

# Relatório

Edileuton Henrique de Oliveira - GRR20091930

*eho09@inf.ufpr.br*

<sup>1</sup>Universidade Federal do Paraná(UFPR)

**Resumo.** *Este relatório apresentará uma proposta de um sistema, para navegação de robôs móveis em ambientes dinâmicos. Ele será dividido em duas aplicações, utilizando as plataformas Android[?]-Arduino[?]. A aplicação que rodará no robô baseado no Arduino fará a comunicação com os sensores e motores do robô, e repassará as informações coletadas para aplicação Android via bluetooth. A aplicação que rodará no dispositivo Android será responsável por tomar decisões a partir dos dados coletados, pelo robô e pelos sensores do tablet, e assim determinar a localização do robô, e realizar a construção e atualização do mapa do ambiente, e o planejamento do trajeto a ser utilizado.*

## 1. Introdução

Robôs móveis são sistemas incorporados no mundo real que se movem autonomamente e interagem com ele para realizar suas tarefas[?]. A utilização de tais robôs são, até certo ponto, freqüentes hoje em dia, e as tarefas a eles designadas estão aumentando em complexidade. Eles podem ser utilizados em várias aplicações, como limpeza, cortar grama, detectar riscos, explorações de ambientes desconhecidos, vigilância autônoma, e assistência a idosos ou pessoas incapazes. Robôs podem cooperar pra realizar tarefas em comum, como encontrar e resgatar sobreviventes de um terremoto em território urbano [?].

Planejamento de caminho, Navegação, Localização, Evitar obstáculos, Controle de motores construção e atualização de mapas

Dentro do problema navegação em robótica, podemos destacar a construção de mapas, pois, para navegação do robô, é essencial que o robô conheça o ambiente na qual ele realizará sua tarefa. Um mapa é uma representação espacial utilizada para descobrir a localização de elementos especiais [?]. Tipicamente, existem duas abordagens para a representação de mapas de ambiente [?]: mapas métricos e mapas topológicos. Mapas métricos contêm informação da geometria do ambiente, da posição dos objetos e distâncias entre esses. Os mapas topológicos não possuem qualquer informação sobre a geometria do ambiente, eles são representados por elos conectados a nós [?].

Nos nós o robô pode tomar certas ações, como por exemplo, virar à direita ou à esquerda, e reconhecer marcos para se localizar no ambiente. O reconhecimento dos marcos visuais dá ao robô uma localização qualitativa no ambiente, obtendo sua posição em termos do quanto está mais próximo ou mais distante do alvo. Na navegação com mapas geométricos a localização é quantitativa, sabendo o robô a sua posição exata no ambiente [?].

Na navegação autônoma, os elementos representados em um mapa podem ser utilizados para diversas finalidades. Por exemplo, eles podem ser usados para planejar um caminho entre a posição atual do robô e seu destino [?], especialmente se o mapa

representar as áreas por onde ele for permitido navegar. Os mapas podem também ser utilizados para localizar o robô: comparando os elementos sensorizados no ambiente com aqueles registrados no mapa, o robô pode inferir o lugar ou possíveis lugares onde está.

Há um conjunto de algoritmos de localização que utilizam uma rede de sensores sem-fio ('*Wireless sensor networks*' - WSNs). Onde o cálculo da localização depende das informações de localização dos nodos (que podem ser estáticos ou móveis) da rede[?]. A posição de um certo nodo, pode ser obtida, a partir da posição ou direção dos nodos conhecidos por ele, para isso, geralmente é necessário algum tipo de hardware especial.

Tempo de chegada ('*Time of Arrival*' - TOA) [?], ângulo de chegada ('*Angle of Arrival*' - AOA) [?], diferença de chegada de dois sinais diferentes ('*Time Difference of Arrival of Two Different Signals*' - TDOA) [?] e força do sinal recebido ('*Received Signal Strength*' - RSS) [?], são alguns das abordagens feitas por algoritmos que utilizam WSNs, para cálculo da localização.

Na navegação do robô é ideal que ele possa mover-se de um ponto inicial, até a posição objetivo, com a capacidade de evitar obstáculos, fazendo o melhor trajeto possível, levando em consideração a suavidade do trajeto, distância percorrida, etc. Por isso, planejamento de trajetos, vem sendo extensivamente estudado, e é um dos principais problemas dentro da robótica.

Existem um grande número de métodos para resolver o planejamento de movimentos. Entretanto, nem todas generalizações, deste problema podem ser aplicadas. Por exemplo, alguns métodos necessitam que espaço de trabalho seja bidimensional e os obstáculos poligonais. Apesar de muitas diferenças externas, os métodos são baseados em algumas técnicas gerais: *roadmap*, decomposição de células, e campo potencial. Na maioria deles cria-se uma estrutura especial e aplica-se algoritmos de busca em grafos como A\*[?] e Dijkstra[?].

Como sabemos o Android OS [?] é uma plataforma de fácil desenvolvimento, com documentação farta [?], e possui drivers para câmera, wifi, acelerômetro, bluetooth, microfone e GPS, o que torna o desenvolvimento para robôs menos custoso[?], por isso, da sua utilização. Do sistema de navegação, a aplicação que rodará no Android será o cérebro do sistema, ela será responsável por tomar decisões analisando as informações disponíveis. Esta aplicação coletará a força de sinal wifi dos *Access Points*, para determinar a posição do robô no mapa. Ela deverá ser capaz de planejar o trajeto a ser realizado pelo robô, e repassar os comandos de movimentação para este, a partir do mapa topológico e das informações obtidas do robô.

### 1.1. Objetivo

O objetivo deste trabalho é implementar um sistema de navegação para robôs móveis em ambientes dinâmicos, utilizando as plataformas Android e Arduino. E no mesmo, apresentar uma solução para os problemas de construção e atualização de mapas, localização e planejamento de caminhos.

Método de representação de mapas utilizado nesse trabalho será similar ao proposto em [?], onde o mapa é um *grid*, no qual cada célula um valor, que indica o grau de incerteza de haver um obstáculo. O mapa será construído a partir de uma imagem, ela será dividida em células, em cada célula será aplicada a função de transformação de

Hough[?], e assim será atribuído um valor a célula. A atualização do mapa será feita através das informações coletadas pelo sonar do robô.

Nesse trabalho para fazer a localização do robô será implementado o método empírico sugerido no artigo[?], o método é parte de uma técnica de localização baseada em RSS, a qual é uma característica do sinal transmitido, muito utilizada em técnicas de localização por não demandar *hardware* extra.

O planejamento de trajeto do robô será feito aplicando o algoritmo A\*[?] no grafo do mapa topológico.

## **1.2. Organização do trabalho**

A primeira parte aborda o tópico construção de mapas, apresentando as principais técnicas de construção de mapas e mostra um pouco sobre o problema de auto localização e mapeamento simultâneos.

A segunda parte mostra como pode-se obter a localização de um nodo, utilizando redes de sensores sem fio, e trás um resumo do método de localização a ser utilizado no sistema de navegação.

A terceira parte apresenta o problema planejamento de trajetos, e mostra algumas técnicas utilizadas na solução desse problema.

A quarta parte mostra como será implementado o sistema proposto nesse trabalho, para as plataformas Android e Arduino.

A quinta parte faz a conclusão desse trabalho e apresenta uma breve discussão sobre trabalhos futuros.

## **2. Construção de Mapas**

A tarefa de mapeamento corresponde à atribuição de valores aos elementos do mapa, relacionando cada um a uma certa posição nele [?]. O tipo ideal de mapa a ser utilizado ou construído por um robô móvel depende da tarefa e do ambiente onde este esteja inserido. Também depende das características do robô, tais como os tipos de sensores que ele tem, bem como a forma com ele se move[?]. Há duas abordagens principais para a representação de mapas de ambiente [?]. São os mapas métricos e topológicos, que serão discutidos a seguir.

### **2.1. Construção de Mapas Métricos**

No caso da construção de mapas métricos, o objetivo é obter um mapa detalhado do ambiente, com informações sobre a forma e o tamanho dos objetos e os limites das áreas livres para a navegação, tais como corredores, quartos, trilhas e estradas. Para representar essa complexa e detalhada informação, o mapa é geralmente dividido em uma densa rede de forma que cada célula contenha informações sobre a ocupação desse espaço por um objeto e, possivelmente, outras características ambientais que estão sendo mapeadas [?].

Como o mapa métrico salva informações no próprio domínio do sinal do sensor, a tarefa de localização é executada de uma forma simples. Além disso, devido à alta densidade de informação nesses mapas, o resultado da localização é mais preciso e menos sujeito a ambigüidades [?]. Como os mapas métricos demandam uma grande quantidade de informações, eles exigem muita capacidade de processamento e armazenamento.

Atualmente grande parte dos sistemas de mapeamento assumem que o ambiente é estático durante o mapeamento. Se uma pessoa anda dentro do alcance dos sensores do robô durante o mapeamento, o mapa resultante conterá evidências a respeito de um objeto na localização correspondente. Além disso, se o robô retornar para esta localização e varrer a área uma segunda vez, a estimativa da posição será menos precisa, uma vez que as novas medições não contêm nenhum vestígio correspondente àquela pessoa. A exatidão reduzida do mapa resultante pode ter uma influência negativa no desempenho do robô.

### **2.2. Construção de Topológicos**

No caso de mapas topológicos, o objetivo é construir uma estrutura relacional, normalmente um grafo, de forma geral, registram informações sobre determinados elementos ou locais do ambiente, chamados marcos [?]. Assim, marcos e relações são os elementos dos mapas topológicos. Essas relações podem ser de vários tipos. Eles podem indicar o deslocamento relativo entre dois marcos, a existência de um caminho entre eles, etc. É fácil perceber que as informações contidas no mapa topológico é menos detalhado e distribuído de forma dispersa, concentrando-se apenas em alguns pontos de interesse. Assim, é muito seletivo com relação à informação que ele registra [?].

O mapa topológico também demanda que o processamento dos dados dos sensores do robô, sejam feitos de forma mais abstrata, a fim de extrair informações sobre as localidades mais relevantes que devem ser consideradas como marcos.

### **2.3. Problema de auto localização e mapeamento simultâneos**

O problema de auto localização e construção de mapas de ambiente simultâneos (*'Simultaneous Localization and Map Building' - SLAM*) consiste em um robô autônomo iniciar

a navegação em uma localização desconhecida, em um ambiente desconhecido e então construir um mapa desse ambiente de maneira incremental, enquanto utiliza o mapa simultaneamente para calcular a sua localização [?]. O SLAM tem sido tema de várias pesquisas na área de robótica, a grande vantagem do SLAM é que elimina a necessidade de estruturas artificiais ou um conhecimento topológico a priori do ambiente. Há 3 abordagens principais que são utilizadas no problema do SLAM.

O primeiro e mais popular deles é baseado na teoria da estimação usando filtro de Kalman [?]. ele fornece uma solução recursiva para o problema da navegação e uma maneira de calcular estimativas consistentes para a incerteza na localização do robô e nas posições dos marcos do ambiente, com base em modelos estatísticos para o movimento dos robôs e observações relativas dos marcos do ambiente[?].

O segundo método consiste em evitar a necessidade de estimativas absolutas da posição e de medições precisas das incertezas para utilizar conhecimento mais qualitativo da localização relativa dos marcos do ambiente e do robô para a construção do mapa global e planejamento da trajetória[?].

O terceiro método é bastante amplo e afasta-se do rigor matemático do filtro de Kalman, ou do formalismo estatístico, retendo uma abordagem essencialmente numérica ou computacional para a navegação e para o problema de SLAM. Essa abordagem inclui o uso de correspondência com marcos artificiais do ambiente[?].

## **2.4. Problema de navegação e mapeamento completo do ambiente**

No artigo [?] - "*A Bioinspired Neural Network for Real-Time Concurrent Map Building and Complete Coverage Robot Navigation in Unknown Environments*", Luo e Yang propõem um modelo de mapeamento e navegação de robôs em ambientes desconhecidos e dinâmicos. Eles utilizam robôs limpadores para exemplificar o modelo. No qual o robô deve percorrer todo o local, para fazer sua limpeza.

O mapa topológico do ambiente é discretizado em células, onde as células podem assumir o formato de quadrados, retângulos e triângulos. Utilizando quadrados e retângulos, o robô limpador tem 8 direções possíveis para deslocamento, enquanto que utilizando células triangulares, o robô possui 12 direções possíveis, assim permitindo que o robô navegue por caminhos mais curtos e flexíveis. Essas células formam uma rede neural, e cada célula pode assumir um estado: desconhecido, limpo, sujo e *deadlock*.

O mapeamento do ambiente é feito com o robô limpador percorrendo o ambiente em "zigzag" e monitorando a atividade neural da rede. A dinamicidade da atividade neural, representa as mudanças do ambiente. As áreas sujas e com obstáculos ficam no topo e no vale da atividade da rede neural, respectivamente. As áreas sujas atraem o robô, através da propagação da atividade neural. O planejamento do caminho para evitar colisões é feito em tempo real, baseado na dinamicidade da atividade neural da rede e na posição anterior do robô, assim ele é capaz de percorrer o ambiente, limpando as áreas sujas.

### 3. Localização utilizando WSNs

WSNs é uma rede de sensores que coleta informações em um campo monitorado. Este tipo de rede tem sido utilizada em várias aplicações, incluindo rastreamento de objetos, resgate, monitoramento de ambiente, entre outros [?]. Localização é um tópico muito importante, pois, muitas aplicações das WSNs dependem do conhecimento das posições dos sensores. Na maioria das WSNs os sensores são estáticos, enquanto em algumas aplicações modernas os sensores devem ser móveis.

Há um conjunto de algoritmos de localização que utilizam WSNs. Onde o cálculo da localização depende das informações de localização dos nodos (que podem ser estáticos ou móveis) da rede. Esses algoritmos podem ser divididos em duas categorias: *range-based* e *range-free*. Algoritmos do tipo *range-free* [?] [?], geralmente, requerem que os nodos conhecidos, estejam dentro do raio de comunicação dos sensores em comum, eles tem sido muito explorados pois não precisam de *hardware* extra. Os algoritmos do tipo *range-based*, geralmente necessitam de algum tipo de *hardware* especial, onde a posição de um certo nodo, pode ser obtida, a partir da posição ou direção dos nodos conhecidos por ele. Este trabalho se foca em mostrar algoritmos do tipo *range-based*.

#### 3.1. Localização baseada em TOA

TOA é o tempo medido em que um sinal (rádio frequência, acústicos, ou outros) chega a um receptor pela primeira vez [?]. O valor medido é o tempo de transmissão somado ao atraso do tempo de propagação. Este atraso,  $T_{i,j}$ , entre a transmissão do sensor  $i$  e recepção do sensor  $j$ , é igual a distância entre o transmissor e o receptor,  $d_{i,j}$ , dividido pela velocidade de propagação do sinal,  $v_p$ . Esta velocidade para a rádio frequência é aproximadamente  $10^6$  vezes mais rápida que a velocidade do som [?], ou seja, 1 ms corresponde a 0,3 m na propagação sonora, enquanto para a rádio frequência, 1 ns corresponde a 0,3 m [?].

O ponto chave das técnicas baseadas em tempo de a capacidade do receptor de estimar com precisão o tempo de chegada em sinais na linha da visão. Esta estimativa é prejudicada tanto pelo ruído aditivo quanto por sinais de multipercurso.

#### 3.2. Localização baseada em AOA

Existem duas maneiras nas quais sensores medem o AOA [?]. O método mais comum é usar um vetor de sensores e utilizar a matriz de processamento de sinal nos nós sensores. Neste caso, cada nó sensor é composto de dois ou mais sensores individuais (microfones para sinais acústicos ou antenas de sinais de rádio frequência), cujas posições com relação ao centro do nó são conhecidos. A AOA é estimada a partir das diferenças de tempos de chegada de uma transmissão do sinal em cada um dos elementos da matriz de sensores.

A segunda abordagem para a estimativa do AOA, usa a razão RSS entre duas (ou mais) antenas direcionais localizadas no sensor. Duas antenas direcionais apontadas em direções diferentes, de tal maneira que suas hastes se sobrepõem, podem ser usadas para estimar a AOA a partir da relação entre seus valores individuais RSS.

#### 3.3. Localização baseada em TDOA

Os algoritmos de TDOA fazem a medição da diferença no tempo de recepção de sinais de diferentes estações de base ('Base Station's' - BSs), sem a necessidade de uma

sincronização de todos os participantes BSs e terminais móveis(MT)) [?]. Na verdade, a incerteza entre os tempos de referência dos BSs e do MT pode ser removida por meio de um cálculo diferencial. Por isso, apenas os BSs envolvidos no processo de estimativa de localização deve ser bem sincronizados.

Para um par de BSs, digamos i e j, o TDOA,  $T_{ij}$ , é dada por  $T_{ij} = TBS_i - TBS_j$ , onde  $TBS_i$  e  $TBS_j$  são os tempos absolutos tomados pelo estouro de chegada nos BSs i e j, respectivamente. Supondo que o MT está em LOS (*line-of-sight*) com ambos os BSs, i e j, o MT deve situar-se em uma hipérbole. Uma segunda hipérbole, onde o MT deve estar, pode ser obtido através de uma medição adicional de TDOA envolvendo um terceiro BS. A posição do usuário pode ser identificada como o ponto de intersecção das duas hipérbolas. A solução do sistema de equações pode ser encontrado com um método iterativo e minimização dos quadrados mínimos [?].

### 3.4. Localização baseada em RSS

A localização baseada em RSS apresenta-se como uma das únicas a não necessitar de hardware extra, e ser uma técnica de fácil aplicação. É uma técnica duas vezes mais econômica que as demais técnicas conhecidas, tanto em termos de equipamento quanto em baixo consumo de energia[?].

O funcionamento da RSS é baseado na medição do sinal no receptor e o valor obtido indica a distância até o transmissor [?]. Tal forma de medição por muitas vezes é descartada, pois não leva em conta o ambiente onde está sendo aplicada a medição; logo, corresponde a um resultado não confiável, obstruções e obstáculos proporcionam erros de grande relevância, sendo esse um dos maiores problemas dos métodos de localização baseados em RSS[?].

### 3.5. Proposta de localização utilizando RSS de Access Points Wifi

Em [?] - "*Radar: an in-building RF-based user location and tracking system*", Bahl e Padmanabhan propõem o método empírico, para localização baseada em rss dentro de um prédio. O método consiste em realizar uma série de medições da força de sinal das estações bases, e a partir dessas medições, a localização pode ser determinada fazendo uma triangulação dos dados coletados.

No artigo [?], para obter a posição de um nodo, são medidas as forças do sinal wifi de 3 Access Points (rss1, rss2, rss3), utiliza-se a distância Euclidiana, para fazer a comparação com os dados previamente coletados  $\sqrt{(rss1 - rss1')^2 + (rss2 - rss2')^2 + (rss3 - rss3')^2}$ , e assim encontrar o ponto que mais se aproxima dos parâmetros coletados naquele local.

## 4. Planejamento de trajetos

O problema planejamento de trajetos, consiste em encontrar o melhor trajeto possível, de um ponto de origem até um ponto de destino que não resulte em colisão com obstáculos [?]. Dependendo da quantidade de informação disponível sobre o ambiente, que pode ser completamente ou parcialmente conhecido ou desconhecido, as abordagens de planejamento podem variar consideravelmente. A definição de melhor trajeto também pode variar. Esse tópico vem ganhando mais relevância em áreas, além da robótica, como computação gráfica, sistemas de informações geográficas e jogos [?].

A computação geométrica tem um papel especial dentro do desenvolvimento do planejamento de trajetos. As principais abordagens utilizando geometria computacional são: *roadmap*, decomposição de células, e campo potencial. Elas serão apresentadas a seguir.

### 4.1. Roadmap

A técnica de *roadmap* para planejamento de trajeto consiste em capturar a conectividade do espaço livre do ambiente em uma rede de curvas de uma dimensão, denominada *roadmap* [?]. Uma vez que a rede é obtida, ela é vista como um conjunto de caminhos. O planejamento de trajeto pode ser feito conectando as posições iniciais e finais do robô ao *roadmap* e buscar neste um caminho entre estes dois pontos.

Vários métodos foram propostos, formando *roadmaps* a partir de estruturas de computação geométrica como diagrama de Voronoi e grafo de visibilidade.

#### 4.1.1. Diagrama de Voronoi

Um diagrama de Voronoi é um estrutura geométrica que representa informações de proximidade sobre uma série de pontos ou objetos [?]. Dada uma série de sites ou objetos, o plano bidimensional em que o robô se locomove geralmente contém obstáculos, cada um destes obstáculos pode ser representado por polígonos côncavos ou convexos. Para encontrar o diagrama generalizado de Voronoi para esta coleção de polígonos, podemos calcular o diagrama através de uma aproximação, convertendo os obstáculos em uma série de pontos [?].

Primeiramente as faces dos polígonos são subdivididas em uma série de pontos. O próximo passo é calcular o diagrama de Voronoi para esta coleção de pontos. Após o diagrama de Voronoi ser calculado, os segmentos do diagrama que intersectam algum obstáculo são eliminados. E então utiliza-se um algoritmo de busca em grafos para encontrar o melhor caminho. Este método gera uma rota que na sua maior parte permanece equidistante dos obstáculos, criando um caminho seguro para o robô se locomover, apesar de não ser o caminho mais curto. O caminho encontrado pode ser melhorado como é mostrado em [?], no qual pode-se deixá-lo mais suave, retirando desvios desnecessários.

#### 4.1.2. Grafo de visibilidade

Um grafo de visibilidade é obtido gerando-se segmentos de reta entre os pares de vértices dos obstáculos [?]. Todo o segmento de reta que estiver inteiramente na região do espaço



livre é adicionada ao grafo. Para executar o planejamento de trajetória, a posição atual e o objetivo são tratados como vértices, isso gera um grafo de conectividade onde utiliza-se algoritmos de procura para se encontrar um caminho livre. O caminho mais curto que for encontrado no grafo de visibilidade é o caminho ótimo para o problema especificado[?]. Mas os caminhos encontrados no grafo podem tocar obstáculos. Para utilizar um método de navegação com grafo de visibilidade é necessário que o mapa seja completo e bem definido.

#### **4.2. Decomposição de células**

O método de decomposição de células consiste em dividir o espaço livre do robô em regiões simples (células) [?], de forma que um caminho entre quaisquer duas configurações em uma mesma célula possa ser obtido. Um grafo não dirigido representando a relação de adjacência entre as células é construído. Os vértices que compõem este grafo são as células extraídas do espaço livre do robô. Há uma aresta entre dois vértices se e somente se as células correspondentes a ele são adjacentes. Uma busca é feita nesse grafo e seu resultado é uma sequência de células denominada canal. Um caminho contínuo pode ser retirado do canal. A qualidade de caminho obtido, é diretamente afetado pelo grau de decomposição de células.

#### **4.3. Campo potencial**

A idéia por trás do método de campo potencial, é assinalar uma função similar ao potencial eletrostático para cada obstáculo, e então criar uma estrutura topológica de espaço livre do robô na forma de vales de potencial mínimo [?]. O robô move-se em direção a objetivo, já que este, gera um campo potencial que atrai o robô. Os obstáculos geram uma força de repulsão, que impede que o robô colida com obstáculos. Assim a direção do movimento do robô é determinada pela força proveniente do campo potencial nesta determinada configuração. Um problema desse método, é que no caminho utilizado, o robô pode ficar preso e nunca atingir o objetivo.

## 5. Implementação

Os experimentos serão realizados no departamento de informática da Universidade Federal do Paraná, devido a grande quantidade de Access Points disponíveis e a disponibilidade da planta do departamento. A implementação do sistema de navegação será dividida em duas etapas:

### 5.1. Etapa 1

Uma aplicação escrita em C será utilizada para criação do mapa topológico do departamento. Essa aplicação recebe como entrada:

- Uma imagem contendo o mapa.
- Tamanho da célula.

A imagem será dividida em células. Em todas as células, serão aplicadas a função de transformação de Hough [?], da biblioteca OpenCV para detecção de linhas. Esta aplicação devolve uma matriz, onde célula possui um valor que indica o grau de incerteza de haver um obstáculo: 0 (sem obstáculo) ou 4 (com obstáculo). Essa matriz será embarcada na aplicação Android da etapa 2.

O tamanho ideal da célula, será definido via experimentos. Em [?] o tamanho da célula tem o tamanho do robô. O artigo [?] sugere que o tamanho da célula deve comportar todo o obstáculo, lembrando que a quantidade de células pode influenciar no tempo de processamento do planejamento da trajetória do robô [?].

Uma segunda aplicação Android, será utilizada para coletar amostras, como descrito no método empírico do artigo [?]. As tabelas resultantes dessa aplicação, também serão embarcadas na aplicação Android da etapa 2. As linhas da tabela conterão as coordenadas, RSS e nome do AP da amostra daquele local. Vale ressaltar, que número de amostras tem grande impacto na precisão do cálculo da localização [?].

### 5.2. Etapa 2

A aplicação Android dessa etapa, possui 2 componentes principais: uma *Activity* [?] que fará a interação com o usuário, e um *Service* [?] que fará o processo de navegação do robô. O robô e o tablet, se comunicarão via *bluetooth*, ou seja, para o funcionamento do sistema, o tablet deve estar pareado com o robô.

A *Activity* irá obter a posição do clique em relação à imagem do mapa, e obter as coordenadas da posição de destino do robô e repassar para o *Service*. O usuário deve clicar no botão conectar para que a aplicação possa iniciar uma conexão *bluetooth* com o robô. Haverá uma imagem com o mapa do ambiente, o usuário deve apontar o local no qual o robô deve atingir. E então clicar no botão iniciar.

Um *Service* será iniciado, e executará os seguintes passos:

- O mapa topológico será carregado na memória.
- A posição do robô será calculada, medindo o RSS [?] dos APs e aplicando o método descrito em [?].
- Em seguida será aplicado o algoritmo A\* [?] na matriz gerada na etapa 1. A função *h* (heurística) levará em consideração o valor da célula (células com valor inferior a 3 serão descartados do caminho) e a sua distância euclidiana, em relação à célula destino.

- Com o trajeto definido, um comando (vetor de bytes indicando distância e direção) será enviado ao robô.
- O Service aguarda o recebimento dos dados do robô.
- Se nenhum obstáculo é encontrado um novo comando é enviado, e processo se repete, até que o robô atinja seu objetivo.
- Se o robô encontrar um novo obstáculo, a célula onde este foi encontrado é incrementado em 1. Então o processo de descoberta de trajeto é iniciado novamente.
- Se por acaso o robô verificar está célula de novo e não houver obstáculo, o valor da célula é decrementado em 1.
- Se o robô ficar preso, então é adotado um procedimento descrito em [?], muros virtuais são criados, para que o robô evite este caminho em uma segunda passagem.

Aplicação Arduino faz a interface com os motores, o sonar e o adaptador *bluetooth*. Basicamente, ela executa o comando recebido, e então envia os dados coletados pelo sonar e aguarda um novo comando.

## 6. Trabalhos Futuros

As possibilidades de trabalhos futuros são enormes. Primeiro poderia implantar o modelo de propagação de sinal, proposto no artigo [?]. Nele são levados em consideração diversas variáveis para sua elaboração, sendo que a mais importante delas é a quantidade de obstáculos que estão entre o transmissor de sinal (Access Point) e o receptor (terminal móvel). Com base nesse parâmetro, uma equação da distância em função da força de sinal recebido poderá ser deduzida, que será útil para os cálculos da posição e rastreamento do terminal móvel. Ou ainda combinar técnicas baseadas em RSS com as tecnologias já largamente empregadas, tais como o GPS.

Poderia também, ao contrário do sistema proposto, tratar a possibilidade de não haver um mapa do ambiente, e utilizar técnicas que lidam com o SLAM, como nos artigos [?][?][?]. E utilizar um algoritmo de planejamento de trajetos mais robusto como o proposto em [?], que utiliza diagrama de Voronoi para criar um *roadmap*. E ainda ao invés de usar um simples sonar para detectar obstáculos, utilizar a câmera do tablet.

Podemos aumentar a escala e ao invés de utilizar um simples robô de 40 cm dentro de um prédio, utilizar o sistema de navegação em um carro, em um ambiente muito maior como pode ser visto em [?].

## 7. Conclusão

Os robôs vem sendo muito utilizados na automatização de tarefas, e nesse trabalho podemos perceber que uma tarefa simples, como deslocar um robô de um lugar a outro, de forma autônoma, exige técnicas complexas de estatística e de várias áreas da computação como inteligência artificial, geometria computacional e processamento de imagens.

O conceito por trás desse trabalho é muito promissor, pois, a idéia de poder clicar em um mapa, ou falar o endereço onde você deseja ir, e seu celular ou tablet dirige seu carro para você, sem precisar tocar no volante, é extremamente interessante.

## Referências

- [1] Android, “Android home page.” [Online]. Available: <http://www.android.com/>
- [2] Arduino, “Arduino home page.” [Online]. Available: <http://arduino.cc/>
- [3] A. H. P. Selvatici, “Construção de mapas de objetos para navegação de robôs,” in *Advances in Computer Science*. The Digital Library of Theses and Dissertations of the University of São Paulo, 2009.
- [4] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, “Deployment of mobile robots with energy and timing constraints,” in *IEEE TRANSACTIONS ON ROBOTICS*, VOL. 22, NO. 3, 2006.
- [5] P. R. G. D. OLIVEIRA, “Auto-localizaÇão e construÇão de mapas de ambiente para robôs móveis baseados em visão omnidirecional estéreo.” The Digital Library of Theses and Dissertations of the University of São Paulo, 2008.
- [6] C. Luo and S. X. Yang, “A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments.” *IEEE TRANSACTIONS ON NEURAL NETWORKS*, VOL. 19, NO. 7, 2008.
- [7] Z. Wang, Y. Wang, M. Ma, and J. Wu, “Efficient localization for mobile sensor networks based on constraint rules optimized monte carlo method.” *Computer Networks*, 2013. [Online]. Available: [dx.doi.org/10.1016/j.comnet.2013.06.010](http://dx.doi.org/10.1016/j.comnet.2013.06.010)
- [8] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, “Global positioning system: Theory and practice,” in *Advances in Computer Science*. Springer Verlag, 2001.
- [9] D. Niculescu and B. Nath, “Ad hoc positioning system (aps) using aoa.” *IEEE INFOCOM 2003*, 2003. [Online]. Available: [http://www.ieee-infocom.org/2003/papers/42\\_04.pdf](http://www.ieee-infocom.org/2003/papers/42_04.pdf)
- [10] A. Savvides, C. Han, and M. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors.” *MobiCom*, 2001.
- [11] P. Bahl and V. N. Padmanabhan, “Radar: An in-building rf-based user location and tracking system.” *IEEE INFOCOM 2000*, 2000. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=832252&queryText%3DRADAR%3A+An+In-Building+RF-based+User+Location+and+Tracking+System>
- [12] S. C. Yun, V. Ganapathy, and T. W. Chien, “Enhanced d\* lite algorithm for mobile robot navigation.” *2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2010)*.
- [13] P. BHATTACHARYA and M. L. GAVRILOVA, “Roadmap-based path planning using the voronoi diagram for a clearance-based shortest path.” *IEEE Robotics and Automation Magazine*, 2008.
- [14] “Android developers.” [Online]. Available: <http://developer.android.com/index.html>
- [15] S. W. Moon, Y. J. Kim, H. J. Myeong, C. S. Kim, N. J. Cha, and D. H. Kim, “Implementation of smartphone environment remote control and monitoring system for android operating system-based robot platform.”
- [16] OpenCV, “Hough line transform.” [Online]. Available: [http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough\\_lines/hough\\_lines.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html)

- [17] P. Lester, "A\* pathfinding for beginners." [Online]. Available: <http://www.policyalmanac.org/games/aStarTutorial.htm>
- [18] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem." IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 17, NO. 3, 2001.
- [19] X. Zhu, X. Wu, and G. Chen, "Relative localization for wireless sensor networks with linear topology." Computer Communications, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S014036641300176X>
- [20] N. Patwari, A. O. Hero III, and J. A. Costa, "Learning sensor location from signal strength and connectivity.secure localization and time synchronization for wireless sensor and ad hoc network." Advances in Information Security series 30, 2006.
- [21] C. T. Pozzer, "Planejamento de caminho." [Online]. Available: [http://www-usr.inf.ufsm.br/~pozzer/disciplinas/cga.13\\_path.pdf](http://www-usr.inf.ufsm.br/~pozzer/disciplinas/cga.13_path.pdf)
- [22] G. de Lima Ottoni, "Planejamento de trajetórias para robôs móveis," 2000. [Online]. Available: [http://liberty.princeton.edu/Publications/seniorthesis\\_ottoni.pdf](http://liberty.princeton.edu/Publications/seniorthesis_ottoni.pdf)
- [23] D. P. Rodrigues, "Técnicas de navegação." [Online]. Available: [http://www.dca.fee.unicamp.br/~gudwin/courses/IA009/artigos/IA009\\_2010\\_22.pdf](http://www.dca.fee.unicamp.br/~gudwin/courses/IA009/artigos/IA009_2010_22.pdf)
- [24] "Android app component: Activities." [Online]. Available: <http://developer.android.com/guide/components/activities.html>
- [25] "Android app component: Service." [Online]. Available: <http://developer.android.com/guide/components/services.html>
- [26] "Get wifi signal strength of connected network." [Online]. Available: <http://stackoverflow.com/questions/11389492/get-wifi-signal-strength-of-connected-network>
- [27] E. Guizzo, "How google's self-driving car works," A. B. Smith-Jones, Ed. Discovery News, 2011. [Online]. Available: <http://news.discovery.com/autos/future-of-transportation/how-google-self-driving-car-works-111018.htm>