

RASPCAR – Carro Robótico Guiado Remotamente

Gustavo de O. Guidetti, Arthur V. M. Sabatino, Daniela Eloise Flôr,
Eduardo H. M. Cruz

¹ Instituto Federal do Paraná (IFPR) – Campus Paranavaí
CEP 87703-536 – Paranavaí – PR – Brasil

`gustavooguidetti@gmail.com, sabatinoarthur@gmail.com`

`daniela.flor@ifpr.edu.br, eduardo.cruz@ifpr.edu.br`

Resumo. *Nós humanos estamos o tempo todo pensando em projetos para encurtar caminhos, quebrar obstáculos e facilitar a vida moderna. O carro autônomo faz parte destes projetos, uma vez que este pode contribuir para a evolução do trânsito, de modo a reduzir acidentes, facilitar as entregas de lojas e restaurantes, melhorar as condições de transporte de pessoas, entre outros. A partir disso, propõe-se o RASPCAR. O protótipo de carro robótico guiado remotamente, que tem como meta construir e adaptar um robô com funções referentes ao conceito de veículo autônomo a partir de ferramentas de custo e escala reduzida.*

Abstract. *We humans are all the time thinkin about projects to shorten paths, break obstacles and facilitate the modern life. The autonomous car is part of this projects since it can contribute to the evolution of traffic, in order to reduce accidents, facilitate deliveries to stores and restaurants, improve conditions for transporting people, among others. From this, RASPCAR is offered. The remotely guided robotic car prototype, which aims to build and adapt a robot with functions related to the concept of autonomous vehicle from low cost tools and reduced scale.*

1. Introdução

A ideia do carro autônomo vem sendo arquitetada por pesquisadores a cerca de 60 anos; porém, ainda não possuímos uma versão final e concreta para este conceito de automóvel, ou seja, ainda estamos no estágio de desenvolvimento.

Com a ascensão da internet das coisas (IoT) nos últimos anos, o carro de cunho autônomo vem sendo uma realidade cada vez mais próxima de nós, tendo em vista tantas outras coisas que já se tornaram possíveis no ambiente da automação. No contexto deste atual trabalho, apresenta-se então um protótipo de carro remotamente controlado, modelado principalmente a partir de uma placa Raspberry pi model B [Nauman 2018] e um microcontrolador eletrônico que pode ser considerado também um microcomputador por possuir um sistema operacional próprio (Raspibian). Para a identificação do ambiente, foram utilizados sensores ultrassônicos, obtendo a locomoção do carro através da comunicação entre as informações extraídas por estes e o código escrito na linguagem Python de programação, atuante na parte lógica do protótipo.

O RASPCAR tem como meta, além de discutir e popularizar como as gigantes do automobilismo e da robótica estão operacionalizando o conceito de carro autônomo,

construir e adaptar um protótipo com funções semelhantes, a partir de ferramentas de custo e escala reduzida. Prática que inclusive, oportuniza a obtenção de determinadas noções de trabalho deste setor da maneira menos abstrata possível, aguçando os sentidos do ponto de vista científico. As discussões sobre os problemas encontrados, a viabilização de hipóteses de solução, bem como o desenvolvimento seguido de testes, análise de resultados e os respectivos refinamentos permitem atingir os objetivos do projeto.

Neste artigo, iremos visualizar seções sobre: as obras que serviram para a inspiração e expansão dos conhecimentos na área da robótica autônoma (Seção 2) e os métodos utilizados para o progresso do projeto (Seção 3). Além da criação do protótipo RASPCAR e seu desenvolvimento (Seção 4), os resultados alcançados ao fim de todo o trabalho realizado (Seção 5), a conclusão que podemos tirar desta obra (Seção 6) e enfim os planos para o futuro do robô (Seção 6).

2. Trabalhos Relacionados

[Ondruš et al. 2020] diz sobre a importância da indústria automotiva e cita dados europeus como exemplo, quando diz que esta indústria é responsável por 14% da produção total e investimento financeiro do setor europeu manufatureiro. Ele então introduz a ideia de que a autonomia dos veículos teria um grande impacto, pois estes poderiam substituir frotas corporativas e transporte de funcionários, que podem utilizar o tempo anteriormente gasto para ganhar produtivas horas de trabalho. Isso além da clara redução de acidentes (número que será até 80% menor em 2040, segundo previsões científicas). Os níveis de automação em carros também são apresentados no artigo de Ondruš, através da tabela (Tabela 1), publicada em 2014 pela SAE (Society of automotive engineers).

Por fim, são citadas as possíveis tecnologias para a obtenção de um carro autônomo. Entre elas, constam os sensores ultrassônicos que, inclusive, foram escolhidos para a construção do protótipo do RASPCAR referente ao presente artigo.

O trabalho de [Valocky et al. 2019] sugere, além dos sensores ultrassônicos, uma unidade de controle utilizando câmeras no teto da estação de testes, por onde percorrerá o protótipo, integrada a um script para que seja feita a localização do carro, assim enviando comandos para a realização de seus próximos movimentos através de uma conexão via rede wi-fi.

De acordo com [Lima et al. 2013], além da necessidade do robô autônomo de conseguir percorrer ambientes estavelmente, deve-se buscar a flexibilidade, de modo que este consiga lidar com diversas tarefas. Introduzindo a arquitetura híbrida em seu trabalho. Esta mescla a deliberativa (capacidade do protótipo planejar ações futuras tendo conhecimento do ambiente) e reativa (baseada em comportamentos, a partir de uma condição sensorial, extraindo informações do ambiente). O autor apresenta também uma alternativa simples na matéria-prima do protótipo, utilizando um kit LEGO MINDSTORMS, contando com um esquema sensorial devidamente arquitetado com 3 componentes diferentes. Uma bússola e alguns sensores de luz e ultrassônicos, que foram implantados nas laterais ao invés da frente, para que o carro evite a colisão com paredes. Assim mantendo uma distância específica delas, enquanto realiza um percurso seguro e linear em referência a pista ou ambiente.

Segundo [Oltean 2019], nos dias de hoje, já podemos perceber robôs substituindo humanos em diversas profissões, explicando assim a necessidade do desenvolvimento

Tabela 1. Tabela de níveis de automação

Nv.	Nome	Direção e Aceleração	Monitorar o ambiente	Solução de recurso na tarefa de condução dinâmica	capacidade de sistema
0	Sem automação	Motorista Humano	Motorista Humano	Motorista Humano	n/a
1	Assistência de motorista	Motorista Humano e sistema	Motorista Humano	Motorista Humano	Alguns modos de condução
2	Automação parcial	SISTEMA	Motorista Humano	Motorista Humano	Alguns modos de condução
3	Automação condicional	Sistema	SISTEMA	Motorista Humano	Alguns modos de condução
4	Alta automação	Sistema	Sistema	SISTEMA	Alguns modos de condução
5	Automação completa	Sistema	Sistema	Sistema	todos os modos de condução

Fonte: [SAE 2016] [Ondruš et al. 2020]

de robôs móveis. O autor propõe um protótipo de carro com 2 andares, 4 rodas, uma placa Arduino UNO, um Raspberry pi, uma câmera, sensores ultrassônicos, infravermelhos e um peculiar braço robótico. Acoplado com a justificativa de que o transporte de obstáculos e objetos pode vir a ser útil para a resolução de diversas tarefas nas quais o robô pode atuar. O braço é operado junto a um motor que, inclusive, é capaz de desempenhar um torque de 3,17 kg/cm em uma voltagem de 4.8V para o desenvolvimento de suas funções mecânicas. Quanto às funcionalidades gerais do protótipo, pode-se dizer que foram divididas, sendo uma parte delas gerenciadas por uma das placas microcontroladoras e o restante pela outra. Distribui-se para o Arduino a movimentação do robô, a leitura dos sensores infravermelhos para que o carro mantenha seu funcionamento como seguidor de linha, controlar os 4 motores e ler os sensores ultrassônicos para evitar colisão e controlar a distância dos objetos. O Raspberry gerencia a câmera (mapeamento 2D, planejamento de rotas, navegação, detecção de obstáculos juntamente com o formato e cor dos mesmo) e o motor do braço robótico, utilizado para levantar, mover e transportar objetos.

[Wei 2015] diz que um veículo autônomo deve realizar 4 tarefas básicas, sendo elas: Navegação, cobertura, localização e mapeamento, tarefas estas que podem ser resumidas em 3 perguntas: “Como eu chego lá?”, “Onde estou?” e “Onde estive?”. A partir disso, 3 conceitos passaram a ser utilizados com frequência: caminho (conjunto de posições no espaço que será percorrido); trajetória (conjunto de pares caminho/velocidade, utilizados para ir de um ponto a outro) e Planejamento (construção de uma trajetória

livre de acidentes). Então o autor apresenta algoritmos e estratégias para melhor lidar com os obstáculos que o protótipo pode enfrentar do ponto inicial ao ponto final. Assim, apresentam-se os códigos do inseto 1 e 2, que são assim chamados por simularem o comportamento de um inseto de verdade, estes podem então serem descritos pela realização do instinto natural, como insetos fazem. Deverá andar na direção de um objetivo até que um obstáculo seja encontrado, e então deverá ser contornado pelo inseto que assim pode retornar ao seu curso. O algoritmo do inseto 1 tem como característica mover-se em busca do ponto final e, no caso de encontrar um obstáculo antes de chegar, contorná-lo inteiramente. Desta maneira obtém-se a largura do mesmo e a noção da possibilidade de continuar seu curso a partir de algum ponto desobstruído encontrado enquanto circulava o objeto. Caso existente, força o carro a abandonar o obstáculo em direção ao ponto livre e logo em seguida ao objetivo, e caso inexistente, encerra o programa.

O algoritmo do inseto 2, conta com uma melhor dinâmica, uma vez que o carro não precisa contornar todo o objeto para saber se pode prosseguir, e sim fazendo-o assim que encontra um ponto desobstruído, durante seu contorno. Encerra o programa ao alcançar o objetivo ou se encontrar na ausência de desobstruções. Além da parte lógica citada, o autor também fala sobre sensores, especialmente sobre o sensor ultrassônico. Seu funcionamento é explicado como baseado na velocidade média entre a emissão e recepção da reflexão de uma onda transmitida pelo sensor. Ou seja, o sensor faz a emissão das ondas e aguarda o retorno das mesmas após serem refletidas por alguma superfície e então o tempo de retorno destas ondas indica a distância entre o robô e os obstáculos. Logo, as principais características para a análise dos sensores ultrassônicos são: alcance mínimo e máximo, faixa de frequência e precisão. Tendo então como suas maiores fraquezas a dependência da superfície alvo (rugosidade, grau de absorção, ângulo de inclinação) o nível de ruído ambiente, a suscetibilidade à interferência cruzada e seu ângulo de abertura.

A partir da visualização dos trabalhos citados, podemos começar a entender o porquê da robótica móvel autônoma ser uma área de grande interesse para desenvolvimento. A partir da análise dos resultados obtidos por outros pesquisadores, são identificadas características em comum entre eles e consequentemente suas diferenças. Nos permitindo identificar atributos positivos e negativos, para que assim seja traçado o melhor caminho possível para o início da desenvolvimento do protótipo RASPCAR. Entre eles, a constante presença de sensores ultrassônicos mostra a provável eficiência que este componente pode vir a proporcionar em projetos relacionados ao controle de trajeto em carros. Da mesma maneira, o braço robótico apresentado por [Oltean 2019] não soa ser muito interessante para o RASPCAR. Não apenas por ser um aspecto incomum, mas também por demonstrar um comportamento substituível, visivelmente claro através da interpretação dos dados estudados.

3. Metodologia de desenvolvimento

Tendo em mente os conhecimentos necessários para o desenvolvimento do protótipo RASPCAR, foram iniciados os estudos de elétrica e seus componentes, incluindo os microcontroladores. De modo a tornar mais claro o entendimento acerca da maneira como eles funcionam. Assim fazer o uso adequado destes materiais, sem danificar seus circuitos se torna mais fácil. Após isso, o próximo passo a ser tomado é a análise dos trabalhos realizado por outros pesquisadores. Esta etapa serviu, por exemplo, para entendermos o porquê de um componente ser tão popular entre os autores de projetos robóticos de

movimentação autônoma (Como o sensor ultrassônico, por exemplo) e se este deveria ou não ser implementado no RASPCAR. Além de obter uma maior noção a respeito deste tipo de projeto, entre outros.

Finalmente a construção do protótipo começou a ser executada, partindo da montagem e alocação dos elementos e terminando na parte lógica do robô, a programação. O código acabou sendo alterado diversas vezes ao longo do desenvolvimento, buscando sempre o equilíbrio entre uma movimentação suave e o desvio de obstáculos. As páginas científicas da web citadas ao longo do artigo, ocasionalmente complementavam as ideias da escrita, que durante o projeto foi ficando mais adequada para enfim desempenhar a função de um carro robótico autônomo remotamente guiado.

4. Composição do Protótipo

Esta seção apresentará a maneira como foi criado o protótipo RASPCAR, desde a escolha dos materiais que ao fim iriam integrá-lo até a programação de sua parte lógica.

4.1. Material

O material escolhido para o início da confecção do protótipo RASPCAR foi o papelão, pois com base em experiências anteriores e análises realizadas, este foi apontado como o elemento mais adequado para a realização. Visto que ele exerce as mesmas funções das finas placas de madeira usadas em projetos adjacentes, porém com um menor impacto ambiental devido a sua reciclabilidade. Ele também possuía menor peso (atributo diretamente relacionado ao bom funcionamento de um robô, cujo principal objetivo é se movimentar da maneira mais suave possível), e maior facilidade de modelagem. Uma vez que não era necessário o uso de materiais de carpintaria, mas sim tesouras e estiletes para polimento e corte, que podem ser consideradas ferramentas de mais fácil porte e manuseio.

4.2. Montagem

Sabendo que o primeiro passo para a criação de um carro é seu chassi, responsável por ser a base de todas as peças de modo que estas possam ter um lugar a serem alocadas, foi formulado um esboço do protótipo. Este tinha como finalidade de servir como planta para a construção. Neste esboço, um dos critérios que foram seguidos foi evitar ter qualquer espaço não utilizado. Busca-se assim a menor estrutura possível, para prevenir problemas de organização, tamanho e peso excedente, o que futuramente poderia vir a atrapalhar o carrinho no critério de velocidade e mobilidade.

4.3. Componentes elétricos e fiação

Ao fim da etapa anterior, já é possível ver o formato físico do protótipo, a partir da carroceteria de papelão e cola quente. Inicia-se assim a fase de inserção dos componentes que cumprirão com a parte elétrica do robô. A construção elétrica passa a ser arquitetada, de modo que os componentes eletrônicos sejam posicionados da melhor maneira possível, levando em conta seu tamanho e suas funções.

Começou-se pelo sensor ultrassônico [Bóson 2018], que ficou estabelecido onde normalmente localizam-se os faróis e retrovisores de um carro comum, na extremidade frontal, lateral e traseira. A escolha destes pontos faz se lógica ao pensar que a missão

do RASPCAR é identificar, em tempo real, a presença de obstáculos ao redor de si, de maneira que seja possível então executar o melhor movimento para a situação em que se encontra. Após isso, as três rodas são inseridas ao chassi, sendo duas delas acompanhadas de seus respectivos motores, e uma servindo apenas como apoio multidirecional, popularmente chamada de roda castor. Próxima aos motores, encontra-se a ponte H [Bóson 2017], que assim foi alocada para que seja possível sua conexão tanto com eles, quanto com o Raspberry pi [Raspberry 2020], que foi posicionado na superfície contrária da placa de papelão. Este é alimentado através de um dos reguladores de tensão conectado ao botão que permite ou bloqueia o uso das baterias, para que a tensão a percorrer o microcontrolador nunca seja alta o suficiente para danificá-lo, enquanto o outro regulador reproduz a mesma tarefa na alimentação da ponte H. Além destes itens, também constava no chassi um Arduino UNO [Arduino 2020], pois, no início do projeto, acreditava-se que este poderia ter determinada utilidade através da biblioteca firmata [Hoefs 2013]. Esta possibilidade foi descartada ao notar o atraso de comunicação entre os sensores e a unidade lógica.

A comunicação entre os componentes foi definida através do uso de jumpers (fios condutores) e estanho, que foi o material selecionado para a soldagem dos elementos metálicos presentes. Desta maneira, foram estabelecidos dois circuitos necessários para o essencial funcionamento do protótipo. O primeiro pode ser chamado de circuito dos motores (Figura 1), pois como o próprio nome diz, interliga o Raspberry a todos os outros materiais responsáveis pela movimentação do carrinho. Assim sendo composto por: Motores, microcontrolador, ponte H [Baú 2020] e um regulador de tensão conectado a um botão responsável por cessar a alimentação do robô ao fim do uso, para assim evitar o desperdício de carga das baterias integradas ao chassi. Enquanto o segundo (Figura 2) torna possível o desvio de obstáculos através da detecção destes pelos sensores ultrassônicos [Nakatani et al. 2014] que fornecem as informações de ambiente para a interpretação destas no microcontrolador.

4.4. Análise de funcionamento

Foram então realizados testes para verificar a integridade e o condicionamento das peças conectadas. Nestes testes, foi possível encontrar uma dissemelhança no funcionamento do motor. O lado direito, sob as mesmas configurações de potência, se sobressaía ao esquerdo, o que pode ocorrer devido ao desgaste causado pelo uso deste em outros trabalhos. A partir das análises, foi descoberto que a diferença de velocidades entre os dois componentes é de aproximadamente 60% , logo, para resolver o problema, esta razão percentual foi aplicada em todas as funções de movimento do robô em sua programação. Por exemplo: Sempre que ordenado para traçar um caminho em linha reta, o Raspberry enviava ao motor esquerdo uma potência 60% maior que a enviada ao direito, de modo que este aumento compense suas características assimétricas.

4.5. Programação

Dando continuidade ao desenvolvimento, a programação começa a ser efetuada. A linguagem escolhida para escrita do código, como dito anteriormente, foi Python, uma linguagem que apesar de ser considerada de alto nível, é de fácil aprendizagem e possui uma estrutura de sintaxe simples. Destacando-se como a melhor das alternativas disponíveis. Nos primeiros dias trabalhando com a ferramenta, já foi possível construir uma grande noção de suas características.

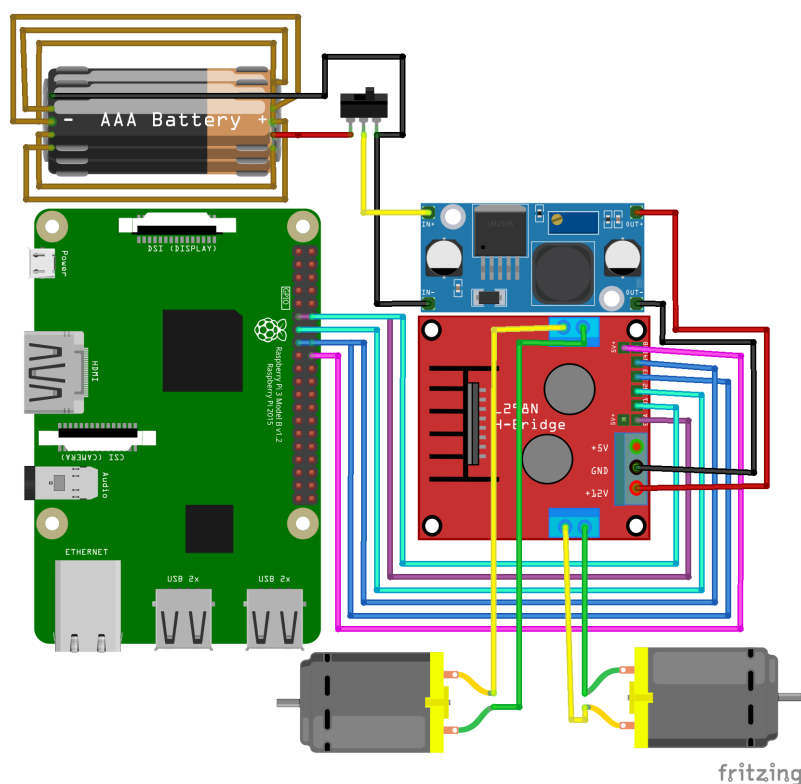


Figura 1. Circuito dos motores.

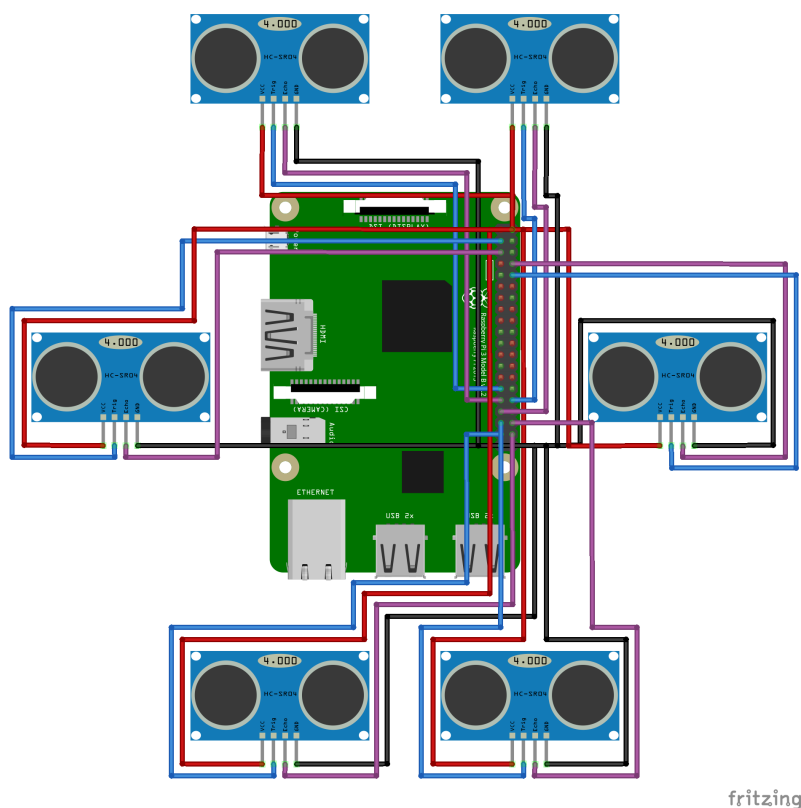


Figura 2. Circuito dos Sensores.

Os comandos para a realização de seus próximos movimentos são transmitidos através de uma conexão SSH (essencial para os objetivos do trabalho), que via rede wi-fi, transporta os dados do computador de programação para o protótipo RASPCAR, desta maneira, dando sentido ao "remotamente" do título do projeto. inicia-se então a fase lógica, que possui como chave para a eficiência do robô, em termos de código, a função "ult". Esta começa definindo as portas dos sensores a serem utilizadas, a taxa de amostragem, a velocidade do som e o "max_delta_t", que é o limite de tempo para o retorno da onda, calculado através da divisão da distância máxima pela velocidade do som.

```

1 def ult(a,b):
2     TRIG = a
3     ECHO = b
4     sampling_rate = 20.0
5     speed_of_sound = 349.10
6     max_distance = 4.0
7     max_delta_t = max_distance / speed_of_sound
8
9     GPIO.setup(TRIG, GPIO.OUT)
10    GPIO.setup(ECHO, GPIO.IN)
11
12    GPIO.output(TRIG, False)
13
14    print ("Sampling Rate:", sampling_rate, "Hz")
15    print ("Distances (cm)")
16
17    GPIO.output(TRIG, True)
18    time.sleep(0.00001)
19    GPIO.output(TRIG, False)
20
21    start_t=time.time()
22    tmp=start_t
23    while GPIO.input(ECHO) == 0 and time.time() - tmp < max_delta_t:
24        start_t = time.time()
25
26    end_t=time.time()
27    while GPIO.input(ECHO) == 1 and time.time() - start_t < max_delta_t:
28        :
29        end_t = time.time()
30
31    if end_t - start_t < max_delta_t:
32        delta_t = end_t - start_t
33        distance = 100*(0.5 * delta_t * speed_of_sound)
34    else:
35        distance = 30
36
37    print (round(distance, 2))
38    return distance

```

Então são escritos dois laços de repetição, designados pela palavra reservada "while". Um para descobrir o tempo em que se inicia emissão da onda ultrassônica e outro para saber quando ela retorna ao "echo". Logo abaixo, uma instrução condicional representada por "if" verifica se a diferença entre as duas informações coletadas não extrapola o limite anteriormente definido pela "max_delta_t", retornando o valor 66, que foi aleatoriamente escolhido para nos informar do erro, se isso acontecer. Se tudo estiver

dentro do esperado a diferença encontrada é multiplicada por 0,5. Pois não queremos saber a distância percorrida pela onda em sua ida e volta. Somente a ida nos interessa, pois ela se encerra onde o objeto se inicia. Então multiplica-se o valor encontrado pela velocidade do som e ao fim por 100, para que a amostra dos resultados seja feita em centímetros a partir do retorno da função.

Por último, as funções definidas no começo são finalmente utilizadas a partir de uma máquina de estados simples que analisa a presença dos obstáculos detectados pelo sensor ultrassônico no ambiente. Os estados integrantes desta são: SPAREDAO e SINICIAL. O carrinho começa no estado SINICIAL, que é onde ele se comportará normalmente até encontrar um objeto através dos dois sensores ultrassônicos frontais com mais de 20 cm de proximidade, assim entrando no estado SPAREDAO onde executará uma virada para a direita caso não exista nenhuma obstrução traseira, assim, quando os sensores deixarem de identificar o obstáculo anteriormente encontrado, o protótipo voltará ao SINICIAL. O robô então define sua movimentação, uma vez que FE e FD significam frente esquerda e frente direita e TE e TD traseira esquerda e traseira direita.

```

1 while True:
2     FE=ult(29,31)
3     FD=ult(32,33)
4     TE=ult(37,38)
5     TD=ult(35,36)
6
7
8     print("FE", (round(FE,1)), "FD", (round(FD,1)), "TE", (round(TE,1)), "
    TD", (round(TD,1)))
9
10    SPAREDAO=1
11    SINICIAL=0
12    estado = SINICIAL
13
14
15    if(estado == SINICIAL):
16        if(FE<20 and FD<20):
17            estado=SPAREDAO
18
19        else:
20            andar_frente()
21
22    elif(estado == SPAREDAO):
23        if(FE<20 and FD<20 and TE>20 and TD>20):
24            viradireita()
25        elif(FE<20 and FD<20 and TE<20 and TD<20):
26            parado()
27
28        else:
29            estado=SINICIAL

```

5. Resultados

Após programar o código base das funções do robô, começam os testes para tornar claro as capacidades atribuídas ao protótipo. Ao executar o código, percebemos que a diferença dos motores, compensada com a alteração de potencial entre eles, continua um pouco apa-

rente. Contudo, a partir da visualização da trajetória realizada pelo robô (Figura 3) podemos observar que a função de virar à direita no estado "SPAREDÃO" ainda afetada pela diferença de potência entre os motores, conseguiu realizar o desvio da parede (Obstáculo 4). Embora a curva não tenha atingido os 90 graus almejados, isso nos diz que mesmo com uma simetria imperfeita, o RASPCAR (Figura 4) se tornou capaz de cumprir seu dever. Podemos então considerar o percurso realizado como um resultado positivo, pois o carrinho o percorreu completamente sem colidir com qualquer objeto, o que é expressamente o objetivo principal do trabalho.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou o desenvolvimento do RASPCAR, o Protótipo Didático de Veículo Automotivo Robótico Autônomo, que foi desenvolvido através de microcontroladores e componentes eletrônicos. Fez-se necessário o uso de determinados conhecimentos em física [Biscuola et al. 2012] e programação lógica durante todo o período do projeto, para a construção de chassi, posicionamento de peças e controle do centro de gravidade do carro. Estes conceitos também possibilitaram a alimentação elétrica, escrita de código, análise e interpretação de resultados na movimentação do robô, técnicas de soldagem, manuseio de elementos eletrônicos. Além de proporcionar certa noção de arquitetura de circuitos, atrito entre materiais, elaboração de fórmulas e resolução de cálculos matemáticos, identificação de erros lógicos, entre outros.

Em vista disso pode-se concluir que é perfeitamente possível a criação de um protótipo de carro robótico autônomo guiado remotamente por meio de componentes de baixo custo e escala, tornando este trabalho o mais didático possível. Proporciona-se

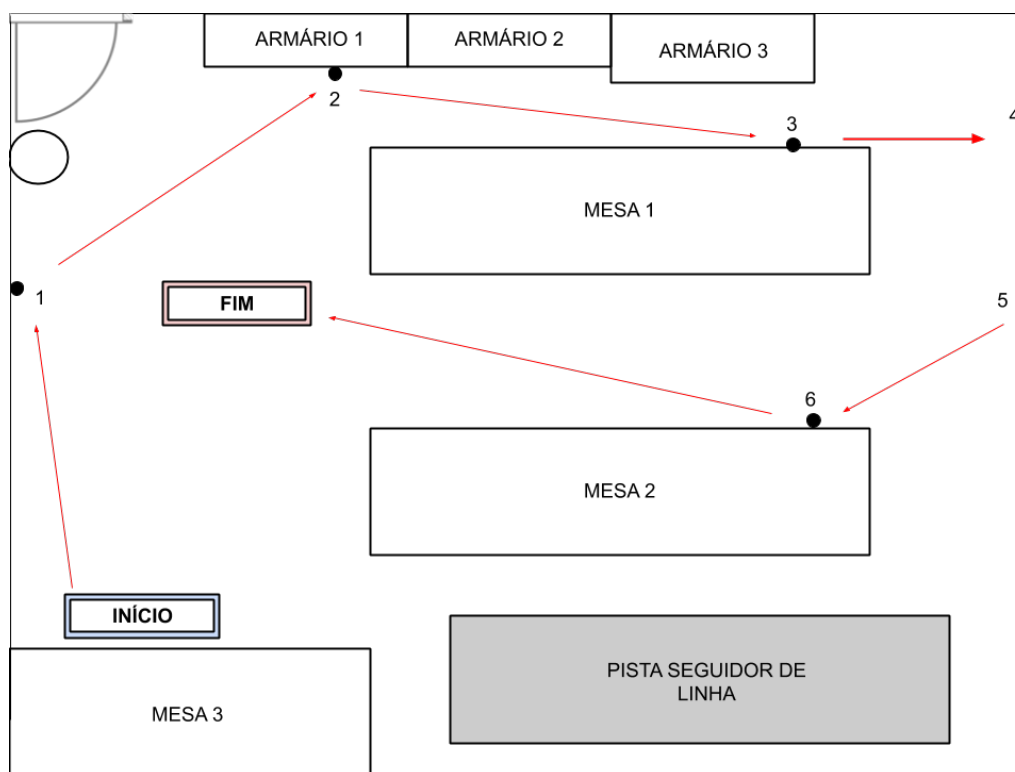


Figura 3. Trajetória de teste

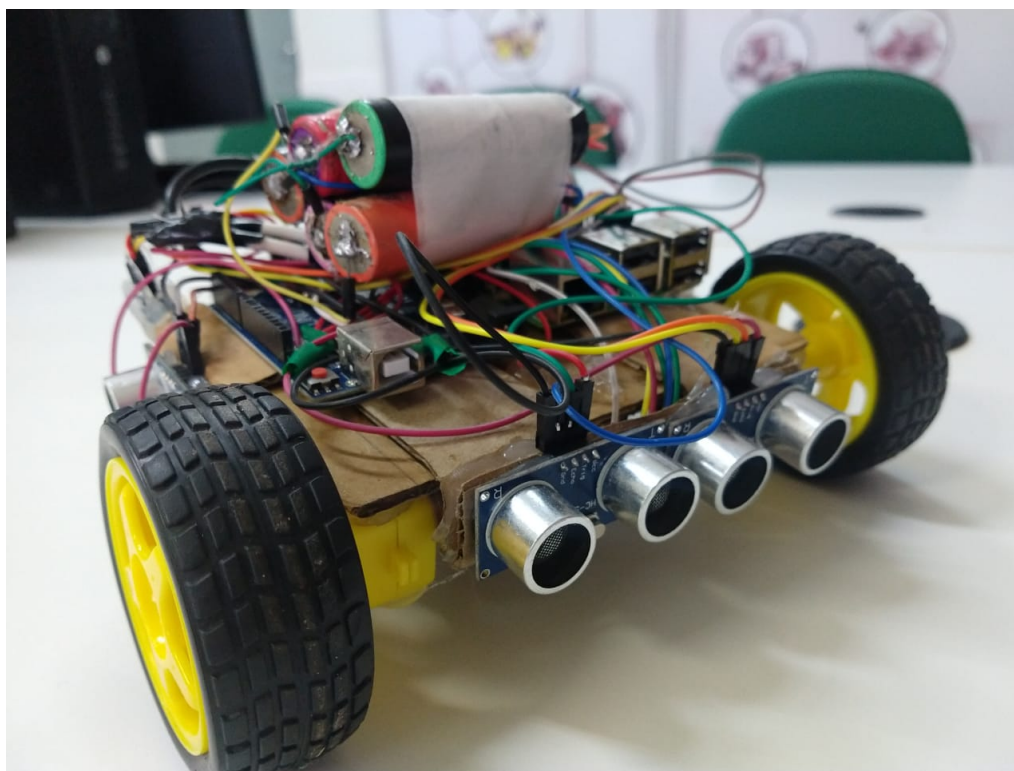


Figura 4. RASPCAR

então um aprendizado que incentiva a pesquisa e o desenvolvimento científico nesta área, através da experiência de trabalhar com a elaboração de projetos eletrônicos. Além de futuramente poder servir para nortear as ideias de novos pesquisadores.

Tendo em vista os atuais resultados, foram planejados aprimoramentos que podem ser futuramente realizados como continuação ao projeto RASPCAR. Por exemplo, tornar sua movimentação melhor, obtendo maior suavidade de movimentos e consequentemente atingindo uma determinada estabilidade de direção. Além disso, realizar pequenas alterações no código também estão nos planos, para tornar este mais organizado e adequado a seus propósitos. Por fim, podemos almejar a construção de um chassi em que o protótipo realmente adotaria a aparência de um veículo, com retrovisores, faróis e outros elementos estéticos para atingir enfim sua forma final como um projeto de carro robótico autônomo simples.

Agradecimentos

Os autores agradecem O CNPq e o IFPR pelo apoio em forma de bolsa de iniciação científica e auxílio financeiro para aquisição de insumos.

Referências

- Arduino (2020). ARDUINO. especificações técnicas do produto. <https://store.arduino.cc/usa/arduino-mega-2560-rev3>. Accessed: 2018-3-24.
- Baú (2020). Baú da eletrônica descrição do produto driver motor ponte-h - l298n. <http://www.baudaeletronica.com.br/driver-motor-ponte-h-l298n.html>. Accessed: 2018-8-24.

Biscuola, G., Boas, N., and Doca, R. (2012). *Topicos De Fisica, V.3 - 3o Ano: ELETRICIDADE, FISICA MODERNA, ANALISE DIMENSIONAL*. SARAIVA - DIDÁTICOS.

Bóson (2017). Como funciona uma Ponte H controle direcional de motores dc. <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/como-funciona-uma-ponte-h-controle-direcional-de-motores-dc/>. Accessed: 2018-8-26.

Bóson (2018). BÓSON TREINAMENTOS EM TECNOLOGIA usando um sensor ultrassônico de distância hc-sr04 com arduino. <http://www.bosontreinamentos.com.br/eletronica/arduino/usando-um-sensor-ultrassonico-hc-sr04-com-arduino/>. Accessed: 2019-10-20.

Hoefs, J. (2013). FIRMATA firmware for arduino. descrição do protocolo e firmware. <https://github.com/firmata/arduino>. Accessed: 2018-8-20.

Lima, L. T. G., Londero, D. J. M., Mantelli, M. F., and Azevedo, L. L. (2013). Robocar: protótipo de robô móvel autônomo. *Anais da V Escola Regional de Informática de Mato Grosso-2014*, page 81.

Nakatani, A. M., Guimarães, A. V., and Neto, V. M. (2014). Medição com sensor ultrassônico hc-sr04. *Departamento de Eletrônica, Curitiba, Brazil*.

Nauman, M. (2018). Drivingmatter: An autonomous rc car using raspberry pi.

Oltean, S.-E. (2019). Mobile robot platform with arduino uno and raspberry pi for autonomous navigation. *Procedia Manufacturing*, 32:572–577.

Ondruš, J., Kolla, E., Vertal', P., and Šarić, Ž. (2020). How do autonomous cars work? *Transportation Research Procedia*, 44:226–233.

Raspberry (2020). RASPBERRY PI especificações técnicas do produto. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Accessed: 2018-3-3.

SAE (2016.). Standard SAE J3016 Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.

Valocky, F., Orgon, M., and Fujdiak, I. (2019). Experimental autonomous car model with safety sensor in wireless network. *IFAC-PapersOnLine*, 52(27):92–97.

Wei, D. C. M. (2015). *Método de desvio de obstáculos aplicado em veículo autônomo*. PhD thesis, Universidade de São Paulo.