PROJETO E DESENVOLVIMENTO DE UM ROBÔ AUTÔNOMO SEGUIDOR DE TRILHA

Nunes, L. F.; Oliveira, P. B.; Olipe, H; Cunha, M. J.; Vincenzi, F. R. S.; Morais, J. S.; Morais, A. S.
Núcleo de Controle e Automação
Universidade Federal de Uberlândia, Faculdade de Engenharia Elétrica, Uberlândia – MG,
aniel@eletrica.ufu.br

***Resumo* - O presente artigo tem como objetivo apresentar detalhadamente as etapas do projeto e desenvolvimento de um robô autônomo seguidor de trilha.**

***Palavras-Chave* – Robô Autônomo, Seguidor de Trilha, Arduino, Controle.**

DESIGN AND DEVELOPMENT OF AN AUTONOMOUS LINE FOLLOWER ROBOT

***Abstract* - This paper aims to present in detail the design and development stages of an autonomous line follower robot.**

***Keywords* - Autonomous Robot, Line Follower, Arduino, Control.**

NOMENCLATURA

PD	Controlador Proporcional derivativo.
Kd	Constante Derivativa.
Kp	Constante Proporcional.
Vdc	Tensão de alimentação do motor.
rpm	Rotações por minuto do motor.
Vsa	Tensão de saída do Arduino.
Vled	Tensão do diodo emissor infravermelho
Re	Resistor do emissor
Cnt	Coeficiente negativo de temperatura

I. INTRODUÇÃO

Cada vez mais a robótica aliada às técnicas de inteligência artificial e várias formas de controle, têm evoluído e se tornado uma potente geradora de oportunidades para o avanço da ciência. Atualmente, os robôs já não são mais fixos no seu ambiente e tampouco exclusivos na indústria.

Hoje já se fala em robôs que interagem com os seres humanos, facilitam a sua vida, proporcionam diversão e comodidade, tomam decisões não pré-programadas, aprendem com movimentos, erros e acertos e exploram ambientes desconhecidos. Um dos grandes avanços das técnicas de controle aplicadas à robótica foi a criação de máquinas autônomas, as quais têm simplificado fortemente a vida de grande parte das pessoas evitando situações perigosas e trazendo lazer e comodidade..

Nos tempos atuais, percebe-se um crescente aumento no uso de controladores automáticos aplicados a engenharia, fato que vem diminuindo cada vez mais a mão-de-obra humana para o manuseio de equipamentos e maquinários, considerados, muitas das vezes, perigosos. Por meio da automação, o controle de sistemas busca minimizar o tempo de produção, como também o índice de erros durante a mesma, erros esses considerados como algo indesejável para o sistema.

Um processo controlado em malha fechada reduz a sensibilidade a interferências e ruídos, o que, consequentemente, torna o controle quase ideal. No caso dos seguidores de trilha, o controle em malha fechada é fundamental tanto para seguir a linha como para rejeitar perturbações e não linearidades da iluminação e pequenos defeitos na pista que comprometem a leitura do sensor.

No projeto apresentado neste artigo, utilizou-se um controle de velocidade do carrinho em malha aberta. Para a diminuição de interferências externas e ruídos, conforme a Fig. 5, na parte II-C, dispõe-se os componentes de modo a tornar mais robusta a medição do sinal. O sistema de sensoriamento foi feito de forma simples utilizando fotodiodos emissores e receptores de luz, processo o qual será melhor apresentado no tópico II-B.

O sistema do seguidor utiliza uma linha branca contínua em um fundo preto no seu caminho durante sua trajetória. Esta linha tem que ser de no mínimo 15 milímetros de largura, a fim de ter um funcionamento com um menor índice de falhas. A captura de dados do sensor é armazenada por um curto período de tempo de forma que, caso o seguidor perca a trilha, ele executa a última ação armazenada até encontrar a linha novamente, que melhor analisado no item III.

II. DESCRIÇÃO DO PROJETO

O projeto do seguidor de trilha foi dividido basicamente em três etapas, sendo elas, mecânica, eletrônica e programação. A alimentação do circuito, motores e microcontrolador, foram promovidos através de uma bateria de Lítio-Polímero 3S, ou seja 11,1V, e 2200 mAh de capacidade de armazenagem de carga.

A. Mecânica

A parte mecânica do seguidor de trilha envolveu, além de uma análise experimental, um breve estudo analítico a cerca de centro de gravidade, análise considerada de suma importância, visto que a falta de estabilidade na planta pode

X CEEL - ISSN 2178-8308
24 a 28 de setembro de 2012
Universidade Federal de Uberlândia - UFU
Uberlândia - Minas Gerais - Brasil

implicar uma instabilidade no controle. Foi feito também um ajuste minucioso nas rodas utilizadas para que houvesse um atrito bom com a pista.

1) Centro de gravidade - Para o carrinho ficar com uma estabilidade próxima do perfeito, decidiu-se por utilizar o centro de gravidade da seguinte forma:

Dividiu-se a distância entre o eixo traseiro e dianteiro em três partes de 4,67 cm. A partir de estudos realizados anteriormente sabe-se que o centro de gravidade de um carrinho deve se localizar na primeira terça parte e entre os eixos. O peso pôde ser distribuído com a ajuda da bateria de lítio polímero com três células, utilizadas para alimentar o circuito, Arduino e motores.

2) Carroceria - A base do robô foi feita com madeira compensada de 4(quatro) mm de espessura. A escolha do material deveu-se ao fato de que o mesmo possui um baixo custo de aquisição, é resistente e extremamente leve. Uma ótima opção para a confecção da carroçaria seria com placas de carbono, estas não pesam quase nada (preocupação devida à limitação de peso do protótipo, proposta pela organização do torneio, de 900g) e são abruptamente resistentes, sendo estas utilizadas principalmente em aeromodelos.

3) Rodas – As rodas traseiras são de carrinho de bebê com 15 cm de diâmetro e massa equivalente a 110g, cada uma. Para a roda dianteira, utilizou-se uma roda “boba” na frente, a qual possui a funcionalidade de direcionar o carrinho. Esta, poderia ter sido confeccionada com retentores, rodas de 1” e alguns arames de aço, formando assim uma bequilha que também gira 360º em torno de seu eixo.

O diâmetro maior na roda traseira possibilitou que o carrinho percorresse um maior caminho, porém, dificultou um pouco o controle, visto que as rodas possuem uma inércia muito grande.

4) Motores - Foram utilizados dois motores de redução 50:1, 174 rpm, alimentação de 7,2 Vdc, torque de 7,13 Kg/cm e possui encoder, contudo este não foi utilizado devido ao tempo limitado para desenvolver o projeto. Cada motor foi acoplado ás rodas traseiras, fazendo com que o movimento de uma fosse independente da outra.

Fig. 1. Motor DC utilizado no projeto.

5) Sensor - Na placa de sensor, fizemos uma proteção utilizando cola quente e chapa de fenolítio para que os componentes do mesmo não sofressem com os possíveis impactos que viessem a ocorrer. Esta foi acoplada à carroçaria do robô utilizando dobradiças de ferro, assim, na rampa o vetor de sensores pôde se movimentar de tal forma que se mantinha sempre o mais próximo possível da linha branca.

B. Eletrônica

A parte eletrônica do projeto foi relativamente simples e de baixo custo. Tal fato foi facilitado pela plataforma de aprendizagem Arduino que além de ser simples e barata, possibilitou a fácil implementação do controlador.

1) Microcontrolador - Nesse projeto foi utilizado o Arduino UNO, uma plataforma open-source de computação física baseada no microcontrolador ATMega-328, que engloba um software e um hardware. Tal plataforma tem sido destaque em muitos projetos tecnológicos, porém ainda é pouco utilizado por grande parte dos pesquisadores, devido, principalmente, a um preconceito pela facilidade.

Uma das grandes vantagens do Arduino em relação a alternativas para controlar um sistema, são os *Shields*, placas que aperfeiçoam as capacidades da plataforma que podem ser plugadas no mesmo. Atualmente existe uma vasta opção de *Shields* como, *ethernet*, *bluetooth*, motor shield, LCD, sensor ultrassônico, dentre tantos outros

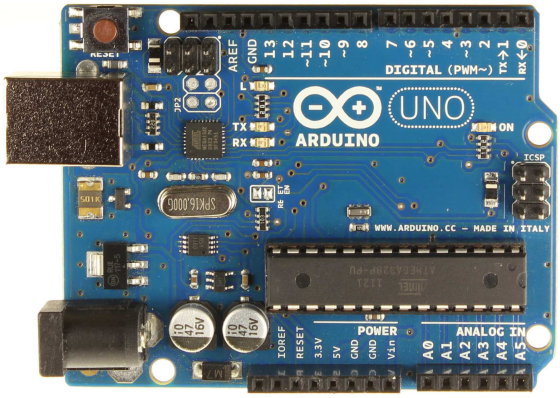

Fig. 2. Plataforma Arduino.

2) Ponte-H - Para controlar os motores por PWM, utilizou-se a ponte-H L298N a qual possui entrada para dois motores e é alimentada com tensão de 12V. O dispositivo, além de facilitar o controle PWM dos motores, permite que os mesmo sejam controlados independentemente. Os fios do motor são conectados na ponte-H e da mesma saem outros dois fios que, neste caso, foram conectados aos pinos de entrada digitais PWM 9 e 10 do Arduino. Os fios para alimentação do motor saíram também do L298N e foram conectados à bateria.

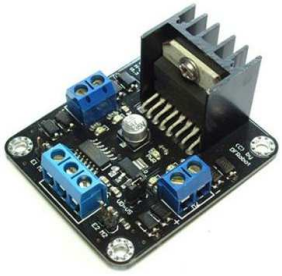

Fig. 3. Ponte-H L298N.

3) Sensoriamento - Foram utilizados treze receptores e outros treze emissores de luz (fotodiodos), oito resistores

de 220Ω e cinco de 2200 Ω, uma placa de fenolite. A placa de sensor foi confeccionada de tal modo que na curva de noventa graus quatro receptores e emissores de luz ficassem mais a esquerda, e um par de componentes mais a direita, sendo assim, o sensor poderia identificar a linha branca na horizontal caracterizando a curva de 90º.

Os outros oito fotodiodos foram posicionados no centro do sensor para que a linha pudesse ser identificada com grande precisão. A proteção envolvendo o sensor foi feita para que houvesse uma menor incidência de luz do ambiente, assim a leitura não foi prejudicada, e para proteger os componentes de possíveis pancadas.

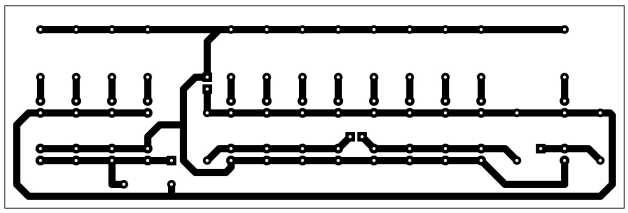

Fig. 4. Esquema da placa de sensor.

O dimensionamento dos resistores dos emissores foi bem simples. Partindo do pressuposto que o LED (fotodiodo) consome de 10 mA a 25 mA, em media, e que a queda de tensão em um LED infravermelho é de 1,6V através da formula U=RI temos que:

U = Vsa - Vled

U = [5 V (saída Arduino) – 1,6V] = 3,4 V

I = 15 mA

Re ≈ 220 ohms

Foi colocado um resistor para cara emissor devido ao coeficiente negativo de temperatura (cnt).

(Cnt = quanto maior a temperatura de um semicondutor menor a resistência.). Portanto para que todos os LED's sejam controlados da mesma maneira é necessário um resistor pra cada. Caso fosse um resistor para todos, um LED provavelmente puxaria mais corrente, pois é um componente real e não ideal. Com isso, este aumentaria sua temperatura, reduziria a resistência e esse ciclo se repetiria até o fotodiodo emissor queimar.

Os resistores utilizados no conversor A/D foram definidos empiricamente conforme obtivéssemos uma melhor variação de tensão. A leitura do sensor foi feita através de um conversor A/D no caso do Arduino é de 10 bits (0 a 1023). Dependendo dessa leitura e com o controle tem-se a reação do motor.

O Arduino capta as informações de um sensor analógico através da função analogRead() e pode enviar as informações para o sensor através da função analogWrite(). Dessa forma, quando o seguidor de trilha fizer uma curva, esse movimento será salvo na memória da plataforma e enviará para o sensor, e assim irá fazendo para cada curva. Como sensores analógicos captam um sinal de 0 até 1023 bits, a grandeza medida foi designada a partir do shield LCD, pega-se os valores da pista e o controle sabe quando o carrinho esta na linha branca ou na parte preta.

Os sensores digitais funcionam através de um sinal HIGH ou LOW, sendo que HIGH seria 5V e LOW 0V, processo análogo ao de uma chave,. A leitura das entradas digitais são feitas através da função digitalRead().

C. Programação

O software utilizado para compilar os programas, *processing*, é disponibilizado no próprio site do Arduino, sendo esse gratuito e multiplataforma. O ambiente gráfico do mesmo é de fácil compreensão e autoexplicativo.

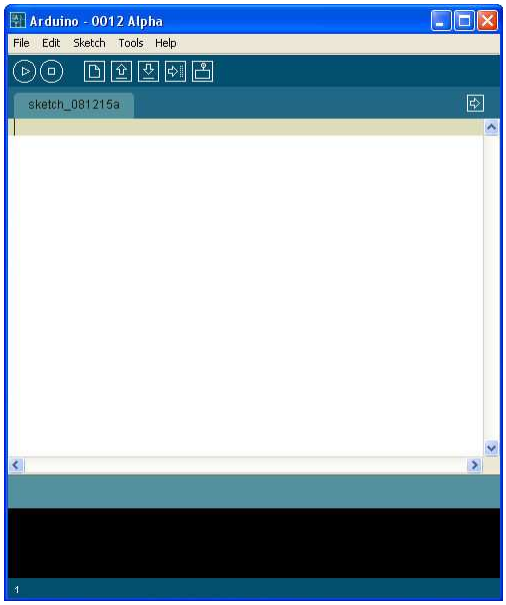

Fig. 5. Ambiente gráfico do Processing.

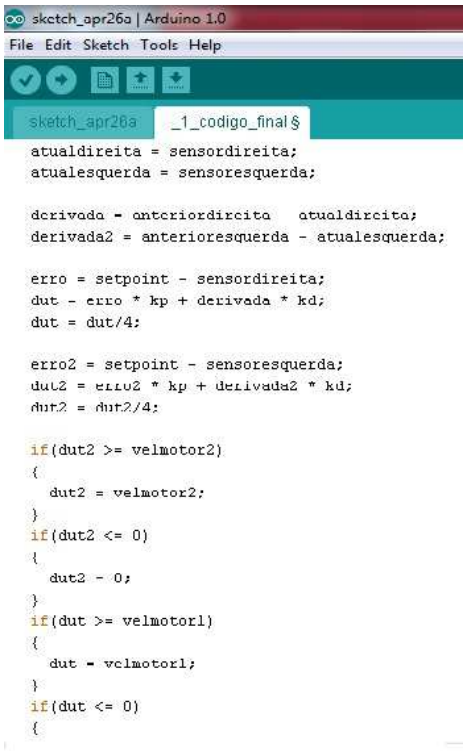

Fig. 6. Linhas de código relacionadas ao controle PD.

O Arduino comunica com o computador através de uma porta USB, assim pode ser feito o upload do programa para o hardware. A linguagem utilizada é bastante simples sendo uma implementação do Wired e muito parecida com C/C++.

Na figura abaixo segue um trecho do código relacionado ao controle PD. Um dos grandes desafios da competição foi

fazer com que o seguidor de trilha fizesse uma curva de 90º sem que o mesmo saísse da pista ou perdesse a trilha. Porém, a lógica para codificação foi relativamente simples, baseado na posição de cada diodo situado nos extremos do sensor, e da leitura PWM de cada motor, assim quando o sensor detectasse a curva, uma roda se move para trás em certa velocidade, e a outra para frente.

A IDE do Arduino inclui automaticamente todas as bibliotecas básicas que utilizamos em C/C++, além de mais uma gama de bibliotecas para controle de motores de corrente contínua, servomotores, displays de LCD, recepção de sinais de sensores, entre tantos outros.

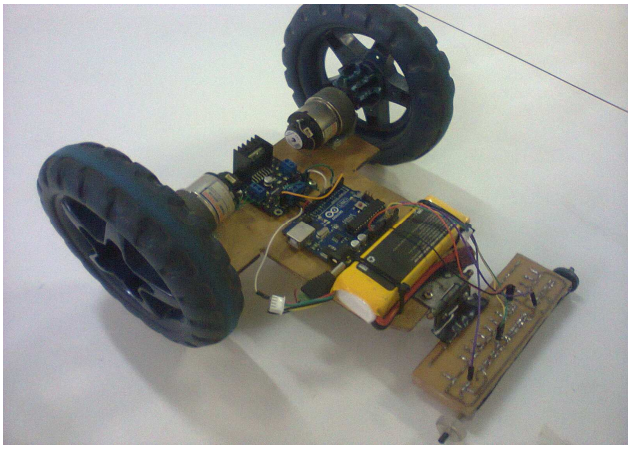

Fig. 7. Projeto do seguidor de trilha concluído.

III. ESTRATÉGIA DE CONTROLE

O controle do carro seguidor de trilha foi feito com um proporcional derivativo (PD). Para ajudar a encontrar os parâmetros exatos que deixam o controle com melhor resposta, sem oscilações, foi utilizado um método empírico, um critério denominado Ziegler-Nichols, busca-se a melhor ajustar os parâmetros do controlador que proporcionem a melhor reposta possível, preferencialmente rápida e sem oscilações. Os métodos empíricos garantem uma boa aproximação para a parametrização do sistema com erros de no máximo 25%.

O método de Ziegler-Nichols foi desenvolvido a partir de simulação de resposta de processos típicos, e ajustando os parâmetros para obter respostas adequadas. Apesar de não ser um método sofisticado, ainda é muito utilizado e permite o projeto rápido de controladores. No caso desse robô, utilizamos um projeto baseado na resposta da planta.

Inicialmente zeram-se as constantes proporcional Kp e derivativa Kd. Aumenta-se a constante proporcional até que o sistema comece a oscilar. No momento em que o mesmo atinja este ponto, é necessário inserir um derivativo no sistema para corrigir a oscilação. Aumenta-se a constante derivativa até um valor onde ela corrija a oscilação. Foi necessário um leve ajuste na constante proporcional para acelerar o sistema. Realiza-se este processo até se obter valores satisfatórios, neste caso obteve-se Kp = 14,5 e Kd = 110000.

Testes práticos foram realizados para identificar na pista quais os valores lidos pelos sensores correspondiam a cada região, linha branco ou pista preto. A partir desta análise definiu-se o *setpoint* (referência) do sistema.

IV. CONCLUSÕES

O projeto descrito nesse artigo foi de grande valia para os envolvidos no mesmo, pois proporcionou uma gama de conhecimentos que, a priori, nenhum componente no grupo possuía. Conceitos como controladores, novas ferramentas para desenvolvimento de robôs e sistemas autônomos para controle de máquinas, foram explorados minuciosamente para que tudo fosse feito de acordo com o que atual mercado tecnológico e científico exige, como praticidade, processos automatizados, diminuir a exposição de humanos a perigos, dentre outros.

O custo total de construção foi o menor possível, porém mantendo um bom desempenho. Logo se percebe que é totalmente possível desenvolver alguns tipos de robôs com gastando em média cerca de R$ 250,00.

Ficou claro durante o desenvolvimento do protótipo que o Arduino é uma tecnologia que pode e irá trazer muitos frutos aqueles interessados em desenvolver tecnologias inovadoras. Um equipamento completo, *open-source* e o mais importante, uma ferramenta muito simples e poderosa.

AGRADECIMENTOS

Os autores do artigo agradecem ao Laboratório de Controle Digital e Robótica (LCR) pelo auxilio ao emprestar ferramentas para confecção do projeto. Agradecemos também o Patrocínio do Airton Borges da Auto Controle Modelismo e ao Giancarlo da Gian Aeromodelos.

REFERÊNCIAS BIBLIOGRÁFICAS.

[1] MARGOLIS, M. Arduino Cookbook. 01 ed. O'Reilly Media Inc., Sebastopol, California, 2011.

[2] Acedido em 13 de Março de 2012 em, http://www.Arduino.cc/en.

[3] SPONG, M.; VIDYASAGAR, M.; HUTCHINSON, S.. Robot Dynamics and Control. Wiley, New York, USA, 2005.

[4] OGATA, K.. Modern Control Engineering. Prentice-Hall ed., Englewood Cliffs, New Jersey, USA, 1970.

[5] BANZI, M.. Getting Startes with Arduino. O'Reilly Books, California, USA, 2009.

[6] Introdução ao Arduino. Pessanha Santos. ASPOF EM-AEL. Acedido em 20 de janeiro de 2012 em, http://www.ebah.com.br/content/ABAAABNcsAC/apresentacao-Arduino.

[7] Introdução ao Arduino. Adaptado por: Cirineu Carvalho Fernandes e Geilton Teles Lopes. Acedido em 20 de Janeiro de 2012 em, http://www.ebah.com.br/content/ABAAABecwAE/introducao-ao-Arduino-get-starter-com-Arduino.

[8] SANTOS, Nuno Pessanha; Arduino, Introdução e Recursos Avançados, 2009; Acedido em 13 de Janeiro de 2012 em, http://pt.scribd.com/doc/57058743/tutorial-Arduino.