



FACULDADE DE TECNOLOGIA SENAI CIMATEC
CURSO DE ESPECIALIZAÇÃO EM CONTROLE E
AUTOMAÇÃO

MAYCON DOUGLAS DA SILVA CARVALHO

ROBÔ AUTÔNOMO
PARA RESOLUÇÃO DE LABIRINTOS BASEADO EM
PLATAFORMA ABERTA

Salvador
2014

MAYCON DOUGLAS DA SILVA CARVALHO

**ROBÔ AUTÔNOMO PARA RESOLUÇÃO DE LABIRINTOS BASEADO EM
PLATAFORMA ABERTA**

Trabalho de conclusão de curso apresentado ao Curso de Especialização em Controle e Automação da Faculdade de Tecnologia SENAI CIMATEC como requisito final para a obtenção do título de especialista.

Orientador: Prof. Doutor Herman Augusto Lepikson.

Salvador

Ficha catalográfica elaborada pela Biblioteca da Faculdade de Tecnologia SENAI CIMATEC

C331r Carvalho, Maycon Douglas da Silva

Robô autônomo para resolução de labirintos baseado em plataforma aberta / Maycon Douglas da Silva Carvalho – Salvador, 2014.

78 f.: il. color.

Orientador: Prof. Dr. Herman Augusto Lepikson.

Monografia (Especialização em Automação, Controle e Robótica) – Programa de Pós-Graduação, Faculdade de Tecnologia SENAI CIMATEC, Salvador, 2014.

Inclui referências.

1. Robótica móvel. 2. Integração de subsistemas. 3. Exploração de labirinto. I. Faculdade de Tecnologia SENAI CIMATEC. II. Lepikson, Herman Augusto. III. Título.

CDD 629.892

MAYCON DOUGLAS DA SILVA CARVALHO

**ROBÔ AUTÔNOMO PARA RESOLUÇÃO DE LABIRINTOS BASEADO EM
PLATAFORMA ABERTA**

Projeto Final de Curso aprovado com nota 8,0
como requisito de Especialista em Automação
e Controle, tendo sido julgado pela Banca
Examinadora formada pelos Professores:

Dr. Herman Augusto Lepikson – Orientador

Faculdade de Tecnologia SENAI CIMATEC

Msc. Oberdan Rocha Pinheiro – Professor

Faculdade de Tecnologia SENAI CIMATEC

Salvador, 27 de junho de 2014.

AGRADECIMENTOS

Agradeço primeiramente ao Professor Dr. Herman Lepikson pela orientação e solicitude sempre que procurado, além de sua compreensão e atenção.

Agradeço igualmente a Geovane Mimoso pelas dicas e suportes durante os longos e difíceis meses de trabalho, estendendo sua mão amiga sempre que necessário.

Também a Débora Short por horas ao meu lado tentando corrigir falhas, operar testes e melhorar os resultados, demonstrando sua grande amizade e generosidade, como sempre.

A minha família, que sempre desempenhou papel preponderante em minha vida, dando-me suporte e entusiasmo quando este era escasso.

E a todos que direta ou indiretamente auxiliaram-me na jornada em busca da completude deste trabalho. Serei sempre grato.

Impossível é apenas uma grande palavra usada por gente fraca que prefere viver no mundo como está em vez de usar o poder que tem para mudá-lo. Impossível não é um fato, é uma opinião. Impossível não é uma declaração, é um desafio. Impossível é hipotético. Impossível é temporário.

Muhammad Ali

RESUMO

A robótica móvel representa um campo em amplo desenvolvimento científico no contexto atual. As aplicações de robôs com capacidade locomotora são crescentes e ganham espaço definitivo em diversas atividades humanas. Sob esta perspectiva, compreender e explorar tecnologias que compõem um robô móvel são práticas recomendáveis aos interessados em se aprofundar no campo. No presente trabalho é proposto um modelo simplificado de robô móvel que apresenta a capacidade de explorar um ambiente parcialmente desconhecido como um labirinto, e de forma bem sucedida encontrar seu término. O protótipo do robô é composto por três subsistemas básicos, responsáveis pelo sensoriamento e percepção do meio físico circundante, processamento das informações e a atuação mecânica a fim de produzir movimento. A composição dos três citados gera um sistema com as mínimas especificações para a completude da tarefa definida. A integração dos componentes em forma de um dispositivo complexo é um dos desafios propostos no trabalho uma vez que os sistemas, ao interagirem, apresentam comportamento variante ao esperado e, formam portanto uma plataforma de aprendizagem prática rica.

Palavra-chave: Robótica Móvel, Integração de Subsistemas, Exploração de Labirinto.

ABSTRACT

Mobile robotics now represents a field under wide scientific development. Applications for robots which are able to move around increase consistently and gain definitive ground in several human activities. Under such perspective, grasping a deeper understanding and exploring technologies which compose a mobile robot are highly recommended practices to those who nourish interests in the field. This work addresses a simplified model for a mobile robot that detains the capacity of navigating a partially known environment like a maze, and successfully find its way out. The prototype is made of three distinct basic subsystems, liable for sensing and perceiving the surrounding physical medium, processing data and actuating over mechanical parts in order to produce motion. The assembling of the aforementioned generate a system with minimum specifications for the completion of the assigned task. The integration of the components to form a complex device is one of the challenges faced in the current project since the interaction with each other adds several unexpected behaviors to the prototype, turning it into a rich learning practical platform.

Keywords: Mobile Robotics, Subsystems Integration, Maze Exploration.

LISTA DE ILUSTRAÇÕES

Figura 1: Esquema cíclico das etapas necessárias para a realização de um sistema robótico móvel.....	21
Figura 2: Representação da posição a partir de bases de coordenadas e angulação entre elas.....	25
Figura 3: Diagrama de abrangência dos sistemas especialistas dentro do universo da inteligência artificial.....	34
Figura 4: Exemplo de <i>labyrinth</i>	36
Figura 5: Exemplo de maze.....	36
Figura 6: Protótipo montado.....	44
Figura 7: Arduino Mega 2560.....	47
Figura 8: Sensor ultrassônico HC-SR04.....	48
Figura 9: Esquema do L293D.....	49
Figura 10: Motores e caixa de redução.....	50
Figura 11: Rodas com pneumáticos.....	50
Figura 12: Roda giratória.....	50
Figura 13: <i>Protoboard</i> de 400 pinos.....	51
Figura 14: <i>Jumpers</i> macho-macho.....	51
Figura 15: Plugue de alimentação para Arduino.....	52
Figura 16: Bateria LiPo 3S 1450 mAh.....	52
Figura 17: Chassi montado com rodas e motores.....	53
Figura 18: Modalidades de PWM aplicáveis à plataforma Arduino.....	57
Figura 19: Conformação dos sensores em 45 graus.....	63
Figura 20: Labirinto da simulação.....	66
Figura 21: Robô dentro da faixa de ação frontal com mensagem “Seguir em frente”.....	67
Figura 22: Robô quando muito próximo à lateral esquerda imprimindo “Ajustar à direita.”.....	67
Figura 23: Robô em uma curva para a direita imprimindo a mensagem “Virar à direita”.....	68
Figura 24: Robô em uma obstrução total, imprimindo a mensagem “Girar 180 graus”.....	68
Foto 25: Robô no fim do labirinto, imprimindo a mensagem “Fim do labirinto”.....	69

SUMÁRIO

1 INTRODUÇÃO.....	16
1.1 MOTIVAÇÃO.....	16
1.2 OBJETIVOS.....	17
1.3 METODOLOGIA.....	18
2 FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 ROBÓTICA MÓVEL	20
2.1.1. Locomoção.....	22
2.1.2 Cinemática e localização.....	23
2.1.3 Algumas técnicas de localização.....	26
2.1.4 Percepção e sensores.....	27
2.1.5 Navegação e planejamento.....	28
2.1.6 Planejamento de rotas.....	29
2.1.7 Mapeamento.....	30
2.1.8 Exploração de ambientes	31
2.2 INTELIGÊNCIA ARTIFICIAL.....	32
2.2.1 Sistemas Especialistas.....	33
2.3 ROBÔ APLICADO À RESOLUÇÃO DE LABIRINTOS.....	35
2.3.1 Micromouse.....	37
2.3.2 Técnicas de resolução de labirintos.....	38
2.3.3 O robô e subsistemas.....	39
3 PROTÓTIPO E METODOLOGIA.....	42
3.1 METODOLOGIA LEAN STARTUP.....	42
3.1.1 Mínimo Produto Viável.....	43
3.2 O PROTÓTIPO.....	44
3.2.1 Os subsistemas.....	45
3.2.2 Percepção.....	45
3.2.3 Controle.....	46
3.2.4 Movimentação.....	46
3.3 COMPONENTES.....	47
3.3.1 Arduino.....	47
3.3.2 Sensores Ultrassônicos.....	48
3.3.3 Driver: CI L293D.....	49

3.3.4 Motores e Rodas.....	49
3.3.5 Prototipagem e Circuito Impresso.....	51
3.3.6 Cabos de conexão e Plugue.....	51
3.3.7 Bateria LiPo.....	52
3.3.8 Chassi.....	53
3.4 ORÇAMENTO DE AQUISIÇÃO.....	53
4 A MISSÃO.....	55
4.1 O CONTROLE.....	55
4.2 CONTROLE DOS MOTORES ATRAVÉS DO PWM.....	56
4.3 A ESTRATÉGIA DA MÃO ESQUERDA.....	58
4.4 O ALGORITMO.....	59
5 TESTES E SIMULAÇÃO.....	62
5.1 MODIFICAÇÕES NO PROTÓTIPO.....	62
5.2 TESTES DE MANOBRAS E DIRIGIBILIDADE.....	64
5.3 SIMULAÇÃO DO LABIRINTO.....	65
6 CONCLUSÃO.....	70
6.1 MELHORAMENTOS E VERSÕES FUTURAS.....	71
7 REFERÊNCIAS.....	72
APÊNDICE.....	74

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

A robótica móvel vem crescentemente ganhando espaço e aplicações no cotidiano das atividades humanas. Além disso, a autonomia de tais robôs é o que mais os diferem da robótica convencional. De carros autônomos a máquinas que fazem limpeza doméstica, a demanda por inovação no segmento é algo evidente.

Com o intuito de adentrar o universo da robótica móvel, prezando inicialmente pelo caráter didático, mas também atentando para o grau de integração de sistemas que poderia ser alcançado ao decorrer do desenvolvimento do trabalho, foi então proposta a confecção de um sistema robótico terrestre autônomo que tem a capacidade de explorar um labirinto.

O projeto integra três sistemas pilares de quaisquer aplicações robóticas que são: uma plataforma mecânica (chassis, motores, engrenagens, entre outros), um sistema eletrônico de controle onde é então embarcada a “autonomia” do produto final através de algoritmos especializados e um sistema sensor, que possibilita a captação de sinais oriundos do meio de operação, que são processados a fim de fornecer dados ao sistema de controle.

Dessa forma, a integração dos três sistemas garantirá, ao término da jornada, a consolidação de conceitos práticos de implementação, aperfeiçoamento de conceitos teóricos e a vivência de um projeto real de engenharia que engloba também aspectos não técnicos como planejamento, prazos, orçamentos e aquisição de recursos.

Por fim, a empreitada na produção deste presente trabalho também permitirá a elaboração de um sistema especialista capaz de simular a inteligência humana, mesmo que de forma limitada, para a resolução do problema em questão. Apesar de não ser uma tarefa corriqueira na vida de um ser humano, a exploração de um labirinto pode ser transplantada para diversas outras situações, como exemplo o mapeamento e reconhecimento de um ambiente desconhecido e perigoso à presença humana, em busca de algum objeto ou objetivo, garantindo a segurança, otimização e qualidade para a tarefa.

1.2 OBJETIVOS

O presente trabalho tem como objetivos principais a elaboração, criação, implementação e validação de um sistema robótico terrestre que tenha a autonomia necessária para a exploração de um labirinto simples (sem a presença de *loops*). O sistema robótico é baseado na metodologia de Mínimo Produto Viável, que se baseia na minimização dos custos, tempo de fabricação e montagem e busca também minimizar os intervalos entre testes e verificações do protótipo com o objetivo de validar ou descartar soluções o quanto antes.

Um objetivo preliminar é o levantamento bibliográfico que endosse a desenvolvimento seguinte dos subsistemas necessários, de forma objetiva e clara. Registros *online* e publicações acadêmicas são a fonte básica de informação para o projeto. É sabida da vasta exploração da temática, onde diversas estratégias e práticas foram desenvolvidas e executadas, geralmente em meios virtuais e simulados. Portanto, traduzir algumas delas em um meio físico, pondo-as à prova das limitações dos sistemas reais, será um grande diferencial do projeto.

Em seguida, outro objetivo importante ao projeto é a realização de um sistema físico composto de uma plataforma mecânica responsável pela movimentação do conjunto e um sistema de controle constituído de componentes e subsistemas eletrônicos que compatibilizarão e integralizarão o sistema robótico de forma unificada.

Como complemento à estrutura física do robô, segue-se a necessidade de criação algoritmo proprietário que garanta ao sistema a capacidade de completar a tarefa especializada a ele atribuída. A linguagem C é oportunamente escolhida, com as nuances próprias ao controlador Arduino.

Por fim, ao se constituir um sistema satisfatoriamente operante, é objetivada a redação de um relatório em texto extensivo com a finalidade de registrar o trabalho mas, acima de tudo, fornecer as diretrizes para estudantes, *hobbyistas* ou profissionais que buscam conhecimento nas temáticas principais ou secundárias relacionadas ao tema.

1.3 METODOLOGIA

A metodologia assumida baseia-se na técnica enxuta, que prega a elaboração do Mínimo Produto Viável (RIES, 2011) que constitui a forma mais rudimentar alcançável que conecte o eixo necessidade (problema) solução. Uma das principais premissas da metodologia é de lançar o produto (ou serviço) o mais breve possível, e aperfeiçoá-lo de forma frequente a ponto de entender a fundo a aplicação, na medida em que a mesma é melhorada em sua própria utilização (do inglês *release early, release often*).

Em minúcias, o primeiro passo para a elaboração do produto pretendido será um levantamento do que já vem sendo desenvolvido na esfera acadêmica ou mesmo no mercado. Isso norteará o desenvolvimento acelerado sem necessariamente se preocupar com o conhecimento elementar já desenvolvido por outrem.

Ao término da prospecção de conteúdo e bibliografia serão então planejadas as etapas específicas da construção do protótipo e seus testes.

De forma genérica, passa-se à seleção de componentes eletrônicos e eletromecânicos que comporão a estrutura física do robô. O primeiro aspecto a ser considerado são as especificações de engenharia que cada componente deve apresentar. Segue, um levantamento dos componentes disponíveis no mercado local para que as primeiras implementações e testes possam ocorrer sem a dependência de prazos de entrega de produtos de mercados externos.

Uma vez adquiridos os componentes físicos do protótipo os mesmos serão montados e testados em suas funções básicas uma vez que, por se tratar de um MVP (do inglês *Minimum Viable Product* ou Mínimo Produto Viável) às atualizações devem acontecer no produto final e o mesmo deve apresentar alta confiabilidade do sistema físico para os ajustes apenas ocorrer em nível de código. O intuito da utilização da metodologia MVP é garantir uma abordagem de engenharia de produto sofisticada ao desenvolvimento do protótipo visando menores custos e tempo de validação.

Uma vez garantida a operabilidade do sistema físico, serão então embarcados os códigos que garantem controle e dirigibilidade ao veículo para o que o mesmo se comporte de forma desejada em situações corriqueiras em sua operação. Em seguida, uma vez apresentando bom desempenho, serão desenvolvidos e implementados os algoritmos que dão inteligência ao sistema no processo de tomada de decisão.

Finalmente, ao se concluir as etapas mencionadas anteriormente, será gravada a operação

do sistema em um labirinto, o que facilitará a exposição dos resultados uma vez que as dimensões espaciais da maquete de labirinto impede o seu traslado com facilidade para demonstrações.

Em posse dos resultados, os mesmos serão discutidos em forma de texto extensivo para o registro e depósito como componente avaliativo final da especialização em Controle e Automação da Faculdade de Tecnologia SENAI CIMATEC.

2 FUNDAMENTAÇÃO TEÓRICA

A presente seção tem como foco o levantamento de informações de trabalhos e publicações de autores diversos que contribuam para a construção de um protótipo mais embasado e justificado.

2.1 ROBÓTICA MÓVEL

A robótica em si não é um campo novo. Ela já se apresenta consolidada em alguns campos industriais que devem seu amplo desenvolvimento à utilização de robôs manipuladores que executam tarefas repetitivas, enfadonhas, perigosas ou que mesmo ultrapassam a capacidade humana. Os manipuladores são, atualmente, o principal elo entre a robótica como ciência pura e os campos de aplicação.

Entretanto, apesar de seu papel decisivo na propulsão de diversos ramos industriais, os manipuladores são sistemas robóticos que apresentam grandes limitações no que diz respeito à mobilidade e, portanto à autonomia. Os “ombros” dos manipuladores são fixados em plataformas proprietárias à aplicação de destino, e garante a esses robôs uma angulação definida, poucos graus de liberdade e baixa ou nula capacidade de translação.

Nesse contexto se insere a robótica móvel como um campo de ruptura da robótica convencional desenvolvida até o presente momento. A robótica móvel, apesar do nome sugestivo, excede os limites do simples trato da mobilidade, pois engloba disciplinas como processamento de sinais, lógica de programação, inteligência artificial e mesmo campos de ciências sociais, entre outros.

Robôs móveis são aplicações novas dos conhecimentos elementares produzidos através de outros campos da robótica. Eles podem ser simplificada e resumidos a três subsistemas principais presentes em todo robô móvel dotado de autonomia. O primeiro é o sistema de movimentação, que compreende os processos cinemáticos e de locomoção, a propulsão e o controle dos movimentos. Esse subsistema é munido de informações provenientes do meio externo através do subsistema de percepção composto por sensores e o processamento de sinais adquiridos. Por fim, o subsistema de autonomia da navegação que armazena em

códigos algorítmicos a inteligência necessária para a completude da missão designada ao robô (SIEGWART & NOURBAKHS, 2004).

A Figura 1 apresenta um esquema visual (conhecido como *see-think-act*) para a compreensão do relacionamento cíclico que encerra os três subsistemas mencionados anteriormente. O esquema abaixo apresenta de forma simplificada os estágios vitais dentro do funcionamento de um robô móvel. O ciclo se estabelece pela forma com que o processo é iterativo e se retroalimenta de forma constante com o meio externo (Figura 1).

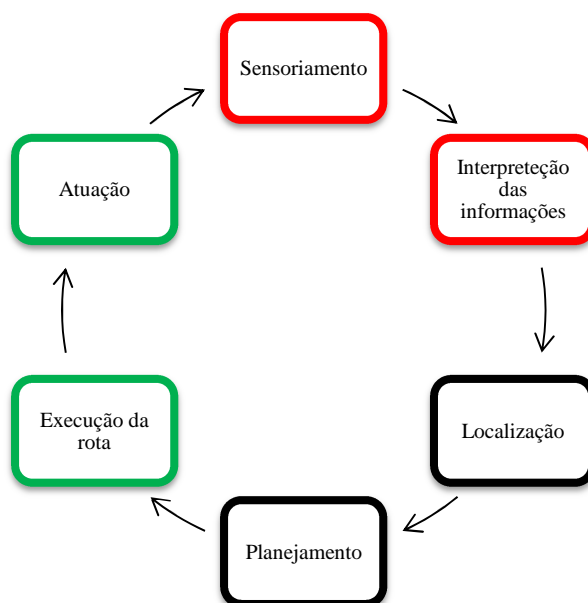


Figura 1: Esquema cíclico das etapas necessárias para a realização de um sistema robótico móvel. Os blocos em vermelho representam o subsistema de percepção, em preto a autonomia e em verde a moção.
Fonte: Adaptado de (SIEGWART & NOURBAKHS, 2004)

A capacidade de desenvolvimento da robótica móvel é crescentemente visível. Há poucos anos o seu desenvolvimento se restringia a ambientes laboratoriais ou aos *hobbyistas* que por ventura desejassem desenvolver sistemas para aplicações pouco comerciais. Nos dias atuais, com a viabilização de sistemas computacionais poderosos e ampliação da aplicação das técnicas de inteligência artificial que aproximam o processamento das informações nos ditos sistemas à cognição humana, tarefas antes insalubres ou mesmo não imaginadas para seres humanos (exploração da superfície de Marte, por exemplo) se tornam possíveis.

Nas próximas seções do presente trabalho serão tratados de forma particular os componentes dos subsistemas acima citados. Os tópicos varrerão a locomoção e cinemática, seguindo pela percepção do ambiente e conseqüente localização, sendo finalizados pelos

relacionados a planejamento e navegação.

2.1.1. Locomoção

Quando se parte para a implementação de um sistema robótico móvel é necessário inicialmente pensar a forma com que a mobilidade será executada. Uma forma prática de entender essa tarefa é observar como os sistemas biológicos naturais se movem em ambientes diversos existentes. A replicação de tais conceitos sob a ótica de engenharia é conhecida como biônica¹.

Nas mais diversas frentes de desenvolvimento de robôs móveis são utilizadas sofisticadas e complexas modalidades de locomoção, sendo possível encontrar robôs que andam, rolam, pulam, escorregam, patinam, nadam, voam e rolam (SIEGWART & NOURBAKSHI, 2004). Todas essas formas são baseadas nas observações dos correspondentes naturais, mas atentando para as limitações de engenharia.

Os sistemas biológicos podem agregar complexidade suficiente aos mecanismos para que esses sejam capazes de suportar torção, pressão, torque, entre outros esforços, sem necessariamente comprometer a integridade da estrutura em si. Isso advém da capacidade celular em miniaturizar e especializar competências a ponto de organismos com pequenas dimensões apresentar grau de complexidade imensamente superior aos sistemas fabricados pelos seres humanos.

Nesse tocante, a limitação primordial na seleção e execução do modo de locomoção é atribuída à engenharia. Os componentes manufaturados não atendem as especificidades das suas contrapartes replicadas. Desse modo, hoje se tem primordialmente dois segmentos amplamente utilizados nas frentes de pesquisa e desenvolvimento da robótica móvel: os robôs articulados e os com roda.

Os robôs articulados têm grande vantagem em ambientes naturais, sem regularidade, com variação de relevo e repleto de obstáculos de formas irregulares. Existem exemplares que apresentam de uma a algumas dezenas de pernas, que agrava consistentemente a dificuldade de controle e estabilidade. Por outro lado, a adição de articulações amplia a liberdade de movimento dos seus portadores.

¹ Ciência que visa estudar, compreender e replicar processos biológicos dos seres vivos com o intuito de desenvolver sistemas técnicos aplicáveis a problemas artificiais ou industriais.

Os robôs estruturados sobre rodas, por outro lado, apresentam grandes desvantagens quando comparados aos citados acima quando se tratam de terrenos irregulares. Mas deve-se levar em consideração que a maior parte dos ambientes construídos pelo homem apresenta as características de lisura, suavidade vertical e pouca ou nenhuma irregularidade. Nesse caso específico os robôs com rodas deslancham em termos de eficiência e atingem os objetivos de sua aplicação com maior facilidade e agilidade (SIEGWART & NOURBAKHS, 2004).

Quando se pensa em robôs sobre rodas vem inicialmente em mente o aspecto estabilidade. Um robô que apresenta no mínimo três rodas distribuídas uniformemente sob o chassi encontra-se estável. É possível construir robôs com duas rodas e o mesmo encontrar estabilidade, mas ajustes de controle devem ser levados em consideração a fim de obter uma margem de estabilidade suficiente para as condições de operação planejadas. Já os veículos com mais de três rodas devem necessariamente apresentar sistema de suspensão que visa manter todas as rodas no chão ao mesmo tempo.

Outros dois aspectos presentes em sistemas com roda são a controlabilidade e manobrabilidade. Os termos representam respectivamente a capacidade de controle e de execução de manobras que o sistema tem sobre os atuadores de forma geral. As duas grandezas se relacionam de forma inversa, ou seja, à medida que se acrescenta em uma tem-se necessariamente perda por parte da outra. Deve haver ponderação no desenho do projeto sempre se referindo às necessidades técnicas e funcionais do mecanismo.

Desta forma, a seleção do sistema de locomoção deve levar em consideração fatores intrínsecos e extrínsecos ao projeto, sendo que o ambiente tem importância fundamental nessa etapa.

2.1.2 Cinemática e localização

A cinemática faz parte do estudo elementar de sistemas mecânicos em geral. Para robôs móveis não seria diferente, apesar de geralmente envolverem graus mais elevados de liberdade. A compreensão precisa do comportamento cinemático de um robô tem impacto direto em seu controle da mesma forma que o desenho mecatrônico demanda um amplo conhecimento de tal fator.

Claramente existem variadas abordagens dentro do campo da robótica móvel que

atendem aos também diversos graus de liberdade. Existem robôs simples, compostos por um sistema diferencial de controle das rodas que provê todas as formas de movimento que é capaz de executar. Outros, por outro lado, usam técnicas apuradas de controle clássico, ou mesmo, em casos menos frequentes, de controle avançado para garantir as demandas mais sofisticadas de movimento.

Além do controle, as manobras devem ser inseridas no projeto uma vez que atributos como centro de gravidade, massa e aderência são cruciais para a efetuação satisfatória dos movimentos planejados. Entretanto o grande desafio que envolve a cinemática de um robô móvel é a estimativa da posição deste. Robôs com estruturas fixadas em algum ponto referencial como os manipuladores não carecem desta referência, uma vez que um conjunto dedicado de sensores e o estudo dos movimentos e localização das junções fornecem informações precisas em confiáveis nesse respeito (SIEGWART & NOURBAKHS, 2004).

Já os sistemas móveis apresentam uma gama consideravelmente maior de possibilidades de localização e não há formas diretas de aferir instantaneamente a posição do robô. Um procedimento a ser executado para tal finalidade é a integração do movimento vetorial no tempo através da técnica denominada *dead reckoning*². O grande problema de tal abordagem são as imprecisões geradas pelo próprio movimento, a exemplo de derrapagens e deslizamentos da roda na superfície, o que faz os sensores fornecerem informações irreais ao sistema de processamento (SIEGWART & NOURBAKHS, 2004).

O estudo cinemático do robô como um todo advém essencialmente do estudo individualizado de cada roda. Os movimentos compostos desempenhados pelo sistema robótico são a combinação do movimento de cada roda objetivando uma dada manobra. Por outro lado, as rodas também impõem limitações particulares aos movimentos do sistema total, uma vez que se uma das rodas não estiver preparada ou desenhada para uma dada modalidade de movimento, as demais terão maiores dificuldades em realizá-la.

A fim de estudar a cinemática de um dado robô é necessário referencia-lo ao seu campo de operação. No caso simplificado para exemplificação, o campo tem sua origem selecionada de forma arbitrária que dá origem, como representado na Figura 2, pela base inercial $\{X_I, Y_I\}$. O robô é considerado um corpo rígido limitado ao seu chassi, ignorando, por vias de simplificação, os graus de liberdade internos ao robô e suas rodas.

O ponto P (ou (x, y)) representa a posição do robô em relação aos eixos referenciais

² Processo pelo qual se estima a posição final de um objeto a partir de um ponto inicial fixado, através da integração das velocidades vetoriais em função do tempo.

arbitrários delimitados a partir o campo de operação, que dá origem à base $\{X_R, Y_R\}$ construída em referência às dimensões longitudinal e latitudinal do robô. Uma última variável é o ângulo θ , que representa a diferença angular entre a base global (campo de operação) e a local (robô).

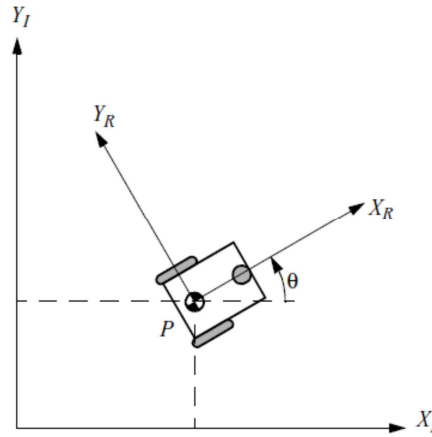


Figura 2: Representação da posição a partir de bases de coordenadas e angulação entre elas
Fonte: (SIEGWART & NOURBAKHS, 2004)

A posição de um robô pode ser descrita a partir dessas três variáveis, organizadas em forma de matriz como segue na Equação 1:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1)$$

Com o intuito de determinar o comportamento cinemático do sistema em referência ao referencial local estabelecido a partir de sua posição é necessário mapear a função da posição atual do robô, o que é possível através da Matriz de Rotação Ortogonal (Equação 2) (SIEGWART & NOURBAKHS, 2004).

$$R(\theta) = \begin{bmatrix} \cos \theta & \text{sen } \theta & 0 \\ -\text{sen } \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

De posse dessa equação é então possível determinar a cinemática do robô em relação a ele próprio como descrito pela Equação 3.

$$\dot{\xi}_R = R(\theta) \cdot \dot{\xi}_I \quad (3)$$

A partir das equações definidas acima é possível estimar a posição momentânea do robô a partir das técnicas de integração comentada nessa seção.

2.1.3 Algumas técnicas de localização

De forma simplificada, como já ilustrado na seção anterior, a localização de um robô consiste do processo de determinação de sua posição e orientação espacial (GONZÁLEZ, RODRÍGUEZ, & GÚZMAN) e é determinante para outros processos embarcados no robô como mapeamento e roteirização, assim como outros.

Como também mencionado na seção anterior, o *deadreckoning* é uma das técnicas mais difundidas dentro as aplicações de robótica móvel, sendo ele simplesmente a integração da velocidade vetorial do robô em função do tempo de navegação.

A odometria se faz uma alternativa também grandemente aceita no meio científico pelas vantagens evidentes que oferece como custo baixo de implementação, boa precisão, entre outras. O conceito baseia-se na integração do movimento das rodas de forma consecutiva, o que inevitavelmente acumula erros como um dos reveses da utilização da técnica, mas que para aplicações mais triviais são aceitáveis em contrapartida das vantagens oferecidas.

A odometria e *deadreckoning* são técnicas consideradas locais ou relativas. (GONZÁLEZ, RODRÍGUEZ, & GÚZMAN). Essas técnicas são, de forma genérica, consistem do incremento da posição e orientação do robô a partir do ponto de partida. Os dados necessários são obtidos através de sensores como *enconders*, giroscópios, acelerômetros, dentre outros.

Além das técnicas supracitadas, outras abordagens vêm ganhando espaço dentro da pesquisa em robótica. As técnicas conhecidas como globais ou absolutas determinam a posição do robô em função de uma referencia global, através da utilização de marcos ou *beacons* (GONZÁLEZ, RODRÍGUEZ, & GÚZMAN).

Um grande exemplo é a localização por GPS que se baseia na informação fornecida por satélites que possibilita a determinação das coordenadas espaciais do objeto (altitude, longitude e latitude). Consiste de uma tecnologia amplamente desenvolvida e acessível, barata

e comercial. Além disto, a grande vantagem atribuída a esta modalidade é que é possível a diminuição consistente dos erros caso medições de realimentação foram possíveis.

Outra forma também considerada local é a determinação de marcos e *beacons* em uma determinada área de interesse, onde o robô atuará. Assim o robô consegue triangular informações de posição em tempo real, possibilitando sua localização em referência ao espaço de trabalho.

De forma geral, as técnicas tidas como globais oferecem vantagens como a não dependência da posição e orientação inicial do robô, assim como mitiga erros sistemáticos de forma considerável. Entretanto, em comparação às técnicas locais, são relativamente mais caras e de maior dificuldade de implementação.

2.1.4 Percepção e sensores

Uma das mais importantes tarefas de um sistema autônomo é adquirir conhecimento sobre o ambiente em que se encontra. Isto é feito a partir das medições tomadas usando vários sensores e então extraindo informação significativa dessas medições (SIEGWART & NOURBAKSH, 2004).

Os sensores associados a um sistema robótico podem ter como funcionalidade perceber o ambiente que circunda o robô ou mesmo obter informações referentes ao estado do próprio robô. Grandezas como temperatura interna, rotação de motores, corrente demandada, entre outras, são geralmente as mais evidenciadas quando se busca entender e monitorar o funcionamento interno.

Entretanto, uma vez que os sistemas robóticos móveis se deslocam em ambientes desconhecidos que apresentam obstáculos e irregularidades, a percepção dos espaços de operação são etapas críticas e fundamentais para a construção de um sistema autônomo.

Os sensores, de forma geral, são classificados em duas frentes (SIEGWART & NOURBAKSH, 2004). Uma primeira faz referência a internalidade ou externalidade da grandeza a ser medida. Os sensores *proprioceptivos* medem grandezas internas ao sistema com a finalidade de acompanhar o seu desempenho e conformidade ao passo em que a operação se desenvolve. Os *exteroceptivos* são destinados ao reconhecimento externo, ou seja, compreender as características do ambiente a fim e tomar decisões e atuar sobre este

(SIEGWART & NOURBAKHS, 2004).

Uma segunda frente refere-se ao aspecto energético. São classificados os sensores em passivos e ativos. Essa denominação se relaciona ao método de sensoriamento. No caso em que a grandeza a ser medida adentra o sensor e desencadeia processos que são interpretados, fornecendo uma leitura desta os sensores se enquadram como passivos. Já no caso dos ativos, os sensores emitem energia para o meio e através da interação desta com determinadas características do ambiente é possível estimar sua magnitude.

Entender as duas frentes classificatórias é um primeiro passo em direção à boa seleção de sensores para uma dada aplicação. Saber que existem sensores dedicados a determinadas aplicações e plataformas também facilita a escolha, uma vez que é uma indústria altamente desenvolvida, com graus de especificidade grandes.

2.1.5 Navegação e planejamento

Na etapa de planejamento e navegação, o aspecto preponderante é a cognição. Essa representa o elenco de processos de tomada de decisão com a ambição de conquistar um determinado objetivo (ou vários deles) para sua missão. Essas decisões devem estar sob o regimento de propósitos que as conecte aos objetivos finais, sendo convergentes e significativos.

A competência de navegação está diretamente relacionada ao aspecto da cognição. Dada uma missão com objetivos e conhecimento parcial sobre o ambiente em qual o sistema robótico estará submetido e sua posição atual, a navegação engloba a capacidade de o robô agir baseado no conhecimento fornecido, tão como nas informações obtidas a partir dos sensores com a finalidade de alcançar a completude da missão o mais eficientemente e com alta confiabilidade. O grande diferencial de um sistema de navegação é exatamente a forma com que ele decompõe a problemática global em subunidades, de fácil resolução (SIEGWART & NOURBAKHS, 2004).

Já o planejamento envolve a elaboração de uma trajetória a ser assumida levando o robô a concluir sua missão, quando executada. O planejamento em si é um processo de prazo mais longo que o momento presente, sendo uma atividade estratégica que aproximará as etapas cognitivas das funções de locomoção e movimento. Essa etapa envolve o levantamento de

informações sobre o ambiente associando-as ao conhecimento parcial prévio existente a respeito do meio de operação e da missão.

Um ponto crucial para a navegação que está intrinsecamente relacionado com o planejamento das rotas é o desvio de obstáculos. A segurança e integridade do sistema só são garantidas quando uma boa abordagem é desenhada com essa finalidade. Diferentemente do planejamento em si, evitar obstáculos ocorre se alicerçando nos sensores em tempo real de reconhecimento ambiental, caracterizando-se como uma atividade tática e reativa.

Ambos, a navegação e o planejamento, transcendem os algoritmos propriamente ditos e demandam de todos os sistemas integração e operabilidade para que os melhores resultados sejam atingidos.

2.1.6 Planejamento de rotas

Como mencionado na seção anterior, o planejamento de rotas consiste da elaboração das trajetórias a serem assumidas pelo robô durante sua navegação. Apesar de soar natural a um sistema robótico planejar os caminhos que percorrerá, esta é uma etapa de grande dificuldade uma vez que não é possível que haja intervenção humana e reajustes a fim de corrigir falhar durante a execução. Em outras palavras, é uma etapa de grande autonomia para o robô.

Existem diversas soluções propostas no campo da robótica móvel, entretanto um dos pontos limitantes da maioria das tecnologias é a grau de complexidade dos algoritmos adotados, uma vez que este cresce exponencialmente com os graus de liberdade existentes no robô (ADÔRNO, AGUIAR, & BORGES, 2005). Da mesma forma, a complexidade dos algoritmos incrementa consequentemente as capacidades do *hardware* utilizado para o processamento, exigindo dispositivos mais vultosos e onerosos, o que acaba por restringir a utilização dos mesmos.

Nas duas últimas décadas os estudos científicos relacionados à temática focaram extensivamente em planejamento de rotas em ambientes conhecidos. Além disto, métodos muito eficientes e precisos foram desenvolvidos para robôs que portam poucos graus de liberdade (KAVRAKI & LATOMBE, 1994), garantindo a sistemas robóticos simples alta performance baseados em dispositivos com capacidade de processamento limitada.

Em geral, as técnicas de planejamento de rotas difundidas no campo científico da

robótica móvel baseiam-se preliminarmente em três abordagens básicas atuais (ADÔRNO, AGUIAR, & BORGES, 2005). Elas são os métodos de mapas de rotas, métodos de decomposição em células e métodos de campos potenciais.

O método de mapas de rotas leva em consideração a aprendizagem sobre o meio o armazenamento desta informação em forma de grafo e nós, sendo mais indicado para ambientes estáticos. O método de campos potenciais são técnicas que fazem analogia a campos potenciais elétricas atribuindo ao objetivo um valor potencial atrativo em relação ao robô e aos obstáculos um potencial repulsivo, fazendo-o evitá-los. Este, por sua vez é largamente utilizado em realizações em espaços tridimensionais, como ar e água. Por fim, o método de decomposição em células consiste de segmentação do espaço de atuação do robô em células adjacentes sem intersecções de área, onde se busca conectar células adjacentes que liguem o robô ao seu destino. Esta técnica, por sua vez, é indicada para ambientes bidimensionais, mesmo que com a presença de obstáculos irregulares (SILVA, 2011).

Em geral, a escolha da abordagem mais apropriada para a aplicação corrente é de suma importância para um bom desenvolvimento e desempenho da plataforma robótica em sua atividade de destino.

2.1.7 Mapeamento

O mapeamento é terceiro pilar do que se conhece no meio científico como “Robótica Exploratória”. Os demais são a localização e o planejamento de rotas e a intersecção das três áreas é conhecida com SPLAM (do inglês *Simultaneous Planning, Localization and Mapping*) (NETO & CAMPOS), tendo o planejamento e localização sido mencionados em seções anteriores.

De forma simplificada, o mapeamento em robótica visa a construção representativa de modelos do ambientes físicos pela utilização de robôs móveis (THRUN, 2002). Assim como o planejamento de rotas, o mapeamento está intrinsecamente relacionado com a autonomia do robô e portanto como a qualidade da execução da tarefa a que se destina, mesmo quando não seja diretamente o mapeamento da área de estudo.

O processo de mapeamento ocorre através a aquisição de dados do meio, através de sensores tais quais câmeras, sonares, lasers, de tecnologia infravermelho, radares, dentro

outros (THRUN, 2002) que estão sujeito a imprecisões e erros, que devem ser devidamente analisados e escolhidos a respeito do meio e condições de medição para melhores resultados.

Por fim, é notável a correlação intrínseca entre o planejamento, localização e mapeamento, sendo estes interdependentes, o que faz da robótica uma ciência complexa e interdisciplinar. O desenvolvimento de uma plataforma robótica passa por iteradas validações dos subsistemas propostos a partir da integração ao conjunto, que insere desafios referentes à interdependência de cada processo realizado pelo robô a fim de atingir a autonomia.

2.1.8 Exploração de Ambientes

A exploração de ambientes é uma das maiores aplicações de robôs móveis autônomos. A perfeita capacidade de executar a tarefa de exploração sem a intervenção humana de forma consistente é um dos balizadores para a categorização de um robô como completamente autônomo (EDLINGER & PUTTKAMER, 1994).

O desbravamento do espaço pelo robô é o resultado a integração dos sistemas de percepção, controle e atuação em seu ciclo contínuo. Desse modo é possível inferir que os aspectos mais importantes para a exploração do ambiente são a localização do robô durante sua movimentação, modelagem apropriada do ambiente, sensores e estratégias de controle da dirigibilidade do robô (STACHNISS & BURGARD).

Grande parte das abordagens de exploração toma como base as fronteiras entre partes do ambiente conhecidas e desconhecidas (STACHNISS & BURGARD) e deste modo visam sempre ampliar o conhecimento da área explorando a porção desconhecida mais próxima da localização do robô. Em geral, uma localização precisa do robô tem valor decisivo na construção de modelos mais adequados do ambiente já conhecido, assim como sua fronteiras e conseqüente impacto na qualidade da tarefa de exploração, o que mais uma vez aponta a grande interdependência das etapas e processos na construção de uma plataforma robótica.

2.2 INTELIGÊNCIA ARTIFICIAL

O homem tenta há milênios compreender quais os processos desencadeiam o pensamento e cognição da espécie *homo sapiens*. Chega a ser provocador a forma com que um punhado de matéria, não composto por elementos diferentes daqueles encontrados nas demais estruturas e organismos que o cercam, ter a capacidade de perceber, entender, prever e manipular um mundo incrivelmente maior e mais complexo do que si mesmo (RUSSELL & NORVIG, 2010).

Por outro lado, o campo da Inteligência Artificial se propõe a abarcar maiores potencialidades. Além de compreender, é objetivada a construção de entidades inteligentes, como réplica dos sistemas biológicos naturais de cognição humana.

Inteligência artificial (IA) teve uma primeira sistematização com o conhecido Teste de Turin. Esse teste propunha um limiar em que, quando interrogados simultaneamente um computador e um ser humano, um avaliador não mais conseguisse distinguir quem era máquina e quem era humano.

Com esta proposição é possível elencar pelo menos seis disciplinas que se relacionem diretamente ou mesmo compreendam grande parte do que hoje a inteligência artificial representa. Elas são (RUSSELL & NORVIG, 2010):

1. **Processamento de linguagem natural:** representa a capacidade de máquina compreender e se comunicar através das linguagens humanas, a exemplo o Português;
2. **Representação do conhecimento:** capacidade de armazenar e que sabe e captura do meio;
3. **Raciocínio automatizado:** utilizar o conhecimento armazenado para responder aos estímulos externos e traçar novas conclusões a partir do que já é sabido;
4. **Aprendizado de Máquina:** capacidade de se adaptar a novas circunstâncias, detectar ou mesmo extrapolar padrões;
5. **Visão computacional:** capacidade de perceber e reconhecer objetos;
6. **Robótica:** capacidade de manipular objetos e se movimentar num campo de operação ou trabalho.

Nessa conjuntura traça-se uma distinta separação entre programas corriqueiros e sistemas inteligentes. Para por em palavras o que expressa tamanha diferença faz-se mister a

definição de um agente racional.

Um agente racional se diferencia de um computador convencional pela sua capacidade de operar autonomamente, perceber o ambiente, persistir por um período prolongado de tempo, adaptar-se a mudanças, criar e buscar metas. Ele age para atingir o melhor resultado ou, quando existem incertezas, o melhor resultado esperado (RUSSELL & NORVIG, 2010).

Um forte pilar no qual se sustenta a inteligência artificial é a marcação de inferências positivas. Uma vez que se tem posse de uma dada missão faz-se necessário salientar as tentativas e incursões com resultados positivos para a conclusão bem sucedida da missão a ponto em que o sistema se auto-avalia e gere um mapa de boas práticas.

Entretanto, nem sempre isso é possível, pois há determinadas situações e circunstâncias onde não há clareza dos objetivos. Nesses casos, alternativas a essa técnica são interessantes, sendo que a inferência positiva é apenas uma de muitos possíveis mecanismos para se alcançar a racionalidade.

2.2.1 Sistemas Especialistas

Os sistemas especialistas (SE) são um subconjunto dos agentes inteligentes advindos da inteligência artificial. Eles têm suas estruturas baseadas em conhecimento, sendo amplamente utilizados para a resolução de problemas complexos. Como o nome sugere, os SE são a transferência das competências de um especialista humano para um sistema que replique suas habilidades. Uma definição formal é como segue:

“Um sistema especialista (*expert system*) é um sistema computacional que emula a estratégia de resolução de problemas de um especialista humano. Apesar de que um SE não pode chegar a possuir a capacidade cognitiva de um especialista humano, na ausência deste constitui-se em uma ferramenta importante para a resolução de problemas” (PAUCAR, 2000).

Existem diversas aplicações em que os sistemas especialistas são consistentemente aplicados em uso corriqueiro nas atividades humanas. Interpretação de textos que ultrapassa a simples tarefa de digitalizar as informações manuscritas, mas também gerar padrões e conclusões acerca do que é proposto; diagnóstico de doenças a partir da sintomatologia aparente; planejamento de rotas a partir de geoposicionamento e conhecimentos sobre a

fluidez do tráfego de veículos. Todos esses casos acima são exemplos notórios de utilização cotidiana dos sistemas especialistas.

Um diagrama, adaptado de (PAUCAR, 2000) é apresentado na Figura 3, visando apresentar o alcance dos sistemas especialistas dentro do universo da Inteligência Artificial.

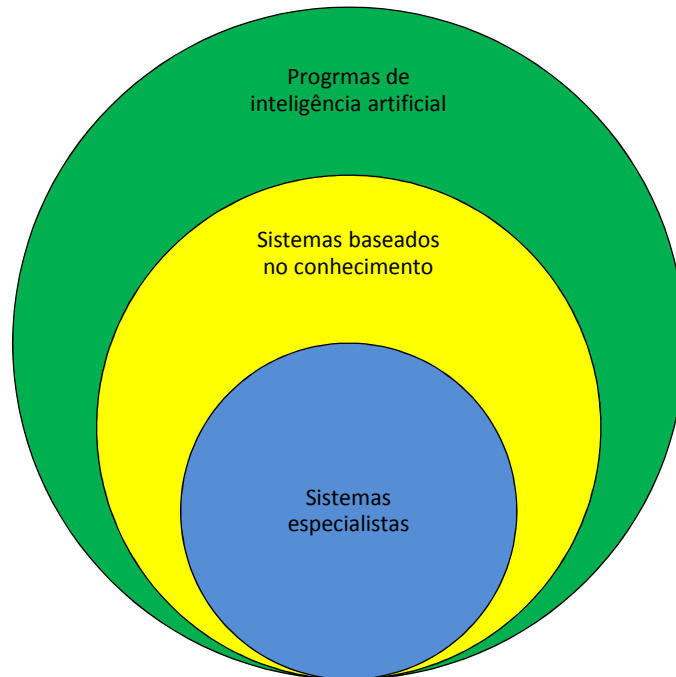


Figura 3: Diagrama de abrangência dos sistemas especialistas dentro do universo da inteligência artificial
Fonte: (PAUCAR, 2000)

O uso de um sistema especialista pode substituir, quando oportuno, um especialista humano ou mesmo complementar sua atuação. Casos críticos onde existe expressivo risco à integridade humana ou mesmo que as condições não são favoráveis ao exercício da função de uma pessoa, aplicar um sistema especialista pode ser salutar.

Além das questões inerentes às condições de trabalho, aspectos de desempenho e integração de informações, a aplicação dos SE apresenta sensível superioridade à utilização da mão-de-obra. Velocidade de processamento de dados e atuação são diferenciais positivos aos sistemas não-humanos.

Por fim, algo também importante a respeito dos sistemas especialistas é que os mesmos podem integrar e agregar conhecimentos de diversas pessoas especialistas em um só ponto de decisão, aumentando a sincronia das informações e garantindo que não haverá divergências na execução das tarefas demandadas.

Entretanto, a contrapartida também é verdadeira. Existem aspectos do emprego da mão-de-obra humana que são, até o presente dia, inacessíveis quando se aplica sistemas

especialistas. Os seres humanos diferenciam das máquinas pela sua capacidade de aprendizado, que se apresenta, por ora, limitada nas máquinas. Além do aprendizado, algo correlacionado é a criatividade que extrapola a mera correlação de padrões executada pelos SE. Adiciona-se a versatilidade inerente ao homem e mesmo a sua elevada percepção do ambiente externo, quando comparado a sistemas computacionais (PAUCAR, 2000).

Além das aplicações e vantagens referentes aos sistemas especialistas, outro fator importante para o real conhecimento dos sistemas especialista é a classificação destes. Seguindo a mesma linha que é possível identificar cinco categorias principais (PAUCAR, 2000):

1. **Representação do conhecimento:** estruturados sobre regras, orientados objetos;
2. **Operação da máquina de inferência:** orientação através de dados, objetivos ou a combinação dos dois;
3. **Interface homem-máquina (IHM);**
4. **Operação *online* (tempo real) ou *off-line*;**
5. **Extensibilidade de conhecimento:** se existe a capacidade de ampliar ou adaptar o conhecimento do sistema a novas necessidades.

Por fim, o sistemas especialistas, apesar de suas limitações, são excelentes alternativas para tarefas que exijam processos decisórios mas que não sejam robustos em demasia. São sistema que apresentam um grande variedade de complexidade, varrendo de algoritmos de dezenas de linhas de código a sistemas orientados a objeto, com diversas classes e grupos de conhecimento.

2.3 ROBÔ APLICADO À RESOLUÇÃO DE LABIRINTOS (MAZE SOLVING ROBOT)

Em termos gerais há duas conceituações possíveis para labirinto. Apesar de a língua portuguesa apenas apresentar um vocábulo para tal denominação, ao estendermos à significação à língua inglesa são achados duas palavras, erroneamente utilizadas de forma intercambiada, mas que representam conceitos de estruturas diferentes. Ambas as definições representam um série complexa e confusa de caminhos, porém apresentam singularidades.

A palavra em inglês *labyrinth*, que em português tem a tradução literal para labirinto, representa um caminho único, sinuoso, sem ramificações, que leva ao centro, como

representado na Figura 4 (DIFFEN, 2014).



Figura 4: Exemplo de *labyrinth*
Fonte: (DIFFEN, 2014)

Em contrapartida, a palavra *maze*, também oriunda do inglês, que apesar de simbolizar algo semelhante, tem seu conceito um tanto mais abrangente. Um “*maze*” é um sistema complexo, que apresenta uma variedade de caminhos e cursos possíveis de serem tomados, exige, em vias de regra, um processo decisório de qual caminho e direção a seguir e é ramificado. A Figura 5 apresenta um exemplo de “*maze*”.

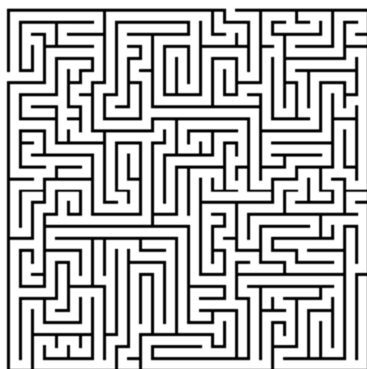


Figura 5: Exemplo de *maze*
Fonte: (UEFAP, 2014)

No caso presente do trabalho, tratar-se-á apenas do segundo caso, sendo adotada a palavra “labirinto” apenas para fazer referências às estruturas denominadas “*maze*” em inglês.

2.3.1 Micromouse

O *Micromouse* é um evento mundialmente conhecido, onde competidores projetam pequenos robôs móveis autônomos com a finalidade de chegar, a partir de um ponto de largar predeterminado, ao centro de um labirinto, representando por uma marca preestabelecida. Os labirintos apresentam especificações técnicas definidas pela competição que compreendem a quantidade de células, espessura e altura das paredes e dados específicos dos robôs (MISHRA & BANDE, 2008).

Além de completar de forma bem sucedida a tarefa, o robô deve, ao passo em que varre o espaço do labirinto, mapeá-lo e definir, em conclusão, o caminho ótimo para o alcance do objetivo. Ao fim, o robô deve executar o caminho ótimo em menor tempo possível.

O evento vem sendo promovido desde a década de 1970 e representa um grande aliado ao desenvolvimento da robótica móvel. Diversos aspectos cinemáticos e lógicos são levados em consideração a fim de obter melhoramentos sensíveis aos objetivos da competição.

Ainda neste tocante, a competição é um ambiente que conjuga entretenimento e ciência, sendo a base de testes de diversos grupos focados em robótica móvel. Os padrões exigentes da competição levam as equipes a ponderar os aspectos que compõe a plataforma robótica, como estrutura mecânica, poder computacional, velocidade de desenvoltura, entre muitos outros.

Deste modo, o *micromouse* se configura hoje como um dos principais catalisadores dos avanços tecnológicos referentes aos robôs de exploração de labirintos, sendo muitas das práticas e algoritmos consagrados oriundos do mesmo (MISHRA & BANDE, 2008). Exemplos destes algoritmos são apresentados na seção seguinte, como o “seguidor de parede”, “regra da mão esquerda” e “flood-fill”.

2.3.2 Técnicas de resolução de labirintos

A exploração de ambientes complexos e desconhecidos através de plataformas robóticas vem ganhando muitas aplicações uma vez que amplia a segurança da tarefa, assim como sua otimização e eficácia. Os labirintos são modelos representativos de ambientes com alto grau de complexidade e com presença de obstáculos, que simulam de certa forma ambientes reais onde os robôs serão expostos a diversas opções de caminhos.

Como mencionado na seção anterior, existem competições, como o *Micromouse* que fundem entretenimento a ciência. Tais eventos funcionam como plataformas de teste para novas técnicas que visam acelerar o processo de resolução de um labirinto que, em outras palavras, significa explorar o ambiente da melhor forma possível em menor tempo. Muitas dessas técnicas se consagram como ótimas para aplicações reais de robótica móvel.

Faz-se mister a apresentação de algumas dessas técnicas a fim de ilustrar a grande gama de opções que o desenvolvedor pode fazer uso, dando devida atenção às especificações do projeto corrente, tanto em função dos dispositivos utilizados na construção da plataforma robótica (que envolve adaptação dos elementos mecânicos, eletrônicos e mecatrônicos), assim quanto em relação às características do meio físico.

Um primeiro exemplo, e também um dos mais difundidos, conhecido como a “lógica do seguidor de parede”. O robô toma como referência a parede esquerda ou direita e a segue até atingir seu objetivo de sair do labirinto ou encontrar seu centro, a depender do desenho da missão (MISHRA & BANDE, 2008). A adoção de qualquer uma das paredes, seja esquerda ou direita, como referencial não impacta nos resultados do algoritmo, ficando ao critério do desenvolver a sua escolha. O algoritmo seguidor de parede exige um moderado grau de processamento, sendo que o sistema de controle deve ter configurações suficientes para manter a avaliação do ambiente em tempo real.

Uma segunda opção, entretanto derivada da primeira supracitada, é a técnica conhecida como “regra da mão esquerda”. Os algoritmos baseados nesta são divididos primordialmente em dois blocos. O primeiro bloco está relacionado com a dirigibilidade do robô, o que significa manter o mesmo se movimentando em trajetória aproximadamente retilínea, o que já acontece de forma integrada nos “seguidores de parede”. Isso se dá através do monitoramento em contínuo, em pequenos intervalos de tempos, da distância entre a lateral do robô e uma das paredes, tomada como referencial.

O segundo bloco compreende a etapa de tomada de decisão, baseado na ideia de nós. Um nó é basicamente um ponto onde um ou mais caminhos, ou a mudança do mesmo é apresentada ao robô. De forma sistemática o robô preferirá sempre tomar o caminho da esquerda, quando disponível, seguido pelo caminho frontal, também quando disponível, deixando por último a direita, em detrimento das outras possibilidades. Caso nenhuma das opções seja disponível, resta ao robô retornar pelo mesmo caminho percorrido anteriormente.

Deste modo esta técnica traz uma dissociação entre as duas partes do algoritmo, permitindo a otimização e melhoramento de cada parte de forma independente. Entretanto, dois problemas são presentes nessa técnica, que requerem dispositivos com maior potencial de processamento, assim como depende da regularidade das paredes e dos caminhos, uma vez que sempre tenderá a manter a mesma distancia destas.

Um último exemplo é conhecido como *flood-fill*. Este algoritmo baseia-se na segmentação do espaço ao redor do robô em células quadradas de lado predeterminado, sendo a célula onde o robô esta associada ao potencial 0, as células à frente, atrás, perpendicularmente à esquerda e direita são atribuídas com o potencial 1. As células logo à frente, atrás, perpendicularmente à direita e esquerda das de potencial 1 são atribuídas com potencial 2, a menos que a célula já esteja associada a um potencial menor, e assim por diante (MISHRA & BANDE, 2008).

Com toda a área do labirinto coberta por potenciais numéricos, o robô começa a exploração dos potenciais de forma crescente até que toda a área demarcada seja explorada ou o objetivo seja alcançado neste entretempo.

Esses são alguns exemplos de abordagens elaboradas com intuito a exploração e completude de missões em labirintos. Cada abordagem tem seu foco, seja ele tempo, menor espaço, menor custo computacional, ou a composição destes, entre outros.

2.3.3 O robô e subsistemas

Um robô destinado à realização da tarefa de navegação em um labirinto deve, como todo sistema robótico móvel, apresentar sistemas periféricos com funções específicas (percepção, locomoção, entre outros) integrados em torno de um centro de processamento e tomada de decisão, vulgarmente referido como seu “cérebro”.

Em sua grande maioria, os Robôs para Resolução de Labirintos (do inglês Maze Solving Robot ou MSR) apresentam seu sistema locomotor estruturado sobre rodas (frequentemente pneumáticas). Em termos cinemáticos, geralmente usa-se a menor quantidade de rodas possível, ou seja, três. Três rodas garantem movimentos compostos bidirecionais, entretanto apenas duas dessas devem ser necessariamente motorizadas, sendo a terceira uma roda guia, que facilitará a operação das outras.

Em termos de percepção, esses robôs devem perceber o ambiente basicamente em aspectos de distância, sendo que não são esperados obstáculos no caminho, apenas a identificação das possibilidades de caminhos e seus limites (paredes). Parte dos trabalhos práticos nessa temática sugere como opções sensores sônicos e luminosos (VANNOY, 2009). O ultrassom é uma alternativa barata e de fácil implementação. Para labirintos maiores, que permitem erros sistêmicos na ordem grandeza entre 10 e 20 milímetros, é sem dúvida uma opção viável. Entretanto, no caso de labirintos com dimensões reduzidas, os sensores infravermelhos são mais indicados por apresentarem precisão ampliada em relação aos ultrassônicos.

Outro componente importante para a viabilização de um protótipo são os *drivers*. Os motores, apesar de existirem unidades motoras que exijam tensões próximas aos sistemas digitais de processamento, os mesmos demandam potência, e por consequência, correntes muito superiores às que os sistemas eletrônicos digitais são capazes de oferecer. Por isto, os *drivers* são importantes, pois, apesar de serem controlados pelo sistema eletrônico de controle, extraem de fontes externas (baterias, fontes ou pilhas) a potência necessária para operar os motores de forma adequada.

Existe uma variedade grande de *drivers* disponíveis no mercado, variado em potência controlada, corrente fornecida e número de motores passíveis de controle. Em adição, é possível garantir também que, através dos *drivers*, os motores tenham o sentido de rotação orientado ao desejado do sistema como também as velocidades de rotação desempenhadas à mercê da necessidade.

Por fim, o controle. Como se trata de um sistema autônomo móvel é possível elencar o controle em duas partes distintas. Um primeiro bloco está relacionado à dirigibilidade do robô, ou seja, ele deve seguir uma linha minimamente reta para até que uma manobra seja necessária. Apesar de uma soar primário, é um processo que deve ser cuidadosamente elaborado para que as estratégias de tomada de decisão sejam perfeitamente executadas.

O segundo se relaciona à inteligência embarcada no robô. Ele terá que, a partir de uma

estratégia elaborada por um especialista humano, desempenhar um algoritmo direcionado à completude da exploração do labirinto. Existem várias técnicas disponíveis, algumas mais completas e complexas que outras, mas é possível elaborar códigos relativamente simples que atendam às necessidades da aplicação.

3 PROTÓTIPO E METODOLOGIA

Na seção presente serão discutidos tanto a metodologia empregada na confecção do protótipo e nuances como também este último será decomposto em componentes básicos onde serão abordados os modelos e especificações tão como os motivos levados a escolhê-los.

3.1 METODOLOGIA LEAN STARTUP

A metodologia *Lean Startup* (ou Startup Enxuta) foi elaborado pelo estadunidense (RIES, 2011) em função da necessidade de um modelo diferenciado de gestão de empresas de base tecnológica que se encontram em um momento de mercado mais dinâmico, com exigências em termos de tempo, custos e qualidade cada vez mais crescentes: as conhecidas *Startups*.

O modelo convencional não mais atende às necessidades da nova lógica de negócios trazida principalmente pelas empresas produtoras de *software*. As necessidades mutantes de clientes e concorrentes cada vez mais preparados e competitivos exigem dos empreendedores tecnológicos uma nova abordagem para desenvolverem seus negócios.

O pilar básico da metodologia é o incremento da velocidade e a diminuição de custos em cada iteração realizada com o cliente. Nesse entretempo a empresa deve validar o descartar hipóteses importantes sobre o produto ou mercado (SANTOS, 2009).

A prática considera que tanto a necessidade do cliente quanto o produto que suprirá essa necessidade são desconhecidos e, portanto um processo de descoberta é lançado a fim de obter validações a cerca de ambos. Os procedimentos iterativos para tal finalidade são conhecidos como *Customer Development*, ou seja, a empreitada se dá em duas etapas: a primeira é literalmente identificar quem é o cliente e validar se realmente é um cliente passível de ser atendido pelo negócio. Após a garantia de que é um relacionamento genuíno, o empreendimento pode considerar aquilo um negócio e estruturar mecanismos para escalar o produto através de marketing e afins.

Para o presente trabalho, o uso da metodologia se dará na esfera do produto. A *startup* enxuta entende o produto sempre pela ótica do mínimo produto viável, ou seja, ao entender

que é necessário investir o mínimo e executar as tarefas afim de validar o negócio em menor tempo possível, é traçado um modelo de desenvolvimento de produto que visa propor uma solução ao problema do cliente da forma mais básica possível e lança-lo. Na medida em que o cliente testa a solução e realimenta o desenvolvedor com as deficiências e vantagens de seu produto, o mesmo tem a capacidade de se empenhar em outra interação para atingir o melhor atendimento ao cliente.

Esse relacionamento acaba por poupar tempo e dinheiro, recursos primordiais para negócios altamente inovadores em um ambiente competitivo.

A utilização da metodologia no trabalho visa gerar uma abordagem de engenharia de produto sofisticada que possibilita a o acompanhamento do desenvolvimento do protótipo de forma sistemática, com controle de custos, validações constantes das propostas de soluções para os problemas, mantendo como foco as especificações do projeto.

3.1.1 Mínimo Produto Viável

A metodologia *Startup Enxuta* encerra um método prático de elaboração de produtos em geral denominado Mínimo Produto Viável ou MVP (do inglês *Minimum Viable Product*). Uma definição completa é como segue:

“Para descobrir a combinação Problema/Solução para um produto, a startup deve estabelecer um processo iterativo que permita um aprendizado constante sobre os clientes e outras premissas do negócio. Esse processo é uma combinação de experimentações práticas com investigações qualitativas que buscam extrair dados para comprovar as hipóteses dos empreendedores, bem como as nuances e “porquês” que estão por trás do comportamento dos clientes. No centro da experimentação está o conceito do Mínimo Produto Viável (MVP).” (SANTOS, 2009)

O MVP apresenta as mínimas funcionalidades que permitem o desenvolver aprender sobre a sobre as necessidades dos usuários de seu produto. A proposição gira sobre o mantra proposto pelo Eric Ries que sugere “lançar cedo, lançar com frequência” (do inglês *release early, release often*) que tem como vantagem primordial a possibilidade de obter realimentação real, uma vez que se considera que toda decisão tomada dentro do escritório é apenas uma hipótese, sem validade experimental (SANTOS, 2009).

Dentro da perspectiva do presente trabalho, o MVP aplica-se visando a eximir do

desenvolvimento do protótipo do robô a necessidade de uma pesquisa profunda e detalhada, que consumiria muito tempo e não, necessariamente, garantiria bons resultados.

Na seção seguinte o processo será adotado será explicado em detalhes, elencando também os componentes selecionados para o protótipo.

3.2 O PROTÓTIPO

O robô proposto consiste de sistema com controle diferencial de duas rodas que garantem a capacidade de execução de manobras em 90 graus para direita ou esquerda e 180 graus. Há uma roda giratória auxiliar passiva que fornece estabilidade ao corpo do robô. O sistema incorpora também:

1. Dois motores de corrente contínua que operam na faixa de tensão entre 6V e 9V, abaixo da tensão dos componentes eletrônicos de controle;
2. Um grupo de sensores ultrassônicos que permitem a visualização dos ambientes laterais e frontais;
3. *Drivers* de acionamento e controle de motores DC para a faixa de tensão e corrente especificados pelos demais componentes eletrônicos;
4. Controlador microprocessado com quantidade de entradas e saídas digitais e analógicas suficiente para a aplicação.

O protótipo montado, com todos componentes está representado na Figura 6:

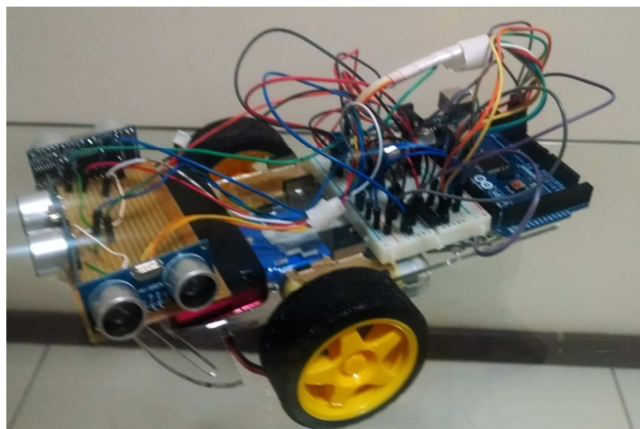


Figura 6: Protótipo montado

3.2.1 Os subsistemas

O robô móvel desenvolvido para o presente projeto conta com as mínimas funcionalidades que configurem um robô móvel. Os componentes foram agrupados em três subsistemas distintos que tem funções específicas dentro do funcionamento global do robô. Os subsistemas foram denominados:

1. Percepção;
2. Controle;
3. Moção.

3.2.2 Percepção

O subsistema de percepção compreende a integração de três sensores ultrassônicos que têm por função indicar ao sistema de controle qual a distância entre o extremo do sensor (que coincide com os limites frontais e laterais do robô) e a parede do labirinto mais próxima.

O sensor frontal é disposto perpendicularmente aos sensores laterais, sendo que os sensores laterais ficam diametralmente opostos entre si. Para facilitar a organização todos os sensores foram fixados em uma placa universal PCB que permite a impressão de um circuito usando solda. As conexões GND e VCC foram interconectadas para facilitar a conexão com a placa do controlador. Os demais conectores são conectados a saídas I/O do controlador, a depender de sua configuração no código fonte.

Foi avaliada a utilização de um motor servo que poderia rotacionar em 90 ou 180 graus, entretanto foi preferida a angulação manual de três sensores pelo baixo custo da unidade do sensor, em comparação com o custo do motor. Além disto, o motor adicionaria maiores dificuldades de controle ao sistema.

3.2.3 Controle

O subsistema de controle foi implementado através da plataforma de desenvolvimento microprocessada Arduino MEGA 2560. Essa plataforma foi selecionada por ser de fácil acesso no mercado local, de custo baixo e com todas as funcionalidades requeridas pela aplicação do MSR.

Também foram adaptados dois *drivers* L293D que apresentam controle de motores DC através de PWM (do inglês *Pulse Width Modulation* ou Modulação por Largura do Pulso) que permite a precisa determinação da velocidade dos motores e o sentido de giro dos mesmos. O CI L293D é um dispositivo bem adaptado a aplicações com Arduino por suportar suas especificações e apresentar sistemas de proteção para correntes reversas para proteção do controlador.

O sistema de controle é alimentado por uma bateria LiPo (lítio-polímero) que tem capacidade de descarga de uma vez seu valor nominal, fornecendo 12 V e carga de 1450 mAh em consumo nominal.

3.2.4 Movimentação

O movimento do robô é executado por dois motores DC com faixa de operação entre 6 V e 9 V, que apresentam a liberdade de girar em ambos sentidos (horário e anti-horário). Os motores apresentam acoplados a si caixas de redução e transmissão de movimento que convertem o movimento axial em radial

Os motores são acoplados a rodas com pneus para aumentar a aderência ao solo e amortecer impactos. As duas rodas, quando associadas ao robô, podem compor movimentos circulares ou oblíquos a depender do sentido e velocidade de rotação.

Uma roda giratória garante a estabilidade do chassi do robô, mas não apresenta movimento ativo. Seu movimento é omnidirecional, a depender do impulso dos motores. A velocidade dos motores é controlada pelos drivers e limitada por um potenciômetro que, além de garantir menores correntes dentro das bobinas dos motores, funciona como um divisor de tensão oriunda da bateria LiPo 12 V descrita na seção anterior.

3.3 COMPONENTES

Nesta seção serão apresentados os componentes utilizados na montagem do protótipo do robô. Os componentes que exijam maiores explanações as terão.

3.3.1 Arduino

O Arduino Mega 2560 é uma placa de desenvolvimento baseada no controlador ATmega2560. Ele apresenta 54 entradas e saídas digitais, sendo que dessas, 15 são utilizadas como saídas PWM. Em adição, 16 entradas analógicas, 4 UARTs que permitem a comunicação serial, um conector USB e *power-jack*, além de outras funcionalidades não diretamente utilizadas no projeto. A placa é do modelo apresentado na Figura 7.

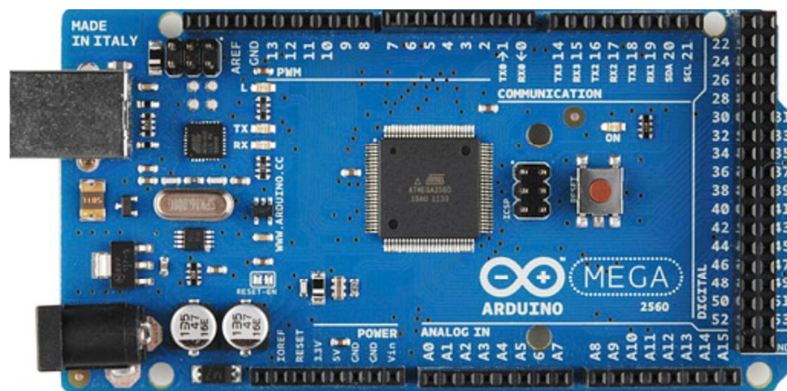


Figura 7: Arduino Mega 2560

Dentre os modelos de controlador disponíveis no mercado o modelo Arduino foi selecionado pela sua facilidade de programação, excluindo a necessidade de um gravador especializado pela presença de *bootloader* na bios do microcontrolador ATmega2560. Além disto, a programação é em C com bibliotecas proprietárias interpretadas do C++.

A quantidade de entradas e saídas digitais superior foi outro fator para sua escolha em detrimento de outros modelos mais simples de Arduino. Entretanto essa escolha se

demonstrou irrelevante uma vez que apenas 12 pinos entradas e saídas (6 PWM e 6 analógicos) foram realmente utilizados.

Outra vantagem é a presença de reguladores de tensão que permitem alimentação direta à placa, sem necessidade de retificar o sinal.

3.3.2 Sensores Ultrassônicos

Sensores ultrassônicos foram selecionados por serem indicados para medição de distâncias não compostas (apenas uma componente). Além disto, o grau de precisão (cerca de 10 mm) fornecido pelos sensores é suficiente para a configuração e tamanho do labirinto assim como para as velocidades de operação do robô.

O modelo escolhido foi o HC-SR04 que apresentam sua comunicação pré-estabelecida com placas Arduino através de bibliotecas proprietárias no compilador da plataforma. Um exemplar do modelo do sensor é apresentado na Figura 8.



Figura 8: Sensor ultrassônico HC-SR04

O processo de medição se dá pelo envio de um feixe de som a partir de um terminal T do sensor para o objeto cuja distância está sendo medida, o mesmo é refletido e captado pelo terminal R adjacente. O tempo de viagem é medido, o que através de manipulações matemáticas, fornece a distancia em centímetros. A velocidade do som no ar é 340 m/s, sendo que a distância deve ser considerada dobrada uma vez que o som a percorre na ida e no retorno, antes de ser percebido pelo receptor do sensor.

3.3.3 Driver: CI L293D

O *driver* selecionado é do modelo L293D. Sua escolha advém da sua capacidade de controlar a velocidade do motor através de PWM (indicado para uso com Arduino) e terminais suficientes para o acionamento de dois motores DC de até 36 W de potência consumida. O CI suporta uma corrente nominal de 1.5 A.

O circuito integrado apresenta duas pontes H, sendo que cada uma delas está disposta em uma lateral. Os pinos entre 1 e 7 são responsáveis pelo acionamento e controle de um dos motores. Os pinos entre 9 e 15 fazem as mesmas operações para o outro motor. Os pinos 8 e 16 são de alimentação. O L239D apresenta o esquema como visto na Figura 9.

Os pinos com simbologia GND são responsáveis pelo arrefecimento do chip, e portanto devem ser conectados, cada um, a fios individuais, que garantam maior transmissão de calor para fora do CI.

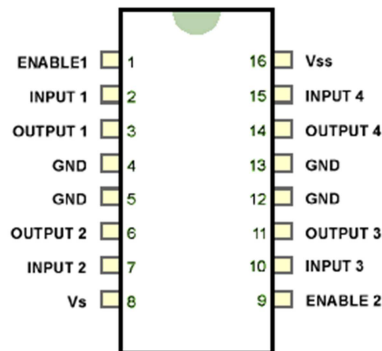


Figura 9: Esquema do L293D

3.3.4 Motores e Rodas

Os motores utilizados foram motores DC operantes entre 6V e 9V como caixa de redução que transmitam movimento axial em radial. A seleção ocorreu pelo fato do mesmo já apresentar plena aplicação para robôs pequenos e também serem vendidos em kits completos, facilitando a montagem e integração dos componentes mecânicos. Os motores associados as caixas são apresentados na Figura 10.



Figura 10: Motores e caixa de redução

As rodas com pneus (para aumentar a aderência ao solo) são vendidas em conjunto com os motores em caixas de redução. Exemplos são apresentados na Figura 11. Além das rodas ativas, foi acoplada ao chassi uma roda giratória omnidirecional para garantir estabilidade ao robô e permitir manobras compostas. Uma imagem ilustrativa é mostrada na Figura 12.



Figura 11: Rodas com pneumáticos



Figura 12: Roda giratória

3.3.5 Prototipagem e Circuito Impresso

A parte referente aos sensores (subsistema Percepção) foi implementada em placa de circuito impresso universal (PCB) e fixando-os aos seus conectores através de solda. Assim foi feito pelo fato do circuito dos sensores ser um sistema seguro, pouco sujeito a sobretensões e sobrecorrentes, com baixa probabilidade de retrabalho.

Já a integração do sistema mecânico ao controlar através do *driver* foi realizada em *protoboard* por facilitar a troca de componentes em caso de defeito. Os motores e o Arduino, por vezes, exigem correntes significativas, que podem eventualmente danificar os componentes do subsistema de controle dos motores. Desse modo, é preferível a prototipagem em placa de 400 pinos, como a apresentada na figura 13.

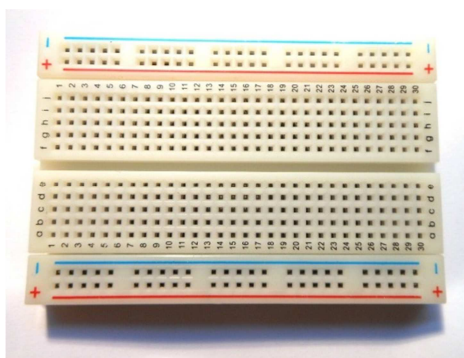


Figura 13: *Protoboard* de 400 pinos

3.3.6 Cabos de conexão e Plugue

As conexões entre componentes foi realidade através de cabos vulgarmente chamados de *jumpers*, com conectores machos em ambas extremidades, como os apresentados na Figura 14.

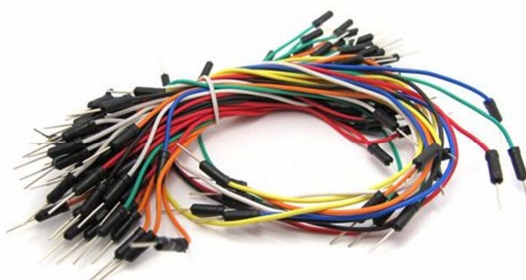


Figura 14: *Jumpers* macho-macho

O Arduino é conectado à alimentação através de plugue 2.1mm, do modelo apresentado na Figura 15.



Figura 15: Plugue de alimentação para Arduino

3.3.7 Bateria LiPo

Inicialmente o projeto foi iniciado com baterias de 1,5 A, com carga nominal de 2700 mAh. Apesar de elas acionarem perfeitamente os motores, a tensão fornecida não atingia o mínimo necessário para o funcionamento ordinário do Arduino. Por isso, foram então substituídas por uma bateria de aeromodelismo com três células, constituídas de lítio e polímeros (LiPo, com tensão de operação de 12V e carga nominal de 1450 mAh. A bateria demonstra autonomia para operar todos os sistemas de forma consistente, apenas demandando um divisor de tensão na entrada dos motores. A bateria é apresentada na Figura 16.



Figura 16: Bateria LiPo 3S 1450 mAh

3.3.8 Chassi

O chassi utilizado é encontrado pré-moldado em lojas de hobbyistas em robótica, e apresenta todas as funcionalidades necessárias. O modelo utilizado é o mesmo da Figura 17.

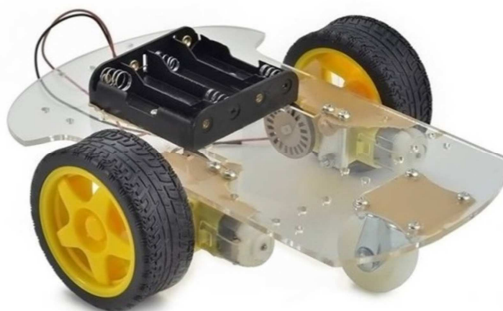


Figura 17: Chassi montado com rodas e motores

3.4 ORÇAMENTO DE AQUISIÇÃO

Tabela 1: Orçamento dos componentes utilizados no prótipo

Produto	Quantidade	Preço Unitário	Preço Total
Chassi completo	1	R\$ 100,00	R\$ 100,00
Bateria LiPo	1	R\$ 85,00	R\$ 85,00
Jumpers x40	2	R\$ 15,00	R\$ 30,00
CIL293D	1	R\$ 8,00	R\$ 8,00
Plugue 2.1 mm	1	R\$ 2,00	R\$ 2,00
Protoboard 400p	1	R\$ 15,00	R\$ 15,00
Arduino Mega	1	R\$ 76,00	R\$ 76,00
Sensor HC-SR04	3	R\$ 10,00	R\$ 30,00
Total			R\$ 346,00

Os produtos listados acima não foram encontrados para pronta entrega na cidade de Salvador. De forma geral, os produtos foram adquiridos pela plataforma Mercado Livre, site especializado em negociação de bens de consumo. Alguns dos vendedores com os quais

foram adquiridas as mercadorias se localizam na cidade de Salvador, entretanto, por não possuírem lojas físicas, as negociações ocorram da mesma forma que para aqueles de outras cidades, dentro ou fora do estado da Bahia.

O processo de compra levou cerca de quarenta e cinco dias, desde a aquisição à entrega dos produtos a domicílio. Atrasos ocorreram devido à paralisação das atividades das agências dos Correios, impedindo, por vezes, o prosseguimento do projeto. Os valores dos fretes não foram incluídos por oscilar a depender da localidade onde habita o projetista, ou mesmo inexistirem, caso todos os componentes sejam livremente comercializados no local.

4 A MISSÃO

O sistema desenvolvido no presente trabalho, como mencionado anteriormente, tem por missão a resolução de um labirinto arbitrário, de forma que seja generalizável a ponto de ser embarcando em diferentes plataformas, com devidos ajustes, e forneça resultados consistentes.

Nesta seção são discutidos o tipo de controle adotado, a estratégia de resolução e o algoritmo desenvolvido.

4.1 O CONTROLE

A estratégia básica de controle adotada para a aplicação robótica do presente trabalho é o liga/desliga (do inglês *on/off*³). O controle executado, apesar de ser um tanto mais abrangente, não se limitando a apenas ligar e desligar, pode ser aproximado desse modelo, uma vez que o controlador alterna entre três condições explícitas diferentes a fim de manter seu curso retilíneo.

São estabelecidas, dentro do curso de um labirinto retangular, três faixas distintas em que o robô, ao executar uma trajetória retilínea poderá se encontrar. Contando com uma largura de 400 milímetros entre paredes adjacentes, um corredor do labirinto deve ter a capacidade de comportar pelo menos duas vezes a largura do robô (cerca de 100 mm) com uma pequena sobra para facilitar as manobras.

As faixas são:

1. Entre a parede esquerda e 100 mm (medidos) longe dela;
2. Entre 100 mm 140 mm (medidos) longe da parede esquerda;
3. Faixa maior que 140 mm (medidos) longe da parede esquerda.

A opção pela parede da esquerda está de acordo com a estratégia especialista adotada para a resolução do problema, conhecida como Regra da Mão Esquerda ou LSRB a ser mais bem explanada na seção 4.3.

Uma quarta condição pode ser adicionada ao sistema a fim de garantir que o robô

³ É a estratégia de controle que se baliza me torno de um ponto de ajuste (*setpoint*). A saída do controlador alterna entre ligado e desligado na medida em que o sinal de erro cruza o ponto de ajuste.

mantenha-se sempre a uma distância segura da parede direita. Como no presente caso as métricas são constantes, a utilização desse condicional, apesar de garantir maior segurança à execução da tarefa, apresenta-se redundante às demais.

A captação dos valores de distância ocorre através dos sensores ultrassônicos. A leitura de tais componentes não é diretamente executada em forma de distância, mas sim em tempo levado para o sinal sonoro percorrer a distância de ida, seja refletido, percorra a distância de retorno e seja então percebido pelo sensor. De forma simplificada, a velocidade do som no ar é tida como constante e aproximadamente igual a 340 m/s. Dessa forma, com simples manipulações algébricas é possível estabelecer uma relação entre o tempo entre o pulso emitido e recebido e a distância entre o sensor e o objeto sentido.

Os valores são obtidos reiteradamente a cada 7 ms e então disponibilizados para o controlador para que o mesmo haja sobre os atuadores, mantendo sempre um percurso aproximadamente retilíneo. Caso o robô se encontre na primeira faixa (menor que 100 mm medidos do sensor longe da parede) o controlador gerará uma diferença de velocidades nos motores, acelerando mais o da esquerda e desacelerando o da direita fazendo o carro tender à direita. Caso o robô esteja na terceira faixa, distante mais do que 140 mm da parede esquerda, o contrário acontecerá e o motor da direita é acelerado enquanto o da esquerda é desacelerado, a fim de obter um movimento para a esquerda.

No último caso, quando robô se encontra na faixa delimitada entre 100 e 140 mm, o mesmo deve proceder com velocidades emparelhadas dos motores que forneçam o movimento mais retilíneo possível. A necessidade da existência dessas faixas é manter o controle realimentado que se ajuste em tempo real a qualquer fator que venha causar perturbações e desequilíbrios ao sistema.

4.2 CONTROLE DOS MOTORES ATRAVÉS DO PWM

A técnica escolhida para a finalidade de controle de velocidades dos motores foi o PWM. Esta é amplamente utilizada em aplicações robóticas por ser indicada para aplicações com motores de corrente contínua.

O PWM, em breves palavras, representa o controle da potência entregue ao sistema sem necessariamente alterar o nível de tensão ao qual estarão submetidos os componentes

elétricos. Isso é obtido através do chaveamento da alimentação do sistema em altas frequências (centenas de vezes por segundo) a ponto de que o sistema é alimentado sem maiores problemas ao passo em que se controla a quantidade de potência entregue.

Na Figura 18 são apresentadas cinco modalidades de modulação disponíveis em dispositivos Arduino. Ainda nela são apresentados os valores de Duty Cycle (ou o percentual do pulso que carrega energia) em valores percentuais junto aos seus correspondentes valores em dentro da lógica do microcontrolador.

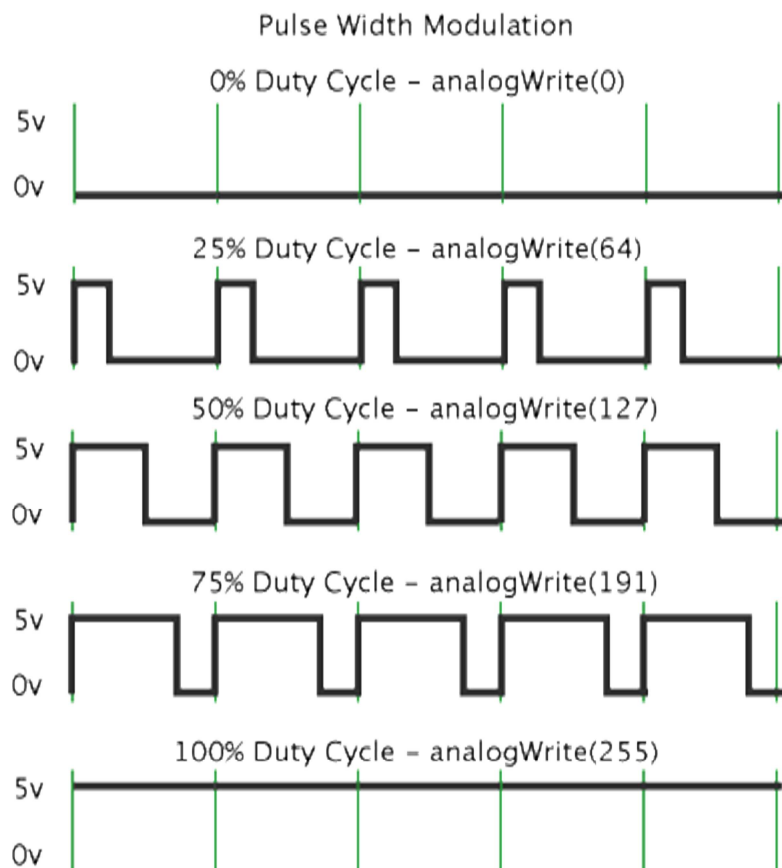


Figura 18 – Modalidades de PWM aplicáveis à plataforma Arduino

Os valores utilizados foram próximos aos valores limiares o que garante maior sensibilidade para pequenos ajustes e variações. Além do controle da operação dos motores, o PWM também oferece acionamento suave dos mesmos.

4.3 A ESTRATÉGIA DA MÃO ESQUERDA

Além da estratégia de controle adotada, é necessário também adotar uma metodologia explícita que garanta ao sistema completar a tarefa designada com sucesso de forma generalizável. Nesse aspecto foi assumida a tática conhecida como Regra da Mão Esquerda ou a Regra LSRB (do inglês Left/Straight/Right/Back) .

A regra da mão esquerda consiste da priorização sistemática de algumas alternativas em detrimento de outras. A sigla em inglês sugere a ordem de prioridades a serem assumidas e são como segue (PANDIAN, KARTHICK, & KARTHIKEYAN, 2012):

1. Se houver possibilidade, vire à esquerda;
2. Caso contrário e se houver possibilidade, siga em frente;
3. Caso contrário e se houver possibilidade, vira à direita;
4. Caso contrário, retorne.

É evidente que a esquerda é sempre a prioridade, por isso a denominação da regra. Apesar de simples, essa metodologia permite a completa exploração de um ambiente desconhecido uma vez que ao se esgotarem todas as possibilidades o robô retornará (um giro de 180 graus) o que inverterá completamente as referências, tornando o que ora era direita em esquerda, e permitindo sua exploração prioritária.

Um ponto desvantajoso na estratégia é sua prisão em *loops*. A regra, como diversas outras táticas adotadas para a navegação de sistemas robóticos, apresenta tal limitação, também conhecida como mínimo local. Vale salientar que no presente trabalho as condições serão estabelecidas como ideias, ou seja, o sistema não encontrará dificuldades ou limitações operacionais, com o intuito apenas de validar sua generalização.

Em um labirinto simples qualquer, são limitadas as possibilidades de bifurcações que podem ser encontradas por um robô explorador. Elas são em número de oito e são como listadas (VANNOY, 2009):

1. Curva a direita;
2. Curva a esquerda;
3. Esquerda ou direita (“T”);
4. Esquerda, direita ou frente (“*Four way*”);
5. Esquerda ou frente;

6. Direita ou frente;
7. Caminho sem saída;
8. Fim do labirinto.

Um robô que apresente um sistema de controle e manobras que cumpra essas oito possibilidades tem a capacidade de resolver qualquer formato de labirinto.

4.4 O ALGORITMO

Os dispositivos Arduino apresentam linguagem proprietária, com bibliotecas específicas para periféricos como sensores, *displays*, *drivers*, entre outros. A linguagem de programação é baseada em C/C++ com alguns pormenores específicos dos Arduino com a finalidade de transformá-la em uma linguagem de mais alto nível, o que facilita na integração de componentes e periféricos eletrônicos.

Os códigos para Arduino são divididos em no mínimo três partes básicas. A primeira corresponde à definição das variáveis associadas às grandezas físicas que os dispositivos tratarão e seus respectivos pinos associados na placa de desenvolvimento. A segunda parte é utilizada para especificar o modo que cada pino se comportará, seja entrada ou saída, ou mesmo acionar e calibrar periféricos ou portas.

A última parte de um código em Arduino consiste no bloco de ação, onde os comandos principais são inseridos a fim de obter o resultado pretendido. Esse bloco é repetido na frequência do *clock* do microcontrolador a menos que períodos diferentes sejam explicitados ao longo do código fonte.

No presente código, apresentando na seção Apêndice, o primeiro bloco é dedicado à definição das variáveis relacionadas ao controle dos motores, sendo que cada motor necessita de três variáveis para seu funcionamento. Uma delas é utilizada para a definição da velocidade de rotação dos motores, e as outras duas são atribuídas à polaridade das ligações elétricas nos motores, que determina o sentido de rotação como horário ou anti-horário.

Além das variáveis dos motores, neste mesmo bloco são definidas as variáveis utilizadas nas medições através dos sensores. Cada sensor apresenta quatro pinos, dois sendo referentes à alimentação e os demais responsáveis pela emissão e recepção do pulso sonoro. Esses dois últimos devem ser definidos em aspecto de tipo de variáveis e ter seus pinos

associados.

No segundo bloco do código, conhecido como *setup*, as mesmas variáveis definidas em termos de entrada e saída. Isso é essencial uma vez que as portas disponíveis no controlador podem funcionar em ambas as modalidades, garantindo maior versatilidade às suas aplicações. Além disto, foi também declarada a porta serial disponível no dispositivo para a comunicação local com o microcomputador e leitura de informações extraídas das leituras dos sensores e entregues aos motores.

Por fim, o bloco chamado de *loop*. Esse pedaço do código é responsável pela execução das instruções sobre as variáveis definidas e outras que foram oportunamente inseridas para transferir informações entre funções.

Inicialmente são criadas as funções de manobras e inicialização de componentes mecânicos. As manobras elaboradas são três, que garantem ao robô a capacidade de resolução integral das oito possibilidades de bifurcações (ou nós) possivelmente encontrados em um labirinto simples. Elas são:

1. Noventa graus à direita;
2. Noventa graus à esquerda;
3. Cento e oitenta graus (retornar).

Essas funções são criadas e ajustadas de forma empírica. O primeiro passo para a sua elaboração é a escolha do sentido da manobra. Uma vez decidido faz-se os motores girarem com mesma velocidade, porém com sentidos de rotação opostos, o que fará o robô girar em um eixo fixo, sem translação. A angulação do giro é diretamente associada ao tempo em que os motores estarão acionados, sendo que quanto maior for este, maior é a angulação desenhada.

É inserida também uma função de inicialização, muito pertinente à aplicação prática do presente trabalho devido à grande inércia apresentada pelos motores em seu acionamento. Seu tempo de operação é pequeno, se limitando a algumas frações de segundo, suficientes para suavizar a partida.

Além desta, está presente a função de parada rápida, que garante aos motores uma parada sutil, onde a inércia dos motores não é recuperada uma vez que todos os pinos são mantidos em nível alto, os motores estão energizados, porém os mesmos não recebem informações de velocidade oriunda das portas PWM do controlador e, portanto não operam, apenas permitindo que as respostas das funções subsequentes sejam rápidas, com curto transiente proveniente da inércia da parada.

O controle da dirigibilidade do carro é também implementado sob funções. Três funções são responsáveis por essa seção do código: a primeira executa o acionamento dos dois motores ao mesmo tempo, em mesma velocidade, fazendo tender a uma trajetória retilínea; a segunda é responsável pelo ajuste para a direita, quando existe a necessidade de afastar o robô da parede esquerda, sendo possível pela imposição de velocidades diferentes nos motores, sendo o motor esquerdo operado em maiores velocidades; a última é o oposto da antepenúltima, sendo o carro ajustado para a esquerda, se aproximada da parede esquerda, pela rotação do motor direito em maior velocidade que o esquerdo.

Por fim, essas funções são livremente utilizadas no bloco de código subsequente, responsável por atribuir inteligência à tarefa e garantir a controlabilidade do carro em sua navegação. A escolha por condensar as ações em funções advém da necessidade de criar um código enxuto que permita a reutilização de blocos de forma irrestrita, sem a necessidade de manter trechos de código redundantes.

5 TESTES E SIMULAÇÃO

Esta seção é destinada à exposição dos testes desenvolvidos a fim de garantir a operabilidade do sistema assim como a simulação realizada para validar o sistema que gerencia as ações do robô.

5.1 MODIFICAÇÕES NO PROTÓTIPO

Uma vez iniciados os testes, foram percebidas necessidades de alteração na morfologia própria do protótipo ou mesmo a substituição de alguns componentes. O primeiro protótipo, como apresentado na seção 3.2, foi constituído através da ideia inicial que se tinha de um corpo funcional. Entretanto, a aplicação prática destituiu alguns dos conceitos estimados, o que corrobora uma vez mais a amplitude da utilização do Mínimo Produto Viável.

Um primeiro ponto a ser explanado é o posicionamento dos sensores. Como comentado anteriormente, os sensores foram primeiramente posicionados de forma retangular, noventa graus entre si.

Esta conformação não apresentou bons resultados durante a operação. Foi percebido que existem dois aspectos cruciais para a determinação da geometria da conformação dos sensores, sendo eles:

1. A capacidade de percepção de vão e aberturas;
2. A capacidade de percepção das paredes e obstáculos quaisquer.

A conformação retangular do protótipo 1 apresentou problemas nos dois aspectos. Referente à percepção das aberturas existentes no labirinto, o mesmo apresentou atrasos, sendo que a estrutura do robô deveria entrar inteiramente no espaço aonde o vão se encontra para percebê-lo. No tocante da aferição das distâncias para as paredes, os sensores só obtêm medidas precisas quando o eixo do robô está alinhado paralelamente às paredes. Caso contrário, pequenas variações angulares do eixo do robô causam grandes variações das medidas de distância.

Dadas as condições acima, a solução elaborada para o problema foi a inclinação dos sensores laterais para um ângulo arbitrário que garanta melhores resultados para os dois

aspectos levantados anteriormente.

Uma série de medições e testes foi desenvolvida a fim de obter um ângulo que fornecesse um equilíbrio entre a percepção dos vãos e das paredes. Foram testados ângulos de trinta, quarenta e cinco e sessenta graus. O primeiro e o último beneficiam de forma exclusiva apenas um dos critérios, oferecendo resultados pobres à contrapartida. Já o ângulo de quarenta e cinco graus garante um equilíbrio entre a qualidade do resultado aferido das paredes e das aberturas do labirinto.

Na Figura 20 é apresentado o protótipo com a nova organização dos sensores.

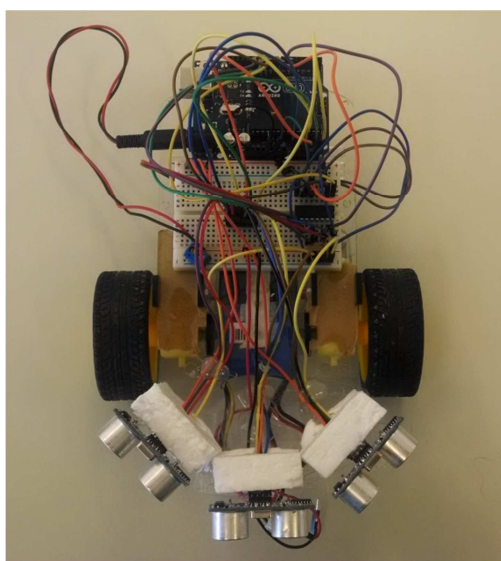


Figura 19: Conformação dos sensores em 45 graus.

Outro ponto que veio a demandar por alterações foi a alimentação do sistema. Inicialmente foi adquirida uma bateria com células compostas por lítio e polímeros, que tem a capacidade de descarga de 1.45 ampère-hora, com taxa de descarga de corrente de *inrush*⁴ de uma vez a taxa nominal. Durante os testes os motores não foram acionados simultaneamente, ou por vezes com velocidade inferiores às determinadas em código. A solução pensada foi a substituição da bateria em questão por uma com capacidade de descarga de acionamento superior à demandada.

No caso específico de motores, a corrente de *inrush* é algumas vezes superior a taxa nominal, então foi pensada numa bateria com cerca de vinte vezes maior capacidade de corrente de acionamento que a nominal (vulgarmente conhecido com 20C). A taxa de

⁴ Corrente demandada para o rompimento de inercia em dispositivos eletromecânicos.

descarga é de 1.5 ampère-hora.

Por fim, devido a um curto-circuito ocorrido durante a etapa de testes, o regulador de tensão da placa de desenvolvimento Arduino Mega 2560 ficou comprometido. A placa em si se manteve operacional, mas a inexistência de um regulador de tensão a mesma só apresentava atuação razoável quando conectado pela porta serial, que extraía tensão regulada diretamente do computador.

A solução definitiva seria comprar um novo dispositivo, entretanto, uma vez que a versão mais simplificada do controlador chamado de Arduino Uno apresenta quantidade de portas analógicas e digitais suficientes para a aplicação do robô, o mesmo foi escolhido, em detrimento do primeiro, devido seu menor custo. A Figura 19, apresentada nesta seção, mostra o segundo protótipo, com sensores reorganizados, bateria 1500 mAh 20C e Arduino Uno.

5.2 TESTES DE MANOBRAS E DIRIGIBILIDADE

Uma das partes fundamentais para a realização da tarefa de resolução de um labirinto é o perfeito controle das manobras de rotação em noventa e cento e oitenta graus assim como o controle da dirigibilidade. A qualidade da execução das manobras e movimentos frontais afetam diretamente a medição distâncias e posicionamento do robô ao seu término assim como o controle da dirigibilidade, uma vez que para grandes angulações em relação às paredes podem levar à perda da calibração e eventuais choques com as paredes do labirinto.

A técnica utilizada para as manobras é através do movimento diferencial dos motores e rodas, em velocidades equivalentes em sentidos opostos para curvas ou retorno e velocidades equivalentes em sentidos equivalentes para movimentos frontais e de ajustes. Para cada manobra, operada sobre um dado terreno, é estipulado um tempo de atuação dos motores. Para o tempo adotado, o motor executará uma angulação total, que deve ser ajustada através do tempo especificado em código para o movimento.

As manobras e os respectivos movimentos dos motores, considerando como referência o motor direito (sentido horário para movimento frontal e sentido anti-horário para movimento de ré) são:

1. Movimento frontal retilíneo: ambos motores no sentido horário e *duty cycle* de 120 em ambos;

2. Movimento de ajuste à esquerda: ambos os motores no sentido horário, motor esquerdo com *duty cycle* de 100 e motor direito com *duty cycle* de 200;
3. Movimento de ajuste à direita: ambos os motores no sentido horário, motor esquerdo com *duty cycle* de 200 e motor direito com *duty cycle* de 100’
4. Movimento de curva à esquerda: motor esquerdo no sentido anti-horário, motor direito no sentido horário, ambos com *duty cycle* de 120 por 800 milissegundos;
5. Movimento de curva à direita: motor esquerda no sentido horário, motor direito no sentido anti-horário, ambos com *duty cycle* de 120 por 900 milissegundos;
6. Movimento de giro em 180 graus: motor esquerda no sentido horário, motor direito no sentido anti-horário, ambos com *duty cycle* de 120 por 2 segundos.

Os dados apresentados foram compilados em forma da Tabela 2, como segue:

Tabela 2: Composição de movimentos do robô

Tipo de Movimento	Motor Esquedo	Motor Direito	Duty Cycle Esquerdo/Direto
Frontal Retilíneo	Sentido Horário	Sentido Horário	120/120
Ajuste à esquerda	Sentido Horário	Sentido Horário	100/200
Ajuste à direita	Sentido Horário	Sentido Horário	200/100
Curva à esquerda	Sentido Anti-Horário	Sentido Horário	120/120
Curva à direita	Sentido Horário	Sentido Anti-Horário	120/120
Girao em 180 graus	Sentido Horário	Sentido Anti-Horário	120/120

Os resultados foram obtidos pelo método de tentativa e erros, ajustando os valores de tempo até que as funções fossem executadas com precisão. Os quatro movimentos básicos estão registrados em vídeo.

5.3 SIMULAÇÃO DO LABIRINTO

Os primeiros testes e experimentos foram bem sucedidos e com resultados registrados para apresentação. Entretanto, por tratar-se de um desenvolvimento empírico de um novo

dispositivo, muitos dos equipamentos foram levados à exaustão e outros perderam a funcionalidade por completo.

No caso específico do sistema mecânico, os *drivers* são sistemas eletrônicos demasiadamente sensíveis e foram necessárias diversas trocas por superaquecimento e consequente fusão dos circuitos integrados. O custo unitário dos chips não é elevado e portanto a substituição foi corriqueira e sem maiores problemas.

Os motores e o chassi, com o uso e choques mecânicos foram perdendo a calibração inicial e passaram a influenciar diretamente nos resultados obtidos. Os motores, que apresentam custo pouco elevados, foram substituídos uma vez que apresentaram elevado escorregamento e consequente elevada corrente de *inrush* que inviabilizava sua utilização dentro da perspectiva do projeto e sistema de alimentação.

Mesmo com novas peças, os resultados, apesar de melhorados, continuaram demandando substituição de peças e estruturas, como o chassi. Devido ao custo elevado, foi optado pela substituição do teste físico pela simulação do sistema de navegação, uma vez que este é o foco central do trabalho, que garante a autonomia e inteligência ao sistema e pode ser facilmente adaptado para quaisquer sistemas mecânicos que supram as condições básicas necessárias para a aplicação.

Portanto, um sistema periférico adicional foi acoplado ao robô sendo capaz de receber o sinal fornecido pelo controlador de forma analógica e interpreta-lo digitalmente em um display de cristal líquido através do uso de uma biblioteca proprietária.

Foram adicionadas *tags* aos blocos do código para substituir a atuação dos motores por uma mensagem correspondente, para validar as ações do veículo dentro do labirinto, que está apresentado na Figura 20.



Figura 20: Labirinto da simulação

As primeiras funções experimentadas foram as de dirigibilidade, apresentadas nas Figuras 21 e 22. O controle da direção do carro ocorre pela captação dos sinais dos sensores que informarão ao controlador a distância lateral para a esquerda.

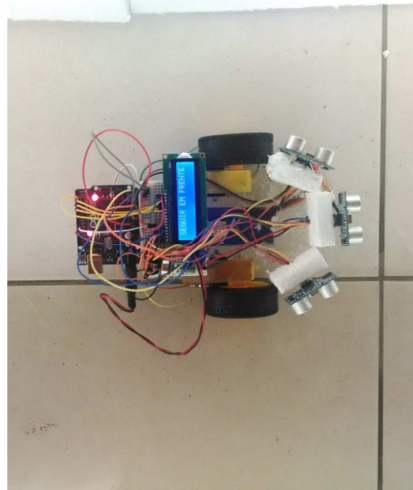


Figura 21: Robô dentro da faixa de ação frontal com mensagem “Seguir em frente”

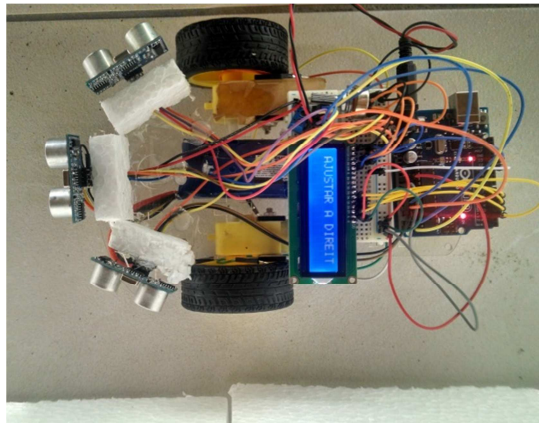


Figura 22: Robô quando muito próximo à lateral esquerda imprimindo “Ajustar à direita”.

As funções responsáveis pelas manobras também foram simuladas. Na Figura 23 o robô se depara com um vão à direita, enquanto os sensores frontal e esquerdo percebem a existência de parede, o que demanda o movimento de noventa graus para a direita.

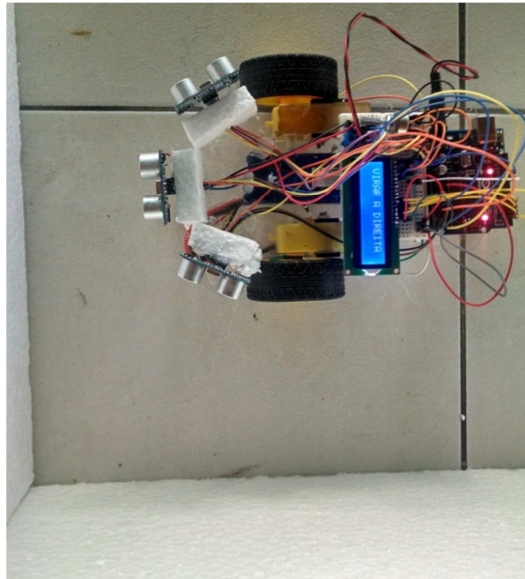


Figura 23: Robô em uma curva para a direita imprimindo a mensagem “Virar à direita”

A situação de obstrução total, conhecido como *dead end*, é percebido pelos sensores e interpretado pelo controlador quando é notada a presença de paredes laterais e distância frontal é inferior a um limite mínimo estipulado para a operação segura, sem choques com a parede do labirinto quando o robô for desacelerado até parar. A Figura 24 apresenta tal situação percebida pelos sensores e representada no *display*.

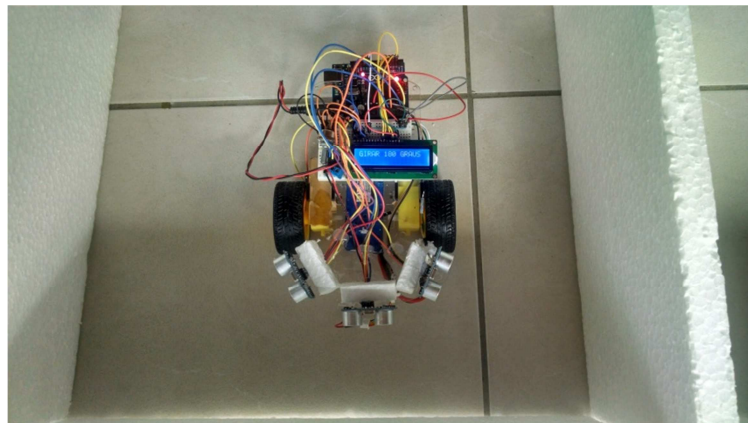


Figura 24: Robô em uma obstrução total, imprimindo a mensagem “Girar 180 graus”

Por fim, uma vez encontrados pelo menos dois vãos “infinitos”, ou seja, que ultrapassem as dimensões sensíveis ao sensor (caso específico do presente trabalho) ou superem as dimensões máximas do labirinto (caso estas sejam conhecidas), o controlador informará aos demais sistemas que o objetivo foi alcançado, finalizando sua operação. Na Figura 25 é apresentada tal situação, onde é impressa a mensagem “Fim do labirinto”.

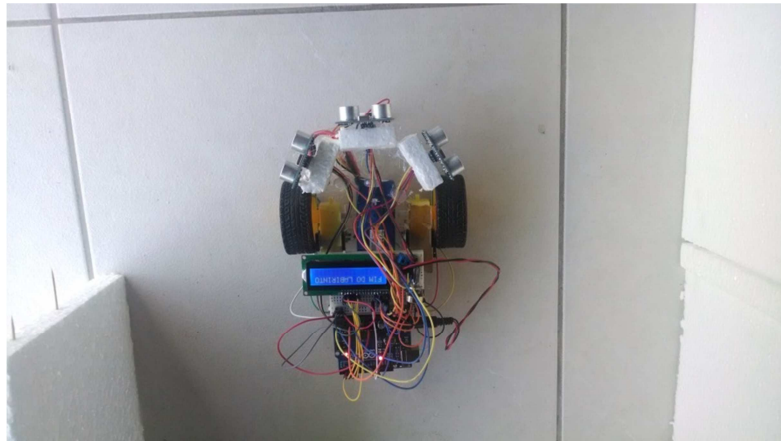


Foto 25: Robô no fim do labirinto, imprimindo a mensagem “Fim do labirinto”

Esses resultados podem ser generalizados para outras situações não explicitamente representadas no texto e para outros modelos de labirinto, através de uma fina calibração do dispositivo para as novas condições.

6 CONCLUSÃO

O trabalho desenvolvido foi uma ampla plataforma de testes para o exercício e captação de conhecimentos diversos que, juntos, formam a ciência robótica. Um projeto de componentes de fácil aquisição e com conceitos teóricos de simples assimilação, mas com um alcance significativo para quem deseja adentrar o âmbito da robótica móvel.

Os principais elementos do protótipo permitiram uma abordagem interdisciplinar de forma integrada e com efeitos globais para o dispositivo. Conhecimentos mecânicos foram os mais difíceis ao longo do trabalho por se tratarem de novos desafios dentro de uma formação elétrica e eletrônica. Essas últimas, por sua vez, foram extrapoladas uma vez que o conhecimento teórico geralmente se mostra inoperante quando diversas variáveis provenientes de ambientes reais são inseridos.

A elaboração do código fonte responsável pela operação do protótipo foi uma oportunidade importante para conhecer a praticidade e versatilidade da plataforma *Arduino* quando aplicado a projetos e processos de automação. Apesar de se tratar de uma aplicação específica em robótica, os controladores *Arduino* são também utilizados em automação residencial e de pequenos processos industriais e a interação com os periféricos e componentes de sistema é padrão, sendo conhecimento largamente intercambiável.

A alimentação do sistema foi uma disciplina que exigiu testes reiterados e especulações diversas, uma vez que afetava diretamente todos os subsistemas do protótipo. A troca de baterias foi influenciada pela corrente de *inrush* dos motores e a conseqüentemente exigiram reprojetos das conexões e cabos, para comportarem maiores correntes. Por outro lado, o controlador, apesar de não demandar altas correntes, apenas apresentara bons desempenhos quando alimentados por uma fonte na faixa entre 9 e 12 volts, exigindo baterias com maiores tensões que as comercializadas localmente. Ao todo foram substituídos quatro tipos de bateria até atender todas as especificações.

Os sensores também tiveram importante papel na construção de um protótipo otimizado uma vez que estes eram a única fonte de informação sobre o meio. Captar esses dados de forma aprimorada oferece claramente benefícios à realização da tarefa.

Com respeito aos resultados, o robô respondeu de forma esperada no que se refere a sistema de controle e algoritmo, uma vez que a simulação obteve resultados significativamente satisfatórios. A tomada de decisão foi precisa, com pequenos ajustes

condizentes com as flutuações dos sensores.

Já a parte física, por ser majoritariamente fabricada a mão, com conexões e ligações frágeis, não se comportou como esperado. A roda omnidirecional, como mencionado previamente, inseriu perturbações ao sistema, por se comportar de forma aleatória, assim como o desgaste elementos mecânicos.

Ainda neste tocante, os motores e chassi foram os maiores desafios para a concretização do trabalho. Na medida em que os testes eram executados, os choques mecânicos e mesmo a utilização repetitiva ao longo dos meses levou à sua deterioração e impactou diretamente no controle da navegação, exigindo medidas alternativas para a validação do sistema como a simulação do movimento no labirinto físico a partir dos sensores, posicionando o robô nos nós.

Por fim, no geral o sistema foi uma plataforma de integração de conceitos pilares ao campo da robótica móvel, ainda permitindo a utilização de uma metodologia sofisticada de desenvolvimento de produtos, atingindo as expectativas didáticas e investigativas do trabalho.

6.1 MELHORAMENTOS E VERSÕES FUTURAS

Alguns pontos de dificuldade sugeriram inovações interessantes para a obtenção de melhores resultados através do protótipo desenvolvido. Os sensores, como base da interação com mundo físico, estão sujeitos às intempéries da operação. Uma indicação para melhores resultados é a fixação dos sensores em estruturas rígidas e leves e parafusá-las diretamente ao corpo do chassi. Isso possibilitará ajustes finos da angulação e a manutenção destas quando ocorrerem impactos físicos inerentes à operação.

Em termos de mobilidade, a roda giratória, apesar de permitir movimentos em todas as direções num plano bidirecional, apresenta dificuldades de posicionamento quando se muda a direção ou sentido da trajetória, o que acaba por dificultar o controle da dirigibilidade do veículo. Uma roda omnidirecional esférica atende às demandas do robô e não apresenta a mesma limitação citada.

Por fim, como uma versão alternativa, que apesar de mais complexa em termos mecânicos, um veículo com quatro rodas controladas diferencialmente oferece a mesma solução ao problema proposto mas de forma mais estável e simplificado no aspecto lógico.

7 REFERÊNCIAS

- ADÔRNO, B. V., AGUIAR, C. S., & BORGES, G. A. (09 de 2005). Planejamento de Trajetória para Robô Omni Utilizando Algoritmo Mapa de Rotas Probabilístico. *VII SBAI*, pp. 1-8.
- DIFFEN. (2014). *Labyrinth vs. Maze*. Acesso em 22 de 05 de 2014, disponível em Diffen: http://www.diffen.com/difference/Labyrinth_vs_Maze
- EDLINGER, T., & PUTTKAMER, E. (1994). *Exploration of an Indoor-Environment by an Autonomous Mobile Robot*. Kaiserslautern: University of Kaiserslautern.
- GONZÁLEZ, R., RODRÍGUEZ, F., & GÚZMAN, J. (s.d.). *Comparative Study of Localization Techniques for Mobile Robots on Indirect Kalman Filter*. Almería: Universidade de Almería.
- KAVRAKI, L., & LATOMBE, J.-C. (1994). Randomized Preprocessing of Configuration Space for Fast Path Planning. *IEEE*, pp. 2138-2145.
- MISHRA, S., & BANDE, P. (2008). *Maze Solving Algorithms for Micro Mouse*. IEEE Computer Society.
- NETO, V. F., & CAMPOS, M. F. (s.d.). *Localização e Mapeamento Simultâneos em Robótica Cooperativa com Planejamento: Uma Revisão*. Belo Horizonte: FUCAPI e UFMG.
- PANDIAN, A. J., KARTHICK, R., & KARTHIKEYAN, B. (2012). Maze Solving Robot Using Freeduino and LSRB Algorithm. *International Journal of Modern Engineering Research*, pp. 92-100.
- PAUCAR, L. (16 de 03 de 2000). *Sistemas Especialistas*. Acesso em 16 de 05 de 2014, disponível em Inteligência Artificial: [http://www.dee.ufma.br/~lpaucar/teaching/ia2000-1/cap4.html#Definição de sistema especialista \(SE\)](http://www.dee.ufma.br/~lpaucar/teaching/ia2000-1/cap4.html#Definição%20de%20sistema%20especialista%20(SE))
- RIES, E. (2011). *The Lean Startup*. Nova Iorque: Crown Business.
- RUSSELL, S., & NORVIG, P. (2010). *Artificial Intelligence: A modern approach*. Upper Saddle River, New Jersey: Pearson Education.
- SANTOS, E. (23 de 12 de 2009). *O que é a Lean Startup*. Acesso em 22 de 05 de 2014, disponível em Manual da Startup: <http://www.manualdastartup.com.br/blog/o-que-e-a-lean-startup/>
- SIEGWART, R., & NOURBAKHSI, I. R. (2004). *Autonomous Mobile Robots*. Cambridge, Massachusetts: A Bradford book - The MIT Press.
- SILVA, B. d. (2011). *Planejamento de Rotas*. São Paulo: ICMC USP.

STACHNISS, C., & BURGARD, W. (s.d.). *Exploring Unknown Environments with Mobile Robots using Coverage Maps*. Freiburg: University of Freiburg.

THRUN, S. (2002). *Robotic Mapping: A Survey*. Pittsburgh, EUA: Carnegie Mellon University.

UEFAP. (2014). *Mazes for Discussion*. Acesso em 22 de 05 de 2014, disponível em UEFAP: <http://www.uefap.com/speaking/exercise/mazes/mazes.htm>

VANNOY, T. R. (2009). *Design a Line Maze Solving Robot*.

APÊNDICE

Código Fonte

```
//DEFINIÇÃO DE VARIÁVEL/PINO
#define vel_motor_esq 10 // vel motor esquerda
#define vel_motor_dir 11 // vel motor direito
#define e1 8 // pinos 8 e 9 controlam o sentido de rotação do motor esquerdo
#define e2 9 //
#define d1 12 //pinos 12 e 7 controlam o sentido de rotação do motor direito
#define d2 7 //

int trigger_frente = A4;
int echo_frente = A5;

int trigger_esq = A2;
int echo_esq = A3;

int trigger_dir = A0;
int echo_dir = A1;

//MODO DE CADA VARIÁVEL/PINO
void setup()
{
  //Serial.begin(9600);

  pinMode(trigger_frente, OUTPUT);
  pinMode(echo_frente, INPUT);

  pinMode(trigger_esq, OUTPUT);
  pinMode(echo_esq, INPUT);

  pinMode(trigger_dir, OUTPUT);
  pinMode(echo_dir, INPUT);

  pinMode(vel_motor_esq, OUTPUT);
  pinMode(vel_motor_dir, OUTPUT);

  pinMode(e1, OUTPUT);
  pinMode(e2, OUTPUT);
  pinMode(d1, OUTPUT);
  pinMode(d2, OUTPUT);

}
//DECLARAÇÃO DAS FUNÇÕES
```

```

void initialize()
{
  analogWrite(vel_motor_esq, 120); //bloco de inicialização para inercia em manobras
  analogWrite(vel_motor_dir, 120);

  digitalWrite(e1, HIGH);
  digitalWrite(e2, LOW);
  digitalWrite(d1, HIGH);
  digitalWrite(d2, LOW);

  delay(200);
}

void stop1()
{
  analogWrite(vel_motor_esq, 0); //bloco de paradas rápidas;
  analogWrite(vel_motor_dir, 0); //
  digitalWrite(e1, HIGH); //
  digitalWrite(e2, HIGH); //
  digitalWrite(d1, HIGH); //
  digitalWrite(d2, HIGH); //
}

void forward ()
{
  analogWrite(vel_motor_esq, 120); // Movimento frontal
  analogWrite(vel_motor_dir, 120); //

  digitalWrite(e1, HIGH);
  digitalWrite(e2, LOW);
  digitalWrite(d1, HIGH);
  digitalWrite(d2, LOW);
}

void adjust_left()
{
  analogWrite(vel_motor_esq, 100); // Ajustar para esquerda
  analogWrite(vel_motor_dir, 200); //

  digitalWrite(e1, HIGH);
  digitalWrite(e2, LOW);
  digitalWrite(d1, HIGH);
  digitalWrite(d2, LOW);
}

void adjust_right()
{
  analogWrite(vel_motor_esq, 200); // Ajustar para direita
  analogWrite(vel_motor_dir, 100); //

```

```

    digitalWrite(e1, HIGH);
    digitalWrite(e2, LOW);
    digitalWrite(d1, HIGH);
    digitalWrite(d2, LOW);
}

void turn_left()
{
    analogWrite(vel_motor_esq, 120); // Virar para esquerda
    analogWrite(vel_motor_dir, 120); //

    digitalWrite(e1, LOW);
    digitalWrite(e2, HIGH);
    digitalWrite(d1, HIGH);
    digitalWrite(d2, LOW);
    delay(800);
}

void turn_right()
{
    analogWrite(vel_motor_esq, 120); // Virar para direita
    analogWrite(vel_motor_dir, 120); //

    digitalWrite(e1, HIGH);
    digitalWrite(e2, LOW);
    digitalWrite(d1, LOW);
    digitalWrite(d2, HIGH);

    delay(900);
}

void back ()
{
    analogWrite(vel_motor_esq, 120); // Retornar em 180 graus
    analogWrite(vel_motor_dir, 120); //

    digitalWrite(e1, HIGH);
    digitalWrite(e2, LOW);
    digitalWrite(d1, LOW);
    digitalWrite(d2, HIGH);

    delay(2000);
}

//BLOCO DE AUTONOMIA

void loop()
{
    long duracao_frente, duracao_esq, duracao_dir, direita, esquerda, frente;

```



```

digitalWrite(trigger_frente, LOW); // função matemática para a transformação do sinal de
tempo do sensor em distância para o obstáculo
delayMicroseconds(2);
digitalWrite(trigger_frente, HIGH);
delayMicroseconds(5);
digitalWrite(trigger_frente, LOW);
duracao_frente = pulseIn(echo_frente, HIGH);
frente = duracao_frente/29/2;

```

```

digitalWrite(trigger_esq, LOW);
delayMicroseconds(2);
digitalWrite(trigger_esq, HIGH);
delayMicroseconds(5);
digitalWrite(trigger_esq, LOW);
duracao_esq = pulseIn(echo_esq, HIGH);
esquerda = duracao_esq/29/2;

```

```

digitalWrite(trigger_dir, LOW);
delayMicroseconds(2);
digitalWrite(trigger_dir, HIGH);
delayMicroseconds(5);
digitalWrite(trigger_dir, LOW);
duracao_dir = pulseIn(echo_dir, HIGH);
direita = duracao_dir/29/2;

```

```

if(frente > 10 && esquerda <= 20 || frente == 0) // se caso houver caminho livre a frente
ele segue esta lógica abaixo:

```

```

{
    if(esquerda > 10 && esquerda < 14)
    {
        forward();
    }
    else if (esquerda < 10)
    {
        adjust_right();
    }
    else if (esquerda >14)
    {
        adjust_left();
    }
}

```

```

else
{
    if(esquerda <= 20 && direita <= 20 && frente <= 10) ;
    {
        initialize();
        back();
    }
}
if(esquerda >20 && direita>20);
{
    initialize();
    turn_left();
}

if(direita <=20 && esquerda>20 );
{
    initialize();
    turn_left();
}
if(direita<=20 && esquerda<=20);
{
    initialize();
    turn_right();
}
if (direita==0&&frente==0 || esquerda==0&&frente==0)
{
    stop1();
}

}

```