

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA ELÉTRICA**



**Mapeamento e Localização Simultânea de  
Ambientes Dinâmicos Aplicados na Navegação de  
Veículo Autônomo Inteligente**

**Elias Ramos Vilas Boas**

**Itajubá, Maio de 2011**

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA ELÉTRICA**



**Elias Ramos Vilas Boas**

**Mapeamento e Localização Simultânea de Ambientes  
Dinâmicos Aplicados na Navegação de Veículo Autônomo  
Inteligente**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências Engenharia Elétrica.

Área de Concentração: Automação e Sistemas Elétricos Industriais

Orientador: Prof. Dr. Leonardo de Mello Honório

Co-Orientador: Prof. Dr. Augusto Santiago Cerqueira

**Maio de 2011**

**Itajubá - MG**

*"Quem não vê bem uma palavra,  
não pode ver bem uma alma".*

***Fernando Pessoa***

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por ter chegado ao fim desta jornada.

Aos meus pais Paulo Marcos Vilas Boas e Joalice dos Reis Vilas Boas pelo apoio, confiança, amor e também pelas palavras de conforto nos momentos de maior dificuldade.

Ao meu orientador, professor Leonardo de Melo Honório pelo estímulo e paciência para realização deste trabalho.

A Ana Sophia, por entender todas as minhas dificuldades, ajudar a superá-las e por ser meu ponto de apoio nas horas difíceis. Sou eternamente grato a você.

Aos meus companheiros de república pelo companheirismo.

E, finalmente, a todos que participaram desta vitória.

## RESUMO

Um dos maiores desafios que a robótica móvel autônoma enfrenta na atualidade é o problema da localização e mapeamento simultâneos, conhecido como SLAM. Esse problema surge por causa da complexidade da tarefa de navegar por um ambiente desconhecido e ao mesmo tempo capturar informações desse ambiente, construir um mapa e se localizar no mesmo. Os erros gerados pela imprecisão dos sensores, que se acumulam com o passar do tempo, utilizados para estimar o estado de localização e mapeamento impedem que sejam obtidos resultados confiáveis após longos períodos de navegação.

Para resolver o problema descrito este trabalho apresenta um algoritmo de SLAM probabilístico onde, o algoritmo proposto procura eliminar esses erros resolvendo ambos os problemas simultaneamente, utilizando as informações de uma etapa para aumentar a precisão dos resultados alcançados na outra e vice versa. Para tal, é utilizado um mapa métrico para representar o ambiente em que o veículo esta inserido. Este mapa é construído de forma incremental utilizando a teoria de Bayes e a estimação da posição do veículo é feita por um Filtro  $\alpha - \beta$  e é corrigida por um método de sobreposição de obstáculo. Para demonstração da metodologia foi utilizado em um veículo autônomo inteligente.

**Palavras-chave:** Veículo Autônomo Inteligente, SLAM. Localização e Mapeamentos Simultâneos, Teorema de Bayes, Filtro  $\alpha - \beta$ , Mapas Métricos, Sobreposição de Obstáculos.

## ABSTRACT

One of the greatest challenges that mobile robotics face nowadays is the matter of simultaneous mapping and location of a vehicle, known as SLAM (*Simultaneous Localization and Mapping*). This problem arises due to the high complexity of navigating in an unknown environment and capture its information at the same time. The errors resulted from the impreciseness of the sensors accumulate with time and affect the estimation of the localization state and mapping. Therefore, they do not permit that reliable data is generated after long navigation periods.

In order to solve this problem, this work presents a probabilistic SLAM algorithm that eliminates these errors, solving both problems simultaneously, using information of one stage to increase precision of the obtained results and vice versa. To do so, a metric map is used to represent the environment where the vehicle is inserted. This map is made incrementally using the Bayes theorem and the estimation of the vehicle location is made by a  $\alpha - \beta$  filter and corrected by an obstacle overlapping method. To verify the suitability of such theory, the generated algorithms were applied to an intelligent autonomous vehicle.

Keywords: Intelligent Autonomous Vehicle, Simultaneous Location and Mapping, Bayes Theorem,  $\alpha - \beta$  Filter, Metric Maps.

## SUMÁRIO

LISTA DE FIGURAS.....	ix
LISTA DE ABREVIATURAS E SIGLAS .....	xi
LISTA DE SÍMBOLOS.....	xii
1 INTRODUÇÃO.....	13
2 ESTADO DA ARTE DE SLAM .....	16
2.1 INTRODUÇÃO.....	16
2.2 EVOLUÇÃO HISTÓRICA .....	17
2.3 LOCALIZAÇÃO E MAPEAMENTO SIMULTÂNEOS (SLAM). 18	
2.3.1 LOCALIZAÇÃO.....	21
2.3.2 MAPEAMENTO .....	22
2.4 PRINCIPAIS TÉCNICAS .....	26
3 METODOLOGIA.....	28
3.1 INTRODUÇÃO .....	28
3.2 TÉCNICA DE LOCALIZAÇÃO .....	29
3.2.1 FILTRO $\alpha - \beta$ .....	30
3.3 MAPEAMENTO .....	33
3.4 MÉTODO DE SLAM PROBABILÍSTICO .....	35
4 ALGORITMO .....	37
4.1 INTRODUÇÃO.....	37
4.2 AQUISIÇÃO DE DADOS .....	37
4.2.1 DGPS.....	38

4.2.2	<i>ENCODERS</i> .....	39
4.2.3	<i>SCANNER LAZER</i> .....	40
4.3	PROCESSAMENTO DOS DADOS .....	41
4.3.1	MAPEAMENTO .....	41
4.3.2	CORREÇÃO POR SOBREPOSIÇÃO.....	45
4.4	FERRAMENTA COMPUTACIONAL.....	47
4.4.1	MAPA DE OBSTÁCULOS .....	47
4.4.2	ABAS QUE COMPÕEM A FERRAMENTA.....	50
5	RESULTADOS .....	53
5.1	INTRODUÇÃO.....	53
5.2	ESTACIONAMENTO 1 .....	53
5.3	ESTACIONAMENTO 2 .....	56
5.4	RESULTADOS.....	58
5.4.1	CIRCUITO 1 .....	58
5.4.2	CIRCUITO 2 .....	64
5.4.3	CIRCUITO 3 .....	67
5.4.4	CIRCUITO 4 .....	70
6	CONCLUSÃO.....	74
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	75

## LISTA DE FIGURAS

Figura 2.1 - a) Localização sem erros b) Localização com erros c) Mapeamento sem erros d) Mapeamentos com erros.....	20
Figura 2.2 - Tipos de Mapas (34).....	24
Figura 3.1 - Representação da Posição do Veículo.....	32
Figura 4.1 - Erro do DGPS.....	39
Figura 4.2 - Leque de varredura do <i>scanner laser</i> .....	41
Figura 4.3 - Posição Obstáculo no Ambiente.....	42
Figura 4.4 - Matriz de Obstáculos.....	43
Figura 4.5 - Matriz de Obstáculos - <i>Grid</i> .....	44
Figura 4.6 - Matriz de Obstáculos – Obstáculo.....	44
Figura 4.7 - Representação da Sobreposição.....	47
Figura 4.8 - Representação da Matriz de Obstáculo em forma de mapa.....	48
Figura 4.9 - Representação do Veículo;.....	48
Figura 4.10 - Área de Varredura do <i>Scanner Laser</i> .....	49
Figura 4.11 - Área de Varredura do <i>Scanner Laser</i> com Obstáculo.....	49
Figura 4.12 - Variação da Tonalidade Conforme a Probabilidade.....	50
Figura 4.13 - Representação da Matriz de Probabilidade.....	51
Figura 4.14 - Representação da Matriz de Obstáculos.....	51
Figura 4.15 - Percurso.....	52
Figura 5.1 - Estacionamento 1.....	54
Figura 5.2 - Estacionamento 1 Vista Laterais.....	54
Figura 5.3 - Estacionamento 1 Vista Superior.....	55
Figura 5.4 - Estacionamento 2.....	56
Figura 5.5 - Estacionamento 2 Vista Laterais.....	57
Figura 5.6 - Estacionamento 2 Vista Superior.....	57
Figura 5.7 - Circuito 1 Configuração.....	59
Figura 5.8 - (a) Percurso; (b) Mapa de Obstáculo;.....	59
Figura 5.9 - (a) Percurso; (b) Mapa de Obstáculo;.....	60
Figura 5.10 - (a) Percurso; (b) Mapa de Obstáculo.....	61
Figura 5.11 - (a) Percurso; (b) Mapa de Obstáculo.....	61

Figura 5.12 - (a) Mapa de Obstáculo <i>Grid</i> 1,0; (b) Mapa de Obstáculo <i>Grid</i> 1,0 sem Correção.....	62
Figura 5.13 - Percurso do Veículo.....	63
Figura 5.14 - Ângulo do Veículo.....	63
Figura 5.15 - (a) Mapa de Obstáculo <i>Grid</i> 0,5; (b) Mapa de Obstáculo <i>Grid</i> 0,5 sem Correção.....	64
Figura 5.16 - Circuito 2 Configuração.....	65
Figura 5.17 - (a) Mapa de Obstáculo <i>Grid</i> 1,0; (b) Mapa de Obstáculo <i>Grid</i> 1,0 sem Correção.....	65
Figura 5.18 - Percurso do Veículo.....	66
Figura 5.19 - Ângulo do Veículo.....	66
Figura 5.20 - (a) Mapa de Obstáculo <i>Grid</i> 0,5; (b) Mapa de Obstáculo <i>Grid</i> 0,5 sem Correção.....	66
Figura 5.21 - Circuito 3 Configuração.....	67
Figura 5.22 - (a) Início Obstáculo Dinâmico; (b) Fim Obstáculo Dinâmico. ....	68
Figura 5.23 - (a) Mapa de Obstáculo <i>Grid</i> 1,0; (b) Mapa de Obstáculo <i>Grid</i> 1,0 sem Correção.....	68
Figura 5.24 - Percurso do Veículo.....	69
Figura 5.25 - Ângulo do Veículo.....	69
Figura 5.26 - (a) Mapa de Obstáculo <i>Grid</i> 0,5; (b) Mapa de Obstáculo <i>Grid</i> 0,5 sem Correção.....	69
Figura 5.27 - Circuito 4 Configuração.....	70
Figura 5.28 - (a) Obstáculo Dinâmico 1; (b) Obstáculo Dinâmico 2. ....	71
Figura 5.29 - Obstáculo Dinâmico 3. ....	72
Figura 5.30 - (a) Mapa de Obstáculo <i>Grid</i> 1,0; (b) Mapa de Obstáculo <i>Grid</i> 1,0 sem Correção.....	72
Figura 5.31 - Percurso do Veículo.....	73
Figura 5.32 - Ângulo do Veículo.....	73
Figura 5.33 - (a) Mapa de Obstáculo <i>Grid</i> 0,5; (b) Mapa de Obstáculo <i>Grid</i> 0,5 sem Correção.....	73

## LISTA DE ABREVIATURAS E SIGLAS

SLAM	<i>Simultaneous Localization and Mapping,</i>
GPS	<i>Global Position System,</i>
DGPS	<i>Differential Global Position System,</i>
IMU	Unidade de Medida Inercial (Inertial Measurement Unit),
CML	<i>Concurrent Mapping and Localization,</i>
ISRR	<i>International Symposium on Robotics Research,</i>
TOA	<i>Time of Arrival,</i>
EKF	<i>Extended Kalman Filter,</i>
ABS	<i>Anti-lock Braking System,</i>
MO	Matriz de Obstáculos
MP	Matriz de Probabilidade,
RU	Restaurante Universitario,

## LISTA DE SÍMBOLOS

$X_k$	Vetor de estado do sistema no instante $k$ ,
$A$	Matriz que define a dinâmica do sistema,
$Z_k$	Dados medidos no instante $k$ ,
$H$	Matriz de que define a saída de medição,
$\bar{P}$	Vetor posição do veículo,
$x$	Posição $x$ do veículo,
$y$	Posição $y$ do veículo,
$\psi$	Ângulo de orientação do veículo,
$M$	Matriz que representa a grade de ocupação,
$m$	Região do ambiente representado pela grade de ocupação,
$C$	Comprimento do veículo,
$L$	Largura do veículo,
$S$	Distância entre o <i>scanner laser</i> e o DGPS,
$d$	Distância do obstáculo ao veículo, fornecido pelo <i>scanner laser</i> ,
$\theta$	Ângulo do obstáculo com referência do <i>scanner laser</i>
$x_0$	Posição $x$ do obstáculo,
$y_0$	Posição $y$ do obstáculo,

# CAPÍTULO I

## 1 INTRODUÇÃO

Localização e Mapeamento Simultâneos ou SLAM (*Simultaneous Localization and Mapping*), pode ser definido como um problema onde um robô autônomo deve, através de seus sensores, monitorar o ambiente ao seu redor buscando identificar obstáculos fixos ou dinâmicos. Com a correta definição do posicionamento destes obstáculos, o robô pode ser capaz de montar um mapa e determinar sua posição dentro deste ambiente. Como tanto a definição da trajetória como a localização dos obstáculos ou marcações são realizados em tempo real (sem a necessidade de nenhum conhecimento prévio), o SLAM é o primeiro passo em busca da autonomia completa, principalmente se a aplicação for de veículos autônomos não tripulados.

Entretanto, esta é uma tarefa complexa que envolve grande grau de incerteza. A primeira fonte de incerteza vem dos próprios sensores utilizados para realizar o SLAM. A literatura mostra uma gama de possibilidades [1] [2] [3], onde câmeras monoculares, com visão estéril, DGPS, sensores laser 2D e 3D, IMU (*Inertial Measurement Unit*) fornecem uma quantidade enorme de dados que podem ser, e na realidade muitas vezes são, incompatíveis. As fontes desta incompatibilidade são diversas. Podem ser advindos de uma câmera identificando uma sombra como um obstáculo não percebido pelo sensor de varredura laser, ou o problema oposto, um sensor laser não identificando algum obstáculo (fachadas de vidro, ou veículos prata) que foram identificados pela câmera ou ainda a simples inclinação do veículo.

Outro ponto de incerteza apontado na literatura [4] refere-se ao fechamento de uma volta, isto é, quando o veículo retorna a algum ponto mapeado anteriormente. Nesta situação, se os obstáculos estáticos identificados pelo mapeamento atual não baterem com os mapeados previamente, todo o mapeamento pode ficar com sua confiabilidade comprometida. Apesar de técnicas de distribuição de erro através da modelagem por grafos são utilizadas para minimizar este problema [5], os resultados ainda são indesejados.

Um terceiro problema, indiretamente relacionado ao fechamento de uma volta é a correta identificação do que é um obstáculo estático ou dinâmico. Em ambientes reais um carro estacionado por dias em um mesmo lugar não é um obstáculo estático. A correta identificação do que é estático é fundamental para que a segunda situação de incerteza não aconteça. Outro problema relacionado à identificação de obstáculos estáticos é a mudança natural do ambiente. Neste sentido mapeamentos mais antigos correm o risco de não estarem devidamente atualizados. Desta forma, além da incerteza do mapeamento, deve-se trabalhar com a incerteza de que se os obstáculos identificados são ou não estáticos, o que traz um novo conjunto de desafios.

Para tratar destes problemas, o presente trabalho apresenta uma técnica hierárquica matricial. O primeiro passo para entender a metodologia é através da conceituação da matriz de obstáculo ou MO. A MO é uma matriz responsável por representar o espaço no mundo observado. Cada ponto nesta matriz é um quadrado representando uma região específica do espaço que pode conter um obstáculo, neste caso associar um valor 1, ou estar vazia associando-se 0. Como a posição dos obstáculos pode variar devido às incertezas é necessário sua captura e tratamento. Para tanto uma segunda matriz, da mesma dimensão de MO irá definir qual a probabilidade de que um dado ponto  $mo_{i,j}$  seja igual a 1. Tal matriz é denominada de matriz de probabilidade ou MP. Para a identificação da posição corrigida do veículo dentro do mapa uma técnica de SLAM probabilístico será utilizada [6], entretanto algumas heurísticas serão adicionadas ao problema para computar a probabilidade de um dado obstáculo ser estático ou dinâmico.

Apesar de que, como os resultados irão mostrar, a abordagem do problema definida como no passo anterior produz bons resultados, é computacionalmente muito pesada em um ambiente externo de grande porte. Ter uma única matriz representando o mundo é uma situação que a memória crescerá ao quadrado do problema, levando rapidamente a um estouro no esforço computacional. Entretanto é bastante óbvio que as diversas partes de um mapa, apesar de estarem fortemente conectadas umas com as outras, não necessitam estar correlacionadas com os obstáculos umas das outras. Por exemplo, se o veículo está em uma rua A, ele pode precisar saber aonde e como chegar à rua B, mas não precisa saber se um poste ou uma casa previamente mapeado vão ou não estar lá. Da mesma forma o veículo só precisa correlacionar obstáculos próximos a ele. Obstáculos distantes serão alocados em diferentes matrizes e estas serão associadas/correlacionadas através de elementos de transição.

Ao abordar o problema desta forma, três grandes problemas são resolvidos; tem-se um modelo computacionalmente eficiente, consegue-se uma boa precisão local e reduz-se o erro global no processo de fechamento de mapas.

Para contextualizar as técnicas apresentadas, este trabalho se divide da seguinte forma. No capítulo 2 será apresentado o estado da arte de SLAM, sua evolução histórica e principais técnicas. O modelo matemático da metodologia utilizada e os algoritmos de implementação utilizados serão apresentados nos capítulos 3 e 4 respectivamente. Por fim, os resultados e as conclusões são apresentados nos capítulos 5 e 6.

## CAPÍTULO II

### 2 ESTADO DA ARTE DE SLAM

#### 2.1 INTRODUÇÃO

Localização e Mapeamento Simultâneos (SLAM) é um dos problemas fundamentais para a navegação autônoma de robôs [7], e por isso é um campo da robótica que vem sofrendo intensa pesquisa. Esse problema pode ser encarado como a dificuldade que um robô possui de, partindo de uma posição desconhecida, em um ambiente também desconhecido, construir um mapa dos locais já visitados e simultaneamente ser capaz de se localizar dentro desse mapa. Uma solução para esse problema permitiria o surgimento de robôs verdadeiramente autônomos, capazes de navegar de maneira segura por ambientes desconhecidos e cumprir objetivos sem a necessidade de auxílio externo de espécie alguma [8].

O mapeamento é um problema que consiste na integração de informações obtidas pelos sensores de um robô dentro de uma determinada representação. Este problema pode ser descrito pela pergunta "Com que o mundo se parece?". Os aspectos centrais de mapeamento são a representação do ambiente e a interpretação dos dados do sensor. Em contraste com isso, a localização é o problema de estimar a posição do robô em relação a um mapa. Em outras palavras, o robô tem de responder à questão: "Onde eu estou?". Normalmente, distingue-se entre uma posição de trajetória, onde a posição inicial do robô é conhecida, e localização global, em que não é dado nenhum conhecimento *a priori* sobre a posição de partida.

SLAM é definido como o problema de construção de um mapa e, ao mesmo tempo localizar o robô dentro desse mapa. Na prática, esses dois problemas não podem ser resolvidos de forma independente um do outro. Antes que um robô possa responder à questão com qual ambiente este se parece, a partir de um determinado conjunto de observações, ele precisa saber de que locais essas observações foram feitas. Ao mesmo tempo, é difícil estimar a posição atual de um robô sem um mapa. Portanto, SLAM é muitas vezes referido como o

problema do ovo e da galinha: Um bom mapa é necessário para a localização, enquanto uma estimativa exata da posição é necessária para construir um mapa.

## 2.2 EVOLUÇÃO HISTÓRICA

Durante a década de 80, alguns trabalhos apresentaram as bases para que se pudesse alcançar uma definição probabilística para o problema de mapeamento [9]. Smith e Cheesman, como descrito em [10], foram os primeiros a estabelecer uma base estatística relacionando os marcos (“*landmarks*”), que são estruturas do ambiente utilizadas pelo robô como ponto de referência para se localizar durante a navegação, identificados através de incertezas geométricas. Isso significa que, para um mapeamento ser consistente, a correlação entre as estimativas de diferentes marcos deveria aumentar de precisão, ou seja, crescer caso houvesse sucessivas observações.

Com o trabalho de Smith [11], foi mostrado que estando o robô em um ambiente desconhecido coletando observações de marcos para o mapeamento, as estimativas destes marcos são correlacionadas devido ao erro de localização do robô. Com isto, tornou-se possível estabelecer uma solução comum para os problemas de localização do robô e do mapeamento do ambiente, pois foi definido um vetor de estado que continha a posição do robô juntamente com a localização de cada marco observado. Para cada nova observação realizada, havia uma atualização desta estrutura. Este novo mapa foi chamado de mapa estocástico.

Crucialmente, este trabalho não olhou para as propriedades de convergência do mapa ou o seu comportamento em estado estacionário. Na verdade, foi amplamente assumido, visto que os erros estimados no mapa não convergiram, e em vez disso, exibiram em um comportamento aleatório, com um crescimento de erros ilimitado. Assim, dada a complexidade computacional do problema de mapeamento e sem o conhecimento do comportamento de convergência do mapa, os pesquisadores, ao invés de focar em uma série de aproximações para o problema de mapeamento consistente, os quais forçam que as correlações entre os marcos sejam minimizadas ou eliminadas, reduziram o filtro completo para uma série de marcos desassociados (por exemplo, [10] e [12]). Também por estes motivos, o trabalho teórico sobre o problema de mapeamento e localização combinado chegou

a um impasse temporário, com o trabalho muitas vezes focado em mapeamento ou localização como problemas distintos.

A descoberta conceitual veio com a percepção de que o problema de mapeamento e localização combinado, uma vez formulado como um problema de estimação simples, era realmente convergente. Mais importante ainda, foi reconhecido que as correlações entre os marcos, que a maioria dos pesquisadores tentaram minimizar, eram na verdade a parte crítica do problema e que ao contrário, quanto mais estas correlações cresciam, melhor eram a solução. A estruturação do problema SLAM, a convergência dos resultados e a criação do acrônimo SLAM foram apresentados pela primeira vez em 1995, no Simpósio Internacional sobre Robotics Research, em um trabalho de pesquisa de robótica móvel [13]. A teoria fundamental sobre a convergência e muitos dos resultados iniciais foram desenvolvidos por Csorba [14]. Vários outros grupos já trabalhavam com mapeamento e localização, notavelmente no Instituto de tecnologia de Massachusetts [10], Zaragoza [15] [16] ACFR em Sidney [17] [18] e outros [19], e começaram trabalhando seriamente com SLAM - também conhecido como *concurrent mapping and localization* (CML) - em ambientes internos, externos e subaquáticos.

Atualmente, os trabalhos estão focados na melhoria da eficiência computacional que abordam questões como associação de dados ou fechamento de voltas. Um ponto importante foi o Simpósio Internacional de Robotics Research em 1999 (ISRR'99) [10] onde foi realizada a primeira sessão SLAM e onde foi alcançado um grau de convergência entre os métodos baseados em filtro de Kalman e métodos de mapeamento e localização probabilística introduzidos por Thrun [20].

Há muito mais o que dizer sobre a história das descobertas que foram feitas em SLAM, mas os fatos que estão mencionados aqui são talvez os mais importantes [10].

## **2.3 LOCALIZAÇÃO E MAPEAMENTO SIMULTÂNEOS (SLAM)**

Localização é um problema muito importante na navegação de robôs autônomos e está relacionado com a capacidade do robô em conhecer a sua posição, em um sistema de coordenadas, através do uso de sensores. Similarmente, o mapeamento relaciona a capacidade de distribuir, dentro desse mesmo sistema de coordenadas, as estruturas que se encontram ao

redor do robô. Fornecer um mapa confiável ao robô do ambiente em que está inserido, nem sempre é possível, pois este pode mudar durante a navegação ou até mesmo ser desconhecido, cabendo ao robô gerar e explorar esse mapa [21]. Com o mesmo princípio, nem sempre é possível fornecer uma localização precisa do veículo a cada instante, pois existem situações onde sistemas absolutos de posicionamento, como o GPS, não funcionam ou não fornecem resultados confiáveis [22].

Nesses casos, o robô deve ser capaz de estimar a sua localização e de mapear o ambiente através dos seus próprios sensores embarcados. As informações geradas pelos sensores podem possuir erros por causa das limitações físicas dos mesmos ou por fenômenos que não podem ser modelados devido a sua complexidade ou imprevisibilidade. Esses erros fazem com que as estimativas geradas pelo robô sejam imprecisas, limitando assim a utilização em algoritmos posteriores que dependam de resultados mais precisos de mapeamento e localização.

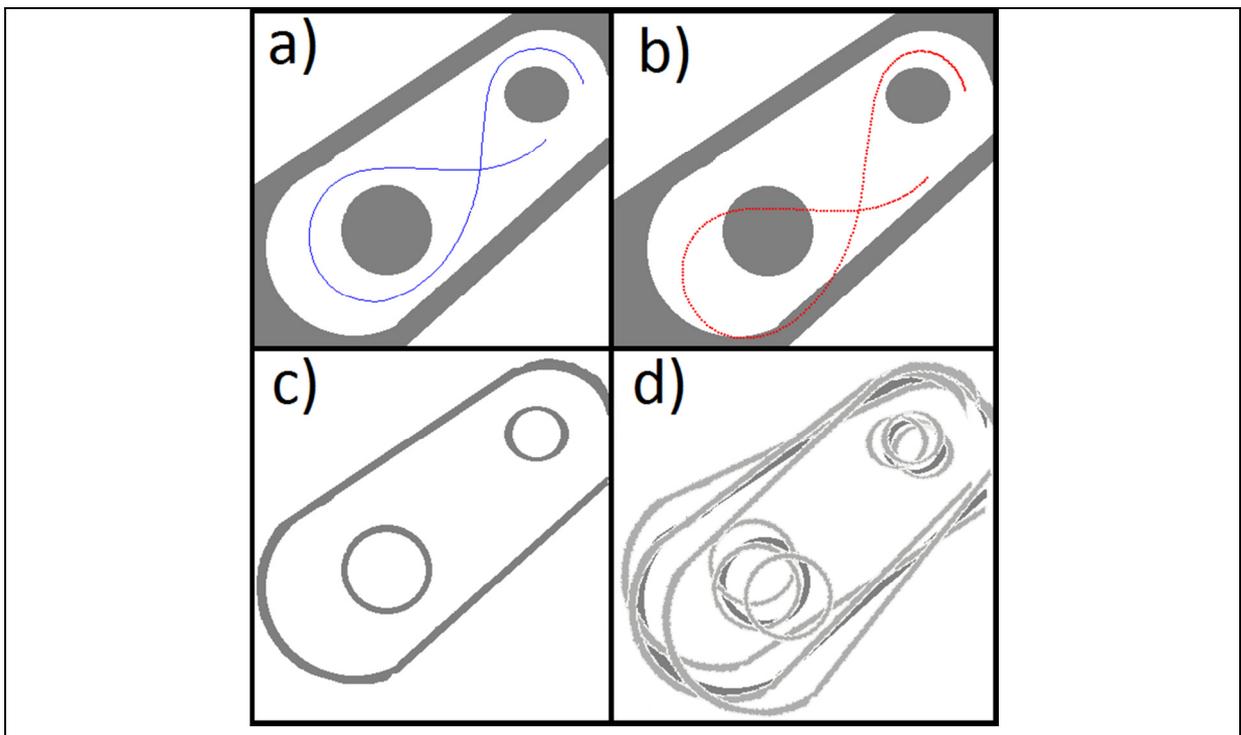
A primeira tentativa proposta para resolver a problema dos erros foi construir sensores cada vez mais precisos, que possuiriam um erro inerente cada vez menor, proporcionando assim um resultado mais confiável. O aumento da precisão dos sensores fez com que estes se tornassem cada vez mais caros e com isso esse tipo abordagem deixou de ser atrativa, sendo necessários investimentos cada vez maiores para gerar aumentos de precisão cada vez menores. É obvio que será impossível eliminar completamente os erros inerentes aos sensores, por causa de fatores que estão fora do controle do projetista. Esse tipo de abordagem apenas oculta o problema ao invés de eliminá-lo. Em uma navegação suficientemente prolongada esses erros se tornariam grandes o suficiente para comprometer os resultados finais.

Sendo impossível eliminar esses erros e com o surgimento da chamada robótica probabilística [23], observou então que os erros passaram a ser incorporados aos modelos utilizados, que até então apresentavam características determinísticas e onde os estados de localização e mapeamento do robô apresentavam valores únicos. A obtenção de estimativas precisas se torna então uma questão de minimizar as incertezas contidas em cada uma das variáveis de interesse, o que pode ser feito a partir do acúmulo de informações [24] visando eliminar os erros aleatórios gerados pelos ruídos apresentados pelos sensores.

O problema de mapeamento e localização possui solução simples quando tratados de forma separados, ou seja, quando o outro já estiver resolvido, como é o caso da localização do

robô dado o mapa do ambiente ou o mapeamento dado a localização do veículo a cada instante. Esse fato acontece porque ambos os problemas não podem ser resolvidos de forma independente, pois estão vinculados entre si. Para o mapeamento é necessário a localização do robô para que possam ser distribuídas as informações obtidas pelos sensores em um sistema de coordenadas, por sua vez, para a localização é necessário o mapa do ambiente para que através das informações obtidas pelos sensores possam estimar a posição do robô no ambiente.

Dessa forma, caso uma dessas estimativas apresente algum erro, esse erro será propagado para a outra, devido ao caráter iterativo do processo, esse erro tende a acumular, invalidando os resultados finais. Por isso, torna-se necessário tratar simultaneamente as incertezas envolvidas no processo procurando minimizá-las a cada iteração, evitando assim que os erros gerados se acumulem. A Figura 2.1 apresenta situações onde o erro é acumulativo gerando uma localização imprecisa e um mapeamento que não corresponde ao real invalidando todo o trabalho.



**Figura 2.1 - a) Localização sem erros b) Localização com erros c) Mapeamento sem erros d) Mapeamentos com erros.**

## 2.3.1 LOCALIZAÇÃO

Um componente fundamental para qualquer sistema de navegação autônoma é a capacidade de se localizar dentro do ambiente no qual se encontra [25]. Para que um robô navegue, partindo de um ponto e chegando a outro, é necessário que o mesmo receba sinais de controle que façam com que ele se movimente pelo ambiente. Em navegação autônoma, o robô deve ser capaz de calcular iterativamente quais são os sinais de controle necessários para sair de uma posição e chegar ao seu objetivo de forma segura e eficiente. Dessa forma, o robô autônomo deve monitorar o seu deslocamento através de sensores, e estimar o seu posicionamento no ambiente, podendo definir se está aproximando ou não do seu destino.

Algumas técnicas utilizadas para localização são descritas a seguir.

### 2.3.1.1 ODOMETRIA

Odometria é o método de navegação mais utilizado para o posicionamento de um robô móvel, isso se deve ao fato de ser um modelo simples de localização que envolve apenas parâmetros internos do robô, e não depende da obtenção de informações do ambiente ao seu redor. Sabe-se que a odometria fornece uma boa precisão de curto prazo, é barato e permite taxas de amostragem muito altas. Um sistema de odometria consegue, com o auxílio de *encoders*, determinar a rotação dos eixos dos atuadores responsáveis pela movimentação do robô em um determinado intervalo de tempo, e integrando esses valores consegue calcular qual o deslocamento realizado nesse intervalo. No entanto, a ideia fundamental de odometria é a integração das informações do movimento incremental ao longo do tempo, o que leva inevitavelmente à acumulação de erros. Particularmente, o acúmulo de erros de orientação irá causar erros de posição de grande porte que aumentam proporcionalmente com a distância percorrida pelo robô, como mostrado na Figura 2.1 b).

### 2.3.1.2 GPS

Outra técnica utilizada para obter a localização é o GPS (*Global Positioning System*). O GPS é um dos sensores utilizados para obter uma informação precisa e absoluta sobre a posição do veículo autônomo. Esse sistema utiliza o conceito de tempo de chegada (TOA - *time of arrival*) para determinar a posição do receptor, neste caso, a do robô autônomo [26]. Devido a grande e complexa interação de vários fatores descritos em [27], a precisão do GPS é

de cerca de 15 metros. Com uma imprecisão tão alta, torna-se necessário a utilização de outras técnicas que complementam o GPS, como o caso do filtro de Kalman e da odometria, podendo ser incorporado no sistema outros sensores para aprimorar a estimação da posição. A maior desvantagem dessa técnica é que só pode ser utilizada em ambientes abertos.

### **2.3.1.3 ESTIMATIVA DE LOCALIZAÇÃO**

A estimativa de localização do robô, cuja incerteza possui um limite superior, só pode ser obtida através de referências absolutas de posicionamento, ou seja, referências que não dependem do estado de localização anterior [28]. Outra possibilidade é a utilização de estruturas fixas no ambiente em que o robô pode utilizar como ponto de referência para se localizar [29]. Essas estruturas são conhecidas como marcos, e devem possuir propriedades que permitam a sua detecção e posterior reconhecimento [30].

A posição desses marcos no ambiente não depende de valores de localização do robô e por isso a estimativa obtida será absoluta, limitada apenas pela incerteza relativa à posição do marco no ambiente e do sensor utilizado para realizar a observação. Essa estimativa pode ser incorporada àquela obtida incrementalmente pela odometria e à fornecida através do DGPS de maneira a gerar uma quarta estimativa mais precisa do que as três iniciais [22].

Não é sempre que um mapa confiável do ambiente pode ser fornecido anteriormente à navegação, seja porque o próprio ambiente ainda não foi explorado ou porque ele ainda não é conhecido, e nessas situações o robô deve também construir incrementalmente o mapa que utilizará para se localizar e movimentar, sendo essa a base para o surgimento do problema do SLAM.

### **2.3.2 MAPEAMENTO**

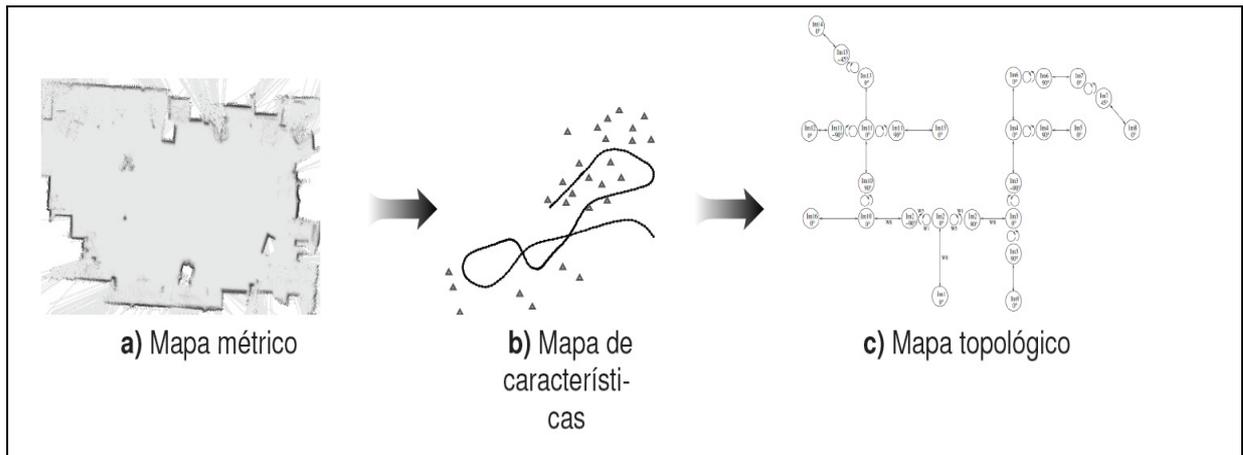
Para que um robô autônomo consiga cumprir os seus objetivos de forma autônoma e segura é necessário que o mesmo seja capaz de construir um modelo do mundo ao seu redor e a partir desse modelo tomar suas decisões. Para construir o modelo do mundo ao seu redor o robô tem que ser capaz de identificar os objetos que estão a sua volta, se possível antes de colidir com eles. Para que isso seja possível é necessária a utilização de sensores que tem a capacidade de perceber o ambiente, como por exemplo, *scanner laser*, câmeras e infravermelho que utilizam a propriedade da luz e sonares que utilizam ondas mecânicas.

Conforme o veículo se movimenta pelo ambiente, ele realiza observações das estruturas ao seu redor e decide se simplesmente pode reagir a elas, como desviar de um obstáculo ou armazenar essas observações para que possam ser utilizadas posteriormente [31].

O armazenamento dessas informações é feito através de mapas, que representam o ambiente em que o robô está inserido. O mapa representa o modelo do mundo observado pelo robô através dos seus sensores e este deve conter informações consistentes. As informações armazenadas podem representar de certa maneira uma característica do ambiente, como por exemplo, a coordenada de um obstáculo no mundo. Quando o robô realiza uma observação, inicialmente deve determinar a qual região do ambiente essa observação pertence ou está vinculada e a partir desse momento atualizar essa informação no mapa de maneira que ela reflita o estado observado. Caso a região não tenha sido observada ainda, a informação é adicionada ao mapa e será atualizada novamente quando essa mesma região for observada novamente.

Os sensores responsáveis pelo mapeamento do ambiente não fornecem informações exatas, pois estão sujeitos a erros que não podem ser previstos ou modelados. Dessa forma, as informações que serão adicionadas ao mapa possuem certo valor de incerteza e isso fará com que o mapa obtido seja apenas uma estimativa da realidade [32]. Essa imprecisão pode ser transmitida ao mapa e com isso uma variável pode ser atualizada com um estado que não corresponde à realidade. Esses erros podem ser corrigidos através do acúmulo de observações [33], que tende a fazer com que erros aleatórios sejam eliminados e o estado da variável convirja para o estado real.

Os mapas expressam a informação que eles contem atribuindo valores aos seus elementos. A natureza desses elementos varia dependendo do tipo de mapa, e pode corresponder a um amplo espectro de possíveis estruturas. Assim, a tarefa de mapeamento corresponde à atribuição de valores aos elementos do mapa, relacionando cada um a certa posição. Os mapas na robótica recebem nomes diferentes de acordo com os elementos que eles usam para descrever o ambiente. O mapa ideal a ser utilizado por um robô depende da tarefa a ser executado e do ambiente em que está inserido. Também dependem das características do robô, tais como os tipos de sensores e como o robô se movimenta. Existem vários tipos de mapa e podem ser classificados em: mapas métricos, mapas topológicos, mapas de características e mapas probabilísticos [34] [35]. A Figura 2.2 ilustra alguns dos mapas que serão abordados a seguir.



**Figura 2.2 - Tipos de Mapas (34).**

### 2.3.2.1 MAPAS MÉTRICOS

O objetivo do mapa métrico é representar o ambiente com um alto nível de detalhe, tratando as informações coletadas pelos sensores e relacionando-as com o tamanho e forma dos objetos e os limites das áreas livres para navegação. Para que isso seja possível, os mapas métricos dividem o espaço 2D ou 3D em células regulares, as quais representam características de interesse sobre o local, como por exemplo, ocupação (possui algum obstáculo que impede a navegação), a probabilidade da área estar vazia (pode ser atravessada pelo robô), dados como o conjunto de estados observados desde o início da navegação. Um exemplo clássico de mapa métrico é o mapa de grade de ocupação [36] [34], que representa o ambiente como uma matriz de forma que cada célula contém a probabilidade de que esse espaço esteja ocupado por um obstáculo ou não.

Em geral, os mapas métricos exigem um alto esforço computacional e um grande espaço na memória, pois a atualização do mapeamento é um processo que envolve somente algumas células de cada vez e o mapa tende a crescer conforme o tamanho da área mapeada. No entanto, o mapa métrico salva as informações no próprio domínio do sinal do sensor, assim a tarefa de localização é executada de forma mais simples e devido a alta densidade de informação o resultado da localização é mais preciso e menos sujeito a ambiguidades [34].

### 2.3.2.2 MAPAS TOPOLÓGICOS

Os mapas topológicos também podem ser chamados de mapas relacionais [34] porque fornecem as relações entre os vários locais ou objetos presentes no ambiente geralmente sob a forma de um gráfico. Mapas topológicos registram informações sobre determinados

elementos ou locais do ambiente, chamados marcos. Assim, os elementos dos mapas topológicos são os marcos e as relações. Essas relações podem ser de vários tipos. Elas podem indicar o descolamento entre dois marcos, a existência de um caminho entre eles, os comandos de controle que movem o veículo de um marco ao outro, e assim por diante [37].

Mapas topológicos exigem muito menos memória quando comparados aos mapas métricos, por essa razão são apropriados para mapeamento de grandes áreas. Segundo Dudek [38], a principal vantagem desta abordagem é sua natureza qualitativa e sua relação com a teoria de cognição humana. Além disso, os mapas topológicos são geralmente menos complexos que os mapas geométricos; muito mais eficientes no processo de planejamento de caminhos [39] e mais flexíveis a noção de estado.

### **2.3.2.3 MAPAS DE CARACTERÍSTICAS**

Outra maneira de se armazenar informações coletadas pelos sensores do veículo é através dos chamados mapas de características, que filtram essas informações em busca de padrões relevantes de alguma forma para a aplicação onde esse mapa será utilizado [40]. Esses padrões podem ser apenas algumas estruturas espaciais, tais como pontos, paredes fronteiriças, ou objetos. Por outro lado, esses padrões ou característica espaciais do ambiente têm sua posição absoluta atribuída, bem como outras eventuais informações relacionadas a elas, essas informações estão relacionadas somente às características armazenadas no mapa, ou seja, essas informações não apresentam relações entre características diferentes como é o caso do mapa topológico.

Essa abordagem gera um menor custo computacional de manutenção, pois lida com uma menor quantidade de informações. Contudo, surge o custo decorrente da busca por padrões nas informações que sejam relevantes para a aplicação em questão, assim como a caracterização desses padrões de maneira a permitir a correspondência com estruturas semelhantes. Como as variáveis em um mapa de características não possuem posições fixas no espaço como acontece em um mapa métrico, a incerteza quanto a essa estimativa pode ser incorporada diretamente no parâmetro que a define, sob a forma de uma distribuição de probabilidades similar a utilizada no tratamento do problema de localização [31].

### 2.3.2.4 MAPAS PROBABILÍSTICOS

Outro tipo de mapa são os mapas probabilísticos proposto por Smith, Self e CheeseMan [11] como uma ferramenta para tratar incertezas espaciais originadas das medidas sensoriais e dos movimentos do robô na representação espacial, ao invés de considerá-las como um problema geométrico à parte. Neste tipo de representação, tanto os obstáculos quanto o robô são representados como um conjunto de objetos que são descritos pela localização juntamente com uma matriz de covariância que descreve o relacionamento espacial entre eles [32].

Este tipo de representação é frequentemente usado com EKF (*Extended Kalman Filter*). O principal problema é a necessidade de armazenar uma matriz de covariância necessária para a convergência do mapa. Esta matriz cresce quadraticamente com a quantidade de objetos observados resultando em um alto esforço computacional quando o ambiente a ser mapeado é grande [35].

## 2.4 PRINCIPAIS TÉCNICAS

Nos últimos anos, os problemas de mapeamento e localização de robôs, nos mais diversos tipos de ambientes, se tornaram uma atrativa área de pesquisa. Há duas principais vertentes para a solução do problema de SLAM: uma usa um filtro estocástico conhecido como Filtro Estendido de Kalman ou EKF (*Extended Kalman Filter*) [7] [41] [42] [43], enquanto a outra utiliza filtro de partícula, também conhecido como FastSLAM [44] [45] [46].

Para estes dois tipos de solução, é necessário que se tenham os modelos do robô e de observação de marcos de referência. Assim é possível utilizar a teoria bayesiana para a atualização do mapa. Sendo este mapa um vetor de estado adicionado de ruído gaussiano, a solução computacional mais utilizada é através do EKF. Porém o grande problema deste método consiste no esforço computacional por causa do crescimento do vetor de estado do sistema (ou do mapa estocástico). Para cada novo marco encontrado, um novo elemento é adicionado ao mapa. Contudo, como demonstrado por [7], a convergência da solução é garantida.

É notado em [47] que o problema do SLAM, dado a localização do robô, pode ser decomposto de maneira exata em um conjunto de problemas menores de estimação que podem ser então resolvidos de maneira independente [31]. O FastSLAM [48] utiliza essa propriedade para, ao invés de lidar com uma única matriz de alta dimensionalidade, lidar com um conjunto de matrizes menores que podem ser resolvidas eficientemente e com um escalonamento computacional menor. Ao invés de se lidar com cada marco individualmente pode-se também dividir o mapa construído pelo robô em um conjunto de mapas menores, que são atualizados localmente com um custo computacional menor. Soluções que utilizam essa abordagem são apresentadas em [49], [50] e [19], conseguindo bons resultados localmente, mas com dificuldades em fechar *loops* muito grandes devido à divergência global gerada pelas atualizações locais.

Há outras soluções aplicadas ao problema de SLAM que não necessariamente seguem as principais metodologias que utilizam filtragem.

Duckett [51] propôs a utilização de algoritmo genético para a resolução do SLAM, tratando-o como um problema de otimização global. A desvantagem, segundo o próprio autor, é ainda o esforço computacional gasto para obter uma representação acurada. Frese [52] apresentaram uma solução utilizando a metodologia de minimização de erro quadrático, modificada de forma a permitir atuação em tempo real.

Já Schoroeter [53], dentro de uma filosofia mais simplificada, utilizaram a correção em tempo real dos erros de odometria para a construção do mapa. Pedrosa apresentaram um SLAM geométrico [54] [55], onde no qual o mapa é do tipo topológico, porém enriquecido com informações métricas dos ambientes definidos como locais (salas, quartos e corredores). Tais informações métricas eram obtidas por informações sensoriais e o erro na posição do robô era mantido dentro de limites devido a consideração do conhecimento abstrato inicial da forma geométrica padrão do ambiente local onde o robô se encontrava.

## CAPÍTULO III

### 3 METODOLOGIA

#### 3.1 INTRODUÇÃO

Este trabalho foi desenvolvido utilizando Veículos Autônomos Inteligentes, que são veículos que operam de forma autônoma, ou seja, sem a intervenção de pessoas. Um automóvel normal foi modificado de forma a possibilitar todo o desenvolvimento deste trabalho. Esse veículo mais todo o sistema mecânico, eletrônico e de *software* embarcado formam o que é chamado de Veículo Autônomo Inteligente. O veículo utilizado é um automóvel Chevrolet Zafira modelo Elite ano 2008 que recebeu uma série de adaptações mecânicas para receber todos os equipamentos do sistema de eletrônica, *software* embarcado, atuadores e sensores necessários para sua automatização.

O Veículo Autônomo Inteligente foi estruturado de forma a permitir que através de um sistema de navegação um automóvel comum automatizado possa realizar determinadas ações de forma autônoma e inteligente sobre o ambiente em que está inserido. O ambiente em que o Veículo Autônomo Inteligente está inserido apresenta as seguintes características:

- Apenas variações no ambiente próximas o suficiente para serem captadas pelos sensores instalados no veículo são percebidas;
- O comportamento do ambiente não pode ser previsto já que é um ambiente constantemente alterado por outros agentes além do veículo;
- O ambiente está sempre alterando seu estado mesmo que o veículo não esteja executando nenhuma ação;
- É um ambiente físico real.

Essas características determinam o ambiente como sendo parcialmente observável, estocástico, sequencial, dinâmico e contínuo [56]. Esses fatores resultam em um ambiente complexo que necessita de algumas simplificações para ser interpretado de forma

conveniente. As simplificações aqui adotadas permitem que o problema de mapeamento seja resolvido de forma simples, segura e confiável.

O foco deste trabalho está voltado para a tarefa de mapeamento e para as ferramentas envolvidas para obter a representação desejada do ambiente de trabalho do Veículo Autônomo Inteligente. Porém, durante o processo de exploração e mapeamento do ambiente, o Veículo Autônomo Inteligente pode perfeitamente utilizar as informações recém-adquiridas do espaço local para atualizar a sua posição. Mas diferentemente das abordagens probabilísticas comumente aplicadas, foi utilizada uma metodologia baseada em relações geométricas. Ou seja, o veículo, valendo-se do conhecimento prévio da estimativa da sua posição através de sensores como DGPS e odometria, tendo o formato geométrico formado pelos obstáculos juntamente com a confiabilidade da posição dos obstáculos que estão no ambiente no qual está inserido, e de informações de distância obtidas de seus sensores, determina a sua posição com maior precisão relativa ao ambiente local. Contudo, nada impede que os métodos baseados no EKF sejam aplicados. Assim, garante-se que o desenvolvimento da abordagem apresentada esteja completamente inserido no contexto de SLAM.

Primeiramente na seção 3.2 será abordada a técnica de localização do veículo. Esta localização é feita de forma independente do mapeamento, através de sensores de posição como o DGPS e o método de odometria. A partir de um filtro  $\alpha-\beta$  e os dados coletados estima-se a posição do veículo.

Na seção 3.3 será mostrado como o ambiente em que o veículo está inserido será representado e como as informações dos sensores serão armazenadas, qual tipo de mapa foi utilizado e como é feita a construção e preenchimento do mapa.

Na seção 3.4 será apresentada a técnica de SLAM, que através da primeira estimativa da posição do veículo gera-se um mapa e através desse mapa corrige a posição do veículo, aumentando assim a precisão da posição.

## **3.2 TÉCNICA DE LOCALIZAÇÃO**

Utilizando as leituras atuais dos sensores, o veículo deve ser capaz de determinar e manter atualizada a sua posição e orientação em relação a este ambiente, mesmo que os sensores apresentem erro e/ou ruído.

Primeiramente, uma estimativa da posição do veículo é calculada através de um filtro  $\alpha-\beta$ , desenvolvido em [27], utilizando os dados dos sensores. O filtro  $\alpha-\beta$  é utilizado para combinar dados obtidos de diferentes sensores para encontrar a melhor estimativa da posição do veículo baseada na correção de cada medida individual. Uma breve explicação do filtro  $\alpha-\beta$  é apresentada na subseção 3.2.1.

### 3.2.1 FILTRO $\alpha-\beta$

O filtro  $\alpha-\beta$  é uma versão de estado estacionário do filtro de Kalman. Sob condições de processos de ruído estacionário e uma taxa fixa de medição, o filtro  $\alpha-\beta$  é equivalente ao filtro de Kalman correspondente ao estado estacionário. Devido às condições restritivas, o filtro  $\alpha-\beta$  apresenta estimativas de qualidade inferior, no entanto, a simplicidade e eficiência computacional do filtro  $\alpha-\beta$  justificam a sua utilização em diversas aplicações [19].

Um filtro  $\alpha-\beta$  é um filtro de Kalman para um modelo de movimento com velocidade quase constante sob condições de estado estacionário, que são resultado de uma taxa de dados constante e processos de ruído estacionário.

O modelo de estado é dado por

$$X_k = AX_{k-1} \quad (3.1)$$

onde  $X_k$  é o vetor de estado do sistema no instante  $k$  e  $A$  é a matriz que define a dinâmica do sistema. A equação de medição é dada por

$$Z_k = HX_k \quad (3.2)$$

onde  $Z_k$  é a medida no instante  $k$  e  $H$  é uma matriz que define a saída de medição.

Considerando que  $X_{klj}$  denotam as estimativas do estado e da covariância em  $k$  dadas às medições no instante  $j$ , o filtro é descrito com as seguintes equações

Atualização do Tempo (Predição)

$$X_{k|k-1} = AX_{k-1|k-1} \quad (3.3)$$

Atualização da Medição (Correção)

$$X_{k|k} = X_{k|k-1} + K_k (Z_k - H_{k|k-1}) \quad (3.4)$$

O modelo de estado para o filtro  $\alpha-\beta$  é um vetor de estado simples de dois elementos dado por:

$$X_k = [x_k \ \dot{x}_k]^T \quad (3.5)$$

Onde  $x_k$  é posição e  $\dot{x}_k$  é velocidade. O sistema dinâmico para o filtro  $\alpha-\beta$  é modelado com

$$A = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \quad (3.6)$$

Onde  $t$  é o tempo entre as medições. Substituindo os valores abaixo em 3.3 e 3.4

$$H = [1 \ 0] \quad e \quad K_k = \begin{bmatrix} \alpha & \frac{\beta}{t} \end{bmatrix}^T \quad (3.7)$$

Tem-se:

Atualização do Tempo;

$$x_{k|k-1} = x_{k-1|k-1} + dx \quad (3.8)$$

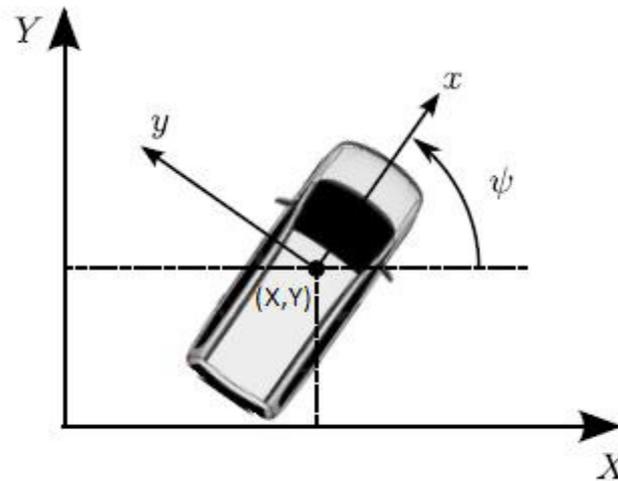
$$\dot{x}_{k|k-1} = \dot{x}_{k-1|k-1} \quad (3.9)$$

Atualização da Medição;

$$x_{k|k} = x_{k|k-1} + \alpha(x_k - x_{k|k-1}) \quad (3.10)$$

$$\dot{x}_{k|k} = \dot{x}_{k|k-1} + \frac{\beta}{t}(x_k - x_{k|k-1}) \quad (3.11)$$

Com base na metodologia explicada o filtro  $\alpha-\beta$  é utilizado para estimar a posição  $(x, y)$  e a orientação  $\psi$  do veículo em coordenadas globais. Obtendo assim, o vetor  $\vec{P}(x, y, \psi)$  do veículo, mostrado na Figura 3.1.



**Figura 3.1 - Representação da Posição do Veículo (56).**

### 3.2.1.1 FILTRO $\alpha-\beta$ APLICADO NA ESTIMATIVA DA POSIÇÃO $(x, y)$

As Equações 3.8 e 3.10 são utilizadas para estimar e corrigir a posição do veículo, respectivamente. A estimativa do valor de  $(x_{k|k-1})$  é dado pela cinemática com base nos dados anteriores  $(x_{k-1|k-1})$  e a correção do valor de  $x$  estimado  $(x_{k|k})$  é calculada considerando o valor de  $x_k$  como o valor obtido pelo DGPS. O valor de  $\alpha$  escolhido foi de 0.15. Esse procedimento é realizado também para  $y$ .

### 3.2.1.2 FILTRO $\alpha-\beta$ APLICADO NA ESTIMATIVA DO ÂNGULO $\psi$

Assim como para a posição, as Equações (3.8) e (3.10) são utilizadas para estimar e corrigir o ângulo do veículo, respectivamente. A estimativa do valor de  $\psi(\psi_{k|k-1})$  é dado pela cinemática com base nos dados anteriores  $(\psi_{k-1|k-1})$  e a correção do valor de  $\psi$  estimado  $(\psi_{k|k})$  é calculada considerando o valor  $\psi_k$  de como o valor calculado utilizando os dados do DGPS, ou seja,

$$\psi = \tan^{-1}\left(\frac{dy}{dx}\right) \quad (3.12)$$

O valor de  $\alpha$  escolhido para este caso foi de 0.10.

### 3.3 MAPEAMENTO

Para que o Veículo Autônomo Inteligente consiga cumprir os seus objetivos de forma autônoma e segura é necessário que o mesmo seja capaz de construir um modelo do mundo ao seu redor e a partir desse modelo tomar suas decisões. Para isto o veículo necessita armazenar as informações coletadas pelos seus sensores, a maneira escolhida foi através dos mapas métricos, que fazem uso de uma estrutura conhecida como grade de ocupação [36].

A grade de ocupação, tratada neste trabalho de forma bidimensional, pode ser modelada como uma matriz  $M^t$  onde cada célula  $m_{x,y}^t$  é uma variável que representa uma região discretizada e bem definida do ambiente e armazena o estado em que ela se encontra em um determinado instante  $t$ .

Cada uma dessas variáveis é caracterizada pelas suas coordenadas no ambiente, e o seu estado indica a probabilidade daquela região estar vazia (pode ser atravessada pelo veículo) ou ocupada (possui algum obstáculo que impede a navegação) dados os estados de localização  $X_{0:t}$  do veículo e o conjunto de observações  $Z_{0:t}$  realizadas desde o início da navegação. Por convenção,  $m_{x,y}^t$  indica o estado ocupado da variável e  $\bar{m}_{x,y}^t$  o estado livre. A representação do ambiente por mapa métrico utilizando grade de ocupação é apresentada em (3.13),

$$M^t = \left\{ \begin{array}{ccccc} m_{0,0} & m_{1,0} & \cdots & m_{x-1,0} & m_{x,0} \\ m_{0,1} & m_{1,1} & \cdots & m_{x-1,1} & m_{x,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{0,y-1} & m_{1,y-1} & \cdots & m_{x-1,y-1} & m_{x,y-1} \\ m_{0,y} & m_{1,y} & \cdots & m_{x-1,y} & m_{x,y} \end{array} \right\} \quad \begin{array}{l} m_{x,y}^t = \{\xi\} \\ \xi_{x,y}^t = p(m_{xy}^t | Z_{0:t}, X_{0:t}) \end{array} \quad (3.13)$$

em que:

- $M^t$ : é uma matriz que representa a grade de ocupação e leva em consideração todas as informações obtidas pelos sensores até o instante  $t$ ;
- $m_{x,y}^t$ : região do ambiente representado pela grade de ocupação com a probabilidade de estar vazia ou não, até o instante  $t$ ;

- $X_{0:t} = \{x_0, x_1, \dots, x_t\} = \{X_{0:t-1}, x_t\}$ : conjunto de dados com todas as posições do veículo até o instante  $t$ ;
- $Z_{0:t} = \{z_0, z_1, \dots, z_t\} = \{Z_{0:t-1}, z_t\}$ : conjunto de dados de todas as observações feitas pelos sensores até o instante  $t$ ;

A atualização da grade de ocupação é feita através do teorema de Bayes [47], um resultado da teoria de probabilidades que permite atualizar iterativamente as distribuições de probabilidades de uma variável aleatória conforme novas informações referentes a ela são obtidas. A formulação do Teorema de Bayes é apresentada em (3.2),

$$p(m_{x,y}^t | Z_t, x_t) = \frac{p(m_{x,y}^t) p(Z_t | m_{x,y}^t, x_t)}{p(m_{x,y}^t) p(Z_t | m_{x,y}^t, x_t) + p(\bar{m}_{x,y}^t) p(Z_t | \bar{m}_{x,y}^t, x_t)} \quad (3.14)$$

em que:

- $Z_t = \{z_0^t, z_1^t, \dots, z_k^t\}$ : indica o conjunto de observações no instante  $t$ ;
- $z_k^t$ : observações realizadas no instante  $t$ ;
- $x_t$ , a posição do robô no instante  $t$ ;

Cada variável do mapa possui uma posição fixa no ambiente, a incerteza inerente às observações fornecidas pelos sensores faz com que seja atualizado um conjunto de variáveis, de acordo com a probabilidade de cada uma delas representar a posição que foi efetivamente observada. Para este trabalho, a probabilidade de acerto das observações realizadas pelo sensor utilizado é considerada alta.

Considerando que o sensor do veículo detecta apenas regiões ocupadas e que os sinais enviados e recebidos viajam no espaço antes de interagir com os objetos detectados, pode-se concluir que os resultados de observação são de regiões ocupadas e todo o espaço intermediário é de regiões livres.

### 3.4 MÉTODO DE SLAM PROBABILÍSTICO

Como a técnica de localização utilizada para estimar a posição de veículo apresenta erros consideráveis para o mapeamento, surgiu então a necessidade de desenvolver uma técnica de SLAM que corrigisse a posição do veículo através do mapeamento existente e ao mesmo tempo atualizar o mapa com os dados dos sensores a partir da posição do veículo corrigida. A técnica utilizada para corrigir o posicionamento do veículo é chamada neste trabalho de método de sobreposição de imagens.

O problema de sobreposição de imagens tem sido amplamente estudado pela comunidade de visão computacional. A maioria dos trabalhos tem-se centrado sobre o problema geral de casar uma imagem de uma posição arbitrária a uma orientação relativa a um determinado modelo. A sobreposição é realizada da seguinte maneira, primeiramente são extraídas características dos dados, em seguida determina a correta correspondência entre as características das imagens e do modelo, geralmente através de uma forma de pesquisa restrita.

A metodologia desenvolvida neste trabalho tem como objetivo realizar a sobreposição de imagens que representam os dados adquiridos a partir de uma leitura do sensor. Ao invés da comparação entre a imagem adquirida e um modelo, a sobreposição é realizada entre duas leituras consecutivas, ou seja, a primeira leitura do sensor é considerada o modelo de imagem e a leitura seguinte deve-se “adequar” a esse modelo. Após uma nova leitura, a leitura anterior é dita o modelo e assim por diante.

Uma observação motivou o desenvolvimento dessa metodologia: o deslocamento da imagem em relação à imagem anterior é pequeno, onde esse deslocamento é o erro gerado pelo método utilizado para estimar a localização do veículo. Esta hipótese é quase sempre verdade para aplicações práticas, particularmente se a posição do veículo for atualizada com frequência. A observação dessas características deu origem a um método que variando a posição do veículo é possível a sobreposição da imagem da leitura atual do sensor com o modelo, o que reduz significativamente o deslocamento.

Se o deslocamento entre a imagem e o modelo é pequeno, então para cada ponto na imagem seu ponto correspondente no modelo é provável que seja o ponto mais próximo em todo o modelo. A determinação da correspondência parcial entre os pontos da imagem e os do

modelo se reduz a uma simples pesquisa para encontrar o ponto do modelo mais próximo ao ponto da leitura atual.

O método desenvolvido se caracteriza por variar a posição do veículo  $(x, y, \psi)$  até encontrar uma posição que minimize o somatório da distância entre os pontos da imagem e do modelo levando em consideração a probabilidade de cada ponto, ou seja, até que a imagem da leitura atual do sensor se sobreponha ao modelo. A equação (3.15) descreve a função objetivo que se deseja minimizar.

$$F_{ob} = \sum_{i=1}^k d_i [2 - p_i], \quad i = \{1, 2, \dots, k\} \quad (3.15)$$

em que:

- $d_i$ : é a distância entre o ponto  $z_i^t$  do conjunto  $Z_{0:k}^t$  da imagem atual dos sensores ao ponto  $z_j^{t-1}$  do modelo;
- $z_j^{t-1}$ : é um ponto do conjunto  $Z_{0:k}^{t-1}$  que representa o ponto do modelo mais próximo ao ponto  $z_i^t$ ;
- $p_i$ : é a média das probabilidades do ponto  $z_i^t$  e do ponto  $z_j^{t-1}$  dadas pela matriz  $M^{t-1}$ ;

Analisando a equação 3.15 pode-se notar que a probabilidade dos obstáculos influencia no cálculo da função objetivo, ou seja, é levada em conta a probabilidade do ponto atual em questão e a do modelo. Quanto menor a probabilidade do modelo e maior a da imagem, mais próximo do local atual o ponto da leitura atual ficará e quanto maior a probabilidade do modelo e menor a probabilidade da imagem o ponto atual se aproximará mais do modelo. Isto leva em consideração não só a ideia de que as imagens devem estar sobrepostas se a probabilidade do modelo for de 100%, como também que elas devem permanecer deslocadas se a probabilidade do modelo for diferente de 100%, para que dessa forma seja levado em conta todo o mapeamento já realizado.

A cada leitura do sensor, o método de SLAM probabilístico é realizado corrigindo a posição do veículo. Com a posição do veículo corrigida, a matriz de probabilidade ( $M^t$ ) é atualizada.

## **CAPÍTULO IV**

### **4 ALGORITMO**

#### **4.1 INTRODUÇÃO**

Uma das dificuldades encontradas no desenvolvimento de sistemas autônomos é o problema da navegação. Navegar entre duas localidades consiste numa tarefa essencial para realização da maioria das missões.

O problema da navegação encontra-se diretamente relacionado com diversas outras tarefas, que devem ser previamente resolvidas para que seja possível navegar, tais como: planejamento de trajetória, localização do veículo e construção de um mapa do ambiente. A primeira tarefa pode ser resolvida sem dificuldade excessiva desde que o mapa e a localização sejam conhecidos precisamente. Portanto, os problemas reais são:

- Determinar a localização do veículo;
- Construir um mapa de seu ambiente.

Neste capítulo será descrito o algoritmo desenvolvido neste trabalho, aplicando a técnica de SLAM para resolver esses problemas. Na seção 4.2 serão descritos os sensores utilizados nesse trabalho para realizar a aquisição de dados. Na seção 4.3 será mostrado como os dados coletados são processados para construir um mapa e posicionar o veículo e os obstáculos neste mapa. E por fim, na seção 4.4 a ferramenta computacional desenvolvida para visualizar a metodologia de SLAM é apresentada.

#### **4.2 AQUISIÇÃO DE DADOS**

O veículo autônomo utiliza sensores para monitorar o ambiente e a si mesmo. O uso adequado dos sensores, bem como a correta interpretação dos dados fornecidos por estes, permitirá ao sistema perceber o ambiente, podendo assim, reagir a ele. Para localizar o veículo

no ambiente e construir um mapa de obstáculos, três tipos de sensores foram utilizados: um DGPS, dois *encoders* e um *scanner laser*.

### 4.2.1 DGPS

O posicionamento do veículo é determinado pelo sistema de posicionamento global - GPS (*Global Positioning System*). Uma variação do GPS é o DGPS (*Differential Global Positioning System*) que é utilizado neste trabalho. GPS Diferencial é uma técnica para reduzir o erro na posição GPS usando dados adicionais a partir de um receptor GPS de referência em uma posição conhecida. A forma mais comum de DGPS envolve determinar os erros em uma estação de referência e transmitir as correções da distância receptor-satélite (*pseudorange*) em tempo real para o receptor, que aplica as correções no processo para determinar a sua posição. A precisão de um receptor DGPS é de cerca de 1 metro.

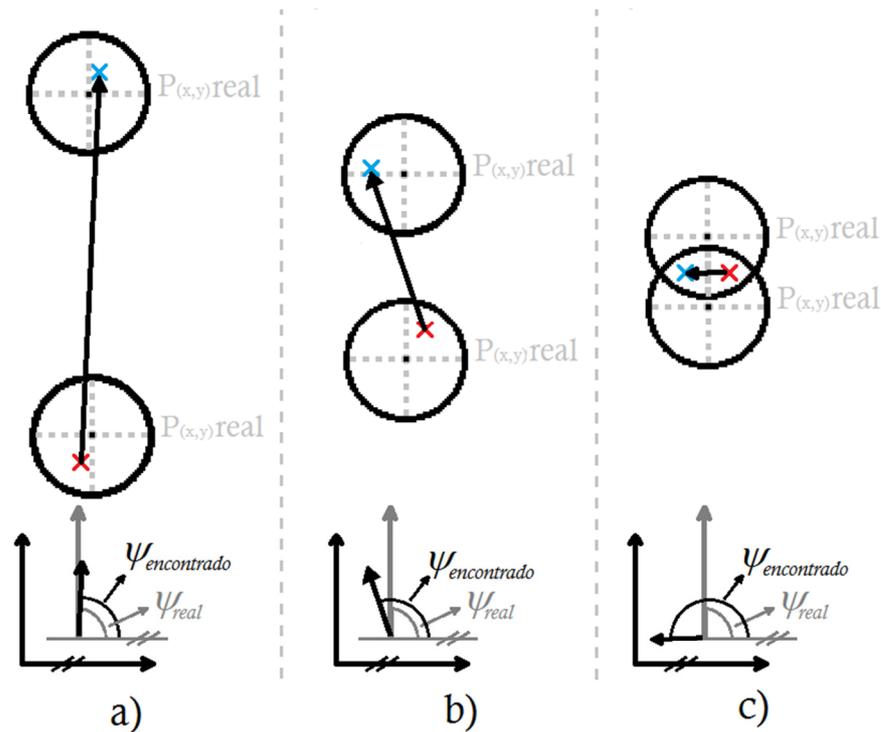
A movimentação do veículo autônomo é restrita a um plano horizontal representado por coordenadas cartesianas. O posicionamento do veículo em relação ao ambiente é representado por um vetor  $\vec{P}$  que contém os valores de sua posição  $X$ ,  $Y$  e o ângulo  $\psi$  formado entre o eixo longitudinal do veículo e o eixo de coordenadas  $X$ , como mostra a Figura 3.1.

$$\vec{P} = (X, Y, \psi) \quad (4.1)$$

O DGPS, através de um pré-processamento dos dados, fornece a posição  $X$  e  $Y$  do veículo, em coordenadas cartesianas, numa frequência de  $10Hz$ , a partir de um ponto de referência escolhido no ambiente. A orientação do veículo é calculada utilizando-se a posição atual do veículo  $(X_i, Y_i)$  e a posição anterior  $(X_{i-1}, Y_{i-1})$  conforme a equação (4.2).

$$\psi_i = \tan^{-1} \left( \frac{Y_i - Y_{i-1}}{X_i - X_{i-1}} \right) \quad (4.2)$$

Por causa do erro de posicionamento do DGPS a orientação do veículo também apresenta erros, como pode ser observado na Figura 4.1, onde  $P_{(x,y)real}$  representa a posição real do veículo e a circunferência que envolve este ponto representa o erro DGPS, ou seja, a posição dada pelo DGPS pode ser qualquer ponto dentro da circunferência. Os pontos em azul (x) e vermelho (x) são a posição atual e a posição anterior dadas pelo DGPS, respectivamente.



**Figura 4.1 - Erro do DGPS.**

O caso *a)* representa uma situação em que o veículo encontra-se em alta velocidade. Nesta condição observa-se que o ângulo encontrado é próximo ao real. No caso *b)* o veículo encontra-se com uma velocidade inferior em relação ao caso anterior e observa-se que o ângulo encontrado possui um erro considerável. No caso *c)* o veículo possui uma velocidade muito baixa, ocasionando uma sobreposição entre as regiões que delimitam a área de erro do DGPS. Nesta situação o ângulo encontrado possui um erro de aproximadamente  $90^\circ$  o que poderia comprometer todo o trabalho.

Analisando a Figura 4.1 pode-se concluir que quanto maior a velocidade do veículo, menor será seu erro de orientação, porém com uma velocidade alta perdem-se informações na construção das matrizes que são o foco principal deste trabalho.

#### **4.2.2 ENCODERS**

Os *encoders* são transdutores de movimento capazes de converter movimentos lineares ou angulares em informações elétricas (pulsos). Armazenando precisamente os movimentos das rodas utilizando *encoders*, o veículo pode determinar o quanto ele se locomoveu e o quanto ele rotacionou. O número de pulsos por rotação da roda determina a resolução do

*encoder*. Conhecendo o raio da roda é possível calcular o movimento do veículo baseando-se nas informações obtidas do *encoder*.

O veículo autônomo em questão utiliza dois *encoders*: um *encoder* de movimento linear e um *encoder* de movimento angular.

- *Encoder* de movimento linear:

O sensor *ABS* (*Anti-lock Braking System*) do veículo foi adaptado como *encoder*. Ele está localizado nas rodas traseiras do veículo é utilizado para determinar seu espaço percorrido e apresenta uma resolução de 29 pulsos por rotação da roda.

- *Encoder* de movimento angular:

Localizado no volante do veículo, este *encoder* é utilizado para determinar sua rotação que pode variar de  $-35^\circ$  a  $35^\circ$  e apresenta uma resolução de 7142 pulsos por grau rotacionado.

A localização de robôs móveis utilizando *encoders* é conhecida como odometria, previamente descrita na subseção 2.3.1.1.

### **4.2.3 SCANNER LAZER**

O *scanner laser* determina a distância dos obstáculos encontrados em uma região de busca. Um espelho giratório desvia o raio laser com uma resolução angular pré-determinada formando assim um leque de varredura de  $180^\circ$ , como mostra a Figura 4.2. Quando o raio é refletido de um objeto a distância do sensor a esse objeto é calculada. A cada leitura do *scanner laser* é varrida uma área de uma semicircunferência cujo raio pode ser ajustado para 8,16,32 ou 80m.



**Figura 4.2 - Leque de varredura do *scanner laser*.**

O *scanner laser* foi configurado para um alcance de 16m e uma resolução angular de 0,5° e apresenta uma frequência de varredura de 10Hz. Com essas configurações, a cada leitura o sensor retorna um vetor de 360 posições contendo a distância dos objetos a cada variação de 0,5 graus do raio laser, ou seja, a posição dos objetos é dada em coordenadas polares.

O *scanner laser* apresenta uma alta precisão, porém para veículos pretos ou pratas o raio laser não é completamente refletido prejudicando a leitura do sensor.

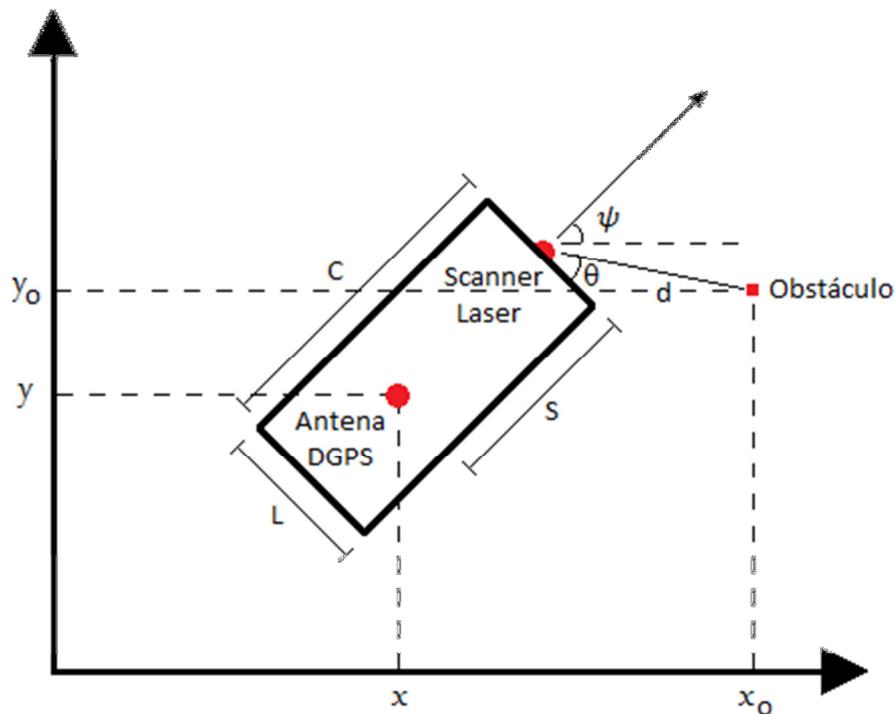
### **4.3 PROCESSAMENTO DOS DADOS**

O algoritmo desenvolvido funciona como um simulador, a cada leitura dos sensores, no caso, dados já armazenados e lidos na mesma frequência de funcionamento dos sensores, o mapeamento parcial é realizado. Para cada nova leitura é feita a correção da posição e orientação do veículo através do método de sobreposição das leituras do scanner e somente após a correção do posicionamento é que o mapa de obstáculos é atualizado.

#### **4.3.1 MAPEAMENTO**

Com as informações coletadas dos sensores é possível construir um mapa de obstáculos. Como a distância de cada obstáculo detectado pelo *scanner laser* refere-se à posição do sensor torna-se necessário posicionar cada obstáculo no ambiente.

A leitura dos dados de *scanner laser* é realizada simultaneamente com a leitura dos dados do DGPS e *encoder*, onde é feito um pré-processamento utilizando um filtro  $\alpha-\beta$  para estimar a localização do veículo como descrito na seção 3.2, para que dessa forma, a posição dos obstáculos possa ser convertida para coordenadas do veículo no ambiente. De acordo com a Figura 4.3, a posição de cada obstáculo no ambiente é calculada através das Equações (4.3) e (4.4).



**Figura 4.3 - Posição Obstáculo no Ambiente.**

$$x_o = x + (d \cos \theta) \sin \psi + (d \sin \theta + S) \cos \psi \quad (4.3)$$

$$y_o = y - (d \cos \theta) \cos \psi + (d \sin \theta + S) \sin \psi \quad (4.4)$$

em que:

- $C$  : é o comprimento do veículo e é a igual 4,33 metros;
- $L$  : é a largura do veículo e é a igual 1,74 metros;
- $S$  : é a distância entre o *scanner laser* e o DGPS e é igual a 2,67 metros;
- $d$  : é à distância do obstáculo ao veículo, fornecido pelo *scanner laser*;
- $\theta$  : é o ângulo do obstáculo com a referência do *scanner laser*;

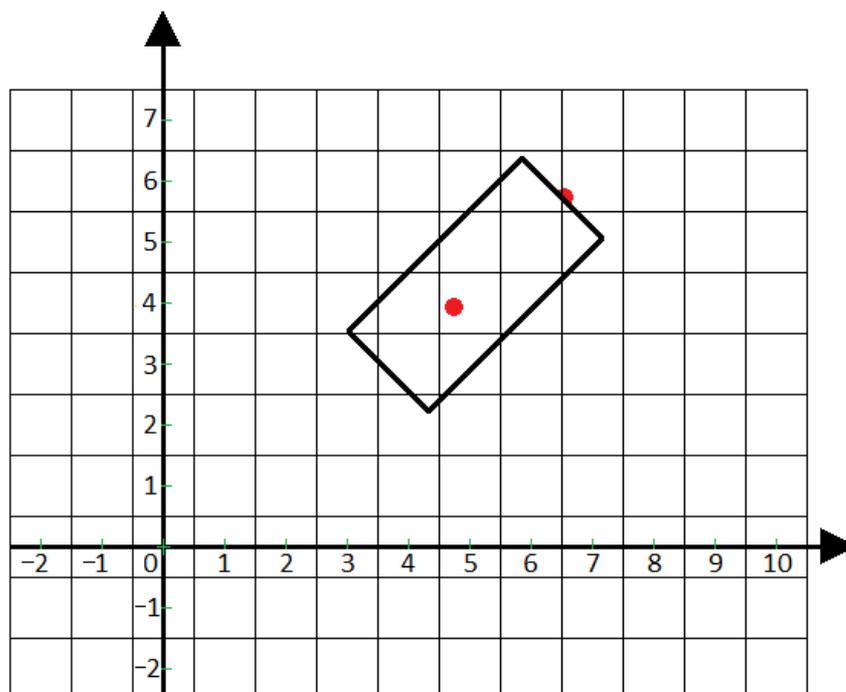
- $\psi$ : é a direção do veículo;
- $(x_o, y_o)$ : é a posição do obstáculo na referência global;
- $(x, y)$ : é a posição do veículo na referência global.

Todos os obstáculos identificados pelo sensor, a cada varredura, passam por esta mudança de coordenadas para que seja possível a construção de um mapa.

#### 4.3.1.1 MATRIZ DE OBSTÁCULOS (MO)

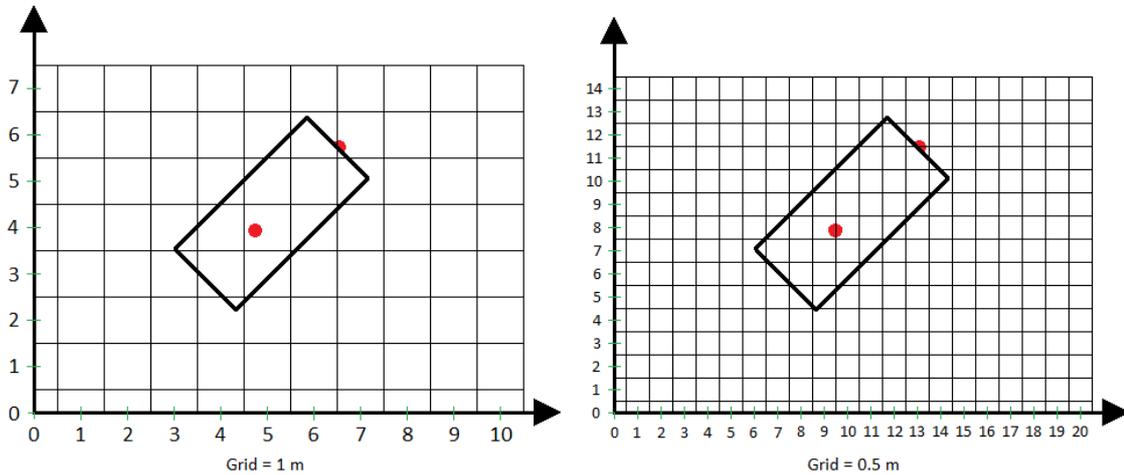
A Matriz de Obstáculos (MO) representa o ambiente em que o veículo está inserido. Cada elemento da matriz corresponde a uma área do ambiente em que pode existir ou não um obstáculo.

A estrutura dessa matriz apresenta características diferentes de uma matriz convencional, ou seja, a contagem do número de linhas e colunas inicia do zero e os elementos podem ser positivos ou negativos para melhor representar o espaço cartesiano. O elemento  $mo_{(0,0)}$  da matriz localiza-se na origem do plano cartesiano, como mostra a Figura 4.4.



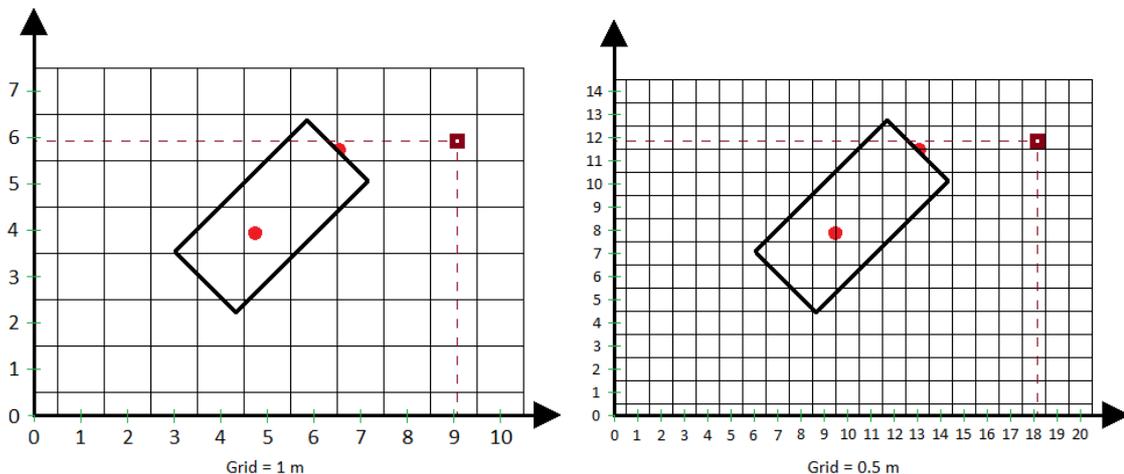
**Figura 4.4 - Matriz de Obstáculos.**

A área do ambiente que cada elemento da matriz representa é quadrada e sua dimensão pode variar de tamanho. A variável que determina a dimensão (em metros  $m$ ) dessa área é denominada de *Grid*. A Figura 4.5 mostra a Matriz de Obstáculos para dois valores diferentes do *Grid*.



**Figura 4.5 - Matriz de Obstáculos - *Grid*.**

Para cada obstáculo, calcula-se então, a qual elemento da matriz ele pertence. Por exemplo: para um obstáculo localizado na posição (9.1m, 5.9m) o elemento da matriz a qual ele pertence será  $mo_{(9,6)}$  para um  $Grid = 1m$  e  $mo_{(18,12)}$  para um  $Grid = 0,5m$ , como mostra a Figura 4.6.



**Figura 4.6 - Matriz de Obstáculos – Obstáculo.**

Após calcular a qual elemento da matriz o obstáculo pertence, o elemento da matriz que contém esse obstáculo receberá o valor 1. Se outro obstáculo pertencer a um mesmo elemento ou se em outra varredura do *scanner laser* o mesmo obstáculo for localizado, esse

elemento receberá um acréscimo de 1, assim essa matriz contém o histórico de todos os obstáculos encontrados pelo *scanner laser*.

#### 4.3.1.2 MATRIZ DE PROBABILIDADE (MP)

A Matriz de Probabilidade (MP) também apresenta a mesma dimensão de *MO* e representa a probabilidade de existir um obstáculo na área do ambiente que cada elemento da matriz representa, ou seja, cada elemento de MP contém a probabilidade de existir um obstáculo ou não naquela região. A probabilidade é dada pelo Teorema de Bayes explicado na seção 3.3.

No início da navegação atribui-se para cada uma das regiões o valor 0, considerando inicialmente que não há obstáculos mapeados. Conforme o robô se movimenta ele coleta informações a partir dos seus sensores e as utiliza para atualizar os estados das variáveis referentes às regiões observadas de maneira a fazê-los convergir para aqueles apresentados na realidade.

#### 4.3.2 CORREÇÃO POR SOBREPOSIÇÃO

Através da análise dos resultados da técnica de localização notou-se que o erro de posicionamento mais significativo ocorre na orientação do veículo. A partir dessas observações, foi possível desenvolver um algoritmo de sobreposição das leituras do sensor priorizando primeiramente a correção da orientação do veículo. Se somente a correção da orientação for suficiente para diminuir o erro do posicionamento a níveis aceitáveis estipulados no algoritmo, não se torna necessário corrigir a posição  $x$  e  $y$  do veículo, diminuindo assim, o esforço computacional de busca. Mas se a correção da orientação não for suficiente é realizada uma nova busca corrigindo também a posição do veículo.

Os passos do algoritmo de sobreposição são descritos a seguir:

Correção de  $\psi$  :

- 1º Passo - A orientação do veículo é variada em  $\pm d\psi$  ;
- 2º Passo - Calcula-se qual ponto do modelo corresponde à menor distância a cada ponto da leitura do sensor;

- 3º Passo - Com os valores das distâncias entre os pontos calculados e o valor da probabilidade de cada ponto, calcula-se o erro (Equação (3.4)) da sobreposição;
- 4º Passo - O posicionamento do veículo é alterado pela orientação  $d\psi$  que apresentou o menor erro. O erro então é avaliado para saber se está dentro dos níveis aceitáveis, se não, o algoritmo volta para o 1º passo, caso contrário o algoritmo é finalizado.

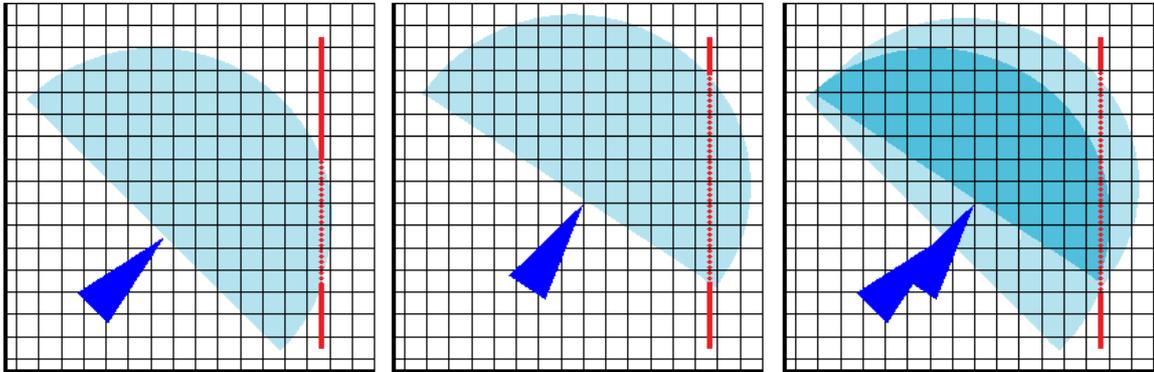
Os quatro passos são repetidos até que o erro estabilize. Se o erro encontrado, for menor que um valor mínimo estipulado, somente com a variação de  $d\psi$ , significa que a posição do veículo está correta, caso contrário torna-se necessária a correção da posição  $x$  e  $y$  do veículo. A correção da posição  $x$  e  $y$  segue o mesmo princípio do algoritmo descrito anteriormente porem com um critério de parada diferente.

Correção de  $x$  e  $y$  :

- 1º Passo - A orientação do veículo é variada em  $\pm dx$ ;
- 2º Passo - Calcula-se qual ponto do modelo corresponde à menor distância a cada ponto da leitura do sensor;
- 3º Passo - Com os valores das distâncias entre os pontos calculados e o valor da probabilidade de cada ponto, calcula-se o erro (Equação (3.4)) da sobreposição;
- 4º Passo - O posicionamento do veículo é alterado para a posição  $dx$  que apresentou o menor erro.
- Os quatro passos são repetidos para  $dy$ .
- O algoritmo é finalizado quando não houver mais variação do erro em ambas as posições  $(x, y)$  ou quando o erro estiver dentro dos níveis aceitáveis, se não, o algoritmo volta para o 1º passo.

A Figura 4.7 exemplifica o método de sobreposição, onde a primeira imagem representa o modelo, a segunda imagem representa a leitura atual e a última imagem mostra a sobreposição das duas imagens. A semicircunferência em azul mostra a área de varredura do

sensor, onde o tom de azul mais escuro representa a área em comum das duas leituras realizadas.



**Figura 4.7 - Representação da Sobreposição.**

Com o posicionamento do veículo corrigido, a posição dos obstáculos encontrados na leitura atual do sensor também é corrigida e a matriz de probabilidade é atualizada.

## 4.4 FERRAMENTA COMPUTACIONAL

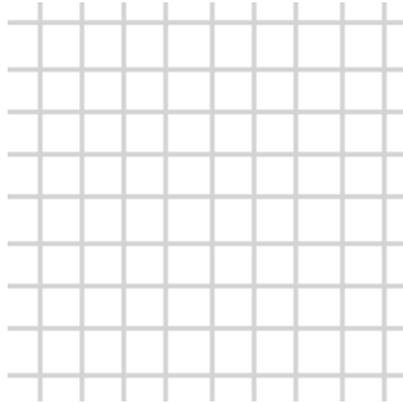
A ferramenta computacional desenvolvida tem como objetivo processar os dados fornecidos pelos sensores para gerar as matrizes descritas na seção anterior, corrigir a posição e orientação do veículo utilizando a metodologia proposta e apresentar os resultados através de um mapa de obstáculos.

### 4.4.1 MAPA DE OBSTÁCULOS

A seguir será apresentado cada elemento que compõe o mapa de obstáculos.

- O mapa:

O mapa é apresentado como uma região quadriculada que representa a matriz de obstáculos ou de probabilidade. Os quadrados podem estar preenchidos, caso exista um obstáculo naquela região, ou não. A Figura 4.8 mostra como o mapa é representado.



**Figura 4.8 - Representação da Matriz de Obstáculo em forma de mapa**

- O veículo:

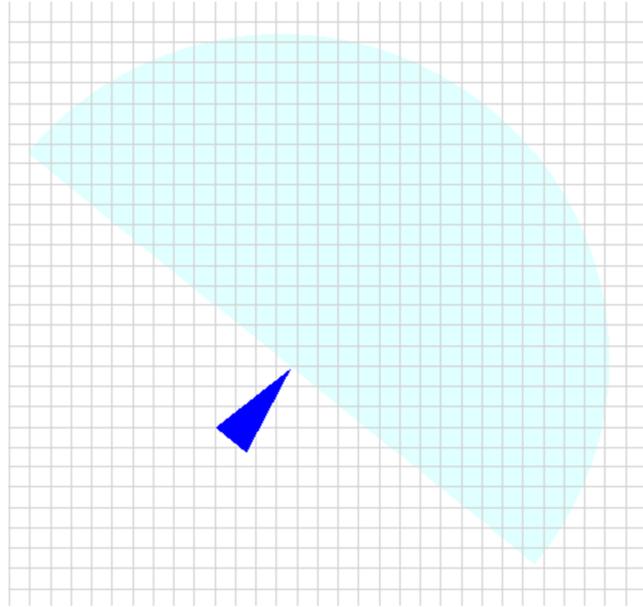
O veículo é representado por um triângulo isósceles, proporcional ao tamanho real do veículo, cuja base do triângulo representa a traseira do veículo e o vértice oposto a base representa a frente. A inclinação do triângulo no mapa de obstáculos indica a direção do veículo. A Figura 4.9 mostra como o veículo é representado no mapa.



**Figura 4.9 - Representação do Veículo;**

- Área varrida pelo *scanner laser*:

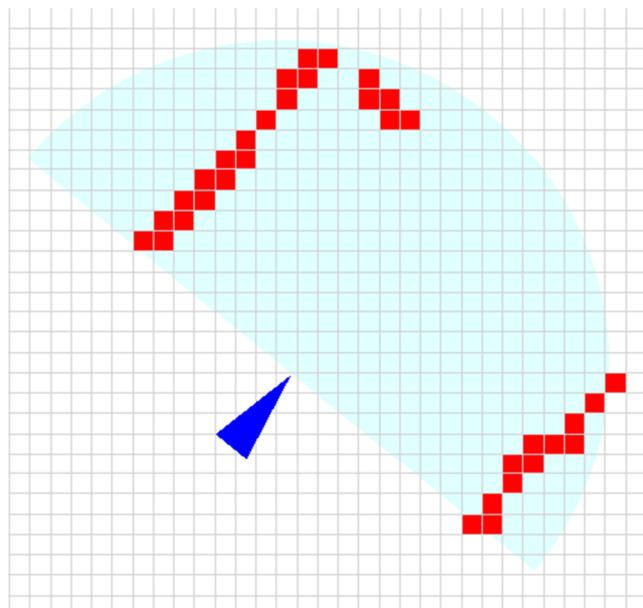
A área varrida pelo *scanner laser* é representada por uma semicircunferência na cor azul claro e o alcance do sensor foi ajustado para 16m, como mostra a Figura 4.10.



**Figura 4.10 - Área de Varredura do *Scanner Laser*.**

- Obstáculo:

O obstáculo é representado por uma área do mapa, que contém o obstáculo, colorida de vermelho, como mostra a Figura 4.11.

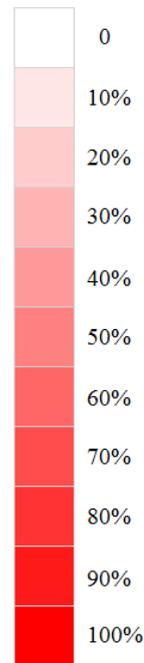


**Figura 4.11 - Área de Varredura do *Scanner Laser* com Obstáculo.**

- Confiabilidade do obstáculo:

A confiabilidade do obstáculo é representada através da variação da tonalidade da cor com que o obstáculo é representado, ou seja, quanto mais forte o tom de vermelho, maior a

chance de existir um obstáculo naquela região e quanto mais fraco o tom, menor a chance. A Figura 4.12 mostra a variação da cor e sua respectiva porcentagem.



**Figura 4.12 - Variação da Tonalidade Conforme a Probabilidade.**

#### **4.4.2 ABAS QUE COMPÕEM A FERRAMENTA**

No programa desenvolvido, informações sobre o mapa de obstáculos podem ser visualizadas em abas distintas. A primeira aba apresenta a Matriz de Probabilidade onde os obstáculos são representados de acordo com sua confiabilidade, como descrito anteriormente. A segunda aba apresenta a Matriz de Obstáculos onde apenas os obstáculos identificados são mostrados sem nenhum cálculo da confiabilidade. E por fim, a quarta aba apresenta o percurso realizado pelo veículo autônomo indicando através de pontos verdes a posição do veículo (Filtro  $\alpha - \beta$ ) e por pontos azuis sua correção utilizando a metodologia proposta.

As Figura 4.13, Figura 4.14. e Figura 4.15, mostram cada aba que compõem a ferramenta.

As Figuras apresentadas mostram a visualização do final de um mapeamento realizado, porém a ferramenta funciona como um simulador, mostrando passo a passo cada movimento do veículo, sua posição e orientação no mapa, os obstáculos identificados a cada leitura do scanner e a construção do mapa a cada movimento.



**Figura 4.13 - Representação da Matriz de Probabilidade.**



**Figura 4.14 - Representação da Matriz de Obstáculos.**



## **CAPÍTULO V**

### **5 RESULTADOS**

#### **5.1 INTRODUÇÃO**

O local onde os testes são conduzidos pode interferir nos resultados. Assim a busca por um lugar vazio não é o mesmo que escolher um lugar onde existam carros, colunas e pessoas transitando por perto. Portanto, os testes foram realizados em ambientes abertos e dinâmicos para se obter vários cenários com diferentes propriedades e tamanhos. Dois ambientes foram selecionados para apresentar os resultados, estes ambientes são estacionamentos da Faculdade de Engenharia.

Estes ambientes são apresentados nas seções 5.2 e 5.3, onde cada ambiente será descrito com suas características como: tamanho, irregularidade do piso, ocupação do estacionamento por pessoas e veículos, construções e objetos que contornam o estacionamento.

Os resultados são apresentados em pequenos circuitos. Cada circuito corresponde a um mapeamento e serão abordados na seção 5.4. O objetivo dessa divisão é obter vários mapeamentos, referentes ao mesmo local, variando somente a posição dos obstáculos dinâmicos.

#### **5.2 ESTACIONAMENTO 1**

O primeiro ambiente é um estacionamento localizado na Faculdade de Engenharia, que será chamado Estacionamento 1. Este estacionamento possui um tamanho de aproximadamente  $100 \times 30m$  e encontra-se ao lado do Restaurante Universitário (RU). Como é mostrado na Figura 5.1.



**Figura 5.1 - Estacionamento 1.**

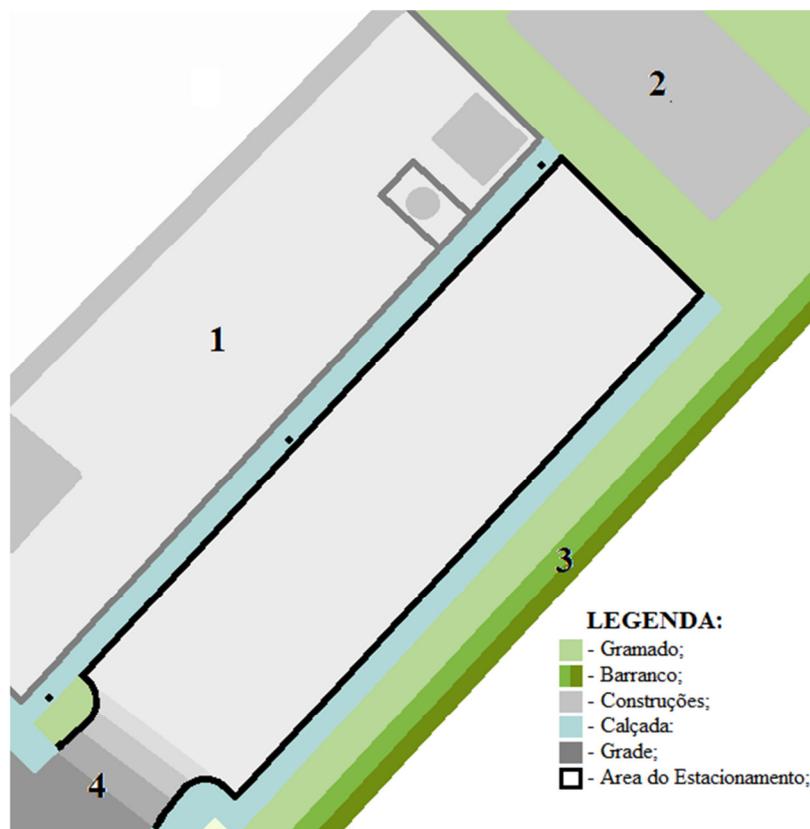
A Figura 5.2 apresenta as vistas laterais do Estacionamento 1, ou seja, as construções e objetos que contornam este estacionamento. Observa-se que no quadro superior esquerdo (SE) está localizado o RU, entre o RU e o estacionamento existe uma grade de proteção. No quadro superior direito (SD) está o prédio da Faculdade de Engenharia. No quadro inferior esquerdo (IE) está um barranco irregular com grama e este possui uma árvore próxima a rampa de entrada e saída do estacionamento. No quadro inferior direito (ID) está a entrada e saída do estacionamento que possui uma inclinação elevada.



**Figura 5.2 - Estacionamento 1 Vista Laterais.**

A Figura 5.3 mostra um desenho da vista superior do Estacionamento 1. Os números 1, 2, 3, e 4 representam as localizações de cada quadros (SE, SD, IE e ID) visto na Figura 5.2 respectivamente. O piso do Estacionamento 1 possui um leve declive que vai da lateral para a parte central. Isso ocorre para que haja um escoamento da água da chuva para os bueiros que se encontram na parte central do estacionamento.

Durante o processo de aquisição de dados, várias situações foram observadas e criadas. Para isto, foi utilizado um veículo Siena da cor cinza como obstáculo estático. Este veículo é posicionado em vários lugares do estacionamento para obter mapeamentos de diversas situações.



**Figura 5.3 - Estacionamento 1 Vista Superior**

Cones e cavaletes também foram colocados no estacionamento para simular obstáculos estáticos de pequeno porte. Estes cones e cavaletes são da cor laranja e possuem ótima visualização pelo *scanner laser*, assim como o veículo utilizado. O objetivo de construir essas situações é analisar o desempenho do sistema em condições reais. Condições estas que podem mudar com o decorrer do tempo em ambientes não controlados.

### 5.3 ESTACIONAMENTO 2

O segundo ambiente é um estacionamento de porte maior, com dimensões em torno de  $100 \times 100m$ . Este ambiente também se localiza na Faculdade de Engenharia ao lado do Galpão da Faculdade de Engenharia Civil. Este estacionamento será chamado de Estacionamento 2 e é mostrado na Figura 5.4



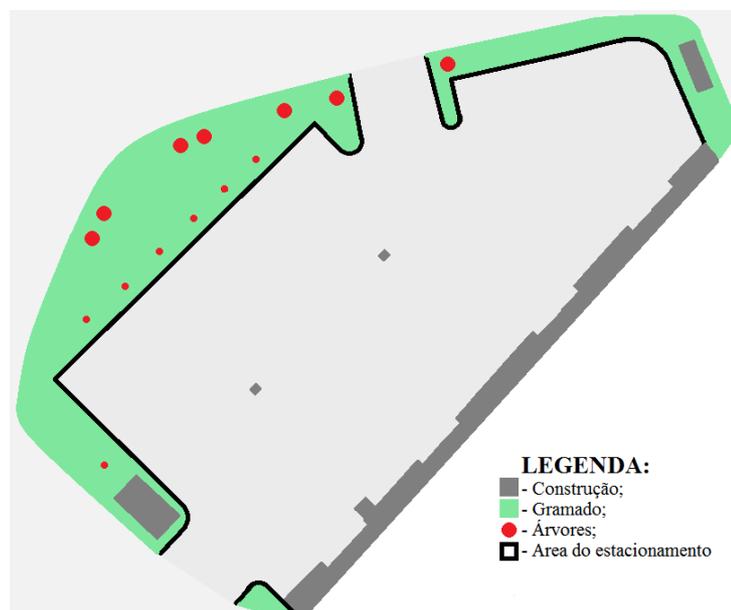
**Figura 5.4 - Estacionamento 2.**

A Figura 5.5 apresenta as vistas laterais do Estacionamento 2, ou seja a construções e objetos que contornam este estacionamento. O número 1 mostra o prédio da Faculdade de Engenharia. O número 2 mostra a saída do estacionamento, uma pequena construção e dois postes. Sendo um poste quase no centro do estacionamento e outro na lateral ao lado da saída. O número 3 mostra as árvores que rodeiam o estacionamento, sendo seis de grande porte e oito de pequeno porte. O número 4 mostra a entrada do prédio da faculdade, três vasos de árvore e uma estrutura de suporte para o poste de iluminação.

A Figura 5.6 mostra uma vista superior do Estacionamento 2 e todos os objetos descritos no parágrafo anterior. O piso desse estacionamento é todo regular e não possui nenhuma inclinação como o Estacionamento 1.



**Figura 5.5 - Estacionamento 2 Vista Laterais**



**Figura 5.6 - Estacionamento 2 Vista Superior**

Durante o processo de aquisição de dados, foram verificadas a existências de vários obstáculos, tantos estáticos quanto dinâmicos. O objetivo desse mapeamento é mapear um ambiente não controlado. Tendo como obstáculos dinâmicos pessoas e veículos que atravessaram o local e outros veículos que estavam estacionados e saíram durante o processo de aquisição de dados. Como obstáculos fixos os itens descritos na legenda da Figura 5.6.

## 5.4 RESULTADOS

Os resultados serão apresentados através da simulação de quatro circuitos, onde cada circuito possui uma particularidade que será explicitada nas subseções seguintes. O Mapa de Obstáculos construído em cada circuito simulado tem como intuito validar o algoritmo proposto.

Para um estudo detalhado de cada circuito, são mostrados instantes de tempo da construção do Mapa de Obstáculos (mapa parcial) para que diferentes situações sejam analisadas. Os seguintes resultados serão apresentados:

**Percurso** - mostra a trajetória que o veículo irá percorrer. Essa trajetória é formada por *Waypoints* representados em verde. Os pontos em azul representam os *Waypoints* já percorridos pelo veículo e estes *Waypoints* podem ou não coincidir com os *Waypoints* em verde porque eles sofreram correção pelo método proposto no capítulo 3. O veículo autônomo é representado pela seta (triângulo) em vermelho. A seta indica a direção e o sentido que o veículo está se movimentando. Os pontos em amarelo são os obstáculos encontrados pelo *scanner laser* na leitura atual.

**Ângulo do Veículo** - compara a variação do ângulo do veículo após percorrer todo o percurso sem utilizar o método de correção da posição e utilizando o método.

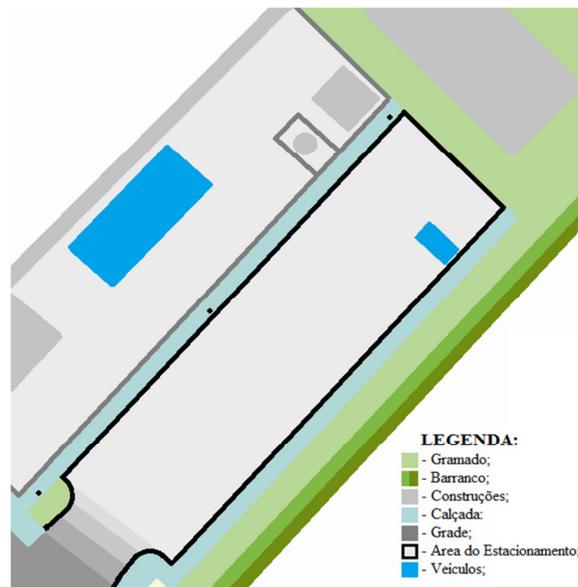
**Mapa de Obstáculos** - mostra os obstáculos encontrados na região percorrida pelo veículo e a confiança de cada obstáculo. O mapeamento pode ser parcial, quando o veículo ainda não completou o percurso ou total.

### 5.4.1 CIRCUITO 1

O Circuito 1 foi mapeado no Estacionamento 1 e sua configuração é apresentada na Figura 5.7. No momento do teste havia um veículo estacionado na parte direita do circuito e outro estacionado dentro da área do RU. Estes veículos estão representados na cor azul. Nenhuma pessoa ou veículo transitou no estacionamento durante a aquisição de dados. Desta forma, não existe nenhum elemento dinâmico nesta amostragem.

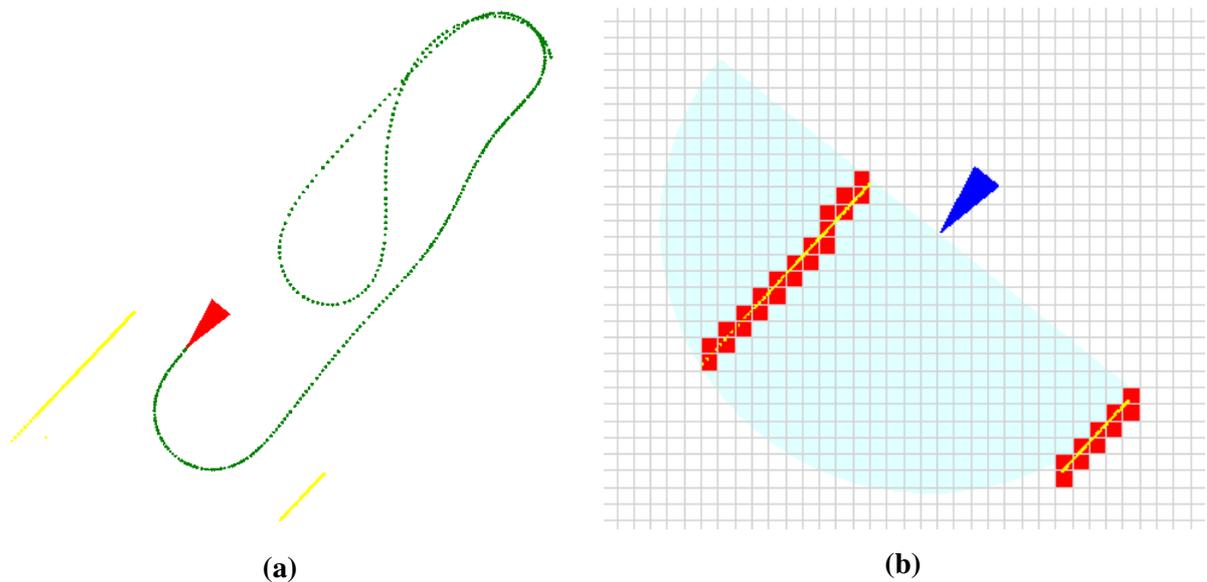
O Circuito 1 possui um comprimento total de 137 metros. O veículo inicia o percurso fazendo uma volta no sentido anti-horário e depois inverte o sentido parando de frente para o

carro estacionado. O percurso possui um total de 364 *Waypoints* e a velocidade média desenvolvida durante o percurso foi de aproximadamente 20Km/h.



**Figura 5.7 - Circuito 1 Configuração**

A Figura 5.8 mostra o início da simulação, onde o veículo identifica a sua esquerda o barranco irregular com uma inclinação de  $60^\circ$  aproximadamente e a direita a grade do RU. A Figura 5.8 (a) mostra a posição inicial do veículo no percurso total que será percorrido.

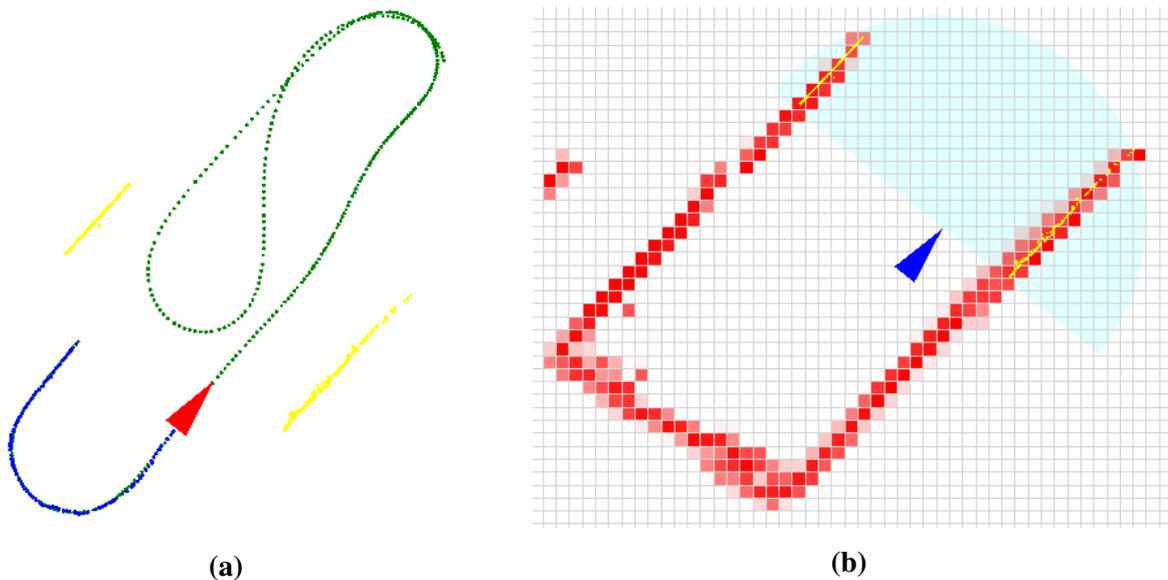


**Figura 5.8 - (a) Percurso; (b) Mapa de Obstáculo;**

A Figura 5.8 (b) mostra o mapa de obstáculos parcial. Os obstáculos preenchidos na matriz apresentam uma tonalidade forte por ser a primeira leitura do *scanner laser*, ou seja, todos os objetos identificados terão uma probabilidade igual a 100%. Conforme o veículo se

movimenta o mapa de obstáculos aumenta e a tonalidade dos obstáculos varia conforme sua confiança. A Figura 5.9 (a) mostra que o veículo fez um retorno de 180° e segue em linha reta, paralelo ao barranco que agora está a sua direita.

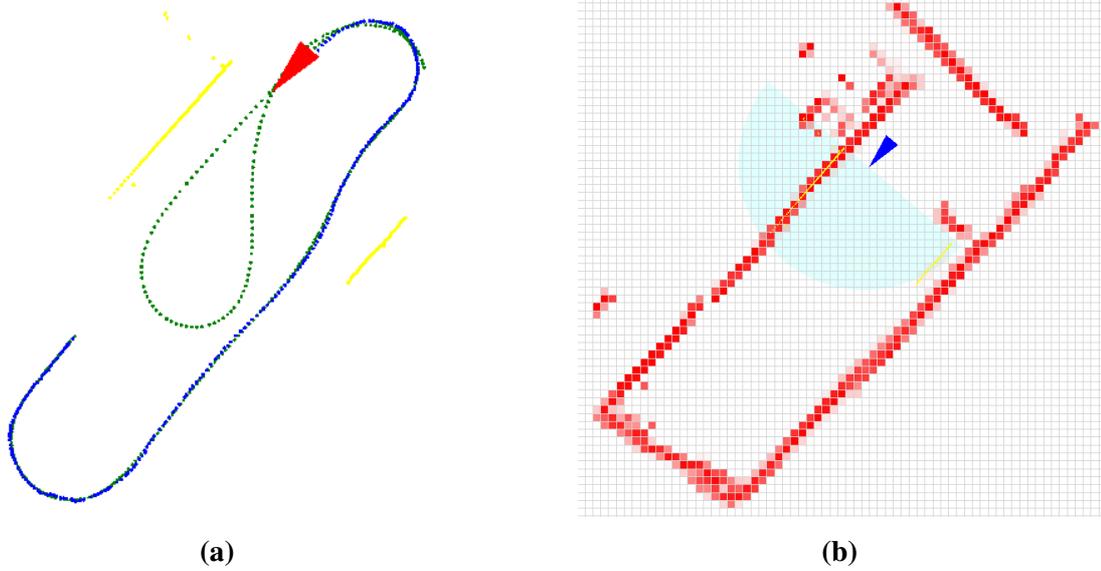
Após a posição do veículo ser corrigida, os obstáculos identificados na atual leitura do *scanner laser* (representados pelos pontos amarelos), mostrados na Figura 5.9 (b), se encontram sobre a linha de maior probabilidade de se encontrar um obstáculo, provando assim, a eficiência do algoritmo de correção da posição do veículo.



**Figura 5.9 - (a) Percurso; (b) Mapa de Obstáculo;**

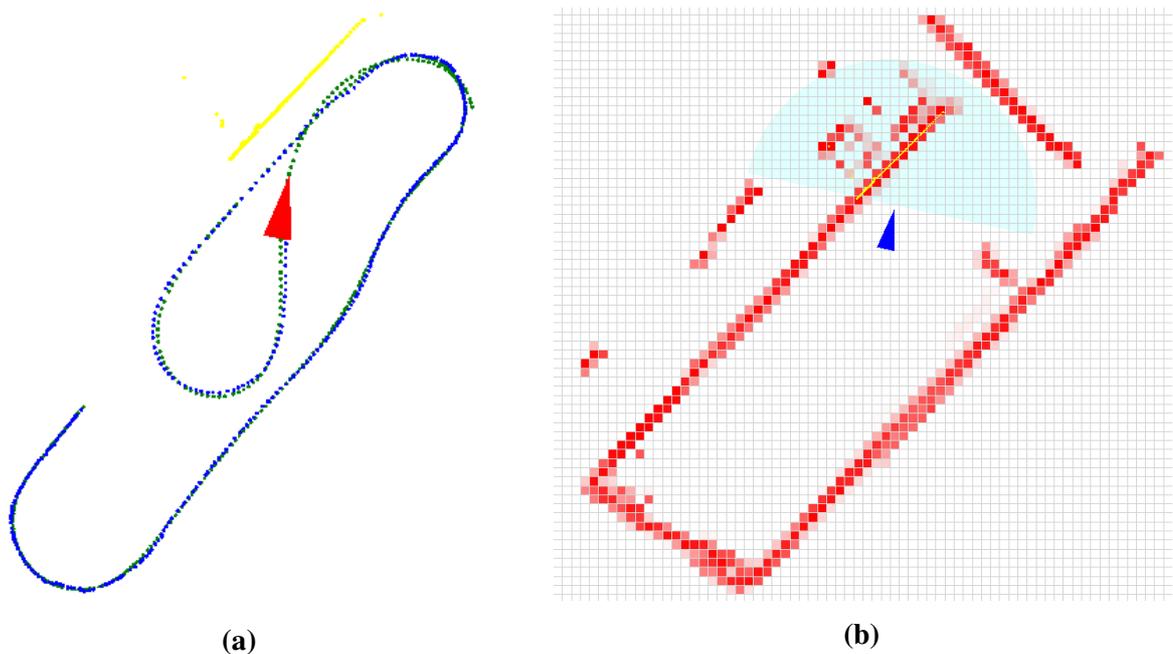
Continuando a simulação, a Figura 5.10 (a) mostra que o veículo já se encontra no mesmo sentido do início da simulação e está próximo de fechar uma volta no estacionamento. Analisando a Figura 5.10 (b) observamos que a parte a direita do veículo foi bem mapeada e mostra com consistência a distribuição de probabilidade.

Entretanto é possível notar que no lado esquerdo do veículo houve uma alteração significativa na altura da leitura do *scanner laser*. Isto ocorre por causa da inclinação da pista, ou seja, quando o veículo está na lateral do estacionamento próximo ao RU, a posição da leitura do *scanner* é alterada fazendo com que o *laser* atinja o piso do estacionamento, gerando uma interpretação incorreta da posição do barranco. Mesmo com este erro de leitura, as distribuições de probabilidade se mostram eficazes para indicar a posição relativa do obstáculo dentro do mapa.



**Figura 5.10 - (a) Percurso; (b) Mapa de Obstáculo.**

A Figura 5.11 (a) mostra que o veículo deu uma volta completa no estacionamento mapeando todo o seu contorno como mostrado na Figura 5.11 (b). Pode-se perceber na Figura 5.11 (b) que o *scanner laser* consegue identificar a grade do RU e o que está atrás dela, visualizando a caixa d'água, a lateral de outro veículo de grande porte estacionado próximo a rampa de carga e descarga e o reservatório de gás. Mesmo com a ocorrência deste fato, a distribuição de probabilidade indica de forma precisa e correta a posição da grade e o que está além dela.

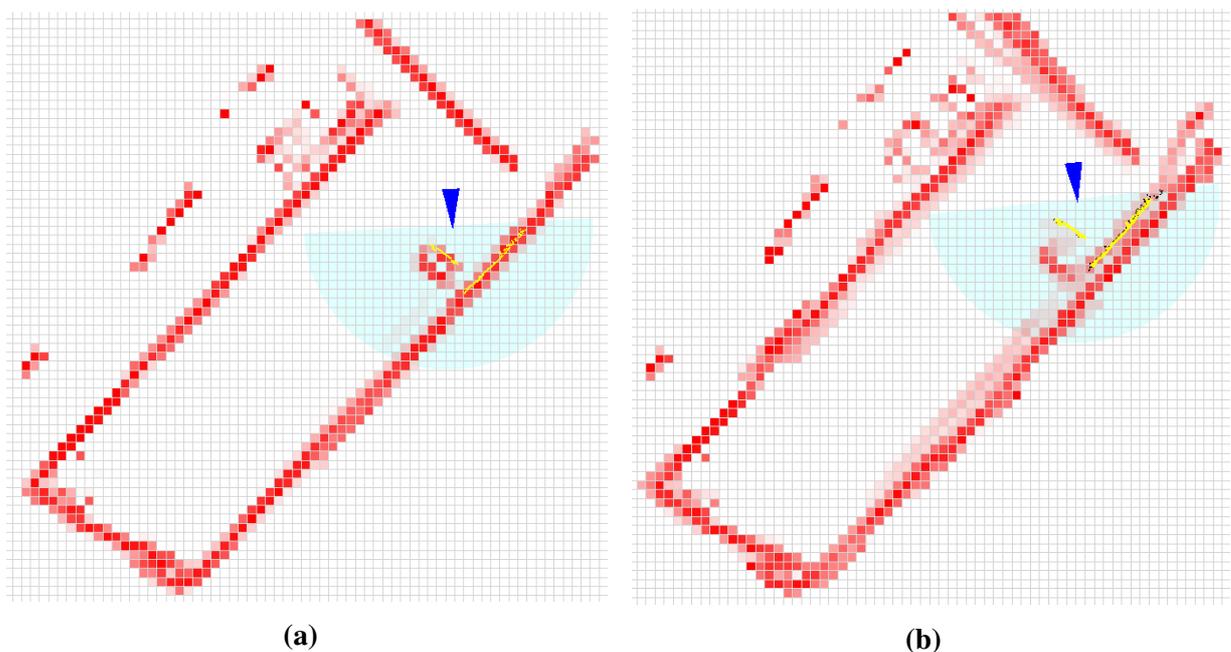


**Figura 5.11 - (a) Percurso; (b) Mapa de Obstáculo.**

Outro ponto importante refere-se ao fechamento de uma volta, ou seja, quando o veículo retorna a um local já mapeado, significa que os obstáculos identificados no fechamento da volta devem estar localizados na mesma posição dos obstáculos mapeados anteriormente, garantido a confiabilidade de todo o trabalho. Mesmo com o problema de desnível da pista descrito anteriormente a distribuição de probabilidade dos objetos encontrados no mapeamento não muito foi afetada.

O fim da simulação é mostrado na Figura 5.12 (a). Analisando a simulação, observa-se que um veículo foi mapeado no lado esquerdo da pista. Como esse veículo não se locomoveu durante a simulação, ele foi considerado um obstáculo estático e possui um alto grau de confiabilidade.

A Figura 5.12 (b) apresenta o resultado da simulação do circuito sem utilizar o algoritmo de correção da posição do veículo.

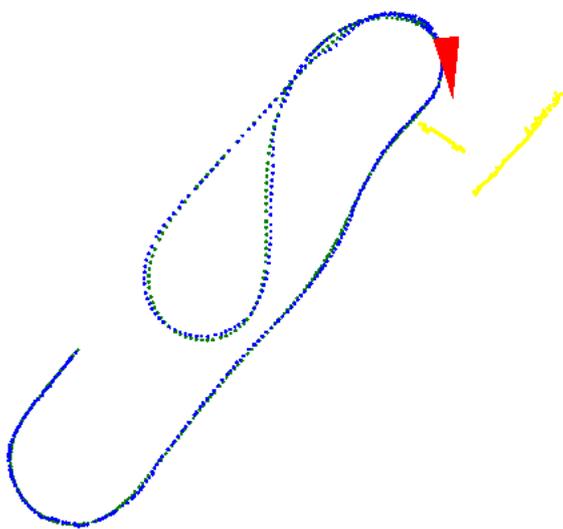


**Figura 5.12 - (a) Mapa de Obstáculo *Grid* 1,0 ; (b) Mapa de Obstáculo *Grid* 1,0 sem Correção.**

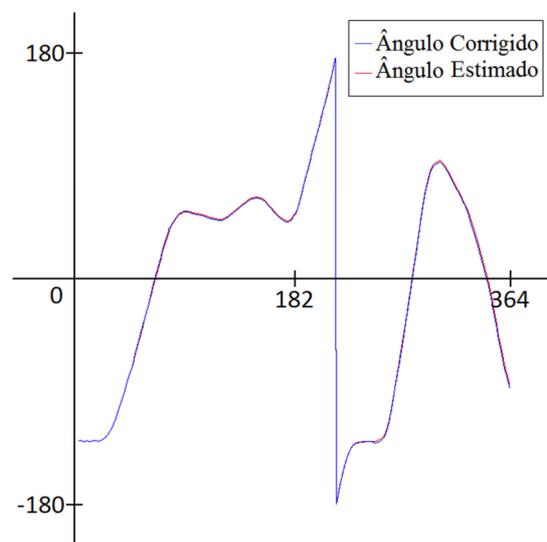
Comparando a Figura 5.12 (a) e (b) percebe-se que sem o algoritmo de correção da posição o mapeamento fica completamente comprometido. Os obstáculos mapeados após o fechamento de uma volta não sobrepõem com os mapeados anteriormente, ou seja, quando o veículo faz um retorno passando por uma área já mapeada os obstáculos não coincidem. Isto ocorre por causa do erro fornecido pelo DGPS, que pode chegar a até um metro, e pelo erro da odometria que tende a acumular conforme com o passar do tempo. Mesmo os *Waypoints*

sendo corrigidos pelo o Filtro  $\alpha-\beta$ , eles apresentam um erro pequeno, porém cumulativo, resultando num mapeamento incorreto como mostrado na Figura 5.12 (b).

Uma comparação entre os valores fornecidos pelo Filtro  $\alpha-\beta$  (posição e orientação do veículo) com os resultados do algoritmo de correção da posição pode ser visto na Figura 5.13 e na Figura 5.14. A Figura 5.13 apresenta os resultados para a posição do veículo onde os pontos verdes representam os valores fornecidos pelo Filtro  $\alpha-\beta$  e os pontos azuis os pontos gerados pelo algoritmo de correção da posição e a Figura 5.14 apresenta os resultados para a orientação do veículo.



**Figura 5.13 - Percurso do Veículo**

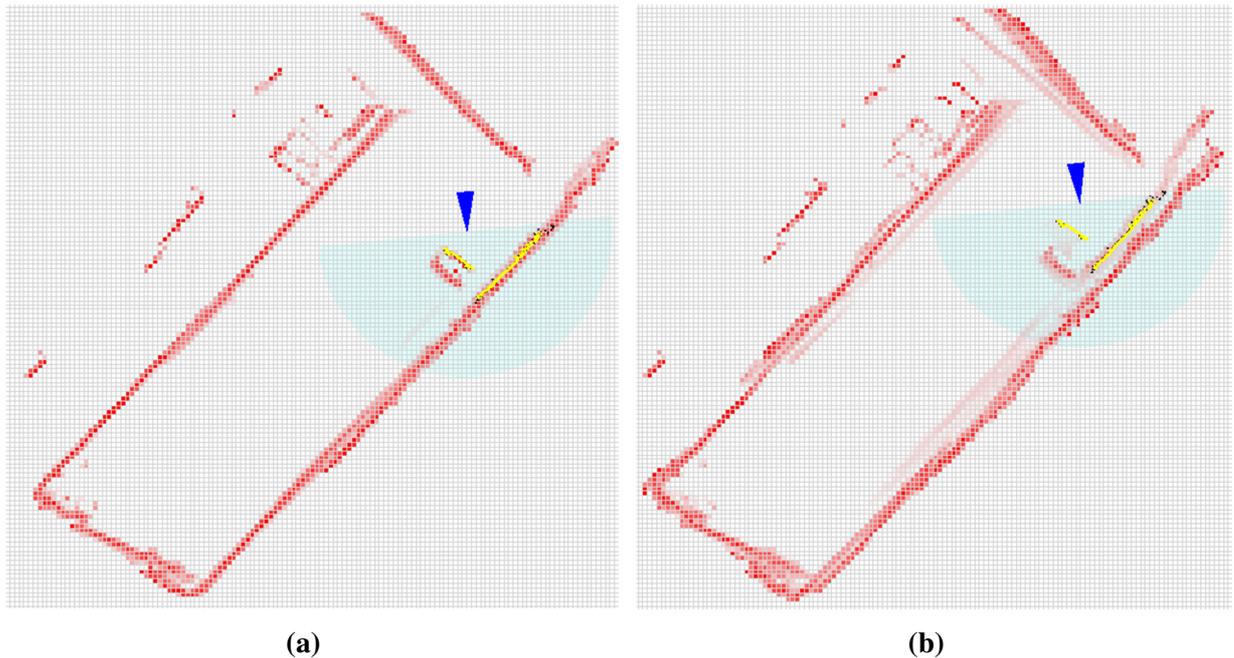


**Figura 5.14 - Ângulo do Veículo**

Analisando as Figura 5.13 e Figura 5.14, observa-se que não ocorre uma grande variação na posição e orientação do veículo, contudo, essa pequena variação pode causar um grande erro na posição do obstáculo. Por exemplo, um erro na orientação do veículo de  $1^\circ$  para um obstáculo a  $10m$  provoca um erro de aproximadamente  $0,35m$  na posição do objeto e esse erro cresce conforme a distância aumenta.

Os resultados apresentados na Figura 5.12 foram calculados para um *grid* de  $1,0m$  enquanto os resultados apresentados na Figura 5.15 foram calculados para um *grid* de  $0,5m$ . Comparando as figuras, observa-se que a precisão do mapeamento para um *grid* de  $0,5m$  é maior, porém o esforço computacional necessário para se obter essa precisão varia ao quadrado para esse problema.

É necessário então avaliar o que se torna mais importante para a aplicação deste algoritmo, a precisão dos obstáculos ou o tempo computacional gasto para o mapeamento.

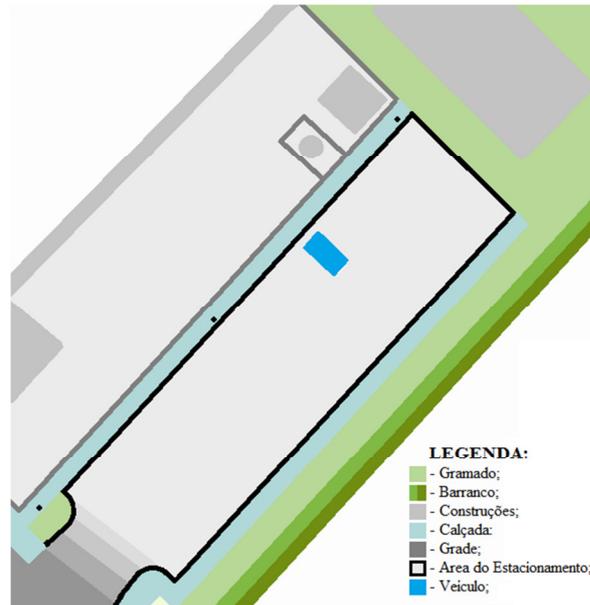


**Figura 5.15 - (a) Mapa de Obstáculo *Grid* 0,5; (b) Mapa de Obstáculo *Grid* 0,5 sem Correção.**

## 5.4.2 CIRCUITO 2

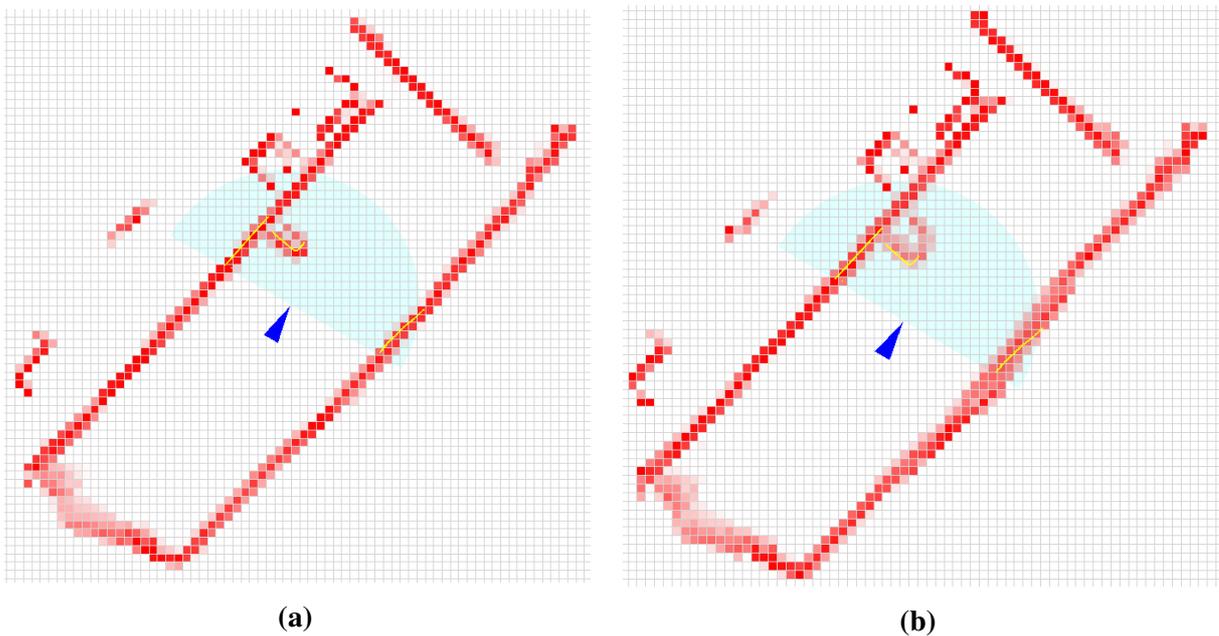
O Circuito 2 também foi mapeado no Estacionamento 1 e sua configuração é apresentada na Figura 5.16. No momento do teste havia apenas um veículo estacionado na parte esquerda do circuito, oposto ao Circuito 1. O veículo está na cor azul. Nenhuma pessoa ou veículo transitou no estacionamento durante a aquisição de dados. Desta forma, não existe nenhum elemento dinâmico nesta amostragem.

O Circuito 2 possui um comprimento total de 118 metros. O veículo inicia o percurso fazendo uma curva no sentido horário e depois inverte o sentido fazendo outra curva e parando após fechar a volta. O percurso possui um total de 583 *Waypoints*. A velocidade média desenvolvida durante o percurso foi de aproximadamente  $10\text{Km/h}$ .



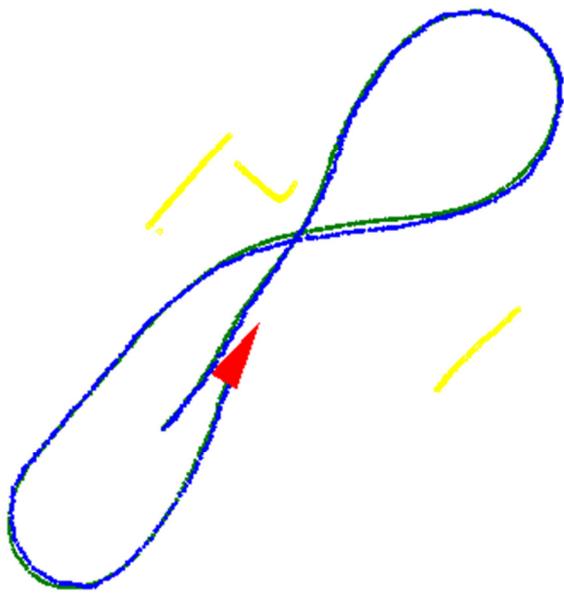
**Figura 5.16 - Circuito 2 Configuração**

O mapa de obstáculos resultante da simulação do Circuito 2 é apresentado na Figura 5.17 (a), com correção da posição do veículo e a Figura 5.17 (b), sem correção da posição.

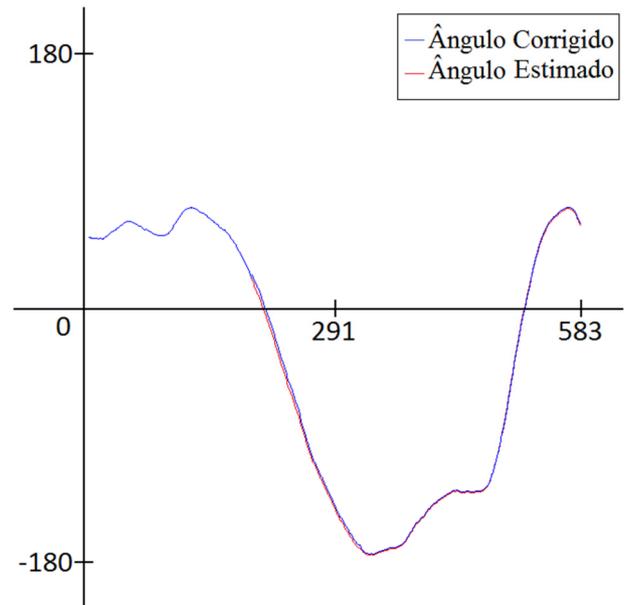


**Figura 5.17 - (a) Mapa de Obstáculo *Grid* 1,0 ; (b) Mapa de Obstáculo *Grid* 1,0 sem Correção.**

A Figura 5.18 mostra o percurso realizado pelo veículo e sua correção, enquanto a Figura 5.19 mostra a variação de sua orientação juntamente com sua correção.

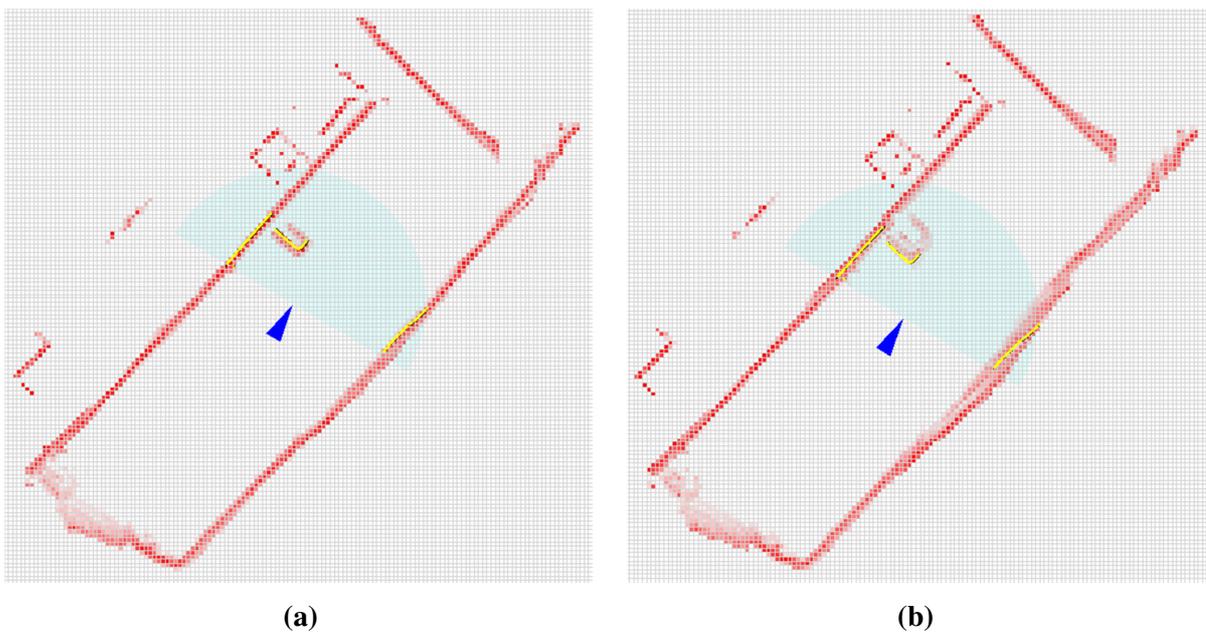


**Figura 5.18 - Percurso do Veículo.**



**Figura 5.19 - Ângulo do Veículo.**

Analisando a Figura 5.18 e a Figura 5.19 observa-se que, um percurso realizado em baixa velocidade apresenta uma amostragem mais densa e próxima da real posição do veículo, fazendo com que o esforço computacional do algoritmo de correção da posição do veículo seja menor. Essa mesma situação pode ser observada comparando a Figura 5.17 (a) e (b), onde o mapeamento sem correção é muito próximo do mapeamento que sofreu correção.



**Figura 5.20 - (a) Mapa de Obstáculo *Grid* 0,5; (b) Mapa de Obstáculo *Grid* 0,5 sem Correção.**

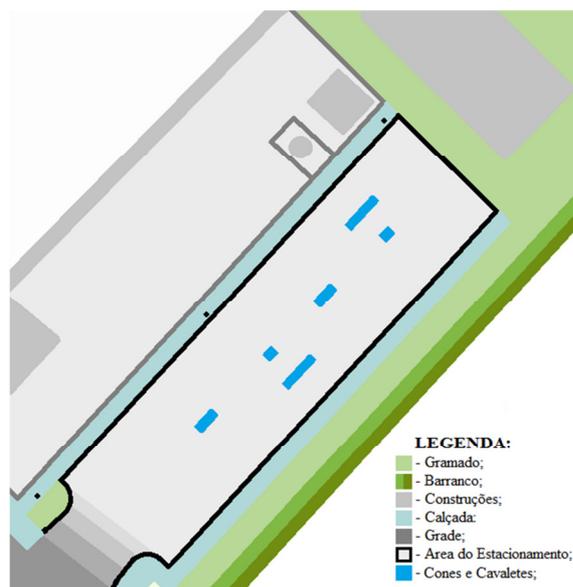
O resultado do mapeamento do Circuito 2 para um *Grid* de  $0,5m$  é mostrado na Figura 5.20 para que seja possível sua comparação, assim como realizado no Circuito 1, com os resultados para um *Grid* de  $1,0m$  apresentados na Figura 5.17.

Comparando os resultados do Circuito 1 com o Circuito 2 observa-se que, quando o veículo se movimenta em baixas velocidades o erro de posicionamento é menor.

### 5.4.3 CIRCUITO 3

O Circuito 3 também foi mapeado no Estacionamento 1 e sua configuração é apresentada na Figura 5.21. Para o teste foram utilizados cones e cavaletes como representação de obstáculos estáticos e para obstáculo dinâmico uma pessoa se movimenta na frente do veículo. Nenhum veículo transitou no estacionamento durante a aquisição de dados.

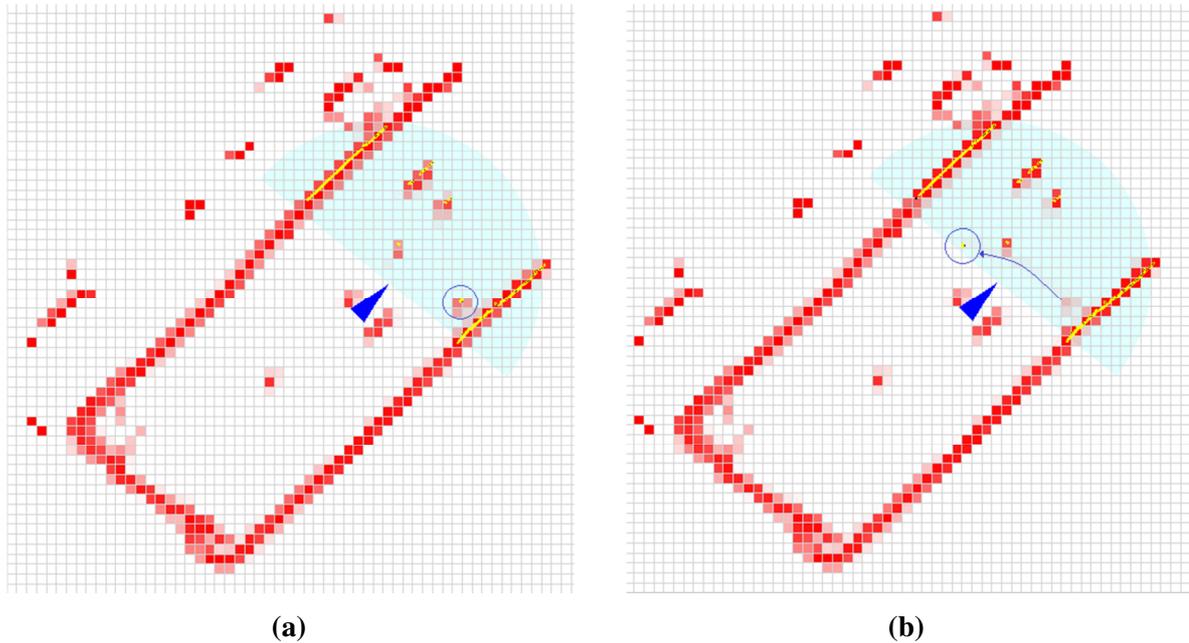
As características do Circuito 3 são: comprimento total de 78 metros, uma curva de  $180^\circ$ , velocidade média de aproximadamente  $20Km/h$  e total de 312 *Waypoints*.



**Figura 5.21 - Circuito 3 Configuração.**

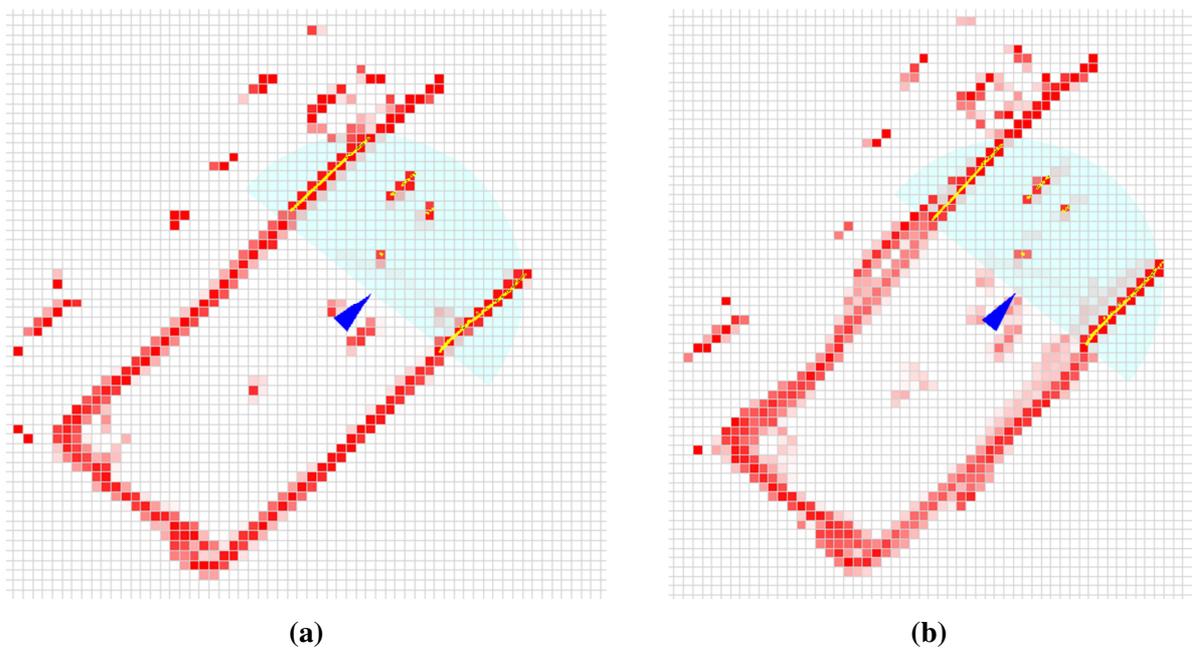
A simulação ocorre igual aos circuitos anteriores, o que difere este Circuito 3 dos demais é a introdução de um obstáculo dinâmico. No *Waypoint* 215 o veículo para e uma pessoa atravessa na frente do veículo passando por todo o campo de leitura do *scanner laser*. O objetivo é analisar o valor de probabilidade fornecido pelo algoritmo para o obstáculo dinâmico.

A Figura 5.22 (a) mostra a simulação quando o obstáculo dinâmico começa a se movimentar. O obstáculo dinâmico está marcado com uma circunferência azul. A Figura 5.22 (b) mostra a posição final e a trajetória feita pelo obstáculo dinâmico



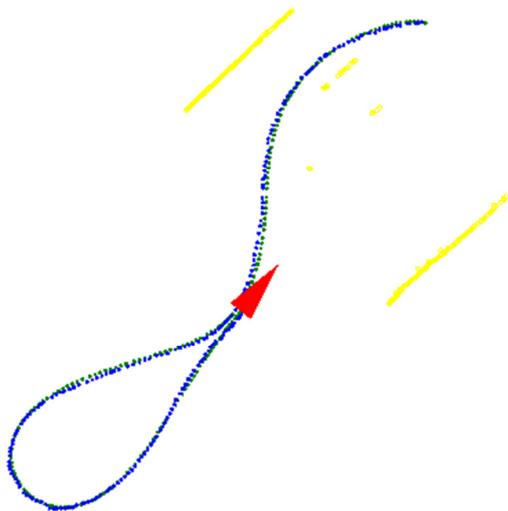
**Figura 5.22 - (a) Início Obstáculo Dinâmico; (b) Fim Obstáculo Dinâmico.**

A Figura 5.23 (a) mostra o resultado final do mapeamento com correção da posição e a Figura 5.23 (b) mostra o mapeamento sem a correção da posição.

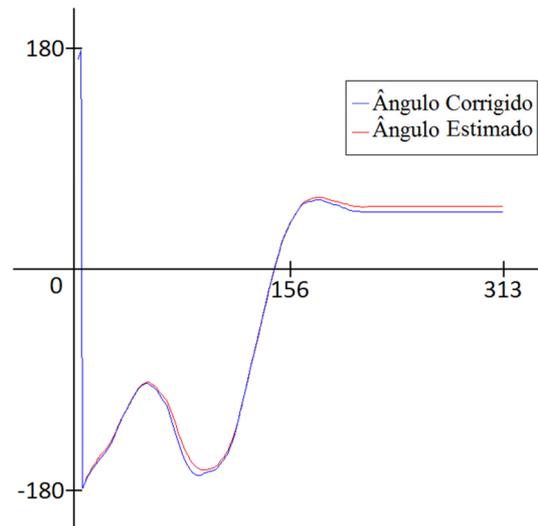


**Figura 5.23 - (a) Mapa de Obstáculo *Grid* 1,0; (b) Mapa de Obstáculo *Grid* 1,0 sem Correção.**

Analisando a Figura 5.23 (a) observa que a trajetória feita pelo obstáculo dinâmico é marcada no mapa de obstáculos com uma cor muito fraca o que significa que a probabilidade de ter um obstáculo naquela região é baixa. Se houvesse uma continuação no mapeamento a tendência era que a região marcada pelo obstáculo dinâmico desaparece no mapeamento.



**Figura 5.24 - Percurso do Veículo.**

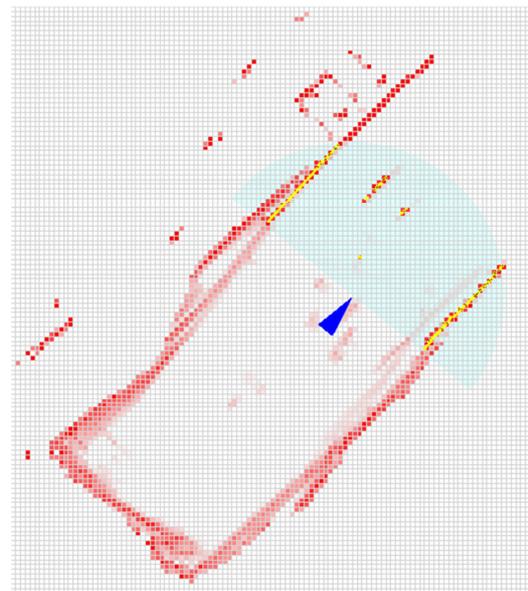


**Figura 5.25 - Ângulo do Veículo.**

A Figura 5.24 e a Figura 5.25 mostram uma comparação entre os valores sem correção e os valores corrigidos da posição e da orientação do veículo, respectivamente. Nota-se, como já comentado anteriormente, que para uma velocidade maior o erro de posicionamento também se torna maior.



**(a)**



**(b)**

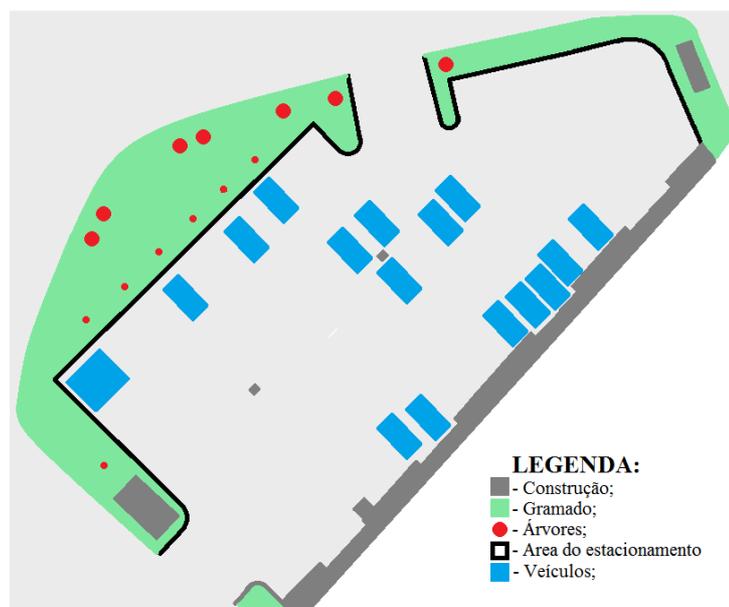
**Figura 5.26 - (a) Mapa de Obstáculo *Grid* 0,5; (b) Mapa de Obstáculo *Grid* 0,5 sem Correção.**

O resultado do mapeamento do Circuito 3 para um *Grid* de  $0,5m$  é mostrado na Figura 5.26. Comparando com os resultados para um *Grid* de  $1,0m$  apresentados na Figura 5.23, observa-se a eficiência do algoritmo de correção da posição para os dois *Grids*, diferindo-os apenas pelo esforço computacional.

#### 5.4.4 CIRCUITO 4

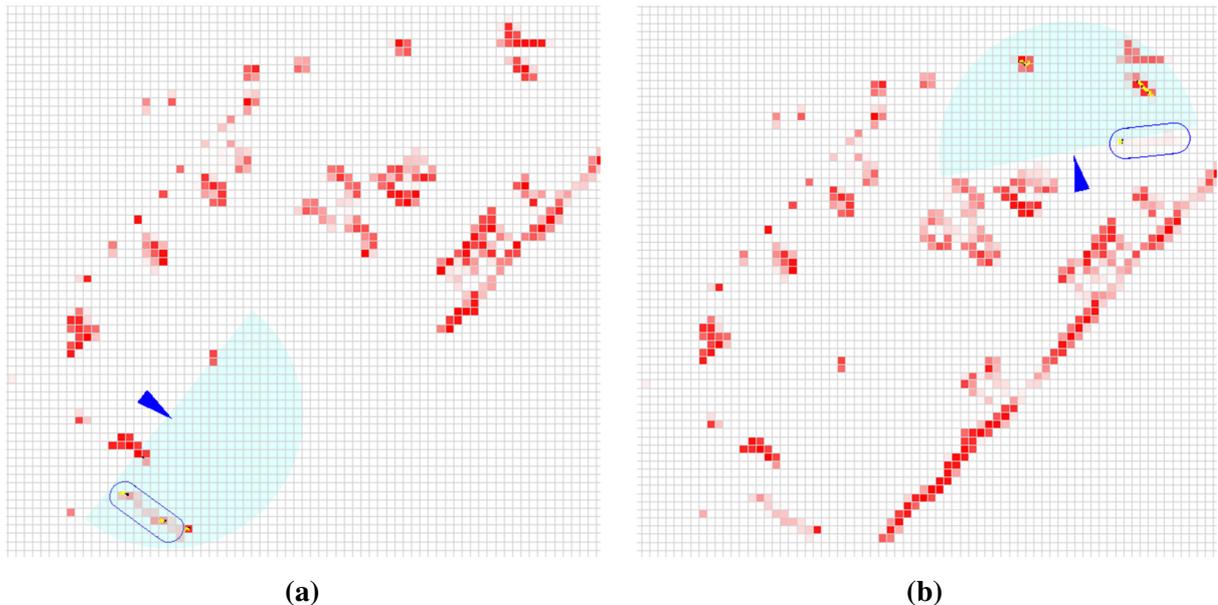
O Circuito 4 foi mapeado no Estacionamento 2 e sua configuração é apresentada na Figura 5.27. Este teste foi realizado em um ambiente não controlado, onde carros e pessoas transitaram livremente no estacionamento. O objetivo dessa simulação é analisar a resposta do algoritmo para uma situação adversa. Durante a simulação várias pessoas atravessam o estacionamento, mas somente três entraram no campo de leitura do *scanner laser*. Um carro que estava estacionado deixou o estacionamento durante a simulação, mas não se movimentou no período em que o *scanner laser* passava por ele.

Para essa simulação o veículo inicia o percurso fazendo uma volta no sentido anti-horário com uma velocidade constante de aproximadamente  $10Km/h$ . Após ter feito uma volta e meia nesse mesmo sentido o veículo faz outra volta e meia no sentido contrário ao anterior com uma velocidade maior de aproximadamente  $20Km/h$ . O Circuito 4 possui um comprimento total de 337 metros, sete curvas de  $180^\circ$  e um total de 1169 *Waypoints*.



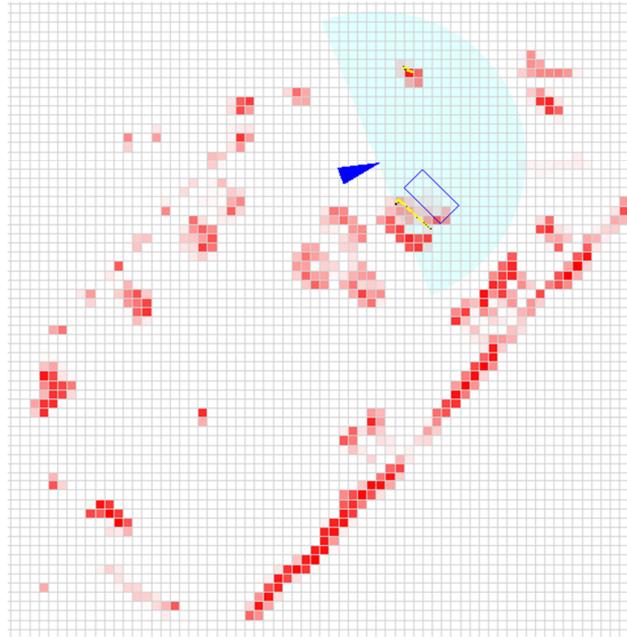
**Figura 5.27 - Circuito 4 Configuração**

A simulação ocorre igual aos circuitos anteriores, o que difere este Circuito 4 dos demais é o mapeamento de obstáculos dinâmicos com o veículo em movimento. A Figura 5.28 (a) e (b) mostram em azul o percurso feito pelos obstáculos dinâmicos. No caso da Figura 5.28 (a) o obstáculo dinâmico está representado por duas pessoas que atravessam o campo de leitura do *scanner laser*, como pode ser visto essas pessoas passam em momentos diferentes e em (b) somente uma pessoa atravessa o campo de leitura do *scanner laser*.



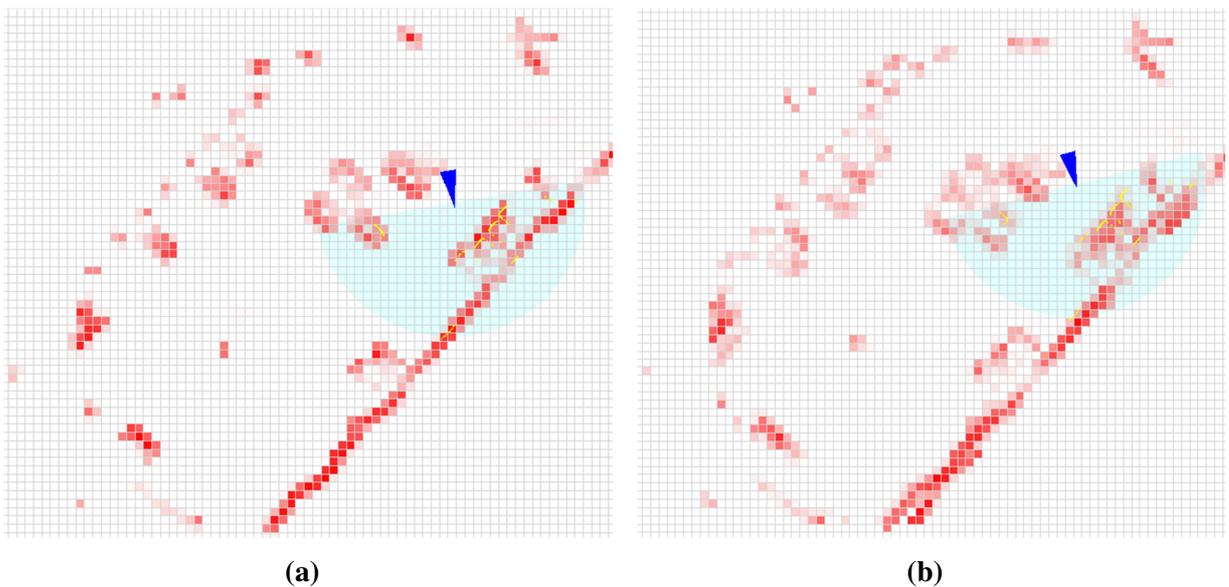
**Figura 5.28 - (a) Obstáculo Dinâmico 1; (b) Obstáculo Dinâmico 2.**

A Figura 5.29 mostra o resultado do mapeamento da região em que se encontrava um carro estacionado que sai durante este mapeamento.



**Figura 5.29 - Obstáculo Dinâmico 3.**

Analisando a Figura 5.28 e a Figura 5.29 percebe-se que a região em que os obstáculos dinâmicos foram marcados a tonalidade tende a diminuir com a continuação do mapeamento, isso mostra que, com o aumento de leituras, a técnica utilizada tende a convergir para uma única situação onde somente existem obstáculos fixos. No final do mapeamento, Figura 5.30, nota-se que as pessoas e o carro descrito anteriormente não são mais visíveis no mapa.



**Figura 5.30 - (a) Mapa de Obstáculo *Grid* 1,0; (b) Mapa de Obstáculo *Grid* 1,0 sem Correção.**



Figura 5.31 - Percurso do Veículo.

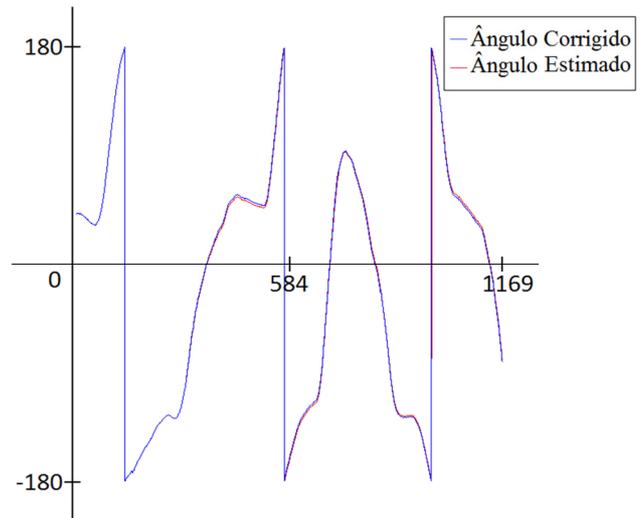
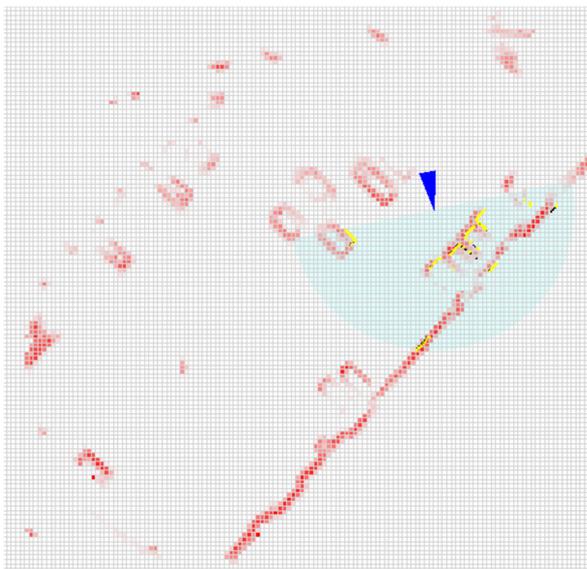
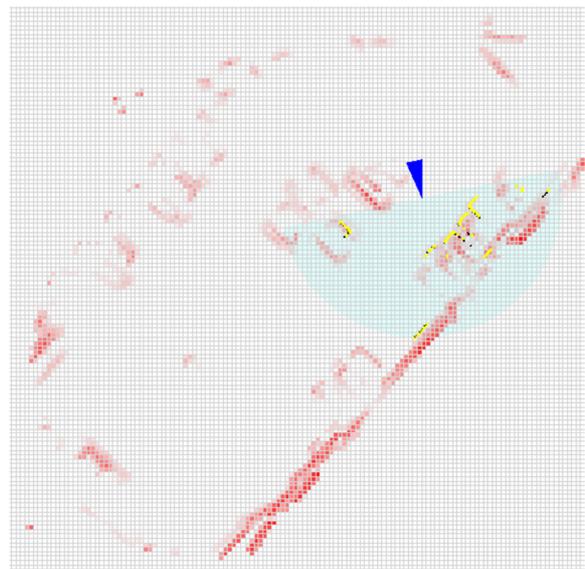


Figura 5.32 - Ângulo do Veículo.

A Figura 5.31 e a Figura 5.32 mostram uma comparação entre os valores sem correção e os valores corrigidos da posição e da orientação do veículo, respectivamente



(a)



(b)

Figura 5.33 - (a) Mapa de Obstáculo *Grid* 0,5; (b) Mapa de Obstáculo *Grid* 0,5 sem Correção.

O resultado do mapeamento do Circuito 4 para um *Grid* de 0,5m é mostrado na Figura 5.33 para que seja possível sua comparação com os resultados para um *Grid* de 1,0m apresentados na Figura 5.30, assim como feito para os outros circuitos.

## CAPÍTULO VI

### 6 CONCLUSÃO

O trabalho apresentado mostrou uma técnica de SLAM Probabilístico. A ideia é construir um mapa e, ao mesmo tempo localizar o veículo dentro desse mapa. Na prática, esses dois problemas não podem ser resolvidos de forma independente um do outro. Desta forma, primeiramente a posição do veículo é estimada e através dessa posição constrói-se um mapa do ambiente com os dados obtidos pelos sensores do veículo. Com o mapa já construído, corrige-se a posição do veículo para uma posição mais precisa e atualiza-se o mapa com os novos valores das posições dos obstáculos.

Para demonstrar esta proposta, um veículo autônomo inteligente foi utilizado. Através da cinemática do veículo e com os dados fornecidos pelo DGPS utiliza-se um filtro  $\alpha - \beta$  para estimar sua posição. Um mapa métrico é construído usando o teorema de Bayes. A técnica de SLAM probabilístico utilizada baseia-se na sobreposição dos obstáculos, onde o obstáculo atual é sobreposto em relação ao obstáculo anterior. Essa técnica de sobreposição leva em consideração a probabilidade dos objetos fornecidos pelo mapa métrico.

Para validar este trabalho os resultados foram comparados com o procedimento onde não ocorre a correção da posição do veículo. Os resultados mostram uma eficiência do método, onde o algoritmo de correção da posição do veículo estima corretamente sua posição fazendo com que o mapa construído tenha uma representação fiel dos obstáculos do ambiente, juntamente com a probabilidade de cada obstáculo.

Tendo em vista o trabalho desenvolvido, apresenta-se como possibilidade de continuidade do estudo: o desenvolvimento de um sistema de navegação que torne possível desviar de obstáculos localizados na rota do veículo identificados através do mapa construído.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

[1] BAILEY, T.; NIETO, J.; NEBOT, E. Consistency of the FastSLAM Algorithm. **International Conference on Robotics and Automation**, Orlando, Maio 2006. 424-429.

[2] BOSSE, M.; ROBERTS, J. Histogram Matching and Global Initialization for Laser-only SLAM in Large Unstructured Environments. **International Conference on Robots and Automation**, Roma, 10 Abril 2007. 4820-4826.

[3] BERTOLLI, F.; JENSFELT, P.; CHRISTENSEN, H. I. SLAM using Visual Scan-Matching with Distinguishable 3D Points. **International Conference on Intelligent Robots and Systems**, Beijing, 9 Outubro 2006. 4042-4047.

[4] ZHAO, H. et al. SLAM in a Dynamic Large Outdoor Environment Using a Laser Scanner. **International Conference on Robots and Automation**, Pasadena, 19 Maio 2008. 1455-1462.

[5] FOLKESSON, J.; CHRISTENSEN, H. Graphical SLAM - A Self-Correcting Map. **International Conference on Robots and Automation**, New Orleans, Abril 2004. 383-390.

[6] COX, I. J. Blanche – An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle. **IEEE Transactions on Robotics and Automation**, v. 7, n. 2, p. 193-201, Abril 1991.

[7] DISSANAYAKE, G. et al. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. **IEEE Transactions on Robotics and Automation**, v. 17, n. 3, p. 229-241, Junho 2001.

[8] GUIZILINI, V. C. **Localização e Mapeamento Simultâneos com Auxílio Visual Omnidirecional**. Dissertação de Mestrado. Universidade de São Paulo. 2008.

[9] DURRANT-WHYTE, H. F. Uncertain Geometry in Robotics. **IEEE Journal of Robotics and Automation**, v. 4, n. 1, p. 23-31, 1988.

[10] DURRANT-WHYTE, H. F.; BAILEY, T. Simultaneous Localization and Mapping: Part I. **IEEE Robotics & Automation Magazine**, p. 99-108, Junho 2006.

[11] SMITH, R.; SELF, M.; CHEESEMAN, P. Estimating Uncertain Spatial Relationships in Robotics. **Autonomous Robot Vehicles**, New York, 1990. 167–193.

[12] RENCKEN, W. D. Concurrent Localisation and Map Building for Mobile Robots Using Ultrasonic Sensors. **International Conference on Intelligent Robots and Systems**, Yokohama, 26 Julho 1993. 2192-2197.

[13] BORTHWICK, S.; DURRANT-WHYTE, H. F. Dynamic Localization of Autonomous Guided Vehicles. **International Conference on Multisensor Fusion and Integration for Intelligent Systems**, Las Vegas, 2 Outubro 1994. 92-97.

[14] CSORBA, M. **Simultaneous Localisation and Map Building**. Tese de Doutorado. University of Oxford,. 1997.

[15] CASTELLANOS, J. A. et al. Experiments in Multisensor Mobile Robot Localization and Map Building. **Proc. 3rd IFAC Sym. Intell. Auton. Vehicles**, 1998. 173-178.

[16] CASTELLANOS, J. A.; TARDÓS, J. D.; SCHMIDT, G. Building a Global Map of the Environment of a Mobile Robot: The Importance of Correlations. **International Conference on Robots and Automation**, Albuquerque, Abril 1997. 1053-1059.

[17] GUIVANT, J.; NEBOT, E.; BAIKER, S. Autonomous Navigation and Map building Using Laser Range Sensors in Outdoor Applications. **Journal of Robotic Systems**, v. 17, n. 10, p. 565-583, Outubro 2000.

[18] WILLIAMS, S. B. et al. Autonomous Underwater Simultaneous Localisation and Map Building. **International Conference on Robotics & Automation**, San Francisco, Abril 2000. 1793-1798.

[19] CHONG, K. S.; KLEEMAN, L. Feature-Based Mapping in Real, Large Scale Environments Using an Ultrasonic Array. **The International Journal of Robotics Research**, v. 18, n. 1, p. 3-19, Março 1999.

[20] THRUN, S.; BURGARD, W.; FOX, D. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. **Autonomous Robots**, 1998. 253–271.

- [21] ENGELSON, S. P.; MCDERMOTT, D. V. Error Correction in Mobile Robot Map Learning. **International conference on Robotics and Automation**, Nice, 12 Maio 1992. 2555-2560.
- [22] NEGENBORN, R. **Robot Localization and Kalman Filters**. Dissertação de Mestrado. University of Utrecht. 2003.
- [23] THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. [S.l.]: The MIT Press, 2005.
- [24] ELFES, A. Sonar-Based Real-World Mapping and Navigation. **IEEE Journal of Robotics and Automation**, v. 3, n. 3, p. 249-265, Junho 1987.
- [25] JENSFELT, P. **Approaches to Mobile Robot Localization in Indoor Environments**. Tese de Doutorado. Royal Institute of Technology. 2001.
- [26] KAPLAN, E. D.; HEGARTY, C. J. **Understanding GPS: Principles and Applications**. 2. ed. [S.l.]: Artech House, Inc, 2006.
- [27] ALVES, A. S. C. **Metodologia e Desenvolvimento de Redes Neuro-Especialista Aplicadas à Navegação Autônomo**. Dissertação de Mestrado. Universidade Federal de Itajubá. 2010.
- [28] BAILEY, T. **Mobile Robot Localization and Mapping in Extensive Outdoor Environments**. Tese de Doutorado. The University of Sydney. 2002.
- [29] BETKE, M.; GURVITS, L. Mobile Robot Localization Using Landmarks. **IEEE Transactions on Robotics and Automation**, v. 13, n. 2, p. 251-263, Abril 1997.
- [30] ALTERMATT, M. et al. SLAM with Corner Features Based on a Relative Map. **International Conference on Intelligent Robots and Systems**, Sendai, 28 Outubro 2004. 1053-1058.
- [31] GUIZILINI, V. C. **Localização e Mapeamento Simultâneos com Auxílio Visual Omnidirecional**. Dissertação de Mestrado. Escola Politécnica da Universidade de São Paulo. 2008.

- [32] BETGÉ-BREZETZ, S. et al. Uncertain Map Making in Natural Environments. **International Conference on Robotics and Automation**, Minneapolis, 22 Abril 1996. 1048-1053.
- [33] NIETO, J. et al. Real Time Data Association for FastSLAM. **International Conference on Robotics & Automation**, Taipei, 14 Setembro 2003. 412-418.
- [34] SELVATICI, A. H. P. **Construção de Mapas de Objetos para Navegação de Robôs**. Tese de Doutorado. Escola Politécnica da Universidade de São Paulo. 2009.
- [35] MACHADO, K. F. **Módulo de Auto-Localização para um Agente Exploratório usando Filtro de Kalman**. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul. 2003.
- [36] ELFES, A. Using Occupancy Grids for Mobile Robot Perception and Navigation. **Computer Magazine**, v. 22, p. 46-57, Junho 1989.
- [37] FOX, D.; BURGARD, W.; THRUN, S. **Probabilistic Methods for Mobile Robot Mapping**. IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments. [S.l.]: [s.n.]. 1999.
- [38] DUDEK, G.; FREEDMAN, P.; HADJRES, S. Using Local Information in a Non-Local Way for Mapping Graph-Like Worlds. **International Joint Conference of Artificial Intelligence**, Chambéry, Agosto 1993. 1639-1647.
- [39] SHATKAY, H.; KAELBLING, L. P. Learning Topological Maps with Weak Local Odometric Information. **International Joint Conference of Artificial Intelligence**, 1997. 920-929.
- [40] BUSCHKA, P. **An Investigation of Hybrid Maps for Mobile Robots**. Tese de Doutorado. Örebro University. 2006.
- [41] GUIVANT, J. E.; NEBOT, E. M. Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation. **IEEE Transactions on Robotics And Automation**, v. 17, n. 3, p. 242-157, Junho 2001.
- [42] ESTRADA, C.; NEIRA, J.; TARDÓS, J. D. Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments. **IEEE Transactions on Robotics**, v. 21, n. 4, p. 588-596, Agosto 2005.

[43] IP, Y. L.; RAD, A. B. Incorporation of Feature Tracking into Simultaneous Localization and Map Building via Sonar Data. **Journal of Intelligent and Robotic Systems** **39**, p. 149-172, 2004.

[44] THRUN, S. A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots. **The International Journal of Robotics Research**, p. 335-363, 2001.

[45] YUEN, D. C. K.; MACDONALD, B. A. An Evaluation of the Sequential Monte Carlo Technique for Simultaneous Localisation and Map-building. **International Conference on Robotics and Automation**, Taipei, Setembro 2003. 1564-1569.

[46] KOUZOUBOV, K.; AUSTIN, D. Hybrid Topological/Metric Approach to SLAM. **International Conference on Robotics and Automation**, New Orleans, Abril 2004. 872-877.

[47] MURPHY, K. P. Bayesian Map Learning in Dynamic Environments. **Neural Information Processing Systems (NIPS)**, 1999.

[48] MONTEMERLO, M. et al. **FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem**. Carnegie Mellon University. Pittsburgh. 2002.

[49] LEONARD, J. J.; FEDER, H. J. S. Decoupled Stochastic Mapping. **IEEE JOURNAL OF OCEANIC ENGINEERING**, v. 26, n. 4, p. 561-571, Outubro 2001.

[50] WILLIAMS, S. B. **Efficient Solutions to Autonomous Mapping and Navigation Problems**. Tese de Doutorado. Australian Centre for Field Robotics, Sydney. 2001.

[51] DUCKETT, T. A Genetic Algorithm for Simultaneous Localization and Mapping. **International Conference on Robotics and Automation**, Taipei, 14 Setembro 2003. 434-439.

[52] FRESE, U.; LARSSON, P.; DUCKETT, T. A Multilevel Relaxation Algorithm for Simultaneous Localization and Mapping. **IEEE Transactions on Robotics**, v. 21, n. 2, p. 196-207, Abril 2005.

[53] SCHROETER, C.; BOEHME, H. -J.; GROSS, H. -M. Robust Map Building for an Autonomous Robot using Low-cost Sensors. **IEEE International Conference on Systems, Man and Cybernetics**, 10 Outubro 2004. 5398-5403.

[54] PEDROSA, D. P. F.; MEDEIROS, A. A. D.; ALSINA, P. J. Uma Proposta de SLAM com Determinação de Informações Geométricas do Ambiente. **Congresso Brasileiro de Automática**, Salvador, 3 Outubro 2006. 1704-1709.

[55] PEDROSA, D. P. F. **Mapeamento de Ambientes Estruturados com Extração de Informações Geométricas através de Dados Sensoriais**. Tese de Doutorado. Universidade Federal do Rio Grande do Norte. 2006.

[56] GONÇALVES, L. M. **Sistema de Navegação de Veículo Autônomo Inteligente**. Dissertação de Mestrado. Universidade Federal de Itajubá. 2010.