



Construção de um Robô Veicular Autônomo

Wesley Silva Dias¹, Bruno Silvério Costa², Camilo Alves Carvalho²

¹Graduando do Curso de Bacharelado em Sistemas de Informação do Instituto Federal da Bahia – IFBA / Campus Vitória da Conquista – Bolsista do CNPq – e-mail: wesleys.dias@hotmail.com

²Professores do Instituto Federal da Bahia – IFBA – Campus Vitória da Conquista – e-mail: {brunosilverio, camilocarvalho}@ifba.edu.br

Resumo: A automação de processos por meio da robótica tem sido há anos objeto de estudos e pesquisas em diversas instituições de ensino e pesquisa, uma vez que, além da contribuição óbvia à sociedade, também acarreta desenvolvimento de áreas estratégicas de conhecimento, como a Engenharia e a Informática. Este projeto visa construir um robô veicular autônomo, modular, que possua heurísticas de descoberta que confirmem certo grau de autonomia. Ele será inicialmente aplicado em sistemas seguidores de linha. Sua carenagem permitirá algum nível de carga física, e sua estrutura de controle será flexível para atuar com diferentes sensores, em conjunto ou separadamente.

Palavras-chave: robótica, arduino, seguidores de linha

1. INTRODUÇÃO

A utilização da automação de processos por meio da robótica tem sido há anos objeto de estudos e pesquisas em diversas instituições de ensino e pesquisa, uma vez que, além da contribuição óbvia à sociedade, também acarreta desenvolvimento de áreas estratégicas de conhecimento, como a Engenharia e a Informática (PAZOS, 2000).

Os Veículos Operados Remotamente (*Remotely Operation Vehicles* – ROVs) são alguns dos primeiros protótipos desenvolvidos nestes estudos. Suas dimensões reduzidas e relativa simplicidade de locomoção – a mecânica dos motores associados a eixos e rodas é amplamente dominada – facilitam sua aplicação em determinadas demandas, tais como localização, mapeamento, inserção em locais de difícil acesso como tubulações, entre outros (ROSARIO, 2005).

Estes dispositivos possuem diferentes modalidades de autonomia. O controle remoto, permitindo que seu controlador esteja fisicamente distante do aparelho, pode utilizar cabos ou transmissão sem fio, por meio de radiofrequência. Também podem ser dotados de heurísticas que dispensem em parte ou totalmente o controle manual, podendo serem citados os seguidores de linha (*line followers*), que usam sensores fotossensíveis para se localizarem e percorrerem um caminho traçado. Outros métodos mais sofisticados, que utilizam outros sensores, também são objetos de estudo (GEORGINI, 2005; GROOVER et al, 1998).

Este projeto visa construir um robô veicular autônomo, modular, que possua heurísticas de descoberta que confirmem certo grau de autonomia. Ele será inicialmente aplicado em sistemas seguidores de linha. Sua carenagem permitirá algum nível de carga física, e sua estrutura de controle será flexível para atuar com diferentes sensores, em conjunto ou separadamente, podendo serem substituídos ou complementados em diferentes fases de sua execução / teste.

2. MATERIAL E MÉTODOS

Para a construção da unidade robótica, foi adotada uma arquitetura baseada na placa Arduino Uno¹. Essa placa, além de possuir custo reduzido, permite a geração de codificações complexas em menor tempo, graças ao *Framework* disponibilizado pelo fabricante. Além de se tratar de uma arquitetura livre e aberta, permite que o desenvolvedor tenha acesso ao projeto original e a seus códigos-fonte.

Objetivando obter uma melhor mobilidade no ambiente, o robô foi construído em formato circular, com duas placas acrílicas sobrepostas em diferentes níveis. A ausência de arestas (ângulos/quinas) facilita o deslocamento do mesmo em ambientes desestruturados em função desta geometria circular, permitindo

¹ <http://www.arduino.cc/>

menos problemas (agarramentos) em relação às diversas configurações ambientais possíveis. O interior vazado do mesmo, possibilita o alojamento de hardwares, bem como permite a acomodação dos cabos necessários interligando os diversos componentes utilizados. A Figura 1 apresenta a vista dos dois níveis da estrutura, interligados por espaçadores plásticos.

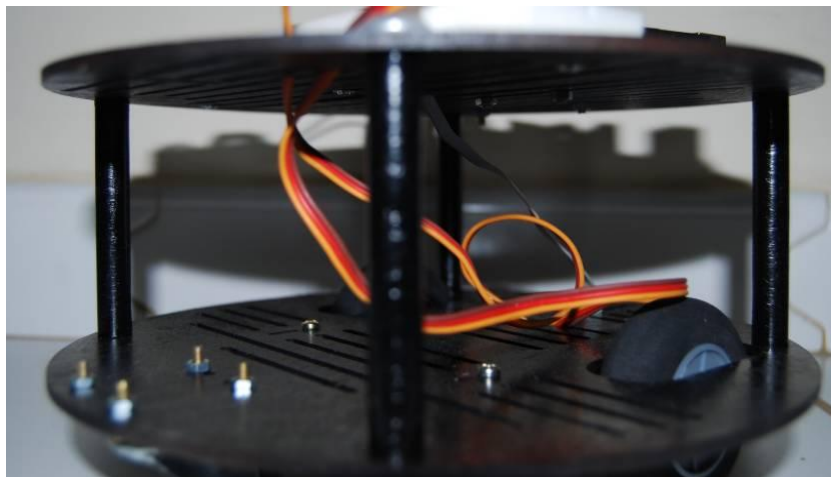


Figura 1 – Visão frontal apresentando níveis e espaçadores

Para sustentar e permitir a locomoção do módulo, foram desenvolvidas duas rodas em espuma de EVA. O processo de desenvolvimento foi simples, tratando-se de um corte circular e acabamento da peça através de um processo de lincamento utilizando uma furadeira comum e uma lixa d'água (RADIO-CONTROLE, 2007). A roda foi então fixada em um carretel plástico após o acabamento. A visão da roda acabada pode ser visualizada na Figura 2.



Figura 2 – Roda de EVA desenvolvida

Esta roda foi afixada por sua vez em um servomotor modificado. Os servomotores permitem em geral 180° de movimento em torno de seu eixo. Este movimento é normalmente limitado por um pequeno batente plástico no interior do servo, além de um potenciômetro interno. Ao se retirar esses elementos, introduzindo-se uma ponte resistiva no lugar do potenciômetro, é possível a adaptação do componente para giros contínuos de 360° (PATSKO, 2006). O servomotor modificado é demonstrado na Figura 3.



Figura 3 – Servomotor modificado

Para a percepção do seu ambiente, especificamente linhas no chão, foi desenvolvida uma pequena placa com sensores de reflexão infravermelha. O tipo de percepção adotada para a confecção do robô é a digital, o que resulta em uma redução na complexidade do *firmware* desenvolvido. Foi adotada a configuração básica de sensores em linha desenvolvida pela Pololu Robotics (POPOLU, 2012). O esquema elétrico do circuito é demonstrado na Figura 4.

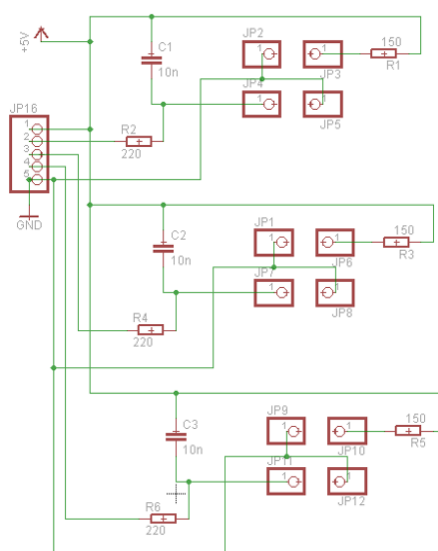


Figura 4 – Esquema elétrico da placa de sensores

A placa desenvolvida é apresentada na Figura 5, exibindo o cabo de conexão, os sensores reflexivos e demais componentes.



Figura 5 – Placa de sensores desenvolvida

O código para o seguimento de linha utilizado foi baseado no algoritmo de inteligência artificial

Agente Reativo Simples. A percepção efetuada pelos sensores é enviada à placa Arduino para processamento. O processamento objetiva determinar se o robô está em um desses estados: fora de uma linha, em movimento para fora da linha pela esquerda, em movimento para fora da linha pela direita ou exatamente em cima da linha. Diante dessas percepções, o agente reativo simples irá efetuar as correções de direção necessárias. Um algoritmo simplificado do processo, utilizado na construção do firmware, é demonstrado na Figura 6.

```
void agente_reativo_simples(Servo motor_dir, Servo motor_esq, int
sEsq, int sDir, int sFrente)
{
  int sE = digitalRead(sEsq);
  int sD = digitalRead(sDir);
  int sF = digitalRead(sFrente);
  int temp=status();
  if ((sE==1) && (sD==1) && (sF==1))
  {
    motor_dir.write(paradoD);
    motor_esq.write(paradoE);
  }
  else if ((sE==0) && (sD==0) && (sF==0))
  {
    motor_dir.write(frenteD);
    motor_esq.write(paradoE);
  }
  else if ((sE==0) && (sD==0) && (sF==1))
  {
    motor_dir.write(frenteD);
    motor_esq.write(frenteE);
  }
  else if ((sE==0) && (sD==1) && (sF==0))
  {
    motor_dir.write(frenteD);
    motor_esq.write(paradoE);
  }
  else if ((sE==1) && (sD==0) && (sF==0))
  {
    motor_dir.write(paradoD);
    motor_esq.write(frenteE);
  }
}
```

Figura 6 - Agente reativo simples implementado no robô

A fim de permitir a autonomia desejada, foi desenvolvido o ambiente completamente observável da Figura 7. O ambiente trata-se de um grafo não-dirigido, permitindo a circulação do robô em qualquer direção sobre a extensão de suas arestas. A fim de possibilitar a sinalização ao robô dos estados intermediários, de início e de fim do percurso, marcas paralelas foram introduzidas nos nós. Para o caso adotado, duas barras paralelas perceptíveis pelo sensor esquerdo indicam estados intermediários, três barras paralelas também à esquerda indicam o estado final, e uma barra perpendicular à aresta, perceptível pelos sensores à esquerda e à direita, o estado inicial.

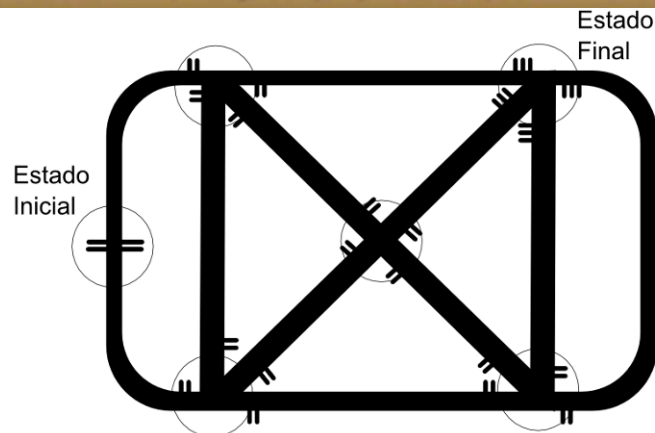


Figura 7 – Ambiente completamente observável utilizado para teste

O deslocamento do robô através do ambiente foi permitido pela adoção de uma heurística simples. Essa heurística permitiu que o robô estabelecesse uma rota que correspondesse a um bom caminho possível do estado inicial ao estado final. Isso foi possível já que o robô, para encontrar o seu estado objetivo, acaba explorando todas as arestas dos estados intermediários. Durante o percurso, o robô registra em sua memória interna as arestas e nós explorados, possibilitando o cálculo heurístico sobre o modelo gerado. A heurística adotada foi a do menor caminho aqulado, sendo descrito pela fórmula da Figura 8.

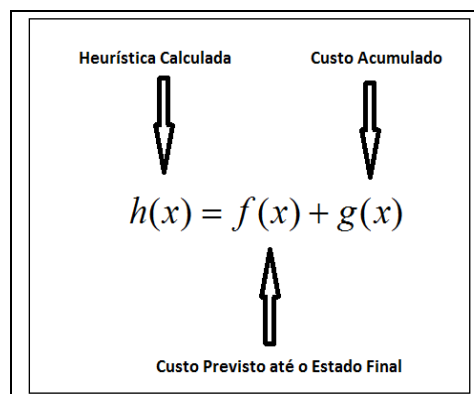


Figura 8 – Heurística utilizada para a determinação do menor caminho

Para a determinação do caminho a ser percorrido, foi utilizado o Algoritmo A*, que recebeu a heurística citada para suas estimativas. Para a efetivação do cálculo do custo previsto da heurística, o algoritmo de busca gulosa também foi utilizado, considerado o mais apropriado ao tipo de busca apresentada (RUSSEL; NORVIG, 2004). Esse bom caminho calculado é fruto da computação sobre a exploração inicial do robô, permitindo sua execução como caminho de volta, ou seja, do estado final à origem, possibilitando a demonstração do resultado obtido

3. RESULTADOS E DISCUSSÃO

A solução mecânica construída apresenta boa desenvoltura para ambiente estruturados, adaptando-se com facilidade ao cenário proposto, isto é, ao seguimento de linhas, visto que sua geometria circular (sem arestas/quinas) permite movimentos rápidos de busca pelos dados no ambiente. A modificação dos servomotores mostrou-se eficaz e possibilitou desenvolver uma solução precisa e de baixo custo para uma maior gama de movimentos (360°) que, aliado à geometria adotada na construção da carenagem, permitiu que as manobras fossem mais diversas e adaptativas (giro em torno do próprio eixo).

O algoritmo para Agentes Reativos Simples demonstrou ser eficaz para todos os testes realizados de seguimento de linha, gerando correções rápidas e precisas das rotas do robô. A placa de sensores possibilitou a percepção de linhas com baixo grau de contraste, o que viabiliza a construção de seguidores de linha em

situações de percepção extremamente precárias. O sistema construído foi eficaz na execução sobre o ambiente controlado de testes, realizando com facilidade os percursos de linha propostos. A heurística utilizada permitiu a realização do cálculo do percurso de melhor caminho, possibilitando ao robô o grau de autonomia desejado. A Figura 9 apresenta uma sequência de fotos exibindo o deslocamento do robô, desde o estado final até o inicial, demonstrando o resultado do Algoritmo A* de busca heurística.

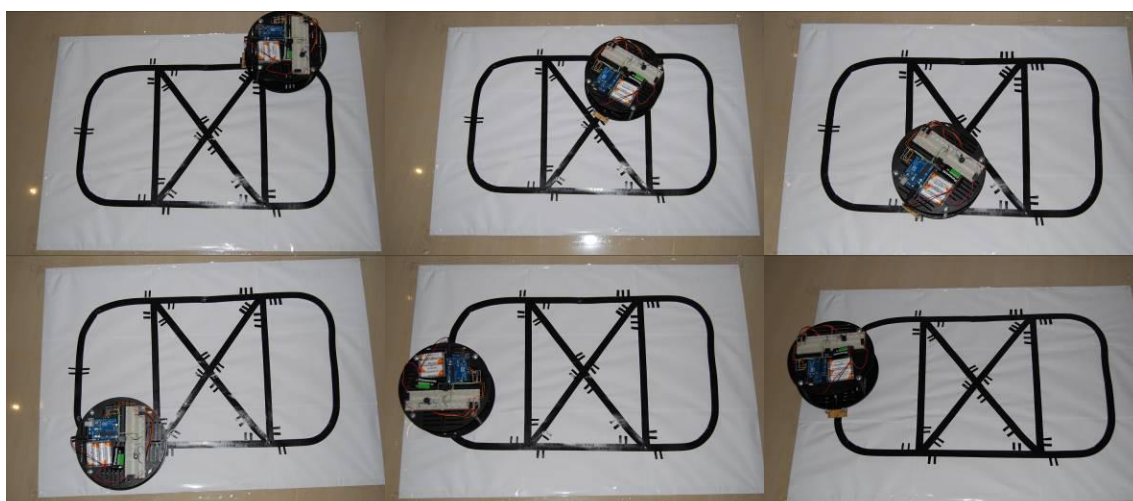


Figura 9 – Sequência de imagens demonstrando o trajeto do robô para o resultado do cômputo de melhor caminho

Para a verificação da eficiência da construção, foram realizados testes utilizando as metodologias de interoperabilidade e de caixa preta, objetivando avaliar a inter-relação entre os algoritmos e componentes mecânicos do projeto (RIOS & MOREIRA, 2006). O protótipo foi submetido a uma sequência de 100 execuções contínuas sobre o ambiente de testes, apresentando os resultados demonstrados pela Tabela 1:

Tabela 1 – Execuções de teste do robô sobre o ambiente estabelecido

Nº de Casos	Tempo de Execução da Busca	Tempo de Execução do resultado da Heurística	Número de retrocessos
41	Entre 40 e 45 segundos	Entre 15 a 20 segundos	Entre 2 a 5
52	Entre 20 e 40 segundos	Entre 20 a 25 segundos	Entre 5 a 7
7	+ de 45 segundos	Entre 15 a 20 segundos	Entre 7 a 8

Pode-se observar pelos dados da tabela que o Algoritmo de Busca A* apresentou-se estável na determinação do melhor caminho possível, exibindo resultados com uma variação pouco significativa. Devido às movimentações bruscas promovidas pelos atuadores frente aos resultados do Agente Reativo Simples, algumas percepções dos vértices ficaram comprometidas, resultando na variação no número de retrocessos. A carga da bateria utilizada para a alimentação do sistema também variou com o tempo de uso, o que resultou na também variação dos tempos de execução.

Dessa forma, o teste de caixa-preta apresentou um resultado satisfatório, visto que o sistema pode ser percebido como um todo unificado, apresentando as saídas previstas para as entradas fornecidas, isto é, foi capaz de apresentar uma solução otimizada para o ambiente fornecido. O teste de interoperabilidade também demonstrou bons resultados, já que os dois algoritmos puderam ser executados conjuntamente para a solução do problema proposto. No entanto, pode ser percebido um comprometimento mínimo do Algoritmo A* em virtude da não suavização das atuações requeridas pelo Agente Reativo Simples.

4. CONCLUSÕES

A utilização de sistemas robóticos seguidores de linha possibilita demonstrar de maneira simples a eficácia dos sistemas autônomos. Processos industriais de variados tipos podem ser automatizados por meio destas soluções, tais como deslocamentos em linhas de produção ou sistemas de estocagem, onde se podem



substituir a utilização de trilhos, principalmente em locais em que não se permite modificações arquiteturas desta natureza. Outra aplicação bastante comum para este tipo de implementação é o acesso a ambientes insalubres ou arriscados, como limpeza em dutos de ventilação.

A utilização de Algoritmos de Busca A* permitiu o estabelecimento de um melhor caminho entre os estados estabelecidos, possibilitando ao robô estabelecer uma otimização ao caminho inicialmente estabelecido. A adaptabilidade ao ambiente que, apesar de ser completamente observável pelo agente, também é desconhecido inicialmente, possibilitou a demonstração da eficácia deste algoritmo na construção de sistemas autônomos. Apesar do algoritmo não retornar sempre o melhor caminho, como foi demonstrado, apresenta uma possível solução à otimização de percursos.

A proposta deste trabalho também contribui pela utilização de uma plataforma promissora para programação de sistemas embarcados: o Arduino, dotada de um microcontrolador ATmega328², aberta em sua documentação e com capacidade de processamento suficiente para algoritmos de complexidade razoável, tal como o de Agente Reativo Simples e de Busca A*, que permite a criação de uma solução elegante, de baixo consumo de processamento e complexidade reduzida de programação. Isso demonstra que os referidos algoritmos são adequados, quando utilizados conjuntamente, na exploração e otimização de percursos em ambientes totalmente observáveis.

O método linear de adaptação demonstrado pelo Agente Reativo Simples às modificações de percepção, demonstrou bons resultados (conforme Tabela 1), realizando no entanto, correções bruscas do movimento. Essas correções acabaram por conferirem um certo grau de aleatoriedade ao experimento, gerando variações no número de retrocessos do procedimento de Busca A* vinculada. Dessa forma, propõe-se para trabalhos futuros a implementação da aplicação de funções de transição com curvas par suavizar os movimentos de correção, bem como ampliar a quantidade e diversidade de sensores para explorar outros tipos de autonomia, tais como radiofrequência – RFID³ ou mesmo câmeras de vídeo, com heurísticas de reconhecimento visual.

REFERÊNCIAS

GEORGINI, M. **Automação Aplicada**. São Paulo: Érica, 2005.

GROOVER, M. P.; WEISS, M.; NAGEL, R.N.; ODREY, N.G. **Robótica: Tecnologia e Programação**. McGraw-Hill, 1998.

McROBERTS, M. **Arduino Básico**. São Paulo: Novatec, 2011.

PATSKO, L. F. **Tutorial Adaptação de Servomotores**. Disponível em <http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_mecanica_-_adaptacao_de_servo-motores.pdf>. Acesso em 02/05/2012.

PAZOS, F. **Automação de Sistemas e Robótica**. Rio de Janeiro: Axcel, 2000.

POPOLU. **QTR-8RC Reflexive Array Sensor**. Disponível em <<http://www.pololu.com/catalog/product/961>>. Acesso em 02/05/2012.

RADIO-CONTROLE. **Como fazer rodas para aeromodelismo**. Disponível em <<http://radio-controle.com/como-fazer-rodas-para-trem-de-pouso-by-rafael>>. Acesso em 02/05/2012.

RIOS, E.; MOREIRA, T. **Teste de software**. 2ª Ed. Rio de Janeiro: Alta Books, 2006.

ROSÁRIO, J. M. **Princípios de Mecatrônica**. São Paulo: Pearson/Prentice Hall, 2005.

2 <http://www.atmel.com/devices/atmega328.aspx>

3 <http://www.rfid.org/>



RUSSEL, S. J.; NORVIG, P. **Inteligência Artificial**. 2ª Ed. Rio de Janeiro: Elsevier, 2004.