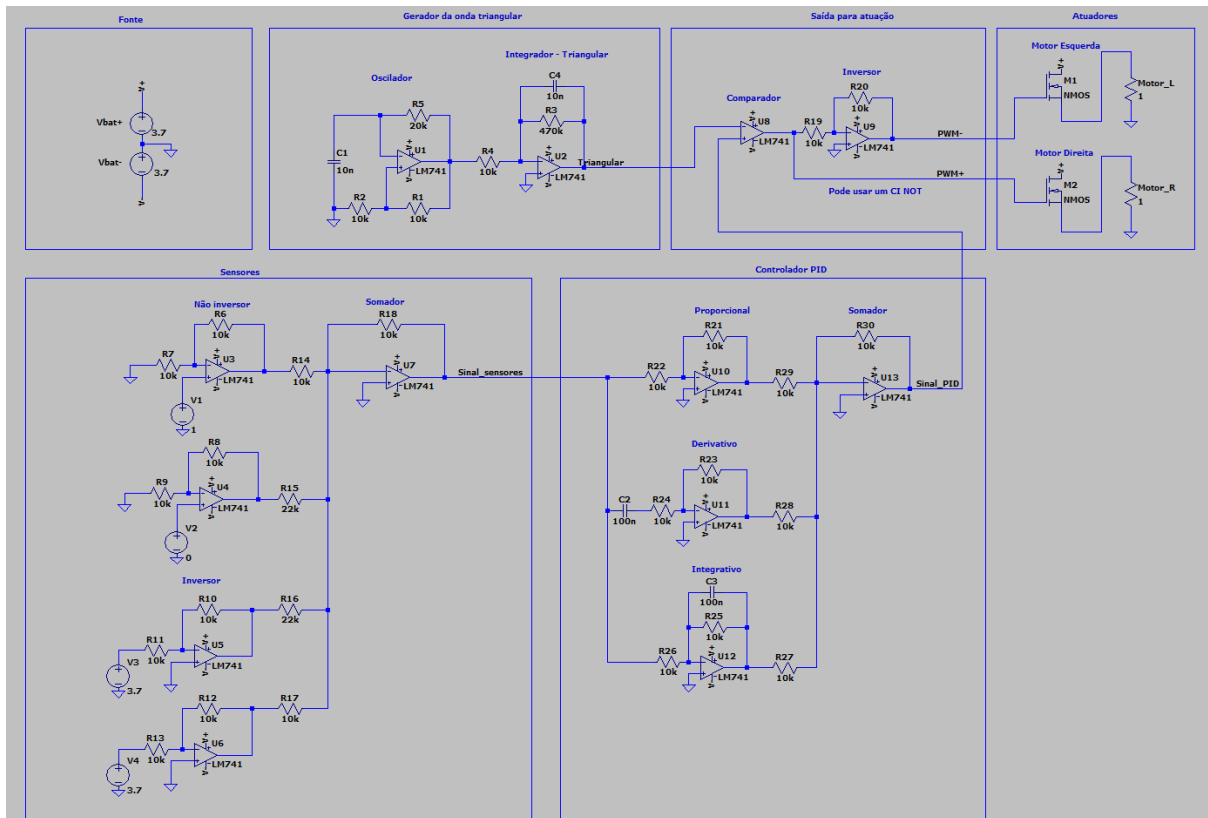


Figura 4: Circuito completo inicial - *LTSpice*.

Analisando a Figura 4, pode-se perceber alguns blocos que definem circuitos especializados dentro do circuito. Será falado em detalhes cada uma das etapas nos subtópicos de 3.1 a 3.6 seguindo a ordem de visualização da esquerda superior à direita inferior da Figura 1.

3.1 Fonte

A peça primordial para se começar a pensar em um circuito elétrico, sem dúvidas é a bateria. Para o projeto fora escolhido duas baterias de Lion-Ion de 3.7V 18650 de tensão nominal devido sua alta durabilidade de carga e custo baixo. Essas baterias serão utilizadas em série como mostrado na

Figura 2 possuindo um *Tap* central onde se está a referência de terra do circuito, necessário para se utilizar os *Amp-Ops*.

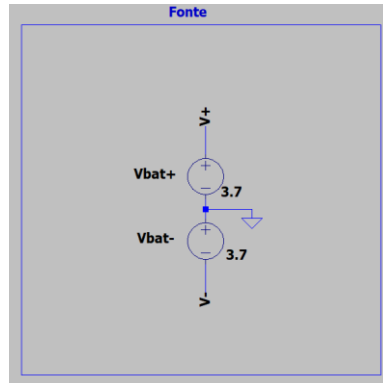


Figura 5: Configuração das baterias. Conexão Série com *tap* central.

3.2 Gerador de onda triangular

O gerador de onda triangular foi realizado utilizando os *Amp-Ops* TL082 na configuração de *Schmitt Trigger*, gerando um sinal de onda quadrada. A saída do TL082 entra em um LM741 na configuração de integrador, que realizando a etapa de integração da onda quadrada, cria uma rampa oscilatória crescente e decrescente.

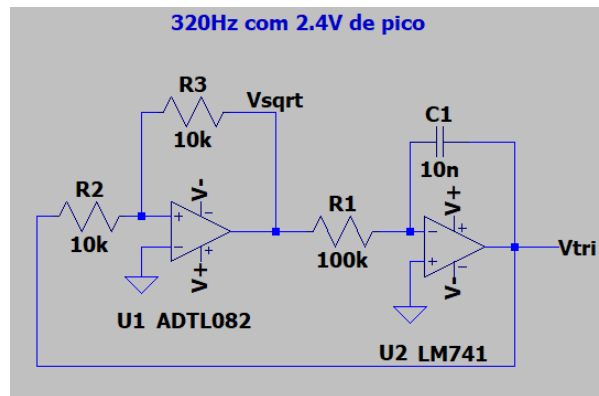


Figura 6: Gerador de onda quadrada (V_{sqr}) e triangular (V_{tri})

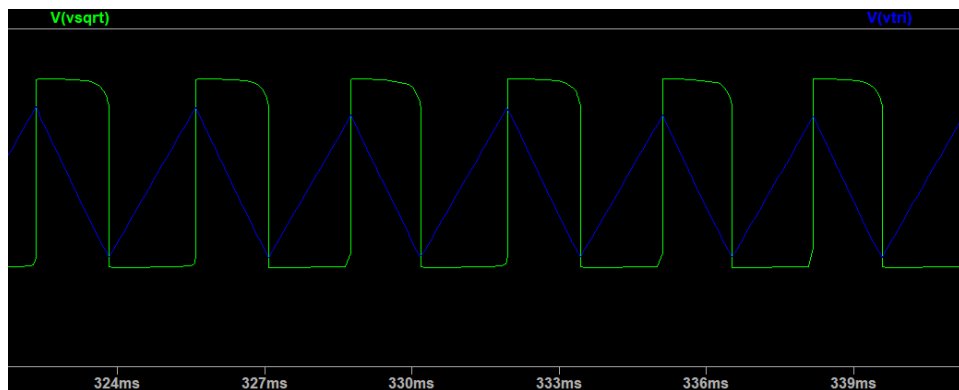


Figura 7: Gerador da onda triangular portadora

3.3 Saída para atuação e atuadores

Para se fazer a comparação dos sinais de controlador com a onda portadora, utilizou-se um *Amp-Op* LM741 que fará a modulação PWM e o mesmo *Amp-Op* para fazer a onda inversa de atuação (Figura 8). Dessa forma, tem-se dois sinais PWM complementares, que irão ativar dois MOSFETs IRF640 que farão ativação dos motores.

A potência entregue para cada motor será pela relação do *duty cycle* da onda PWM.

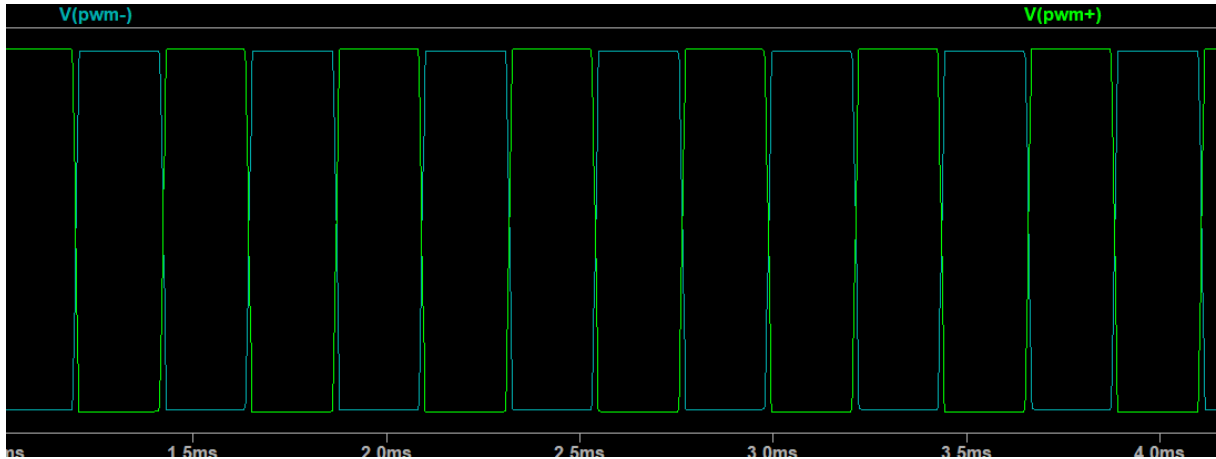


Figura 8: Onda PWM dos motores (Ondas complementares)

3.4 Sensores

Para fazer a aquisição dos valores de linha, fora utilizado os sensores infravermelhos TCRT5000. Esses módulos são dotados de um led infravermelho e um foto transistor que capta a reflexão da luz infravermelha do chão. A Figura 9.A ilustra a imagem do sensor, a Figura 9.B ilustra o esquemático do sensor e a Figura 9.C ilustra um exemplo de montagem.

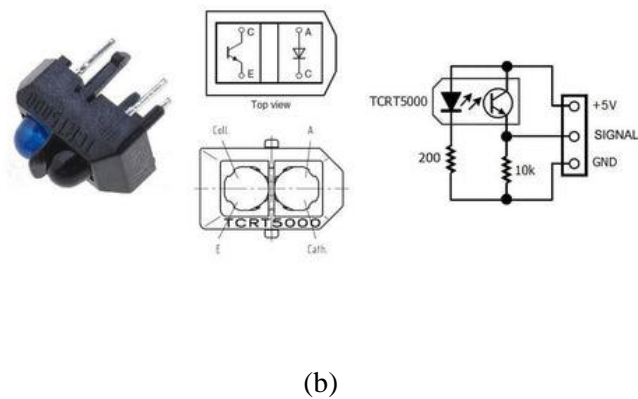


Figura 9: (a) Imagem TCRT500 (b) Imagem técnica TCRT500 (c) Esquema de montagem

A fim de dar um ganho nos sensores como ilustrado na Figura 2, foi utilizado um *Amp-Op* LM324 para dar um ganho positivo e negativo com valores de 2 para os sensores externos e 1 para os sensores internos. A etapa de soma dos sinais é realizada por um *Amp-Op* LM741.

A Figura 10 ilustra o circuito com mais detalhes. Os sensores foram representados como uma fonte de tensão.

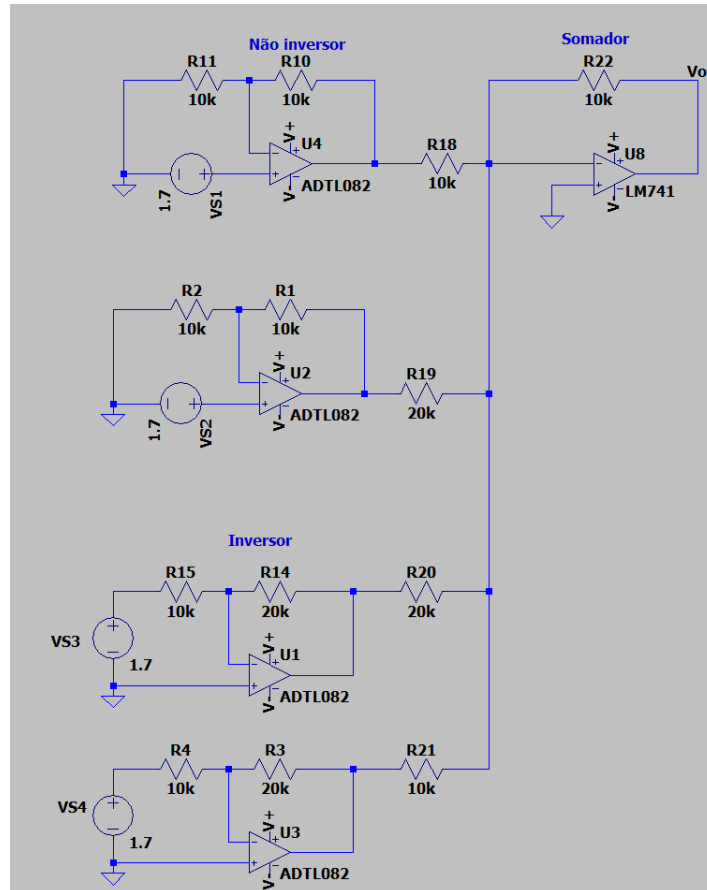


Figura 10: Circuito somador.

3.5 Controlador PID

A lógica de controle PID possui o intuito de trazer o sinal de erro para zero o mais rápido possível através da manipulação matemática o valor erro mudando a velocidade das rodas do robô. O *Amp-Op* utilizado para essa etapa foi o LM324 (4 LM741 internos). Figura 11 ilustra a configuração de montagem.

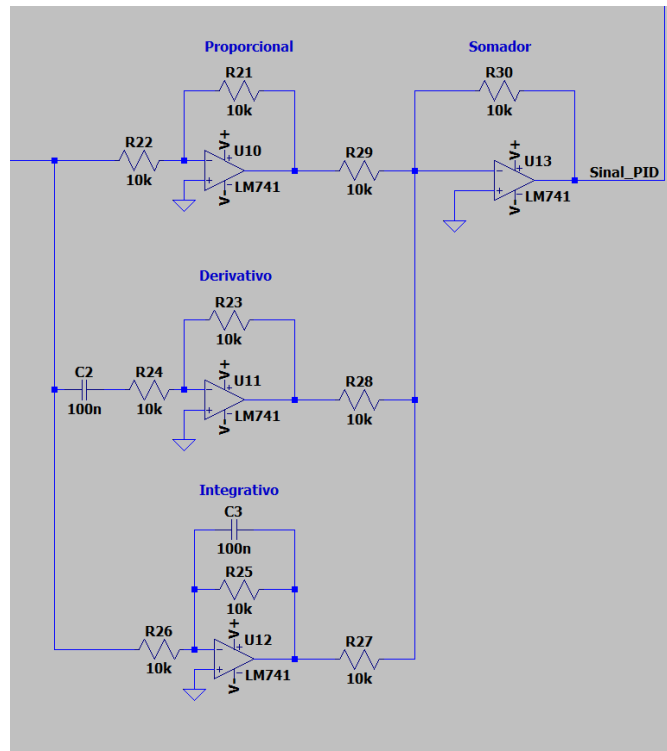


Figura 11: Configuração do PID utilizando um LM324

CAPÍTULO 4 CONCLUSÃO

Após realizado a simulação das etapas de desenvolvimento, foi possível se comprovar os valores obtidos na prática, sendo possível se mudar a velocidade dos motores de forma não apenas proporcional ao erro de medição, mas também com as etapas de correção integrativas e derivativas. Foi possível se verificar as ondas de portadoras e atuação dos motores. Para comprovação dos resultados elétricos, foi utilizado osciloscópios e multímetros.

A etapa conseguinte à de desenvolvimento, foi a etapa de criação da Placa de Circuito Impresso (PCI) utilizando o *Proteus 8* e *EasyEDA* (Arquivos em anexo).