

# Desenvolvimento de Rotas para Monitorar Ambientes Internos

Heitor L. Polidoro  
*Universidade de São Paulo*  
*ICMC - Instituto de Ciências Matemáticas*  
*e de Computação*  
*São Carlos - SP, Brasil*  
*heitor.polidoro@gmail.com*

Dr. Prof. Denis F. Wolf  
*Universidade de São Paulo*  
*ICMC - Instituto de Ciências Matemáticas*  
*e de Computação*  
*São Carlos - SP, Brasil*  
*denis@icmc.usp.br*

**Resumo**—A robótica móvel é uma área de pesquisa que está obtendo grande atenção da comunidade científica. O desenvolvimento de robôs móveis autônomos, que sejam capazes de interagir com o ambiente, aprender e tomar decisões corretas para que suas tarefas sejam executadas com êxito é o maior desafio em robótica móvel. O desenvolvimento destes sistemas inteligentes e autônomos consiste em uma área de pesquisa multidisciplinar considerada recente e extremamente promissora que envolve; por exemplo, inteligência artificial, aprendizado de máquina, estimação estatística e sistemas embarcados. Dentro desse contexto, esse trabalho aborda o problema de navegação e monitoramento de ambientes utilizando robôs móveis. Dada uma representação do ambiente (mapa topológico) e uma lista com urgências de cada uma das regiões do mapa, o robô deve estimar qual o percurso mais eficiente para monitorar esse ambiente. Uma vez que a urgência de cada região não visitada aumenta com o tempo, o trajeto do robô deve se adaptar a essas alterações. Entre as diversas aplicações práticas desse tipo de algoritmo, destaca-se o desenvolvimento de sistemas de segurança móveis inteligentes.

**Keywords**—Robótica Móvel; Monitoramento; Ambiente Interno;

## I. INTRODUCTION

A robótica é uma tecnologia utilizada para auxiliar ou substituir o homem em tarefas que o ser humano não é capaz de executar; em tarefas que podem ser automatizadas; em tarefas em ambientes com alto grau de periculosidade,

como fundo do mar, incêndios, desarmamento de bombas, áreas com contaminação radioativa ou com gases tóxicos e em tarefas repetitivas como uma linha de produção industrial, por exemplo.

Dentre as muitas definições de robô podemos destacar:

- Dispositivo ou máquina que realiza funções normalmente associadas a seres humanos [1];
- Máquinas que, além de serem capazes de reproduzir tarefas e movimentos implícitos em sua construção, complementam a parte mecânica com dispositivos eletrônicos inteligentes [1];
- Órgão mecânico versátil equipado com atuadores e sensores sob o controle de um sistema computacional [2] e
- Simplesmente um agente artificial e ativo cujo ambiente é o mundo físico [3].

Inicialmente os robôs eram utilizados somente para automação industrial, fixados em posições específicas na linha de montagem, os braços robóticos podem mover-se a uma grande velocidade e precisão para realizar tarefas repetitivas. No entanto esses robôs apresentam uma fundamental desvantagem que é a falta de mobilidade. Um manipulador fixo tem alcance limitado que depende de onde foi colocado enquanto que robôs móveis são capazes de locomoverem-se pela fábrica [4]. Com a evolução tecnológica os robôs passaram a ser utilizados em outras

áreas como: medicina de precisão, ambientes perigosos, ambientes insalubres, na área de entretenimento, serviços domésticos, etc. Nesse contexto surgiram as pesquisas para o desenvolvimento de robôs móveis autônomos, que sejam capazes de atuar em ambientes reais e reagir a situações desconhecidas de forma inteligente [5].

Entre todas essas diversas aplicações da robótica móvel, pode-se citar o robô Sojourner da NASA, que explorou e enviou fotos e outras muitas informações do planeta Marte para a Terra [6], e o robô desenvolvido pela universidade Carnegie Mellon, chamado Groundhog, que explora minas abandonadas, que além do risco de desabamento, em muitos casos também contém gases tóxicos [7].

A robótica consiste em uma área multidisciplinar de pesquisa, envolvendo desde elementos de engenharia mecânica, elétrica, computação até áreas de humanas como psicologia e estudos comportamentais. O que diferencia a robótica móvel de outras áreas de pesquisa em robótica, é a sua ênfase nos problemas relacionados com locomoção em ambientes complexos, que se modificam dinamicamente, compostos tanto por obstáculos fixos quanto por obstáculos móveis. Para operar nesses ambientes o robô deve ser capaz de adquirir e utilizar conhecimento sobre o ambiente, tais como: estimar posições dentro do ambiente (sua posição, de um obstáculo, de um *landmark*, de uma meta), reconhecer obstáculos, e responder em tempo real às situações que podem ocorrer nesses ambientes. As tarefas de perceber o ambiente, localizar-se no ambiente e mover pelo ambiente são problemas fundamentais da robótica móvel [8].

A robótica móvel, além de ser uma área de grande potencial científico, tem atraído a atenção de empresas de tecnologia que cada vez mais investem no desenvolvimento de produtos; como por exemplo os robôs que realizam trabalhos domésticos autonomamente, entre os quais o aspirador de pó Roomba da IRobot [9] e o robô cortador de grama Robomow da Friendly Robotics [10]. Ambos apresentam sucesso comercial.

Robôs móveis podem ser classificados quanto

à sua mobilidade como: humanóides, com pernas, com rodas, com esferas, aéreos e aquáticos. Robôs com rodas são mais simples de serem construídos e mais fáceis de controlar, as rodas permitem uma maior praticidade de locomoção e dão um maior suporte estático aos robôs terrestres [3]. A autonomia de robôs móveis é essencial em ambientes remotos, tais como outros planetas, onde, devido à distância, o tempo de comunicação entre o robô e seu operador não permite a execução de ações em tempo real.

Normalmente, a navegação é a principal tarefa a ser executada por um robô móvel. Esta consiste na localização do robô no ambiente, planejamento de um caminho entre a posição inicial e o destino final e a execução do movimento pelo caminho planejado. Esta tarefa pode ser mais, ou menos complexa, dependendo do ambiente em que o robô se encontra [11] e das informações que ele possui.

A motivação deste projeto é desenvolver um sistema de monitoramento de ambientes internos. Existem diversas aplicações práticas para esse tipo de aplicação. Dentre elas, pode-se citar o desenvolvimento de um robô vigia que monitora um ambiente, dando ênfase para áreas de maior importância; um robô que monitora a temperatura e umidade de ambientes onde esse fator é crítico; um robô que faz coleta de lixo ou correspondência; um robô para fazer limpeza de ambientes; um robô para fazer entregas de remédios a pacientes em um hospital e etc.

Este projeto tem como objetivo o desenvolvimento de uma estratégia eficiente para determinar uma sequência de áreas a serem visitadas em ambientes internos com a finalidade de monitoramento destes ambientes utilizando um robô móvel. O problema a ser resolvido consiste na divisão de um ambiente previamente conhecido em áreas de interesse. A cada uma dessas áreas é atribuído um valor (peso) referente à sua importância de monitoramento. A prioridade com que o robô deve visitar determinadas áreas é calculada com base na importância dessas áreas e no tempo decorrido desde a sua última visita. Áreas de maior importância devem ser

visitadas mais freqüentemente.

## II. DESENVOLVIMENTO E RESULTADOS

Esta dissertação tem como objetivo o desenvolvimento de uma estratégia eficiente para determinar uma seqüência de áreas a serem visitadas em ambientes internos com a finalidade de monitoramento destes ambientes utilizando um robô móvel. O problema a ser resolvido consiste na divisão de um ambiente previamente conhecido em áreas de interesse. A cada uma dessas áreas é atribuído um valor (peso) referente à sua importância de monitoramento. A prioridade com que o robô deve visitar determinadas áreas é calculada com base na importância dessas áreas e no tempo decorrido desde a sua última visita. Áreas de maior importância devem ser visitadas mais freqüentemente.

Para determinar a trajetória do robô foram criadas duas soluções, uma *Offline* (seção II-C1) e outra em Tempo Real seção (II-C4), essas soluções serão confrontadas em dois critérios de avaliação (seção II-A), um baseado no Grau de Urgência Total e o outro baseado na freqüência relativa de cada sala.

### A. Critérios de Avaliação

Os algoritmos e estratégias serão então testados e avaliados. As avaliações consistem na comparação dos algoritmos e estratégias. Serão utilizados dois critérios de comparação, um critério comparando a freqüência relativa de cada sala com sua prioridade relativa, onde o melhor resultado é aquele em que a freqüência relativa se aproximar mais da prioridade relativa, pois se uma sala (a) possui uma prioridade com valor duas vezes maior que uma sala (b), a sala (a) deve ser visitada com o dobro de freqüência do que a sala (b). Esse cálculo é feito somando as diferenças quadráticas entre a prioridade relativa e a freqüência relativa de cada sala ( $\sum_{i=0}^n (P_i/P_t - F_i/F_t)^2$ ). Isto indica que o algoritmo ou estratégia se manteve fiel à definição do problema: *Áreas de maior importância devem ser visitadas mais freqüentemente.*

O segundo critério é um gráfico mostrando a progressão da somatória dos graus de urgência

de todas as salas ( $\sum_{i=0}^n U_i$ ), no qual o melhor resultado consiste em manter o menor valor da somatória dos graus de urgência; mostrando que o algoritmo ou estratégia levou o robô às salas com maior eficiência. O grau de urgência  $U$  é calculado multiplicando-se a prioridade relativa da sala  $P/P_t$  pelo tempo decorrido desde a última visita  $t$  ( $U_i = P_i/P_t \times t_i$ ). A casa visita o  $t$  é zerado.

### B. Metodologia

Para a solução do problema descrito, o robô tem uma descrição completa do ambiente em que atua (mapa). Foram utilizados somente dois sensores: Odômetro para localizar o robô no mapa e o LASER para o desvio de obstáculos. O controle do robô foi desenvolvido utilizando a biblioteca Player/Stage. Para desvio de obstáculos utilizou-se a técnica VFH que já vem implementada no Player e se mostrou suficiente para o projeto.

Cada ambiente tem um mapa topológico. O robô deve utilizar esse mapa para se locomover de uma sala para outra no ambiente. Para os algoritmos e estratégias determinarem a seqüência de salas a serem visitadas, é considerado um grafo completo com todas as salas, pois para determinar o melhor caminho entre uma sala e outra será utilizado o algoritmo de Dijkstra no mapa topológico.

Como um dos critérios de avaliação está relacionado ao tempo que o robô fica sem visitar as salas é possível deduzir que a solução seja um ciclo.

Considera-se:

- Um ambiente com  $S$  salas;
- Um ciclo hamiltoniano  $C$  qualquer;
- $C_i$  é a  $i$ -ésima sala visitada no ciclo;
- Uma velocidade constante do robô (tanto linear quanto angular);
- $\Delta t_i$  o tempo para sair da sala  $i$  e chegar na sala  $i + 1$ .

Como o tempo de viagem entre as mesmas salas é constante todos os  $\Delta t_i$  são constantes, portanto o tempo total do ciclo  $T$  é constante

( $\sum_{i=1}^S \Delta t_i = cte$ ). Se o tempo do ciclo é constante, o tempo que o robô demora para revisitar cada sala é constante igual a  $T$ . Mas isso não é interessante para solução do problema pois o grau de urgência de cada sala é diferente, então se uma sala tem uma prioridade muito alta, seu grau de urgência vai ser muito alto até o robô revisitar-lá.

Para resolver esse problema, basta fazer com que o robô revise essa(s) sala(s) mais de uma vez no ciclo. Então supondo um ciclo de tamanho  $n$  sendo  $n \geq S$  o tempo total do ciclo continua constante ( $\sum_{i=1}^n \Delta t_i = cte$ ). Portanto a solução do problema consiste em encontrar esse ciclo.

### C. Soluções

Foram definidos dois tipos de soluções: uma solução **offline**, ou seja, a solução é calculada em um computador e depois informada ao robô qual sequência de sala ele deve seguir, essa sequência não é alterada. E a outra é uma solução **tempo real**, ou seja, o robô define para qual sala deve ir durante a execução do algoritmo, baseando suas decisões no que está acontecendo no momento. A solução **offline** é dividida em duas partes: **Gerador**, que cria possíveis sequências de salas ótimas, e o **Avaliador** que analisa e computa uma nota para as sequências de salas criadas pelo **Gerador**.

1) *Offline*: Esse método consiste em explorar as diversas possíveis combinações de sequências de salas para encontrar a sequência ótima antes de informar ao robô qual sequência de salas deve seguir. Um programa chamado **gerador**, seguindo uma determinada heurística, gera as possíveis sequências de salas, essas sequências são fornecidas ao programa **avaliador** que, baseado no critério de maior Grau de Urgência Total, analisa a sequência e retorna uma avaliação ao **gerador** que tomará a decisão de: descartar a sequência, preservar a sequência para gerar futuras sequências ou guardar a sequência como possível sequência ótima.

2) *Gerador*: O programa chamado **gerador** é utilizado para gerar os candidatos à sequência de salas ótima do mapa analisado. Esses

candidatos são avaliados pelo programa **avaliador** e de acordo com a avaliação o **gerador** irá guardar a sequência como possível ótima, descartar ou continuar utilizando essa sequência para gerar novas sequências.

O programa **gerador** utiliza uma classe (tipo de variável) chamada *Agente* contendo os seguintes atributos:

- **vertice**: Vértice no qual o agente se encontra no momento;
- **caminho**: Vetor de salas que guarda a sequência de salas que o agente percorreu até chegar no vértice atual;
- **avaliacao**: Avaliação da sequência de salas do agente;
- **tempo**: Tempo em segundos que o robô levou para percorrer a sequência de salas até o vértice atual.

O **gerador** recebe apenas um parâmetro como entrada: o nome do mapa que quer achar a sequência ótima. O programa inicia criando uma sequência percorrendo as salas na ordem numérica (Sala 1, sala 2..) e classificada como possível ótima. A seguir o programa cria um *Agente* em cada sala do mapa para explorar as possibilidades do robô começar em cada sala, esses *agentes* são organizados em uma fila de candidatos.

O laço principal do programa consiste em retirar um *agente* da fila e para cada sala  $i$  é criado um novo *agente*, simulando que o *agente* retirado da fila navegou até a sala  $i$ . Em cada novo *agente* é atualizada a sequência de salas visitadas e o agente é reavaliado. Se a avaliação for um valor negativo o *agente* é inserido na fila de *agentes*. Se o valor retornado for 0 (zero) o *agente* é descartado. Caso o avaliador retorne um valor positivo e esse valor é menor que a avaliação da atual sequência ótima, o novo *agente* contendo a sequência melhor é guardado como possível sequência ótima. Como o algoritmo somente insere na fila os *agentes* que podem gerar sequências ótimas, o critério de parada é quando não existe mais *agentes* na fila, o programa então retorna o *agente* com a sequência ótima.

Exemplo: Em ambiente com quatro salas o gerador vai criar um *agente* e coloca-lo em cada sala. Então a fila de *agentes* ficaria da seguinte forma (*Agente/caminho*):

$$(A/1)(B/2)(C/3)(D/4) \quad (1)$$

Seguindo o laço principal, o *gerador* remove o primeiro *agente* da fila, no caso é o *agente A*, cria uma cópia do *agente A* para cada sala, e simula que esses *agentes* navegaram até sua sala correspondente, atualizando o caminho que esse *agente* percorreu para chegar a essa sala e a avaliação desse caminho.

$$(B/2)(C/3)(D/4)(E/1,2)(F/1,3)(G/1,4) \quad (2)$$

Após feito isso o *gerador* recomeça o ciclo, removendo o primeiro *agente* da fila (no caso *agente B*), criando outros *agentes* e inserindo-os na fila de acordo com a avaliação adquirida (*agentes H, I e J*).

$$(C/3)(D/4)(E/1,2)(F/1,3)(G/1,4) \quad (3)$$

$$(H/2,1)(I/2,3)(J/2,4)$$

O *gerador* permanece nesse ciclo até que não exista mais *agente* na fila.

3) *Avaliador*: O programa chamado *avaliador* é utilizado para avaliar os candidatos a sequência de salas ótima do mapa analisado gerado pelo programa *gerador* (Seção II-C2) e retorna para o *gerador* um valor numérico. Baseado nesse valor o *gerador* irá guardar a sequência como possível ótima, descartar ou continuar utilizando essa sequência para gerar novas sequência.

O *avaliador* recebe três parâmetros como entrada:

- O nome do mapa onde o caminho será avaliado;
- Um vetor de vértices, representando a sequência de salas a ser avaliada;
- Um valor opcional de limite para a avaliação.

Com esses parâmetros o *avaliador* inicia a simulação do robô navegando pelo mapa. A

velocidade linear, que determina a velocidade com que o robô anda para frente ou para trás, é definida pela constante `SIMULACAO_VEL` como 1 m/s. A velocidade angular, que determina a velocidade com que o robô vira, é definida pela constante `SIMULACAO_ROT` como 0,5 rad/s. O tempo que o robô demora para visitar uma sala é definido pela constante `VISITAR_SALA` como 5 s. Todos esses valores foram definidos arbitrariamente.

Durante a simulação, a cada visita de sala o *avaliador* mede o Grau de Urgência Total e salva o maior valor, esse valor vai ser a avaliação do caminho neste mapa. Após a simulação o *avaliador* verifica se a avaliação do caminho for maior que o limite fornecido o valor 0 (zero) é retornado, se nenhum limite for fornecido o *avaliador* utiliza o valor do maior inteiro do compilador. Se o caminho não visita todas as salas o *avaliador* retorna a avaliação com valor negativo, se todas as salas foram visitadas o *avaliador* verifica se o caminho é um *loop*, ou seja, se o caminho começa e termina no mesmo vértice, caso contrário ele também retorna a avaliação em valor negativo. Se o caminho for um *loop*, visita todas as salas e o valor da avaliação for menor que o limite fornecido (ou menor que o maior inteiro do compilador), o valor da avaliação, maior Grau de Urgência Total em todo o percurso, é retornado pelo *avaliador*.

Para critério de parada o *avaliador* compara o estado das salas, ou seja, se para cada sala, o Grau de Urgência e o número de visitas é o mesmo ao término do caminho.

4) *Tempo Real*: A solução em tempo real inicia as prioridades de todas as salas em zero, cada sala recebe um valor correspondente à chance dessa sala gerar uma emergência. Emergência é quando a sala solicita que o robô vá visitá-la, como uma lixeira cheia ou a sala não ser visitada por um tempo mínimo determinado por exemplo. A cada emergência gerada a prioridade da sala é acrescida de uma unidade. O robô segue o paradigma de ir à sala de maior grau de urgência, visita a sala e procura a

próxima sala de maior grau de urgência.

A chance de emergência que cada sala recebe é proporcional às prioridades, para fazer o cálculo basta dividir a prioridade da sala pela soma das prioridades do mapa.

$$p_s = \frac{P_s}{\sum_{i=1}^{TotalSalas} P_i} \quad (4)$$

Isso faz com que a soma das chances seja igual a um. Por exemplo, se em um mapa de quatro salas as prioridades forem (Tabela I):

Tabela I  
EXEMPLO DE PRIORIDADES

Sala	Prioridade
1	5
2	1
3	4
4	5

Suas chances de gerar uma emergência ficarão da seguinte forma (Tabela II):

Tabela II  
EXEMPLO DE CHANCES DE EMERGÊNCIA

Sala	Chance
1	0,333
2	0,067
3	0,267
4	0,333

#### D. Resultados

Para a solução em Tempo Real, por se tratar de uma solução baseada em números aleatórios, foram executados 100 repetições em cada mapa.

Para os testes foram criados sete mapas, cada qual com uma peculiaridade diferente como mostra a tabela III

Na sequência são mostrados os resultados em cada mapa:

- Mapa A: Com quatro salas de prioridades iguais (Tabela IV) e ligação entre todas as salas (Figura 1):  
A sequência de salas a serem visitadas gerada pelo gerador foi: 1 2 4 3.

Tabela III  
MAPAS PARA TESTES

Designação	N# de Salas	Prioridades
A	4	Iguais
B	4	Diferentes
C	5	Diferentes
D	5	Diferentes
E	6	Iguais
F	6	Diferentes
G	8	Diferentes

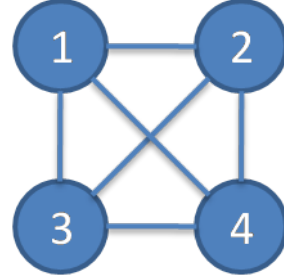


Figura 1. Grafo do Mapa A e B.

As prioridades relativas finais da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa, como mostra a Tabela V.

A Tabela VI mostra que as frequências relativas tanto da solução *Offline* quanto da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa. A tabela mostra também que a solução *Offline* foi melhor que a solução em Tempo Real através da diferença quadrática.

O gráfico (Figura 2) mostra que a solução em Tempo Real depois de estabilizar teve um desempenho inferior à solução *Offline*.

Como o Mapa A é relativamente pequeno (dois metros quadrados) e a sequência ge-

Tabela IV  
PRIORIDADES DO MAPA A.

Sala	Prioridade
1	1
2	1
3	1
4	1

Tabela V  
PRIORIDADES RELATIVAS DO MAPA A.

Sala	Offline	Tempo Real
1	0,25	0,24989
2	0,25	0,24860
3	0,25	0,25224
4	0,25	0,24927

Tabela VI  
FREQUÊNCIAS RELATIVAS DO MAPA A.

Sala	Offline	Tempo Real
1	0,25021	0,25114
2	0,24993	0,25263
3	0,24993	0,24877
4	0,24993	0,24746
Dif. Quad.	0,00000	0,00002

rada pela solução *Offline* é visitar as salas no sentido anti-horário, o robô leva exatamente o mesmo tempo para ir de uma sala à outra. Como as prioridades iniciais do mapa são todas iguais e para gerar o gráfico foram tiradas medidas (Grau de Urgência Total) de dez em dez segundos, coincidiu de o robô a cada dez segundos estar no mesmo lugar no mapa, por isso o gráfico do Mapa A para a solução *Offline* é uma linha.

- Mapa B: Com quatro salas com prioridades diferentes (Tabela VII) e ligação entre todas as salas (Figura 1):  
A seqüência de salas a serem visitadas gerada pelo **gerador** foi: 1 3 4 2 4 3.

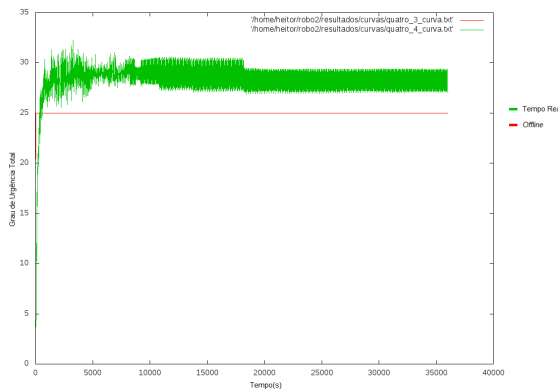


Figura 2. Gráfico do Mapa A.

As prioridades relativas finais da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa, como mostra a Tabela VIII.

A Tabela IX mostra as frequências relativas das soluções *Offline* e em Tempo Real. Nota-se que as frequências relativas das salas 1 e 2 da solução *Offline* são aproximadamente o dobro das suas respectivas prioridades relativas iniciais do mapa, porém, em números absolutos, é uma diferença de aproximadamente 0,08. E as frequências relativas da solução em Tempo Real são próximas às suas respectivas prioridades relativas iniciais do mapa. A tabela mostra também que a solução em Tempo Real foi melhor que a solução *Offline* através da diferença quadrática.

O gráfico do Mapa B (Figura 3), diferentemente do gráfico do Mapa A (Figura 2), ilustra que a solução *Offline* manteve o Grau de Urgência Total dentro de uma faixa, entre dez e trinta aproximadamente, mas seu desempenho também foi superior à solução em Tempo Real.

- Mapa C: Com cinco salas que formam um X (Prioridades na Tabela X) com ligações entre as salas das pontas (Figura 4):  
A seqüência de salas a serem visitadas gerada pelo **gerador** foi: 1 2 4 5 3 2 4 5 3.

As prioridades relativas finais da solução

Tabela VII  
PRIORIDADES DO MAPA B.

Sala	Prioridade
1	1
2	1
3	5
4	5

Tabela VIII  
PRIORIDADES RELATIVAS DO MAPA B.

Sala	Offline	Tempo Real
1	0,08333	0,08470
2	0,08333	0,08331
3	0,41667	0,41141
4	0,41667	0,42058

Tabela IX  
FREQÜÊNCIAS RELATIVAS DO MAPA B.

Sala	Offline	Tempo Real
1	0,16677	0,10130
2	0,16677	0,09994
3	0,33323	0,40206
4	0,33323	0,39670
Dif. Quad.	0,02785	0,00121

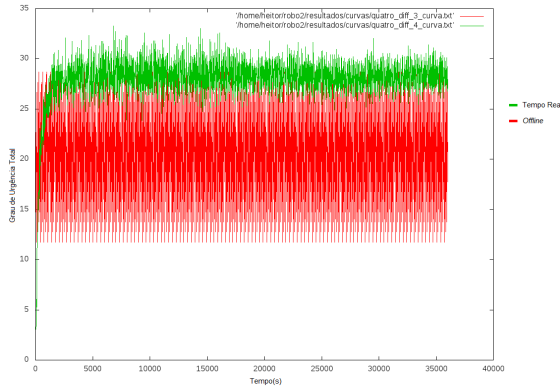


Figura 3. Gráfico do Mapa B.

em Tempo Real foram próximas às prioridades relativas iniciais do mapa, como mostra a Tabela XI.

A Tabela XII mostra que frequência relativa da sala 1 na solução *Offline* foi maior que o dobro da sua prioridade relativa inicial do mapa, entretanto, a diferença numérica absoluta é aproximadamente 0,06. As frequências relativas das salas restantes da

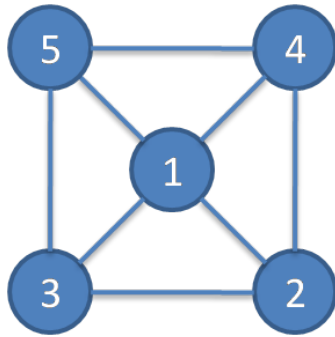


Figura 4. Grafo do Mapa C.

solução *Offline* e as frequências relativas de todas as salas da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa. A tabela XII mostra também que a solução Tempo Real foi melhor que a solução *Offline* através da diferença quadrática.

O gráfico (Figura 5) mostra que a solução em Tempo Real depois de estabilizar teve um desempenho inferior à solução *Offline*.

- Mapa D: Com cinco salas em formato de X (Prioridades iguais ao mapa anterior indicados na Tabela X) com somente a sala do meio ligando as outras (Figura 6):

A sequência de salas a serem visitadas gerada pelo **gerador** foi: 1 2 3 4 5.

As prioridades relativas finais da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa, como mostra a Tabela XIII.

A Tabela XIV mostra que as frequências relativas da solução *Offline* foram aproxi-

Tabela X  
PRIORIDADES DO MAPA C E D.

Sala	Prioridade
1	1
2	5
3	5
4	5
5	5

Tabela XI  
PRIORIDADES RELATIVAS DO MAPA C.

Sala	Offline	Tempo Real
1	0,04762	0,04983
2	0,23810	0,24044
3	0,23810	0,23870
4	0,23810	0,23501
5	0,23810	0,23602

Tabela XII  
FREQÜÊNCIAS RELATIVAS DO MAPA C.

Sala	Offline	Tempo Real
1	0,11134	0,05670
2	0,22234	0,23729
3	0,22200	0,23807
4	0,22234	0,23210
5	0,22200	0,23583
Dif. Quad.	0,00508	0,00012



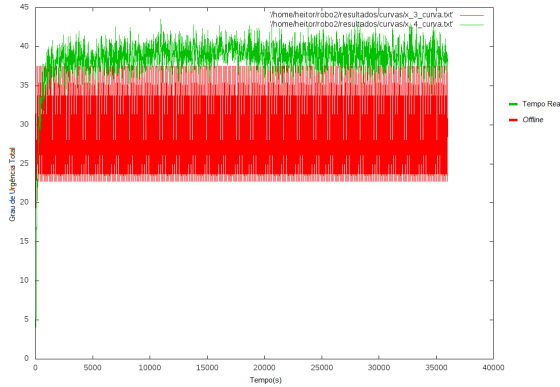


Figura 5. Gráfico do Mapa C.

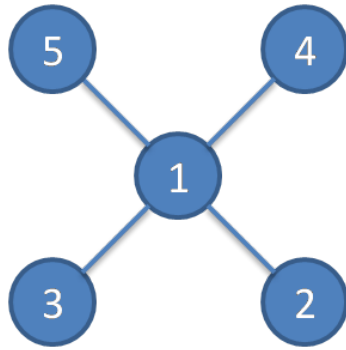


Figura 6. Grafo do Mapa D.

Tabela XIII  
PRIORIDADES RELATIVAS DO MAPA D.

Sala	Offline	Tempo Real
1	0,04762	0,04598
2	0,23810	0,23794
3	0,23810	0,24530
4	0,23810	0,23984
5	0,23810	0,23094

madamente todas iguais, isso se deve ao fato de que a sequência de salas a serem visitadas gerada pelo **gerador** manda o robô visitar cada sala apenas uma única vez dentro do *loop*. Na mesma tabela pode-se ver também que as frequências relativas da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa. A tabela mostra também que a solução

Tempo Real foi melhor que a solução *Offline* através da diferença quadrática.

Tabela XIV  
FREQUÊNCIAS RELATIVAS DO MAPA D.

Sala	Offline	Tempo Real
1	0,20009	0,05481
2	0,20009	0,23908
3	0,20009	0,23633
4	0,20009	0,23275
5	0,19965	0,23703
Dif. Quad.	0,02906	0,00009

O gráfico (Figura 7) mostra que a solução em Tempo Real depois de estabilizar teve um desempenho inferior à solução *Offline*.

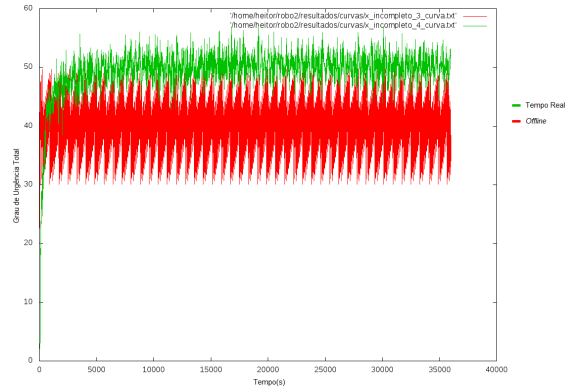


Figura 7. Gráfico do Mapa D.

- Mapa E: Com seis salas de prioridades iguais (Tabela XV) em formato de espinha de peixe (Figura 8):

Tabela XV  
PRIORIDADES DO MAPA E.

Sala	Prioridade
1	2
2	2
3	2
5	2
6	2
7	2

A sequência de salas a serem visitadas gerada pelo **gerador** foi: 1 5 2 6 3 7.

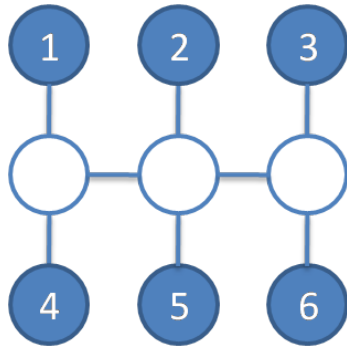


Figura 8. Grafo do Mapa E e F.

As prioridades relativas finais da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa, como mostra a Tabela XVI.

Tabela XVI  
PRIORIDADES RELATIVAS DO MAPA E.

Sala	Offline	Tempo Real
1	0,16667	0,16990
2	0,16667	0,16613
3	0,16667	0,16571
5	0,16667	0,16339
6	0,16667	0,16720
7	0,16667	0,16768

A Tabela XVII mostra que as frequências relativas tanto da solução *Offline* quanto a em Tempo Real foram próximas às prioridades relativas iniciais do mapa. A Tabela XVII mostra também que a solução *Offline* foi melhor que a solução Tempo Real através da diferença quadrática.

Tabela XVII  
FREQUÊNCIAS RELATIVAS DO MAPA E.

Sala	Offline	Tempo Real
1	0,16694	0,16869
2	0,16694	0,16483
3	0,16640	0,16692
5	0,16694	0,16832
6	0,16640	0,16179
7	0,16640	0,16945
Dif. Quad.	0	0,00004

O gráfico (Figura 9) mostra que a solução em Tempo Real depois de estabilizar teve

um desempenho inferior à solução *Offline*.

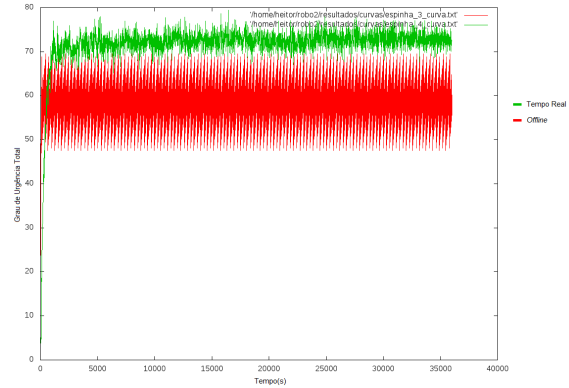


Figura 9. Gráfico do Mapa E.

- Mapa F: Com seis salas de prioridades diferentes (Tabela XVIII) em formato de espinha de peixe (Figura 8):

Tabela XVIII  
PRIORIDADES DO MAPA F.

Sala	Prioridade
1	5
2	1
3	2
5	2
6	1
7	1

A sequência de salas a serem visitadas gerada pelo **gerador** foi: 1 2 6 5 1 7 3 5. As prioridades relativas finais da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa, como mostra a Tabela XIX.

Tabela XIX  
PRIORIDADES RELATIVAS DO MAPA F.

Sala	Offline	Tempo Real
1	0,41667	0,41297
2	0,08333	0,08386
3	0,16667	0,16624
5	0,16667	0,17418
6	0,08333	0,07849
7	0,08333	0,08427

A Tabela XX mostra que as frequências relativas da solução *Offline* foram diferentes

das prioridades relativas iniciais do mapa, visitando as salas 1 e 5 o dobro de vezes que as demais salas, pois na sequência determinada pelo **gerador** as salas 1 e 5 aparecem duas vezes cada uma. Todavia as frequências relativas da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa. A Tabela XX mostra também que a solução Tempo Real foi melhor que a solução *Offline* através da diferença quadrática.

Tabela XX  
FREQÜÊNCIAS RELATIVAS DO MAPA F.

Sala	<i>Offline</i>	Tempo Real
1	0,25000	0,35364
2	0,12500	0,10119
3	0,12500	0,17144
5	0,25000	0,17798
6	0,12500	0,09488
7	0,12500	0,10087
Dif. Quad.	0,04167	0,00488

O gráfico (Figura 10) mostra que a solução em Tempo Real depois de estabilizar teve um desempenho inferior à solução *Offline*.

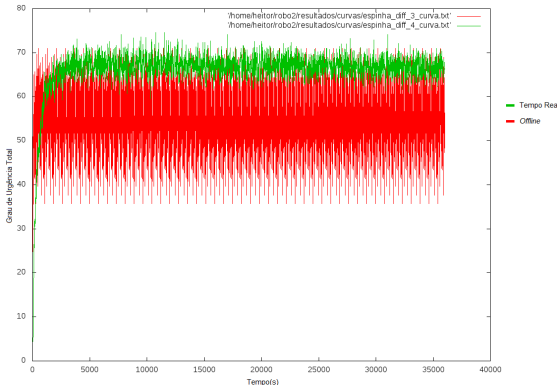


Figura 10. Gráfico do Mapa F.

- Mapa G: Com oito salas de prioridades diferentes (Tabela XXI) simulando um apartamento real (Figura 11):  
A sequência de salas a serem visitadas gerada pelo **gerador** foi: 1 12 9 11 5 7 3 4.

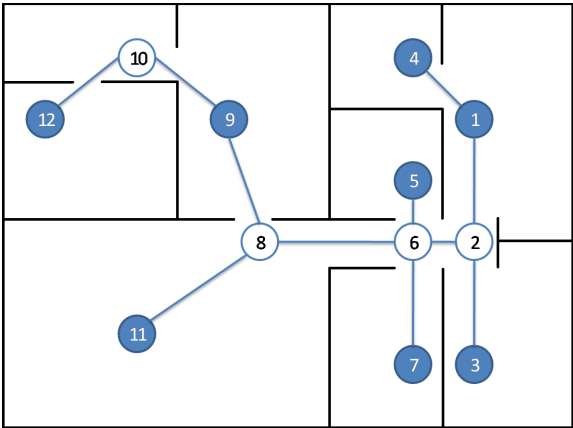


Figura 11. Grafo do Mapa G.

Tabela XXI  
PRIORIDADES DO MAPA G.

Sala	Prioridade
1	5
3	5
4	3
5	3
7	5
9	4
11	4
12	4

As prioridades relativas finais da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa, como mostra a Tabela XXII.

Tabela XXII  
PRIORIDADES RELATIVAS DO MAPA G.

Sala	<i>Offline</i>	Tempo Real
1	0,15152	0,15426
3	0,15152	0,15233
4	0,09091	0,08996
5	0,09091	0,08924
7	0,15152	0,14934
9	0,12121	0,12566
11	0,12121	0,12133
12	0,12121	0,11789

A Tabela XXIII mostra que as frequências relativas da solução *Offline* foram aproximadamente todas iguais, isso se deve ao fato de que a sequência de salas a serem visitadas gerada pelo **gerador** manda o

robô visitar cada sala apenas uma única vez dentro do *loop*, porém as maiores diferenças numéricas são as das sala 4 e 5 que são aproximadamente 0,03. Na mesma tabela pode-se ver também que as frequências relativas da solução em Tempo Real foram próximas às prioridades relativas iniciais do mapa. A Tabela XXIII mostra também que a solução em Tempo Real foi melhor que a solução *Offline* através da diferença quadrática.

Tabela XXIII  
FREQUÊNCIAS RELATIVAS DO MAPA G.

Sala	<i>Offline</i>	Tempo Real
1	0,12525	0,14970
3	0,12475	0,15172
4	0,12475	0,09717
5	0,12475	0,09447
7	0,12475	0,14197
9	0,12525	0,12442
11	0,12525	0,12004
12	0,12525	0,12050
Dif. Quad.	0,00446	0,00016

O gráfico (Figura 12) mostra que a solução em Tempo Real depois de estabilizar teve um desempenho inferior à solução *Offline*.

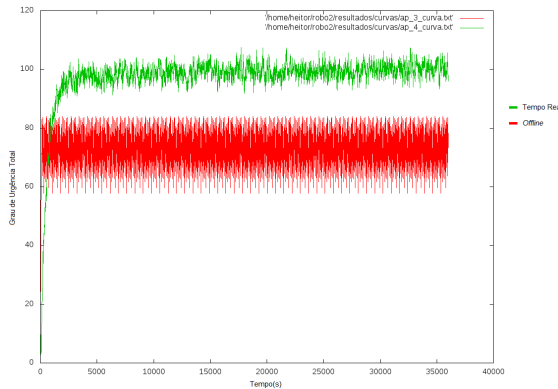


Figura 12. Gráfico do Mapa G.

1) *Análise Comparativa:* Foi calculada uma média do Grau de Urgência Total de cada solução em cada mapa com os últimos dez mil segundos (Tabela XXIV); analisando essa média nota-se que existe uma

tendência de que quanto maior o número de salas mais eficiente será a solução *Offline* em relação à solução em Tempo Real, como mostra o gráfico na Figura 13.

Tabela XXIV  
MÉDIA DOS ÚLTIMOS DEZ MIL SEGUNDOS.

Mapa (N# de Salas)	<i>Offline</i>	Tempo Real
A (4)	25,000	28,150
B (4)	20,980	28,346
C (5)	28,905	38,753
D (5)	39,719	50,326
E (6)	58,527	72,622
F (6)	55,557	66,711
G (8)	73,318	100,217

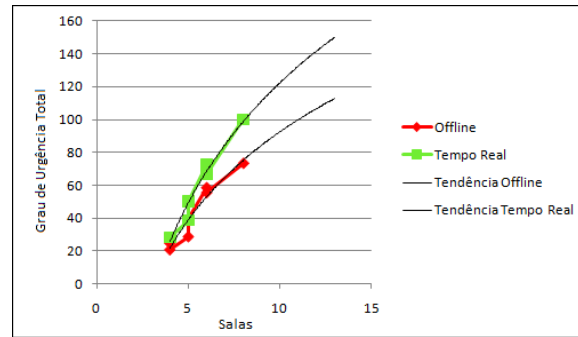


Figura 13. Tendência de Desempenho

### E. Considerações

Analisando os testes pode-se constatar que:

- Segundo o critério de avaliação de frequência relativa, a solução em Tempo Real foi superior à solução *Offline*, pois foi mais eficiente em cinco dos sete testes e na soma total das diferenças quadráticas (0,10812 da solução *Offline* contra 0,00652 da solução em Tempo Real);
- Segundo o critério de avaliação de Grau de Urgência Total a solução *Offline* foi melhor que a solução em Tempo Real em todos os testes realizados e o gráfico da Figura 13 mostrou que existe um tendência de que quanto maior o número de salas maior será a eficiência da solução *Offline* em relação à solução em Tempo Real.

Era esperado que a solução *Offline* tivesse um desempenho pior que a solução em Tempo Real no critério de avaliação de frequência relativa, pois a avaliação das seqüências de salas à visitar é baseada somente no Grau de Urgência Total; e o desempenho das soluções segundo o critério de avaliação de Grau de Urgência Total mostra que o robô, seguindo uma heurística simples, tem um desempenho inferior do que uma solução que determina a seqüência de salas a serem visitadas, baseada no Grau de Urgência Total, antes de colocar o robô no ambiente.

Por fim, o que determina qual solução deve ser utilizada depende da aplicação. Se para a aplicação a frequência relativa for mais importante do que manter o Grau de Urgência baixo, deve-se escolher a solução em Tempo Real, porém se para a aplicação for mais importante manter o Grau de Urgência baixo deve-se escolher a solução *Offline*, que foi baseada no Grau de Urgência Total.

### III. CONCLUSÃO E TRABALHOS FUTUROS

#### CONCLUSAO

Para dar continuidade ao projeto o próximo é explorar a utilização de mais de um robô para realizar a tarefa, cada robô monitorar um conjunto de salas divididas por prioridade ou proximidade, ou todos os robôs podem monitorar todas as salas etc.

#### IV. CONCLUSION

The conclusion goes here. this is more of the conclusion

#### ACKNOWLEDGMENT

The authors would like to thank... more thanks here

#### REFERÊNCIAS

- [1] A. A. D. Medeiros, "Introdução à robótica," Natal, RN, Brasil, Jul. 1998, pp. 56–65.
- [2] J. C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [3] S. J. Russell, P. Norvig, J. F. Canny, and J. F. C. Peter Norvig, *Artificial Intelligence*. Prentice Hall, 2003.
- [4] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.
- [5] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002. [Online]. Available: <http://portal.acm.org/citation.cfm?id=504729.504754>
- [6] NASA, "Mars pathfinder," <http://mpfwww.jpl.nasa.gov/default.html>, 2010. [Online]. Available: <http://mpfwww.jpl.nasa.gov/default.html>
- [7] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, D. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. W., "Autonomous exploration and mapping of abandoned mines," *Robotics & Automation Magazine, IEEE*, vol. 11, no. 4, pp. 79–91, 2004.
- [8] F. J. Heinen and F. Osório, "Sistema de controle híbrido para robôs móveis autônomos," *WTDIA/SBIA - Workshop De Teses E Dissertações D Inteligência Artificial, 2002*, 2002.
- [9] IRobot, "iRobot corporation," <http://store.irobot.com/corp/index.jsp>, 2010. [Online]. Available: <http://store.irobot.com/corp/index.jsp>
- [10] FriendlyRobotics, "Robomow® lawn mower - automatic lawnmower," <http://www.friendlyrobotics.com/>, 2010. [Online]. Available: <http://www.friendlyrobotics.com/>
- [11] D. P. F. Pedrosa, A. A. D. Medeiros, and P. J. Alsina, "Sistema de navegação para robôs móveis autônomos," *Latin American Knowledge Harvester*, Oct. 2001. [Online]. Available: <http://lakh.unm.edu/handle/10229/14268>