

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/242100254>

# NAVEGAÇÃO DE ROBÔS MÓVEIS UTILIZANDO O MÉTODO DE CAMPOS POTENCIAIS

Article · January 2002

CITATIONS

4

READS

639

2 authors, including:



[Anna Helena Realí Costa](#)

University of São Paulo

151 PUBLICATIONS 1,099 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Knowledge Transfer in Reinforcement Learning [View project](#)



AA-REACT [View project](#)

# NAVEGAÇÃO DE ROBÔS MÓVEIS UTILIZANDO O MÉTODO DE CAMPOS POTENCIAIS

Rafael Neto Pacheco

Departamento de Engenharia de Computação e Sistemas Digitais  
Escola Politécnica da Universidade de São Paulo  
[rafael.pacheco@poli.usp.br](mailto:rafael.pacheco@poli.usp.br)

Anna Helena Reali Costa

Departamento de Engenharia de Computação e Sistemas Digitais  
Escola Politécnica da Universidade de São Paulo  
[anna.reali@poli.usp.br](mailto:anna.reali@poli.usp.br)

## RESUMO

Este artigo descreve a implementação de um sistema de navegação para robótica móvel utilizando uma arquitetura baseada em *Motor-Schemas* e no Método de Campos Potenciais. O sistema é constituído por três comportamentos primitivos que, coordenados, dotam o robô da capacidade de navegar na presença de obstáculos em direção a um alvo fixo. Os comportamentos são puramente reativos, possuem codificação contínua e são coordenados dentro do paradigma cooperativo. Um conjunto de sonares e odômetros presentes no robô fornecem a informação sensorial necessária aos comportamentos. A contribuição deste trabalho consiste na definição de modelos e parâmetros adequados, dentro da arquitetura proposta, para uso na plataforma básica do robô móvel Pioneer 2DX, o qual possui capacidade sensorial limitada.

## ABSTRACT

This paper presents an autonomous indoor navigation system for mobile robots using an architecture based on *Motor-Schemas* and on the potential fields method. The system is composed of three primitive behaviors, which enable the robot to move towards a pre-defined goal while avoiding collision with obstacles in the environment. The behaviors defined are purely reactive, their responses are encoded in a continuous functional form, and a cooperative mechanism for behavioral coordination is used. Sonar sensors and encoders are responsible for the robot perceptual stimulus. The contribution of this work lies on the definition of appropriate models and parameters, according to the proposed architecture, for use in the basic Pioneer 2DX mobile robot platform, which has limited sensorial capability.

## 1 INTRODUÇÃO

Robôs móveis são agentes *artificiais ativos*, com *capacidade de locomoção*, imersos no mundo físico real. *Artificiais* por não serem entidades da natureza; *ativos* por atuarem no ambiente, de modo racional e não passivo; *com capacidade de locomoção* por poderem se mover no ambiente [6].

A atuação de um robô está fortemente relacionada à tarefa a ele designada e ao tipo de domínio no qual está inserido. O mundo real é um domínio que apresenta propriedades complexas, exigindo requisitos particulares no projeto e construção dos agentes robóticos.

A atuação de um robô inteligente pode ser reativa, deliberativa ou uma combinação entre elas. A abordagem reativa tem apresentado resultados bastante eficientes quando aplicada a robôs móveis, que atuam em ambientes reais [3,4,5]. Isto se deve por esta abordagem definir uma relação muito clara e direta entre a percepção e a atuação do robô, possibilitando respostas em tempo real dos mesmos. Além disso, a implementação de comportamentos reativos primitivos constitui também uma base adequada para futuras implementações de abordagens híbridas (reativas e deliberativas) para a robótica móvel aplicada a tarefas mais complexas.

A contribuição deste trabalho consiste na definição de modelos e parâmetros adequados ao modelo básico do robô Pioneer, cuja capacidade sensorial é limitada (uma vez que há sonares apenas na parte frontal e os odômetros são imprecisos), de tal forma a implementar o paradigma reativo através de uma arquitetura baseada em comportamentos modelados por *Motor-Schemas*, utilizando o Método de Campos Potenciais.

Três comportamentos primitivos foram construídos para dotar o robô da capacidade de navegar entre obstáculos em direção a um alvo fixo. Estes comportamentos serão aqui detalhados.

## 2 ARQUITETURA BASEADA EM MOTOR-SCHEMAS

A arquitetura utilizada nesta implementação é baseada em comportamentos modelados por *Motor-Schemas* [3]. Um *Motor-Schema* permite, essencialmente, a definição e implementação de um comportamento em dois módulos principais: o módulo de percepção, que é responsável por extrair dos estímulos sensoriais as informações relevantes para o comportamento em questão; e o módulo de codificação do comportamento que, alimentado pelo módulo de percepção, executa o mapeamento dos estímulos sensoriais nas respostas motoras. Os parâmetros que definem uma ação – a qual corresponde à saída de cada *Motor-Schema* – são a magnitude e a direção do movimento. Estes parâmetros são representados por um vetor, e correspondem, respectivamente, à velocidade e à rotação que o robô deve executar, segundo o comportamento em questão.

Neste contexto, definir cada comportamento se resume a dois problemas: como perceber o ambiente e como reagir a esta percepção.

### 2.1 Módulo de Percepção

O módulo de percepção (*perceptual schema*) é composto por algoritmos que extraem dos estímulos sensoriais (i.e. leituras dos sonares, imagens) as informações que tenham significado relevante para um dado comportamento. Caso haja mais de um tipo de sensor, é neste módulo que ocorre a fusão sensorial, ou seja, a interpretação dos diferentes tipos de estímulos do ambiente de forma a compor a percepção global à qual o robô deve reagir. Neste trabalho, apesar de serem usados dois tipos de sensores – sonares e odômetros – não há fusão sensorial, uma vez que estes sensores são utilizados isoladamente.

## 2.2 Módulo de Codificação do Comportamento

A codificação das percepções em ações se dá de forma contínua, utilizando o Método de Campos Potenciais [3,5]. Este método é baseado na construção de campos de força, análogos aos campos elétricos da Física. A construção dos campos pode ser feita através de funções de decaimento que determinam, com base na distância, a magnitude do campo em um determinado ponto. A direção é fornecida pelo tipo de campo, e pode ser radial, constante ou aleatória (no caso de comportamentos de ruído), entre outros.

Para que o comportamento defina a ação a ser executada, basta determinar a direção e a magnitude do campo no ponto em que o robô se encontra em um dado instante (um único vetor), não sendo necessária a computação de todo o campo.

## 2.3 Coordenação dos Comportamentos

A coordenação dos vários comportamentos é feita de forma cooperativa, onde a ação resultante que o robô deve executar deriva das contribuições – em maior ou menor grau – advindas de todos os comportamentos ativos. O Método de Campos Potenciais sugere uma maneira muito simples para a coordenação dos comportamentos: a superposição dos campos. Na superposição, as contribuições (vetores) de cada comportamento são somadas, de maneira que o problema da coordenação se resume a dimensionar corretamente os campos dos diferentes comportamentos de forma que a sua soma resulte no comportamento global desejado para o robô.

## 3 ROBÔ UTILIZADO

O robô utilizado neste trabalho foi o Pioneer 2DX, da ActivMedia Robotics (<http://www.activmedia.com>). Ele possui 8 sonares, dispostos na sua parte frontal, abrangendo 180 graus, que são capazes de detectar objetos entre 10cm e 5m de distância em relação aos sonares. Testes indicaram que objetos de diâmetro inferior a 5cm dificilmente são detectados. A direção de dois sonares consecutivos (com exceção dos laterais) difere em um ângulo de vinte graus. Os odômetros presentes no robô são de 500 marcas (“ticks”), e as rodas têm diâmetros de 19cm.

Suas dimensões são 44 x 38cm. Seu peso é de 9Kg e a máxima carga suportada é de 23Kg.

Foi utilizado um notebook Toshiba Satellite 4025CDT para executar o processamento embarcado nos testes com o robô real.

## 4 DESCRIÇÃO DOS COMPORTAMENTOS

Três comportamentos compõem o sistema implementado para o Pioneer e são aqui descritos: (i) avoidCollision, (ii) moveAhead e (iii) moveToGoal.

### 4.1 Comportamento avoidCollision

Este comportamento tem por objetivo evitar colisões em relação a múltiplos obstáculos.

### 4.1.1 Módulo de Percepção

O módulo de percepção deste comportamento interpreta as leituras dos sonares de modo a identificar os obstáculos presentes no ambiente. Esta interpretação adota a restrição de continuidade espacial do ambiente, no intuito de evitar que duas leituras advindas de um mesmo obstáculo sejam interpretadas como dois obstáculos diferentes. Tal interpretação faria com que o robô reagisse de modo muito intenso ao obstáculo, dificultando a navegação por caminhos estreitos.

A continuidade espacial é determinada por dois fatores: a vizinhança dos sonares e a diferença entre os valores absolutos das leituras. Deste modo, duas leituras correspondem a um espaço contínuo – e, portanto, representam um mesmo obstáculo – se tiverem sido obtidas por sonares vizinhos e se a diferença entre os valores absolutos das leituras (que correspondem às distâncias medidas) não ultrapassar um limiar pré-estabelecido. Este limiar consiste em um parâmetro do comportamento e pode ser ajustado conforme o tipo de ambiente em que o robô deve navegar.

No caso da detecção de um obstáculo por dois ou mais sonares, a posição deste obstáculo é definida como sendo a leitura que indique maior proximidade ao robô.

### 4.1.2 Codificação do avoidCollision

O módulo de codificação deste comportamento, alimentado pelo módulo de percepção, determina qual a ação motora adequada àquela percepção, ou seja, constrói o campo potencial do comportamento, associando cargas repulsivas aos obstáculos.

Como se trata de um campo potencial, este módulo deve fornecer a direção e a magnitude do campo no ponto em que se encontra o robô. A direção é definida como radial. Deste modo, na presença de um obstáculo, o robô sofre uma força repulsiva na direção da reta que une o seu centro de massa ao obstáculo detectado, conforme ilustrado na figura 1. O resultado final do comportamento é a soma vetorial de cada uma das forças devidas a cada obstáculo detectado. A magnitude da força é determinada por uma função de decaimento que depende apenas da distância do centro de massa do robô ao obstáculo em questão. Deste modo, o módulo de codificação deste comportamento é composto pelas seguintes equações:

$$V(d) = e^{(-d + S) / T}$$

$$\phi = -\phi_{\text{robô-obstáculo}}$$

onde V é a magnitude do comportamento, d é a distância do centro de massa do robô ao obstáculo, S é o “stand-off” do robô (distância abaixo da qual a magnitude do comportamento satura em seu valor máximo (normalizado)), T é a constante de tempo da curva de decaimento,  $\phi$  é a direção do vetor resultante do comportamento e  $\phi_{\text{robô-obstáculo}}$  é a direção definida pela reta que une o obstáculo ao centro de massa do robô.

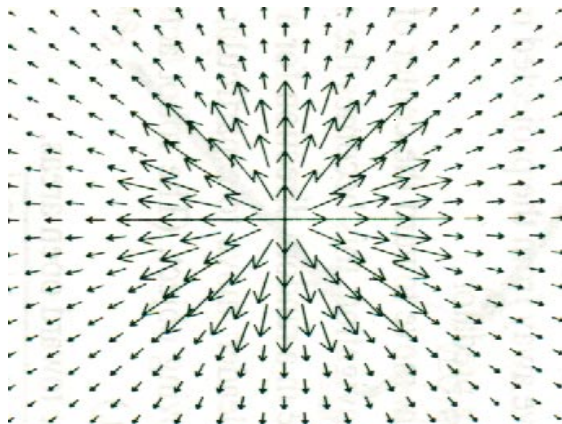


Figura 1. Campo potencial radial repulsivo para um obstáculo puntiforme com decaimento exponencial. [retirado de [3], p. 146]

O uso da exponencial se justifica por esta função apresentar um decaimento controlado (parâmetro T) e por ter a característica de crescer rapidamente em um ponto também controlado (parâmetro S), características bastante adequadas a um comportamento de evitar colisões. O comportamento desta função é exemplificado na figura 2.

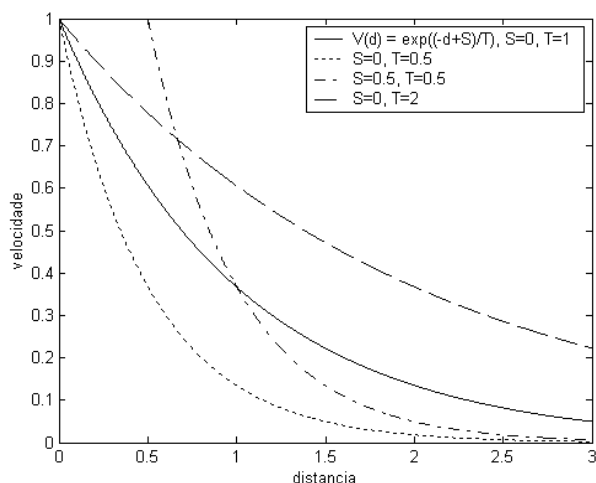


Figura 2. Comportamento da função de decaimento do avoidCollision. Variação dos parâmetros S e T.

A figura 3 ilustra a percepção e a codificação do comportamento avoidCollision na presença de três obstáculos.

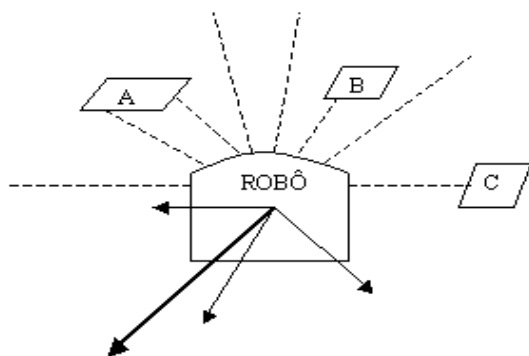


Figura 3. O robô na presença de três obstáculos. Note que, apesar do obstáculo A ser detectado por dois sonares, apenas um vetor é associado a ele. Os vetores associados a cada obstáculo são somados, compondo o vetor resultante (linha de maior espessura).

#### 4.2 Comportamento moveAhead

Este comportamento é responsável por fazer com que o robô sempre tenha uma forte tendência de continuar na direção atual, ou seja, que ele tenha um certo tipo de inércia. Apesar de aparentemente ingênuo, este comportamento tem grande importância no sistema, pois ele compensa a oscilação que pode resultar do comportamento avoidCollision e suaviza significativamente a trajetória do robô, uma vez que contribui para a diminuição da magnitude da rotação executada. Este comportamento também demonstrou importância especial quando o robô navegava por corredores. Neste caso, dado que os sonares abrangem apenas a parte frontal do robô, verificou-se que, sem o comportamento moveAhead, o robô executava um movimento de “zig-zag” entre as paredes. Isto porque, ao perceber uma das paredes laterais, o robô girava para o lado contrário bruscamente, devido à repulsão à parede detectada; no entanto, ao executar este movimento, os sonares agora detectariam a outra parede, causando nova repulsão e gerando a oscilação. Com a introdução do comportamento moveAhead, as rotações foram suavizadas, permitindo que a cada giro para um dos lados, o giro seguinte fosse menor. Desta maneira, a trajetória se estabiliza rapidamente.

O comportamento moveAhead não possui módulo de percepção e o seu módulo de codificação é elementar, consistindo em um campo onde a magnitude é constante (parâmetro do comportamento) e a direção corresponde sempre à direção do robô naquele instante. Portanto:

$$V(d) = cte$$

$$\phi = \phi_{robô}$$

#### 4.3 Comportamento moveToGoal

Este comportamento tem o objetivo de atrair o robô para um determinado ponto no ambiente. A posição inicial do alvo - no sistema global de coordenadas - é informada ao sistema por meio de um agente externo. A posição atual do robô (também no sistema global de coordenadas) é estimada pelo módulo de percepção, tomando como base a posição inicial do robô e acrescentando as informações fornecidas pelos seus odômetros. O módulo de codificação fica encarregado de determinar a direção do movimento, que será a mesma direção dada pela reta que une os pontos que definem as posições do robô e do alvo. A magnitude é constante (parâmetro do comportamento). Portanto:

$$V(d) = cte$$

$$\phi = \phi_{robô-alvo}$$

O resultado desta codificação é um campo radial atrativo de magnitude constante, em direção ao alvo.

Para efeito de visualização, a figura 4 mostra a construção de um campo análogo, porém, diferentemente do comportamento moveToGoal, a magnitude decai com a proximidade ao alvo.

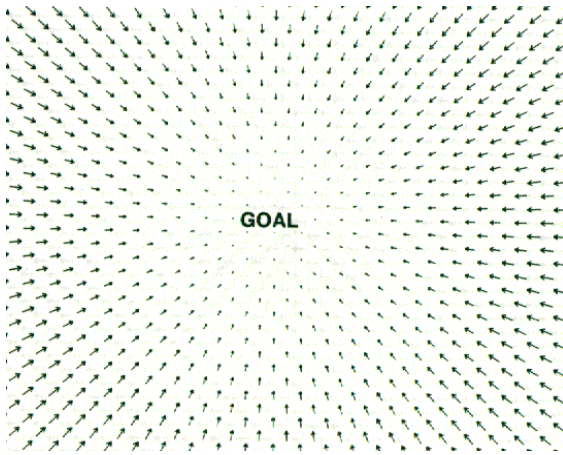


Figura 4. Campo radial atrativo em direção ao alvo. [retirado de [3], p. 148]

#### 4.4 Comportamento Global

O comportamento global é dado pela coordenação dos comportamentos primitivos, que será a soma vetorial das ações definidas por cada um deles. Os Motor-Schemas dos três comportamentos e a respectiva composição vetorial são ilustrados na figura 5.

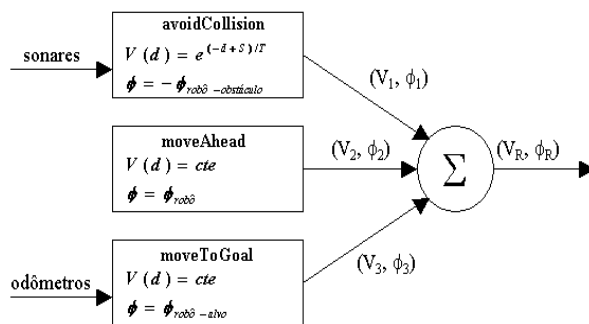


Figura 5. Motor-Schemas dos comportamentos avoidCollision, moveahead, moveToGoal e a composição vetorial.

Para ilustrar a superposição dos campos, a figura 6 mostra o campo e a trajetória resultantes da coordenação dos comportamentos avoidCollision e moveToGoal.

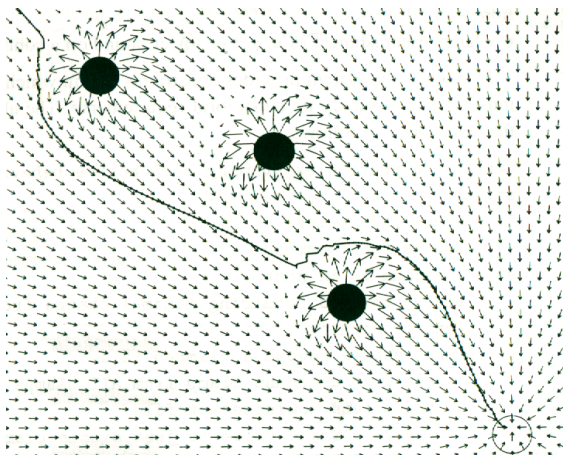


Figura 6. Campos potencial e trajetória resultantes da coordenação dos comportamentos avoidCollision e moveToGoal em ambiente composto por três obstáculos e um alvo. [retirado de [1], p. 116]

## 5 RESULTADOS

A arquitetura foi testada tanto no simulador do Pioneer quanto no robô real, exibindo em ambos os casos uma atuação bastante eficiente. O simulador utilizado foi desenvolvido pela SRI International, e contém modelos de erro realísticos tanto para os sonares quanto para os odômetros específicos do robô utilizado.

A arquitetura reativa proposta foi desenvolvida na linguagem C, através do uso da biblioteca de funções que acompanha o robô, denominada Saphira. A versão da biblioteca utilizada foi a 6.2. Vale dizer que nenhuma modificação no código foi necessária para executar a arquitetura no simulador, uma vez que, aos olhos do programa, o simulador se comporta da mesma maneira que o robô real.

O simulador permite a construção de mundos gráficos fictícios para a navegação do robô. Este artifício gráfico permite a visualização do simulador em funcionamento e, portanto, foi escolhido para mostrar o funcionamento do sistema implementado.

Os parâmetros utilizados no experimento foram os seguintes:

- avoidCollision:  $S=0.3$ ,  $T=0.3$ , limiar de detecção de obstáculos = 50cm;
- moveAhead: magnitude = 0.6;
- moveToGoal: magnitude = 0.4;
- velocidade máxima: 150mm/s;

A escolha dos parâmetros dos comportamentos se mostrou bastante dependente da velocidade máxima permitida para o robô.

Os valores da magnitude dos comportamentos podem assumir valores entre 0 e 1, onde 1 corresponde à velocidade máxima e 0 à mínima. As magnitudes dos comportamentos moveAhead e moveToGoal devem ter sua soma igual a 1, pois desta maneira é possível garantir que no caso em que o robô se dirige em direção a um alvo e se depara frontalmente com um obstáculo (pior caso), a inversão da direção do campo resultante se dará na distância especificada pelo parâmetro  $S$  do comportamento avoidCollision, uma vez que neste ponto a magnitude deste comportamento será máxima (explosão da exponencial), e a direção será diretamente oposta ao vetor resultante da soma dos comportamentos moveAhead e moveToGoal.

O experimento consistiu em colocar o robô inicialmente em uma posição próxima à porta no canto inferior direito de uma sala fictícia com obstáculos, com orientação para o interior da sala, e estabelecer a posição alvo no canto superior esquerdo, extremo oposto da sala. Quando o robô atingir a posição alvo, as posições inicial e alvo serão invertidas, de modo que o robô volte para a posição inicial do experimento. Ao atingir o novo alvo, as posições inicial e alvo serão novamente invertidas, e assim sucessivamente. Deste modo, o robô ficará em “loop” entre os dois extremos da sala.

Na primeira parte do experimento, o robô parte do canto inferior direito da sala e atinge o canto superior esquerdo. Note que o robô desviou com sucesso dos obstáculos, executando uma trajetória bastante eficiente. O ambiente com obstáculos e a trajetória do robô são mostrados na figura 7.

Na segunda parte do experimento, o robô parte do canto superior esquerdo da sala e atinge o canto inferior direito. Observa-se, na figura 8, que o caminho percorrido indica sucesso no desvio dos obstáculos. É importante observar que, conforme esperado, as

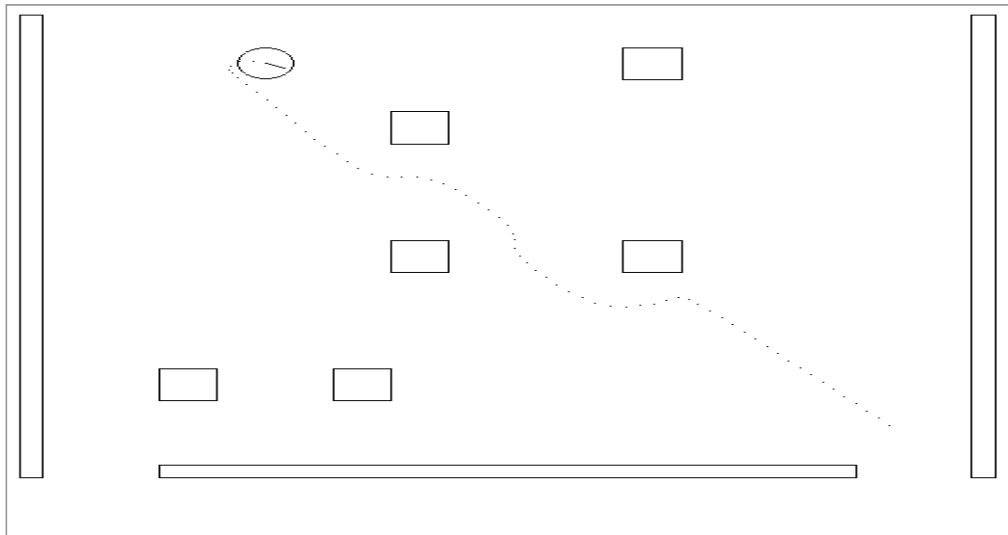


Figura 7. Primeira parte do experimento. Robô executa trajetória eficiente, desviando dos obstáculos e atingindo o alvo.

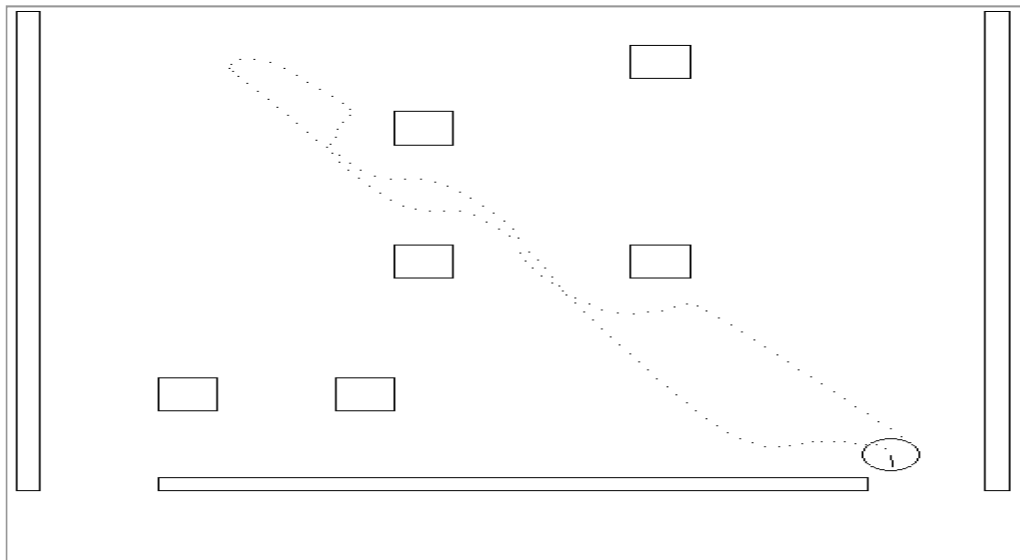


Figura 8. Segunda parte do experimento. Robô volta à posição original, executando trajetória diferente da executada na ida.

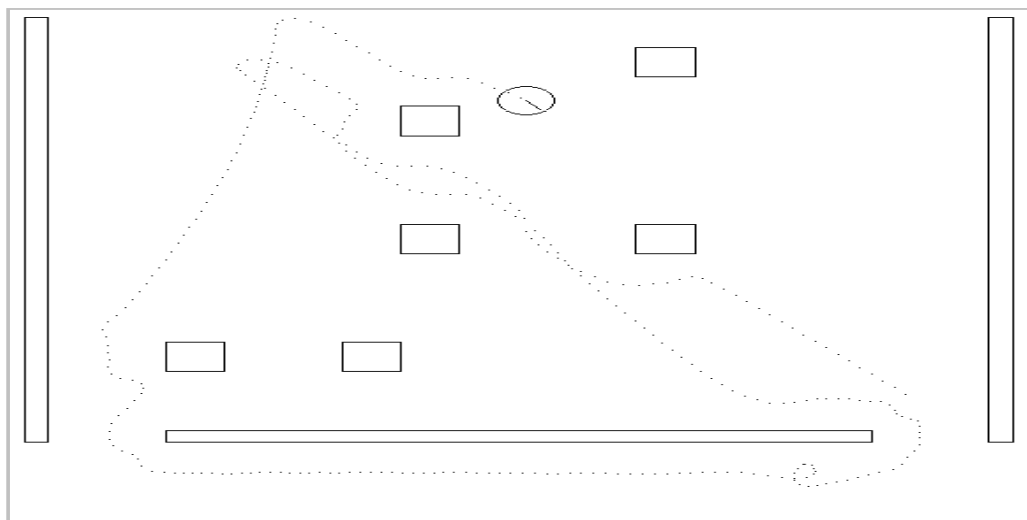


Figura 9. Terceira parte do experimento. Devido ao acúmulo de erros pelo seu sistema de odometria, o robô localiza a posição alvo anterior fora da sala. Devido a atração ao novo alvo e a repulsão à parede, uma trajetória estável, paralela à parede, é executada até a porta, quando o robô consegue reentrar na sala e se dirigir ao alvo.

trajetórias de ida e de volta são distintas, uma vez que o sistema é reativo e que, devido à orientação do robô, as condições de ida e de volta são ligeiramente diferentes.

Um resultado interessante foi obtido na terceira parte do experimento, ilustrado na figura 9. Devido aos erros acumulados por seu sistema de odometria, o robô tem uma percepção incorreta da posição alvo da segunda parte do experimento, localizando-a fora da sala. Ao atingir o ponto que acredita ser a posição alvo, as posições inicial e alvo são invertidas e o robô tenta se dirigir à nova posição alvo, encontrando um obstáculo (parede) à sua frente. Embora o comportamento `moveToGoal` atraia o robô para o alvo, o comportamento `avoidCollision` evita a colisão com a parede, resultando em uma trajetória estável, paralela à parede, até a porta do canto inferior esquerdo do ambiente, quando o robô consegue novamente entrar na sala e se dirigir para o alvo. A trajetória executada nesta última parte do experimento ilustra bem o caráter não-ótimo das trajetórias obtidas em sistemas reativos.

Os sonares do robô demonstraram ineficiência na detecção de quinas. Estas só são detectadas quando o robô já está muito próximo das mesmas, de modo que a colisão não pode ser evitada. Para minimizar este problema foi inserido um mecanismo que verifica, a cada nova percepção do ambiente, se a magnitude da rotação a ser executada pelo robô excede um determinado limiar. Se isto ocorrer, entende-se que houve falha na detecção do obstáculo e este já se encontra demasiadamente próximo do robô, caso contrário a trajetória teria sido mais suave e tal magnitude de rotação não seria necessária. O sistema faz então com que a velocidade translacional seja igual a zero e apenas a rotação é executada, até que uma nova percepção do ambiente seja processada e a magnitude da rotação não ultrapasse o limiar. Com esse mecanismo o robô consegue desviar eficientemente de obstáculos, mesmo quando a sua detecção não é apropriada.

## 6 CONCLUSÃO

A implementação se mostrou, através dos testes, bastante ajustável, devido ao grande número de parâmetros dos comportamentos. Isto era esperado, pois houve a preocupação, durante todo o decorrer do desenvolvimento, de permitir o ajuste dos comportamentos a diferentes ambientes, bem como permitir que estes comportamentos primitivos sirvam de base para tarefas mais complexas, como as que envolvem aprendizado. No entanto, esta flexibilidade traz uma dificuldade no dimensionamento dos parâmetros, principalmente no tocante a velocidade máxima estipulada para o robô. Durante os testes, pôde-se perceber que os parâmetros são extremamente dependentes desta velocidade, o que indica que estes devem ser equacionados também em relação a esta velocidade. Há a intenção, em trabalhos futuros, de eliminar esta dependência, bem como a dificuldade de dimensionar os parâmetros, através do uso de mecanismos baseados em algoritmos adaptativos para gerar automaticamente os parâmetros dos comportamentos.

O ambiente Saphira, que acompanha o robô Pioneer utilizado, mostrou-se bastante apropriado para o desenvolvimento inicial de aplicações. Porém, o desenvolvimento em linguagem C permitiu maior controle sobre a execução do algoritmo.

O simulador que acompanha o Saphira teve importância fundamental no desenvolvimento do projeto, pois facilitou muito a realização dos testes.

O Método de Campos Potenciais visa suavizar a trajetória executada pelo robô, mas para isso é necessário que a capacidade sensorial do robô cubra 360 graus ao seu redor, caso contrário o método tem a sua eficácia diminuída. A capacidade sensorial do nosso robô é apenas frontal e sujeita a falhas, como no caso das quinas. Isto se refletiu, algumas vezes, em mudanças bruscas de direção e variações igualmente indesejadas na velocidade.

## AGRADECIMENTOS

Agradecemos aos usuários da lista de discussão dos robôs e pelos funcionários do fabricante pelas sugestões de implementação e pelos esclarecimentos dados sempre que a documentação era insuficiente. Rafael N. Pacheco agradece ao CNPq/PIBIC pelo suporte. Este projeto está sendo parcialmente financiado pelo NSF/CNPq projeto MAPPEL, Proc. No. 680033/99-8.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ActivMedia Robotics . Pioneer 2 Mobile Robot - Operations Manual, 2000.
- [2] ActivMedia Robotics. Saphira Manual, 1999 .
- [3] ARKIN, R. C. *Behavior-Based Robotics*. MIT Press, 1998.
- [4] MAES, P. (ed.). *Designing Autonomous Agents: theory and practice from biology to engineering and back*. MIT Press, 1993.
- [5] MURPHY, R. R.. *Introduction to AI Robotics*. MIT Press, 2000.
- [6] RIBEIRO, C. H. C.; A. H. R. COSTA & R.A.F. ROMERO. *Robôs Móveis Inteligentes: Princípios e Técnicas*. Em A.T. Martins and D.L. Borges (editors), Anais do XXI Congresso da Sociedade Brasileira de Computação, SBC, Fortaleza, 2001, Vol. 3, pp.257-306.