

## 4 Resultados

A metodologia apresentada foi submetida a ambientes típicos de casas, fábricas e escritórios. A maior parte de locais construídos por seres humanos são compostos de uma grande quantidade de superfícies planas. Neste estudo todos os ambientes são estáticos, ou seja, o ambiente não se altera com o tempo e nem há nenhum tipo de movimentação dentro da região livre do ambiente e composto apenas por superfícies planas.

O protótipo utilizado nos experimentos é composto por um anel de sete sonares dispostos ao longo de  $180^\circ$  espaçados  $30^\circ$  entre si, como mostrado na Figura 34. Cada sonar possui alcance máximo de 200 cm e alcance mínimo de 20 cm, com 1 cm de resolução.

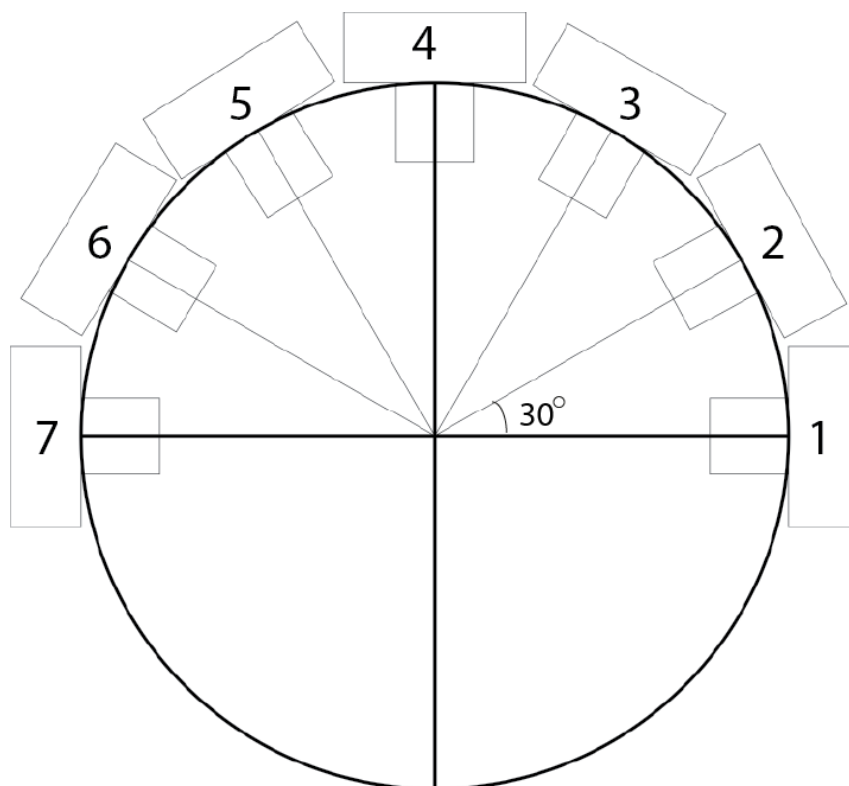


Figura 34 - Modelo do robô usado.

O robô possui duas esteiras paralelas de drive diferencial para locomoção, espaçadas 10 cm e com altura de 3 cm (equivalente ao diâmetro da roda). Os atuadores conectados às esteiras contam com um *encoder* de  $1^\circ$  de resolução. O robô conta ainda com uma bússola eletrônica, também com  $1^\circ$  de resolução.

Para evitar o desenvolvimento de um algoritmo de navegação e planejamento de trajetória, o robô segue uma trajetória pré-estabelecida nas simulações computacionais, e é controlado manualmente nos testes em ambientes reais. O controle manual garante que o robô fará a trajetória desejada, assim, tal trajetória poderá ser reconstruída de forma aproximada em ambiente computacional, viabilizando a apresentação dos resultados.

Como o interesse é comprovar a convergência do algoritmo apresentado, o algoritmo deverá ser capaz de estimar a posição do robô o mais próximo possível da posição real do robô (nas simulações computacionais) ou manter o robô na trajetória percorrida manualmente (nos ensaios com ambientes reais).

#### 4.1 O simulador

Para os testes computacionais um simulador de duas dimensões foi desenvolvido em MatLab, onde o ambiente é descrito por uma matriz de zeros e uns. As posições na matriz iguais a um representam regiões ocupadas (ou paredes) e posições iguais a zero representam espaços sem obstáculos. Logo, um ambiente pode ser facilmente desenhado em uma imagem em preto e branco e ser utilizada como um mapa para o simulador.

Para simular o ruído na transição de estado do robô, devido a derrapagens ou ruídos no próprio *encoder*, foi utilizado o algoritmo apresentado na Tabela 4, (Thrun, 2005).

---

1	Algoritmo modelo_de_locomocão( $v_t, w_t, x_{t-1}, \Delta t$ )
2	$\theta_t = \theta_{t-1} + w_t \cdot \Delta t$
3	$x_t = x_{t-1} + v_t \cdot \cos(\theta_{t-1}) \cdot \Delta t$
4	$y_t = y_{t-1} + v_t \cdot \sin(\theta_{t-1}) \cdot \Delta t$
5	$\delta_{rot1} = \text{atan2}(y_t - y_{t-1}, x_t - x_{t-1})$
6	$\delta_{trans} = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}$
7	$\delta_{rot2} = \theta_t - \theta_{t-1} - \delta_{rot1}$
8	$\hat{\delta}_{rot1} = \delta_{rot1} - N(\gamma_1 \delta_{rot1} + \gamma_2 \delta_{trans}, 1)$
9	$\hat{\delta}_{trans} = \delta_{trans} - N(\gamma_3 \delta_{trans} + \gamma_4 (\delta_{rot1} + \delta_{rot2}), 1)$

---

---

10	$\hat{\delta}_{rot2} = \delta_{rot2} - N(\gamma_1 \delta_{rot2} + \gamma_2 \delta_{tras})$
11	$x_t = x_{t-1} + \hat{\delta}_{trans} \cdot \cos(\theta_{t-1} + \hat{\delta}_{rot1})$
12	$y_t = y_{t-1} + \hat{\delta}_{trans} \cdot \sin(\theta_{t-1} + \hat{\delta}_{rot1})$
13	$\theta_t = \theta_{t-1} + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$

---

Tabela 4 - Modelo de locomoção com erro.

Os parâmetros  $\gamma_1$  a  $\gamma_4$  determinam o quanto o robô se desviará da trajetória real. Foram utilizados valores de 0,04 para  $\gamma_1$  e  $\gamma_2$  e 0,01 para  $\gamma_3$  e  $\gamma_4$ . Em 10 mil amostras deste modelo de deslocamento com os parâmetros especificados acima, em um robô que se desloca a 5 cm/s linearmente, partindo do centro do sistema de coordenadas e alinhado com o eixo Y, com um intervalo de 0,5 segundos, a distribuição de possíveis posições pode ser vista na Figura 35. O robô pode deslocar-se até 2 cm lateralmente e aproximadamente 1cm no sentido do deslocamento.

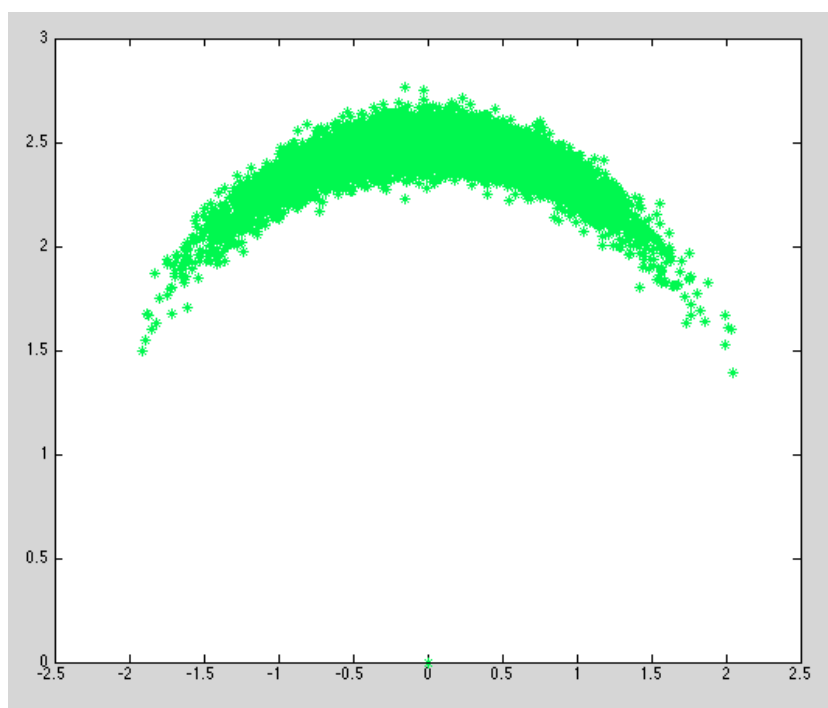


Figura 35 - 10 mil amostras do modelo de locomoção.

Para simular a bússola eletrônica, foi considerado um erro branco de até 4 graus, como mostrado na Figura 36. Dessa forma, o robô real pode ter um erro de até  $4^\circ$ , para mais ou para menos.

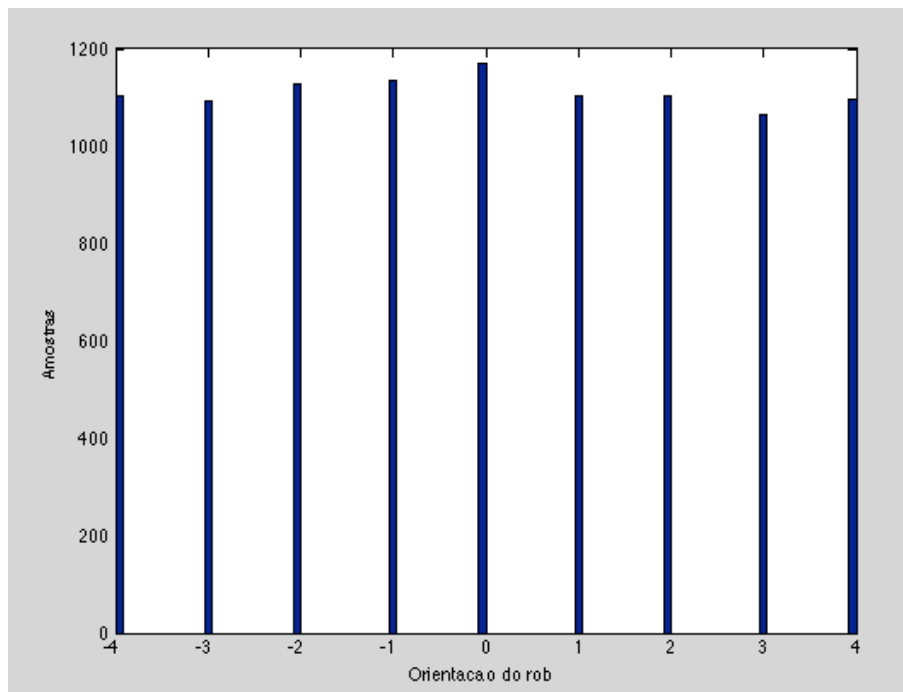


Figura 36 - Histograma da orientação para as 10 mil amostras.

O simulador desenvolvido possui algumas limitações que devem ser destacadas. O simulador não é capaz de simular *specular reflections*, por mais que o ângulo de incidência seja muito grande, em todas as situações as leituras dos sonares sempre serão retornadas com sucesso caso haja obstáculos à frente. O simulador também é indiferente ao problema de *Cross-Talk*, sendo assim, as leituras de um sonar nunca interferirão nas leituras de outro sonar. Estas limitações são decorrentes da dificuldade de simulação da propagação das ondas sonoras, que por si só, já é um processo computacional complexo e custoso. No simulador em questão, a leitura do sonar é feita simplesmente buscando pelo obstáculo mais próximo dentro do cone do sonar. Esta técnica é bastante compatível com o modelo de propagação do som e de como é feita a medição pelo sonar real, ou seja, pelo eco que primeiro retorna ao sonar.

## 4.2 Ambientes de Simulação e Ambientes Reais

Os ambientes escolhidos para a simulação computacional foram três: um quarto quadrado vazio de 2,5 m de aresta, um corredor em laço fechado de 10 m de comprimento e um ambiente típico de escritórios: Figura 37, Figura 38 e Figura 39 respectivamente.

Para ensaios com o robô real foram utilizados três ambientes. Um ambiente no formato de “L”, um corredor de aproximadamente 9 metros de comprimento e

um terceiro ambiente mais complexo (parte do corredor do Departamento de Engenharia Elétrica da PUC-Rio), Figura 40, Figura 41 e Figura 42 respectivamente. Os ambientes estão desenhados em preto e branco de forma bastante simples, captando apenas os principais planos do ambiente.

A apresentação dos resultados será feita comparando a pose estimada apenas com os sensores proprioceptivos (Estimação Simples), a estimação utilizando o LMS-H e a trajetória real do robô.

A Estimação Simples é a estimação da posição do robô utilizando apenas as informações dos *encoders* e bússola, ou seja, utilizando apenas a odometria corrigida pelas informações da bússola. Como as informações dos *encoders* e bússola são inexistente nos ambientes simulados, a Estimação Simples passa a ser simplesmente a resposta do algoritmo de locomoção, mostrado na Tabela 4, corrigido com a orientação do robô acrescido de um erro branco de  $\pm 4^\circ$  para simular o erro das bússolas eletrônicas. Lembrando que o deslocamento do robô simulado e a resposta da Estimação Simples são duas respostas distintas do algoritmo de locomoção, ou seja, o algoritmo é chamado para fazer a locomoção do robô simulado e então é chamado novamente para o cálculo da Estimação Simples. Esse procedimento simula o erro existente entre a leitura dos *encoders* reais com o deslocamento do robô no mundo real.

Para exibição dos resultados, todas as trajetórias exibidas em azul no mapa são as trajetórias reais do robô, as trajetórias em vermelho são as trajetórias do robô utilizando o algoritmo LMS-H e em verde as trajetórias com a Estimação Simples. Serão apresentados ainda em amarelo as retas mapeadas pelo robô e em vermelho os pontos pertencentes às retas mapeadas. Devido à dificuldade de rastrear o robô de forma precisa nos ensaios em ambientes reais, a trajetória real será apenas uma trajetória aproximada, dado que o robô foi controlado manualmente e sua trajetória é conhecida.

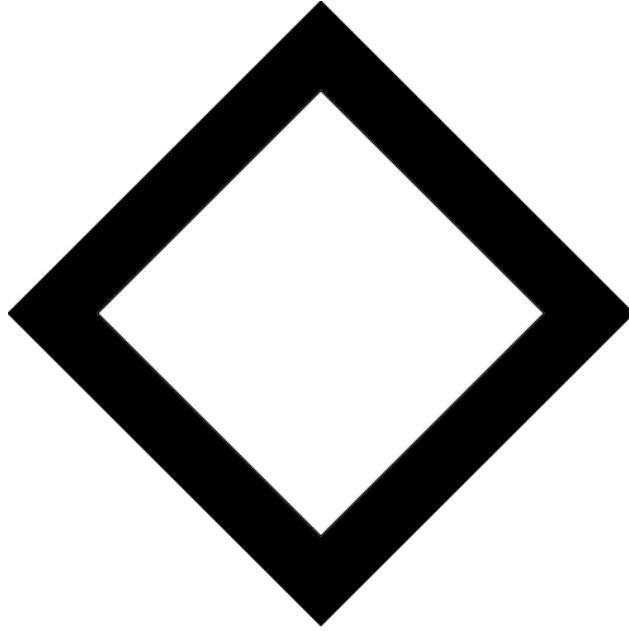


Figura 37 - Quarto Vazio.

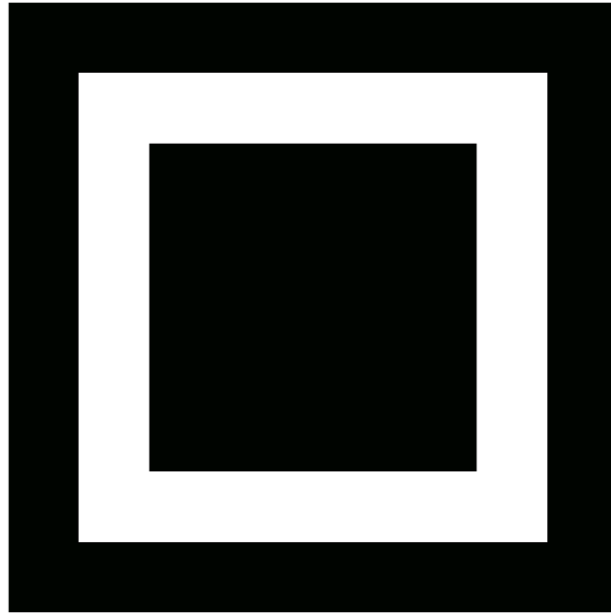


Figura 38 - Corredor em laço fechado.

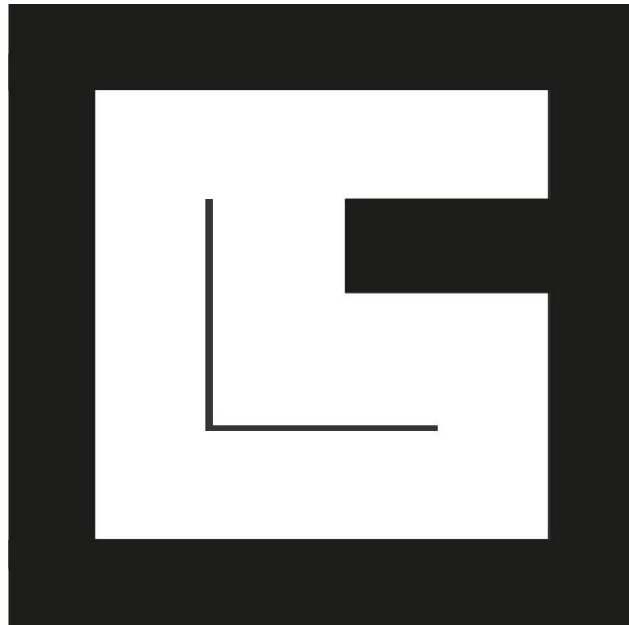


Figura 39 - Escritório.

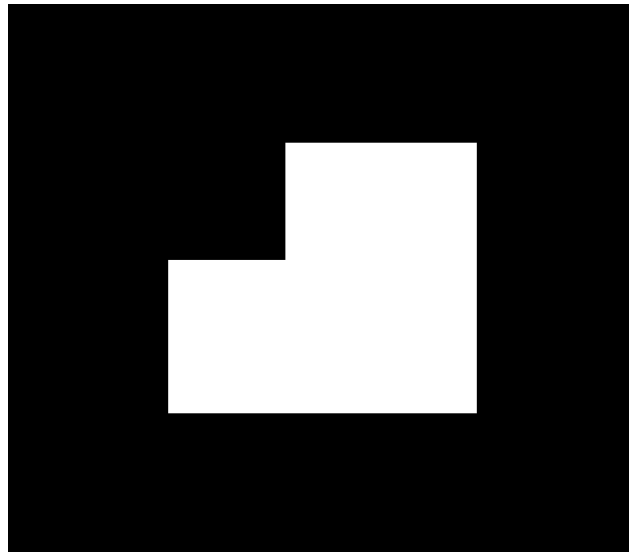


Figura 40 - Ambiente real em "L".



Figura 41 - Corredor.

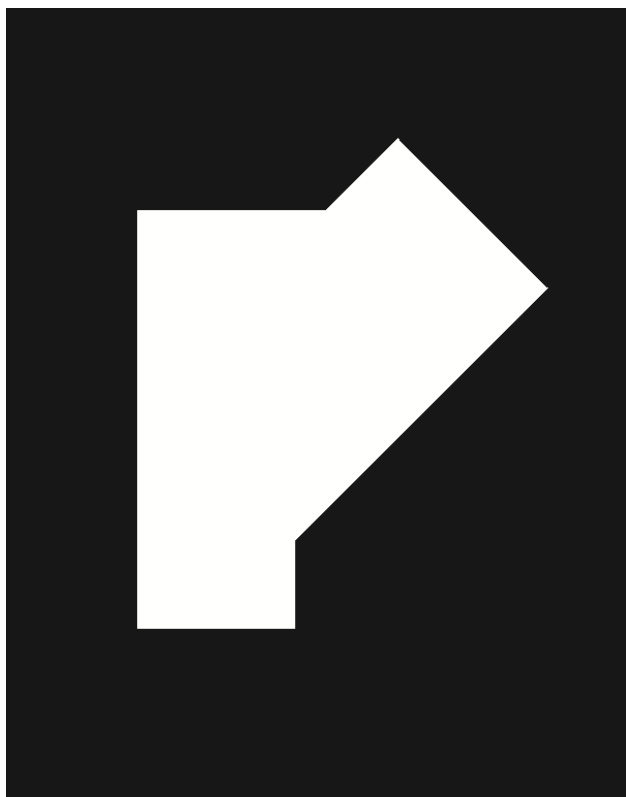


Figura 42 - Ambiente real, parte do corredor do departamento de engenharia elétrica da PUC-Rio.

### 4.3 Parâmetros e Configurações

Em todos os casos, os parâmetros utilizados estão dispostos na Tabela 5.

Descrição	Símbolo	Valor
<b>Configurações do Sonar</b>		
Abertura do Cone do Sonar	$\beta$	20°
Alcance Mínimo	$d_{min}$	20 cm
Alcance Máximo	$d_{max}$	200 cm
Erro Máximo do Sonar	$\varepsilon$	5 cm
Numero de pontos extraídos por leituras	-	1
<b>Configurações do DBSCAN</b>		
Mínimo número de pontos em um cluster	$min_p$	5
Raio da vizinhança	$\phi$	20 cm
Erro médio quadrado máximo da reta	$\epsilon$	5



Configurações do Mapa de Ocupação em Grade		
Tamanho da célula	-	1 cm <sup>2</sup>
Limiar superior – células ocupadas	-	0,5
Limiar inferior – células vazias	-	-0,5
Configuração de Detecção de Retas		
Erro máximo na coordenada $\rho$ da reta quando detectado pelo sonar no sistema de coordenadas de Hough.	$\xi_\rho$	20 cm
Erro máximo na coordenada $\alpha$ da reta quando detectado pelo sonar no sistema de coordenadas de Hough.	$\xi_\alpha$	5°
Covariâncias		
Covariância de deslocamento.	R	$R = \begin{bmatrix} 0,01 & 0 & 0 \\ 0 & 0,01 & 0 \\ 0 & 0 & 0,01 \end{bmatrix}$
Covariância da observação.	Q	$R = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$
Covariância da leitura da bússola.	$\sigma_b$	0,001

Tabela 5 - Parametrização do LMS-H

#### 4.4

##### Ambiente 1: Quarto Vazio

No primeiro experimento, o robô encontra-se em um quarto vazio e deve percorrer uma trajetória circular em um total de cinco voltas.

A Figura 43 mostra a trajetória feita pelo robô real em azul, a estimada pelo LMS-H em vermelha e a Estimativa Simples em verde. As retas em amarelo são as retas mapeadas pelo LMS-H.

A Figura 44 mostra o erro ao longo do tempo para a estimativa da posição do robô. A curva em verde é a Estimativa Simples e a curva em vermelho a estimativa utilizando o LMS-H.

As Figura 45 e Figura 46 mostram os mesmos erros cometidos em  $x$  e  $y$  pelo modelo proposto em vermelho e em preto o intervalo de confiança de 99%. Na maior parte do tempo a estimativa do LMS-H está dentro do intervalo de confiança.

Como esperado, as estimações com bússola e *encoders* tiveram um bom desempenho, contudo, mesmo com a bússola, o acúmulo do erro do *encoder* é inevitável.

Depois do ambiente mapeado de forma correta, o processo de localização via Filtro de Kalman Estendido impede que os erros na estimativa da pose do robô aumentem. O mapeamento do ambiente quando bem realizado garante uma boa localização. Mas, se as primeiras retas mapeadas estiverem muito dissonantes com o ambiente real, é provável que todo o processo de localização e mapeamento seja comprometido.

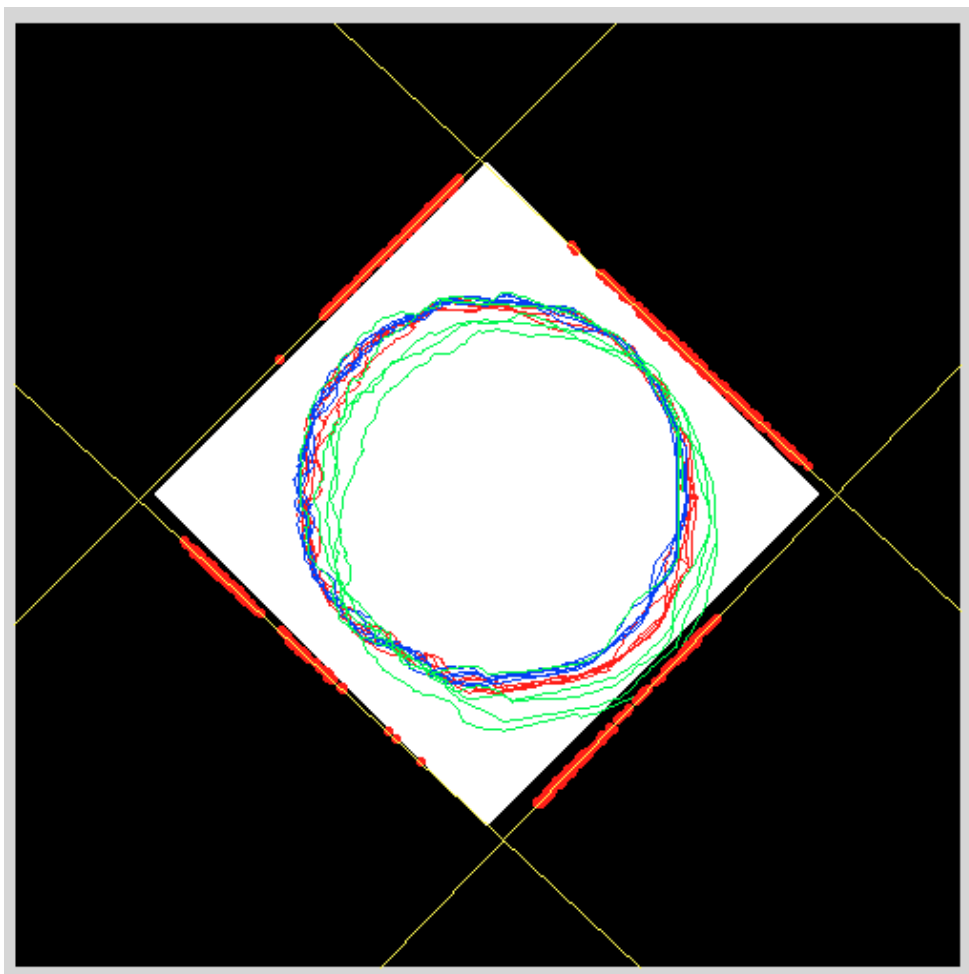


Figura 43 - Trajetórias das estimativas via LMS-H (em vermelho) e Estimativa Simples (em verde) e do robô real (em azul) para o ambiente: quarto vazio.

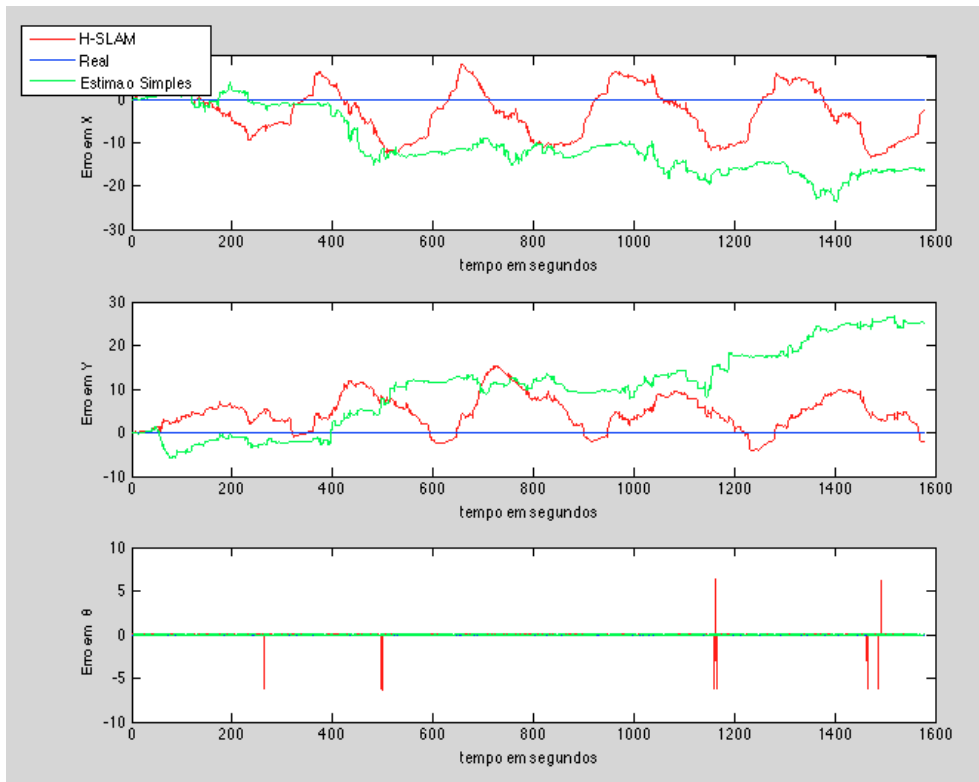


Figura 44 - Erros em X, Y e  $\theta$  da posição do robô no quarto vazio.

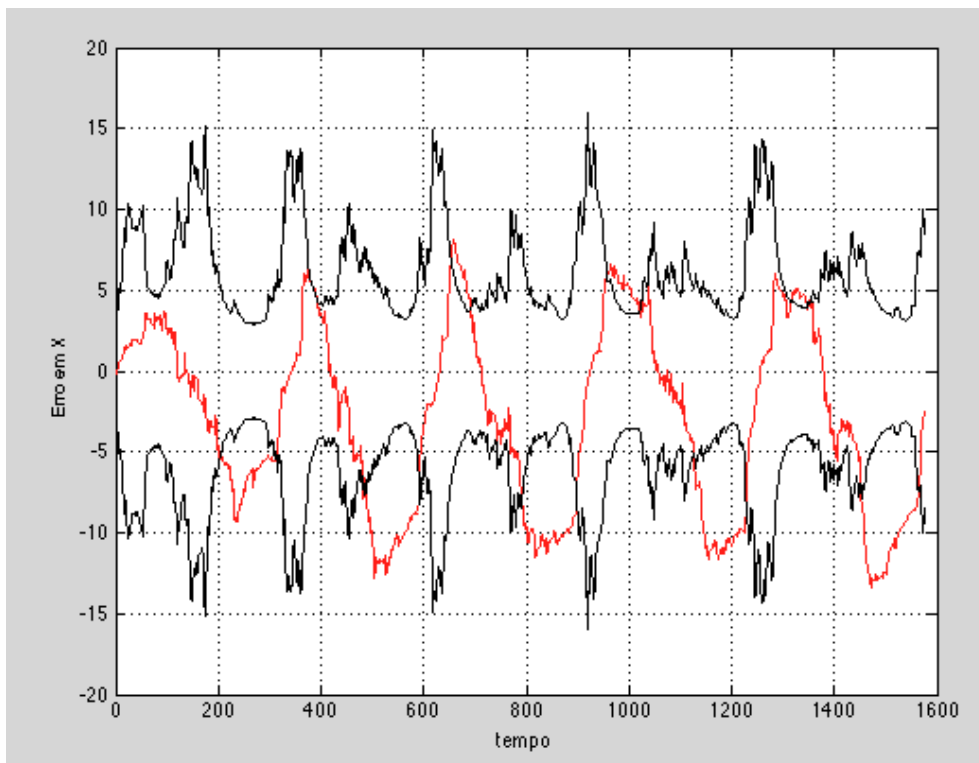


Figura 45 - Erro do modelo proposto em X com o intervalo de confiança de 99% no quarto vazio.

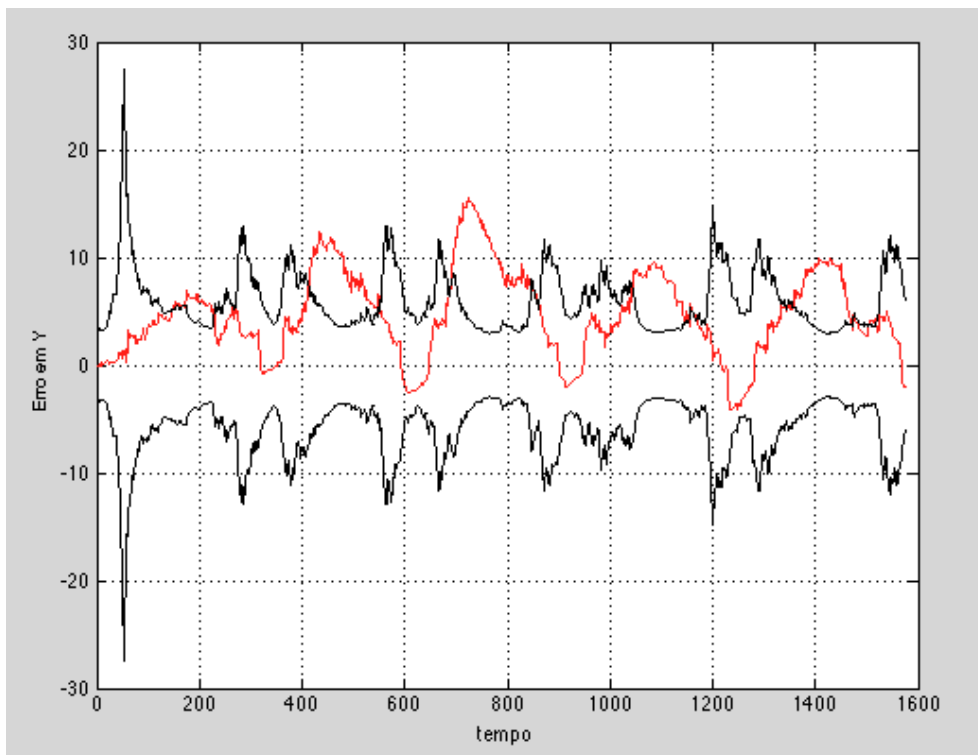


Figura 46 - Erro em Y do LMS-H com o intervalo de confiança de 99% no quarto vazio.

A Figura 47 mostra a distância euclidiana das estimações simples (em verde) e pelo LMS-H (em vermelho). Pode-se verificar o crescente erro da estimativa simples. O erro com o LMS-H não é crescente, pois ele aumenta nos locais onde o ambiente foi mapeado com um erro maior e decresce nos locais onde o mapeamento está melhor.

Quando um mesmo local é observado pelo menos três vezes, o valor do mapa de ocupação em grade é usualmente superior a 0,5, em locais ocupados e inferior a  $-0,5$  em locais vazios. Esses valores, portanto, são boas estimativas para classificarmos as células em vazias ou ocupadas.

O mapa final de ocupação em grade pode ser observado na Figura 48. Se for aplicado o limiar superior de 0,5, é possível observar as regiões ocupadas, como mostrado na Figura 49, onde a região em branco representa a área ocupada e as regiões em preto são regiões “desconhecidas” em relação à ocupação.

Como o robô fez a trajetória apenas no sentido anti-horário, pequenas regiões de “sombra” foram criadas, impossibilitando o robô de detectar todo o plano, entretanto o mapa de retas foi capaz de mapear o ambiente por completo.

Ao aplicar o limiar inferior de  $-0,5$  tem-se a região do ambiente desocupada, no caso, toda a região interna do quarto, como mostrado na Figura

50. Como o ambiente não possui móveis ou obstáculos, o robô foi capaz de mapear bem, praticamente toda a região interna como desocupada.

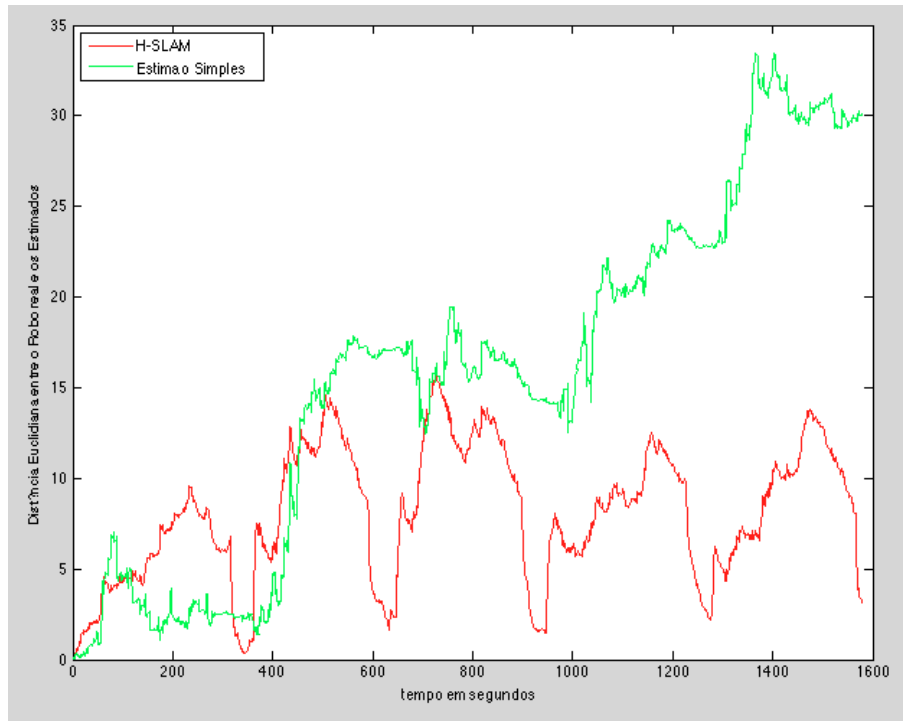


Figura 47 - Erro euclidiano das estimativas no quarto vazio.

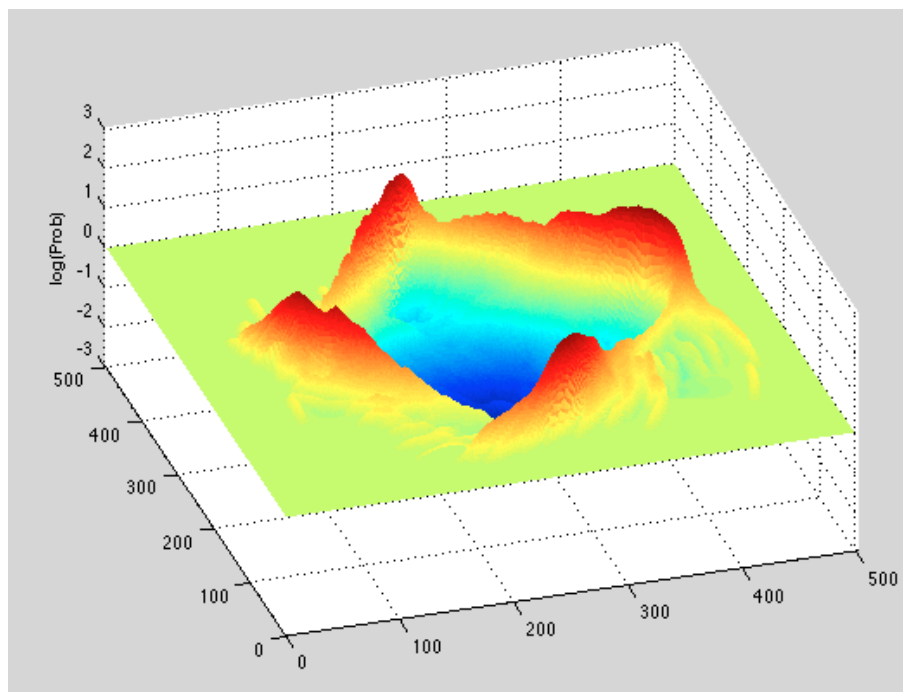


Figura 48 - Mapa de ocupação em grade do quarto vazio.

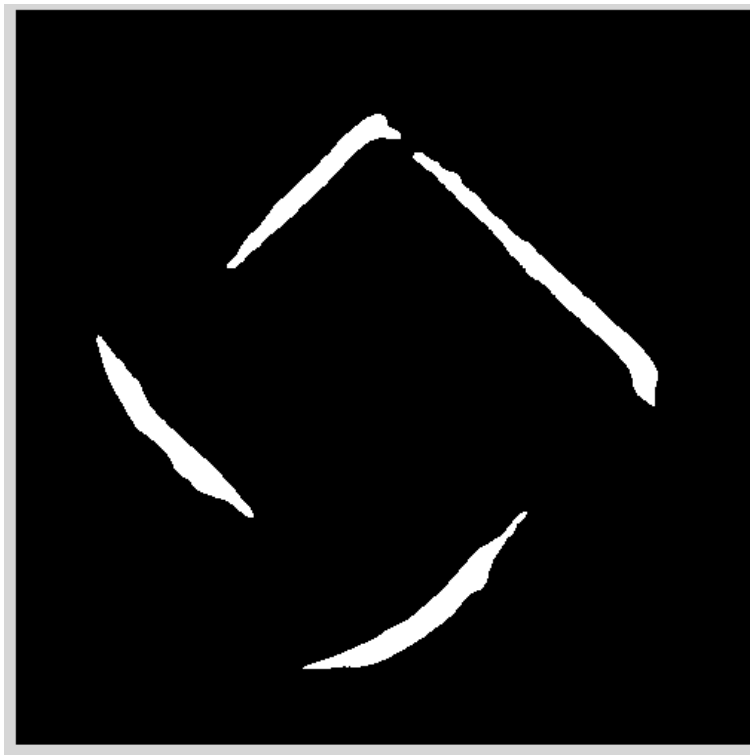


Figura 49 - Região Ocupada para o ambiente quarto vazio. Limiar superior de 0.5.

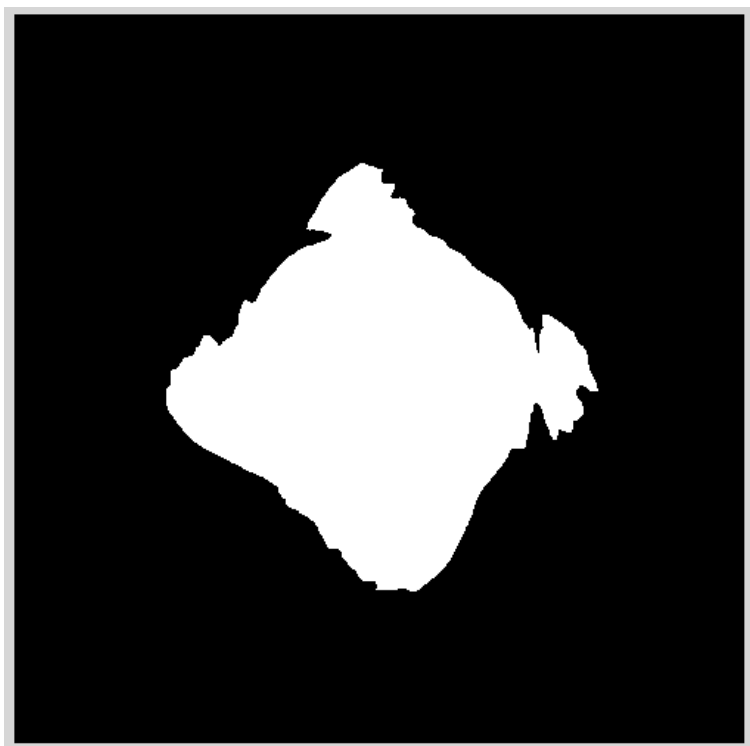


Figura 50 - Região Desocupada para o ambiente quarto vazio. Limiar inferior de -0.5

#### 4.5 Ambiente 2: Corredor em laço fechado

A trajetória realizada pelo robô neste ambiente compõe-se de oito pontos. Um em cada esquina e um no ponto central de cada corredor. O robô inicia a sua trajetória no ponto central do corredor inferior, ponto 1, e segue no sentido horário, passando pelo ponto no canto inferior esquerdo e assim sucessivamente, dando duas voltas no ambiente. A trajetória descrita acima se encontra numerada na Figura 51.

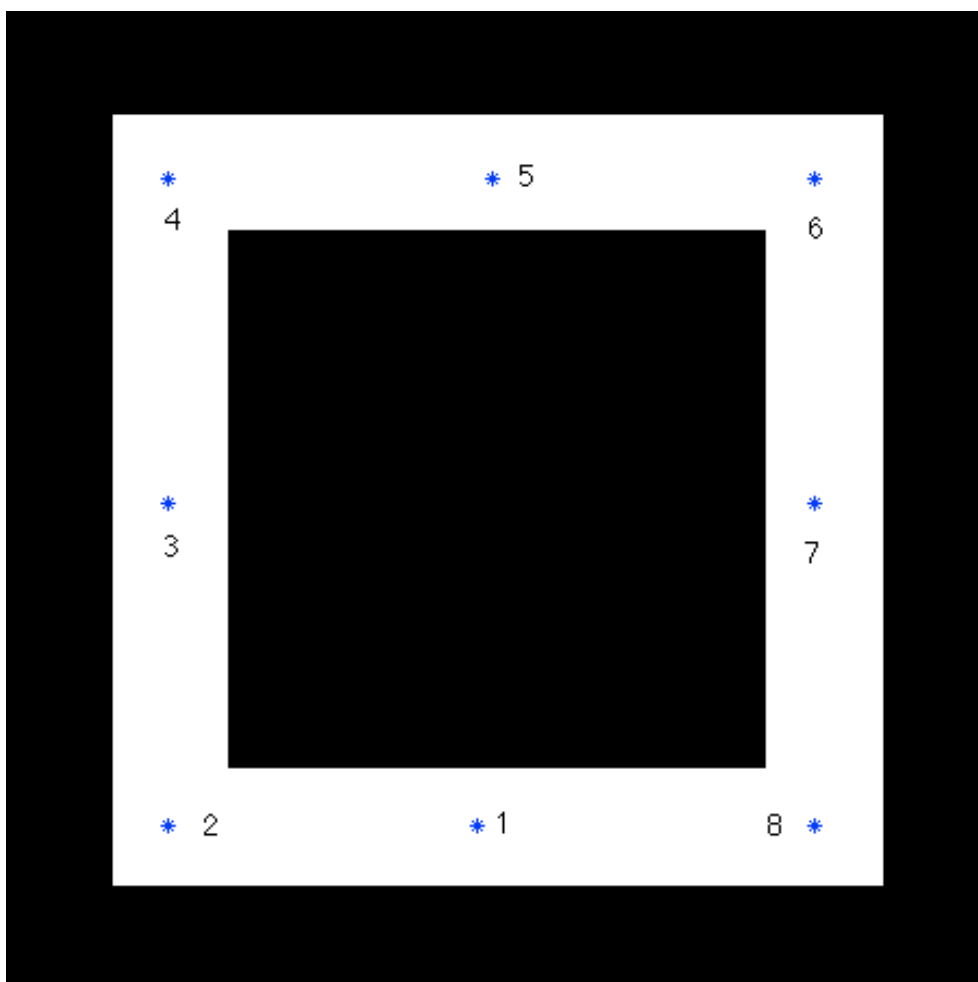


Figura 51 - Trajetória do corredor em laço fechado.

Na Figura 52, a trajetória em verde corresponde à Estimação Simples, a trajetória em vermelho a trajetória com o LMS-H e em azul a trajetória real do robô.

É possível observar que as quatro retas internas não foram mapeadas, isso se dá, principalmente, devido ao sistema de controle utilizado. O sistema de controle utilizado faz um controle bastante rudimentar, corrigindo o erro angular do robô ao próximo objetivo. Quando o erro de orientação estiver dentro de um intervalo

aceitável, então o robô passa a corrigir a distância ao objetivo. Apesar do controle do robô está fora do escopo deste trabalho, é importante mencionar que ele pode privilegiar certas posições do robô, fazendo com que os sonares “vejam“ com mais frequência certos obstáculos, como o que aconteceu neste caso. A correção angular deixava o robô sempre voltado para as paredes à sua esquerda, deixando as paredes à sua direita em um ponto cego do robô.

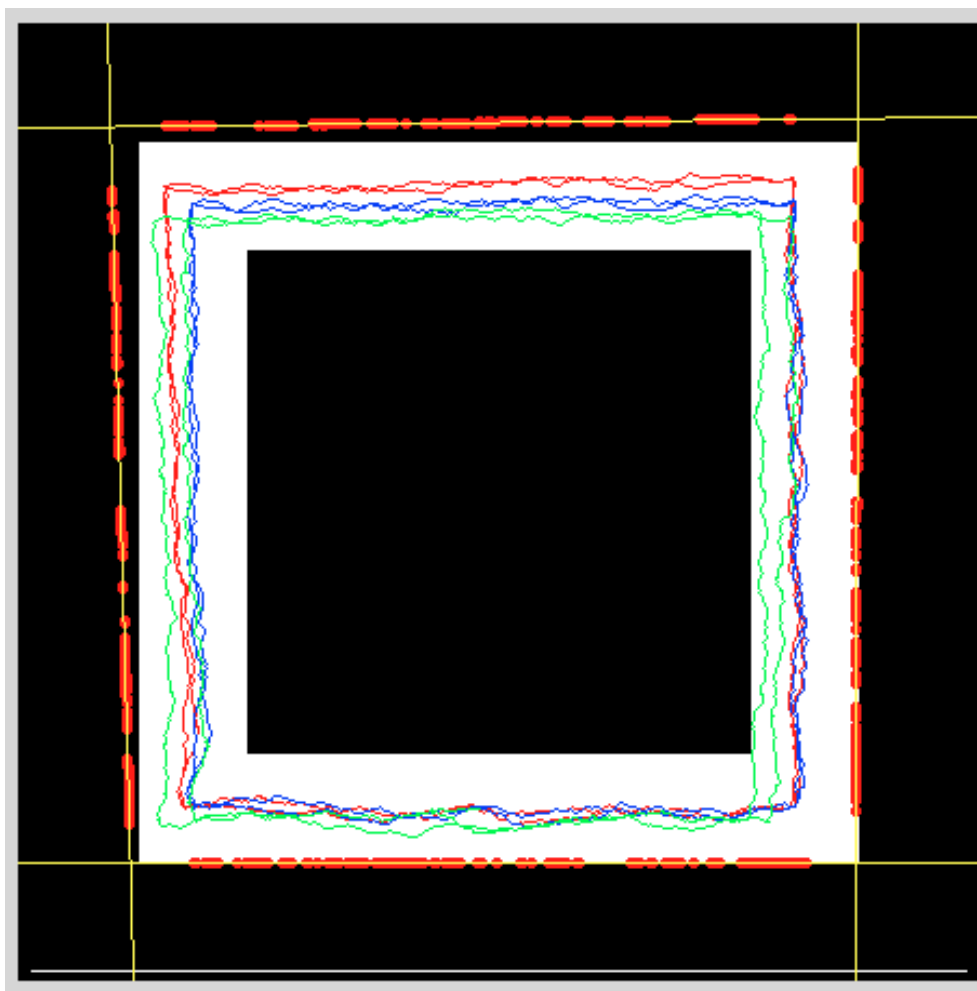


Figura 52 - Trajetórias das estimações via LMS-H (em vermelho) e Estimação Simples (em verde) e do robô real (em azul) para o ambiente: corredor em laço fechado.

Na Figura 53 são apresentados os erros em  $x$ ,  $y$  e  $\theta$  do robô ao longo da trajetória. As Figura 54 e Figura 55 mostram os erros nos eixos  $x$  e  $y$  e o intervalo de confiança de 99%.

É possível observar em alguns momentos que a estimação via LMS-H está fora do intervalo de confiança. Esse comportamento irá ocorrer sempre que as retas forem estimadas com erros muito grandes. Durante a simulação é possível



observar que nos momentos em que o robô está em locais onde a estimação das retas não é boa, o erro de estimação extrapola o intervalo de confiança.

A Figura 56 mostra a distância euclidiana entre o robô real e a estimação via LMS-H e a Estimação Simples. É possível notar que o erro da estimação simples possui uma tendência sempre ascendente, enquanto o LMS-H cresce somente até um determinado ponto, depois tende a oscilar entre 0 e o erro máximo na segunda volta.

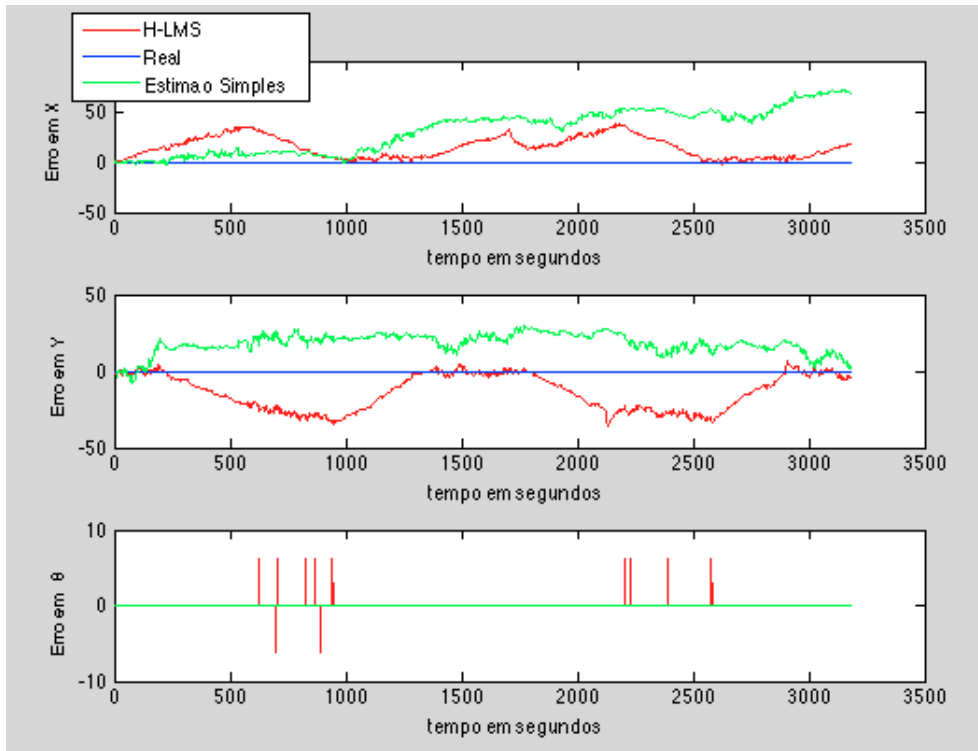


Figura 53 - Erros no eixo X, Y e  $\theta$  do robô no corredor em *Loop*.

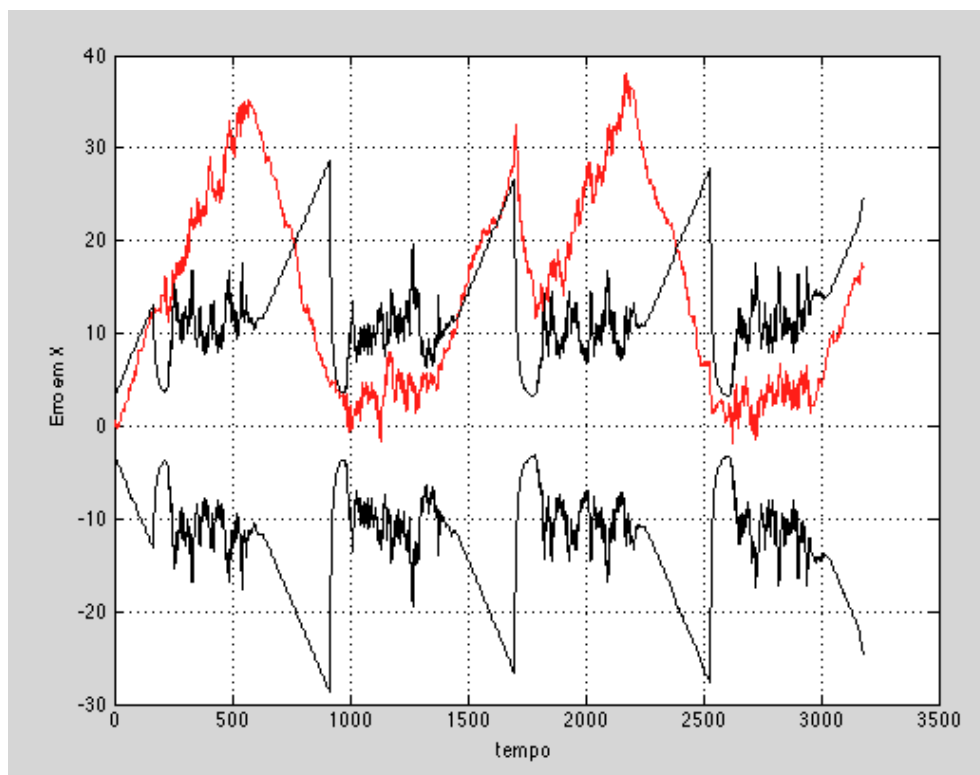


Figura 54 - Erro em X e o intervalo de confiança de 99% no no corredor em *Loop*

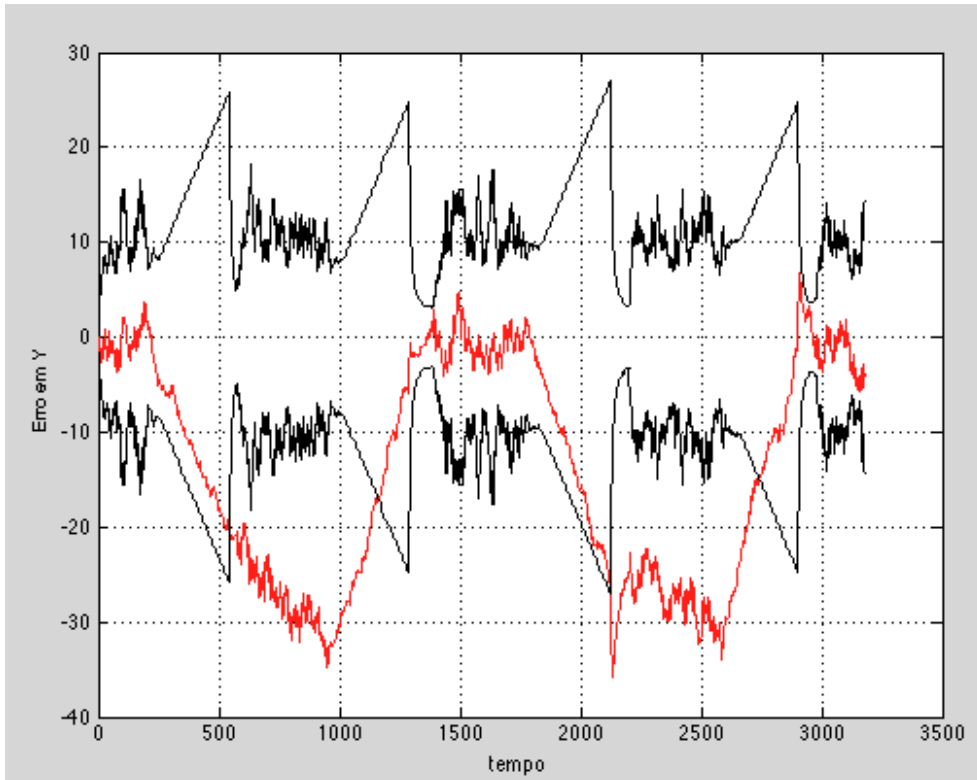


Figura 55 - Erro no eixo Y e a região de confiança de 99% no corredor em *Loop*.

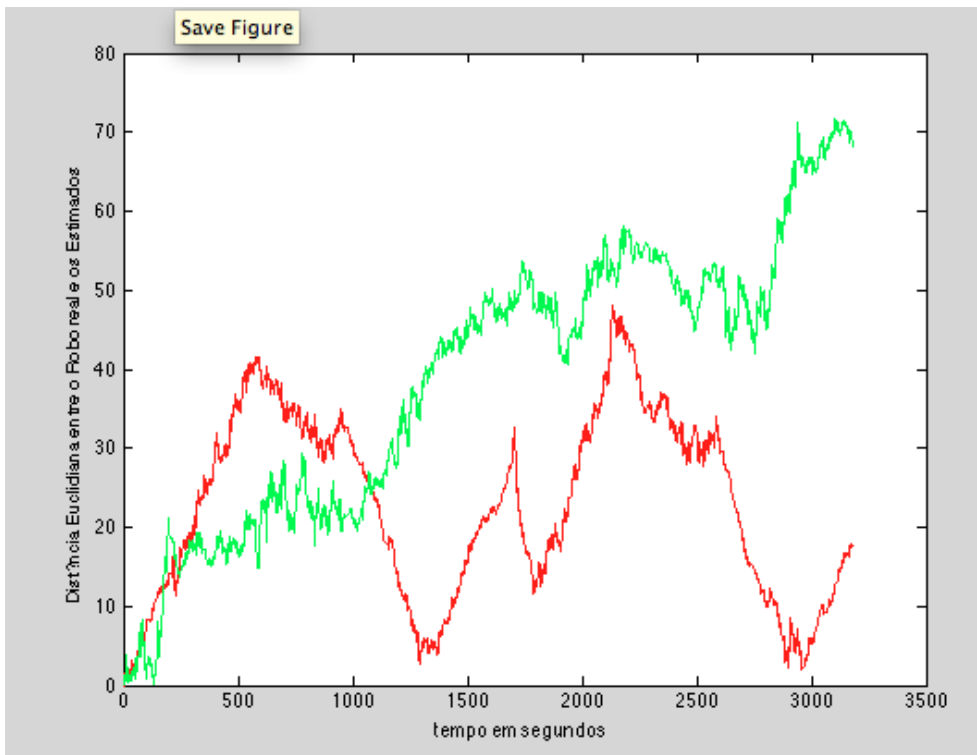


Figura 56 - Erro euclidiano entre o robô real, o robô estimado via LMS-H e a Estimação Simples no corredor em *Loop*.

Este tipo de ambiente é particularmente desafiador para qualquer robô que consiga apenas extrair apenas retas como marcos, independente do tipo de sensor

que estiver utilizado. Isso porque, quando o robô está se deslocando entre duas retas paralelas não há como corrigir o erro gerado ao longo do deslocamento.

Outra dificuldade que o modelo proposto encontra em longos corredores é o crescente erro que uma pequena angulação na reta estimada incorpora na estimação da pose do robô com o passar do tempo. A reta, quando inicialmente criada, possui poucos pontos e isso faz com que a mesma desvie do plano real, inclinando-se em relação a ele. Ao acumular mais pontos com o passar do tempo, a reta tende a ficar bastante próxima do plano real, mas dificilmente ficará exatamente igual. Entretanto, este pequeno erro tende a se agravar para planos muito longos, influenciando diretamente na posição do robô, desviando-o de sua trajetória real.

No modelo proposto optou-se por não limitar as retas extraídas. Este critério possui vantagens e desvantagens. Uma reta ilimitada pode ser erroneamente percebida em um ponto do mapa onde o plano que a gerou não esteja presente naquele local. Entretanto, existem casos em que uma reta sem limite pode ser interessante. Quando o robô ainda não visitou um local pelo qual a reta, outrora encontrada, se estende, como é o caso deste ambiente, há um grande ganho na estimação do robô quando. Como o robô iniciou a trajetória no centro do corredor inferior do mapa, ao dar a volta e entrar no corredor novamente pôde observar e corrigir a sua posição pela reta extraída anteriormente, mesmo sem ter ainda visitado o local.

A Figura 57 mostra o logaritmo dos pesos do mapa de ocupação em grade. Como se pode observar, é possível distinguir claramente os corredores em que o robô fez seu trajeto. Se for aplicado um limiar máximo e mínimo para o mapa de ocupação é possível fazer uma distinção ainda melhor entre as regiões ocupadas e desocupadas. A Figura 58 mostra em branco a região com valores maior ou igual a 0,5 e na Figura 59 toda a região em branco possui valor abaixo de  $-0,5$ . É importante atentar ao fato que locais não contemplados pelo sonar não podem ser classificados nem como ocupado nem como vazio.

No mapa em grade de ocupação é possível observar que tanto o mapa da região vazia como o da região ocupada estão inclinados segundo as retas da Figura 52, o que mostra que o robô de fato utilizou as retas estimadas para localizar-se.

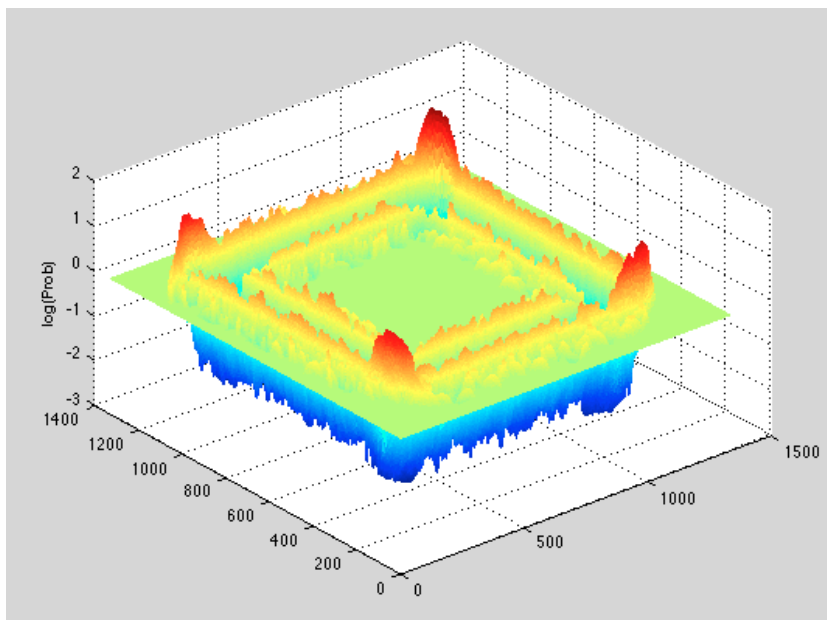


Figura 57 - Mapa de ocupação em grade do ambiente do corredor em *Loop*.

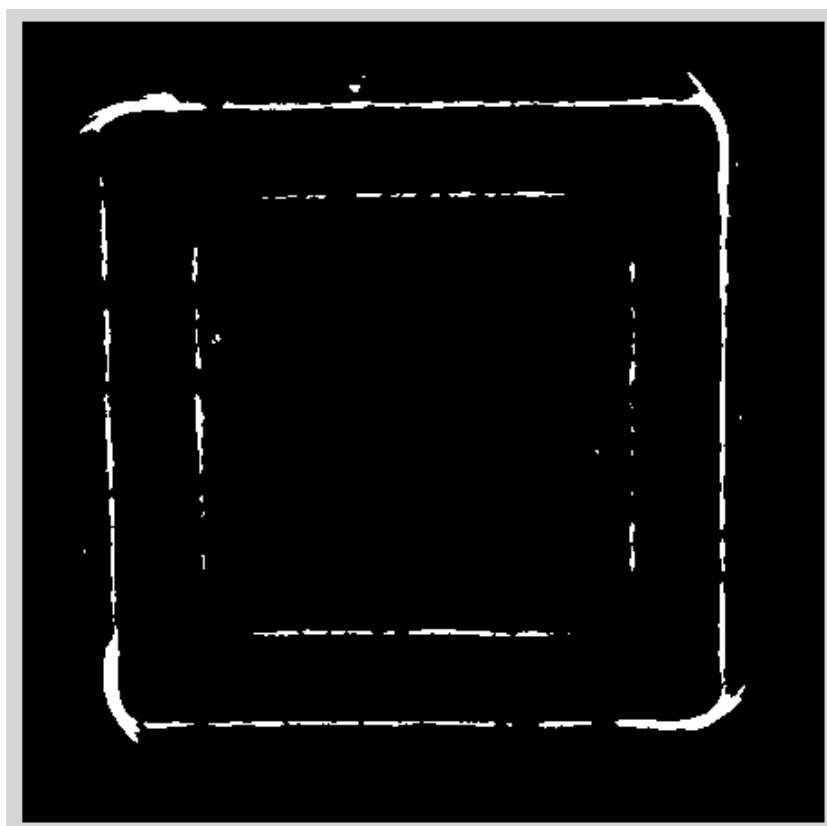


Figura 58 - Região ocupada no ambiente do corredor em *Loop*. Limiar superior de 0.5.

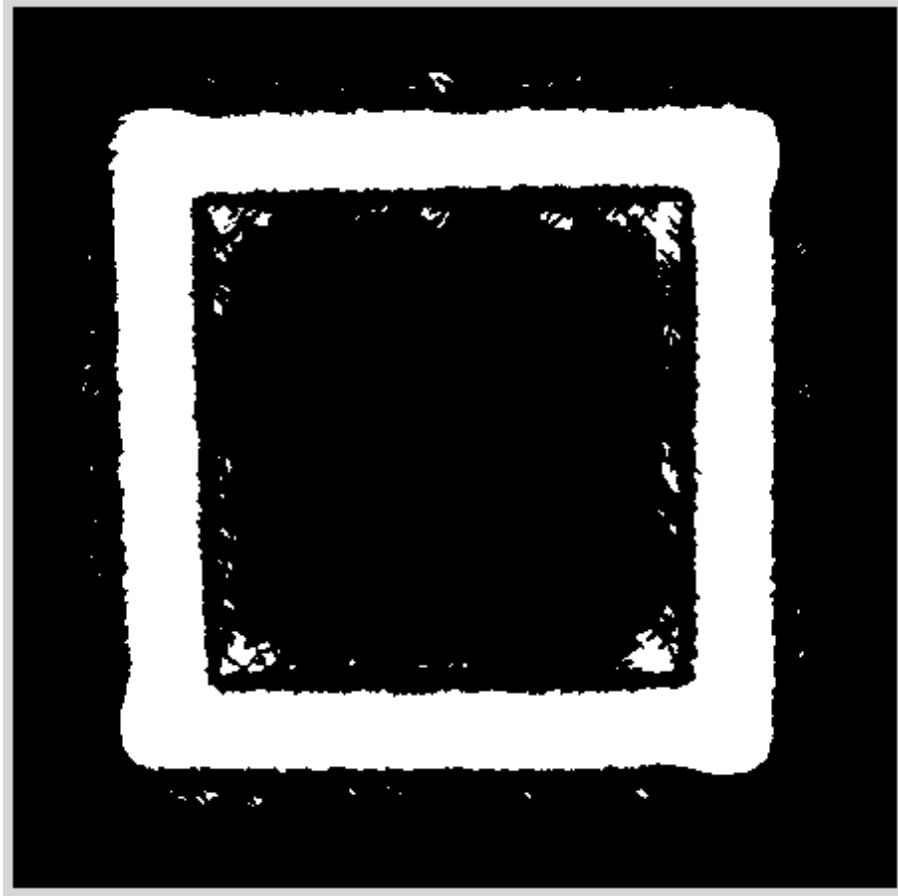


Figura 59 - Região vazia no ambiente do corredor em *Loop*. Limiar inferior de  $-0.5$ .

#### 4.6

##### **Ambiente 3: Escritório**

Este ambiente é o mais complexo dentro os ambientes simulados, é o que possui mais planos e diversos cantos. Outra particularidade deste modelo é a presença de paredes bastante finas, o que é comum em escritórios.

A trajetória realizada pelo robô é mostrada na Figura 60, o robô inicia no ponto 1, localizado no canto esquerdo abaixo no mapa, e segue para os pontos em ordem crescente, dando duas voltas no ambiente.

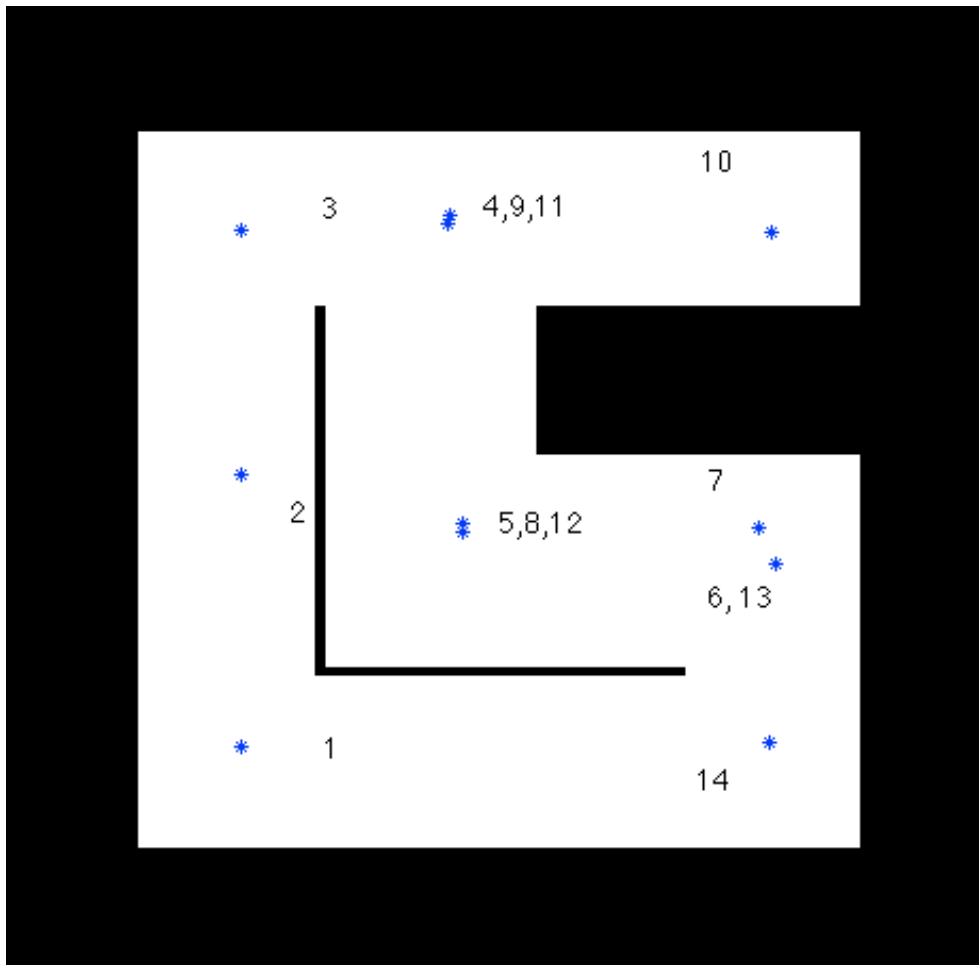


Figura 60 - Trajetória do Robô no ambiente Escritório

O mapeamento e a trajetória podem ser encontrados na Figura 61. Como se pode observar, grande parte dos planos do ambiente foram mapeados corretamente.

Outro fator importante é ver como as paredes finas foram mapeadas como uma única reta. Apesar de não ser o ideal, o mapeamento dessas superfícies com apenas uma reta é conveniente, uma vez que o sistema como um todo não teria acurácia o suficiente para detectá-las como duas retas distintas.

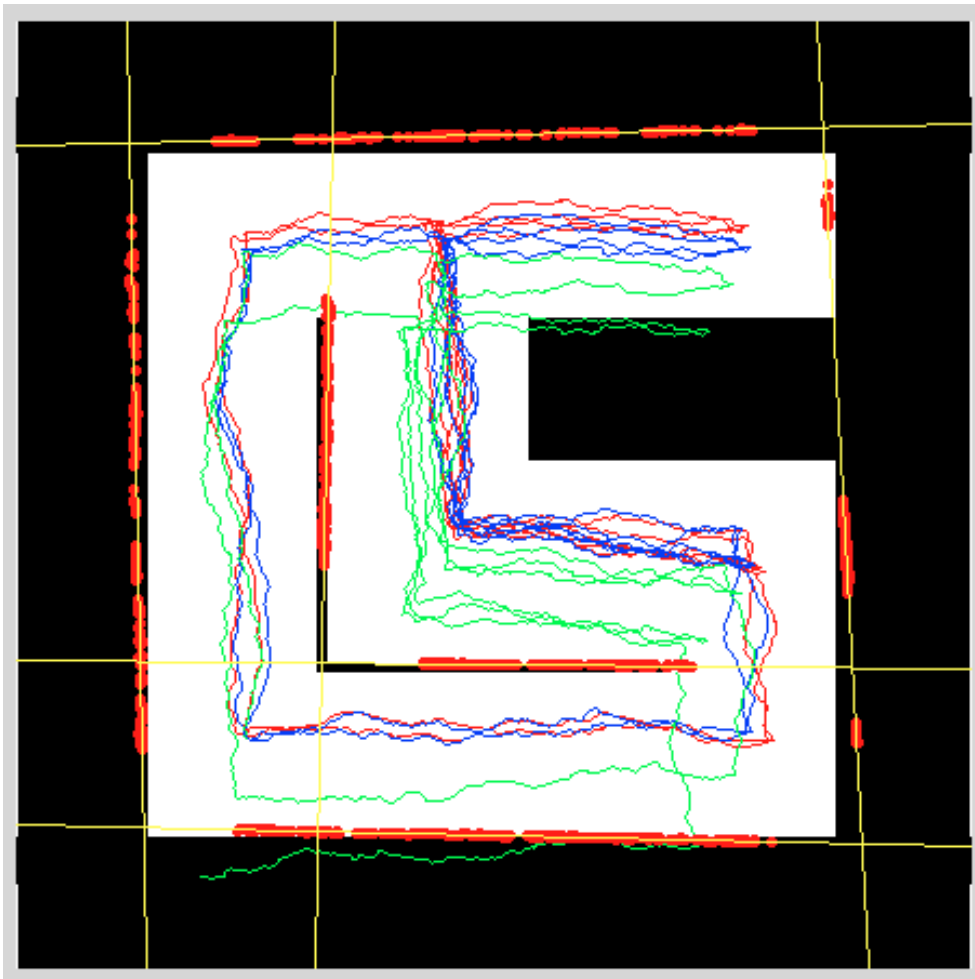


Figura 61 - Trajetórias das estimações via LMS-H (em vermelho) e Estimação Simples (em verde) e do robô real (em azul) para o ambiente: escritório.

Os erros de estimação podem ser vistos na Figura 62. Como se pode observar, o LMS-H manteve um erro bastante pequeno durante as duas voltas, isso se dá principalmente devido à estimação inicial correta das paredes do ambiente. Outro fator importante é notar como o erro da estimação simples aumenta, enquanto o erro via LMS-H permanece pequeno, principalmente na segunda volta.



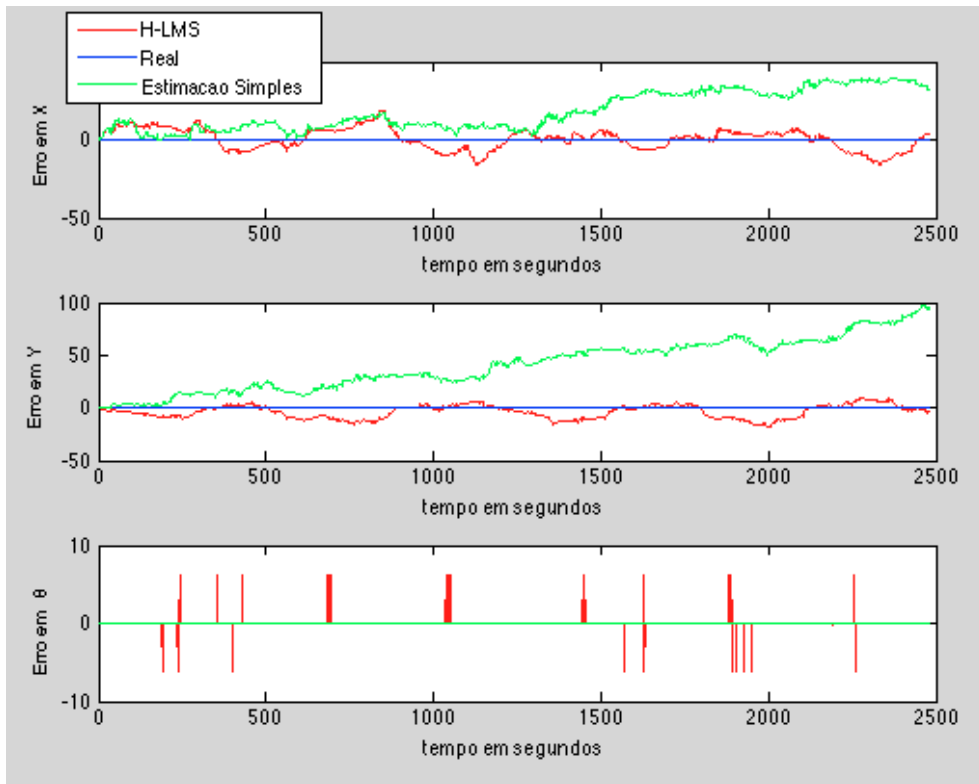


Figura 62 - Erro no eixo X, Y, e,  $\theta$  no ambiente Escritório

Os erros em X e Y permaneceram, em grande parte da trajetória, dentro do intervalo de confiança, como pode ser observado nas Figura 63 e Figura 64.

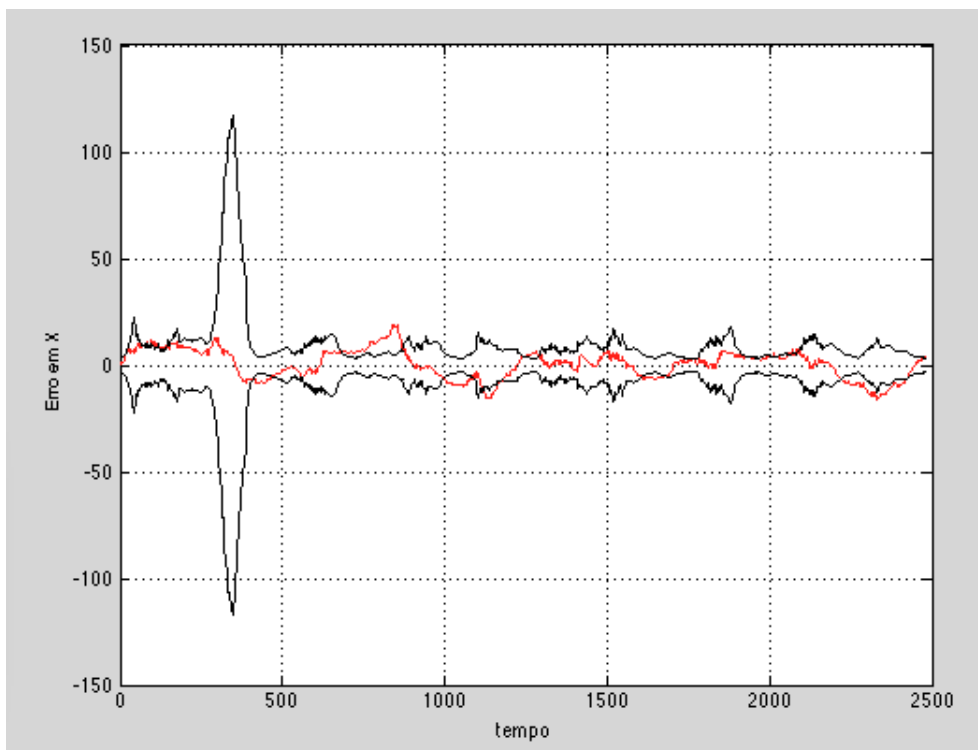


Figura 63 - Erro em X com o intervalo de confiança de 95%.

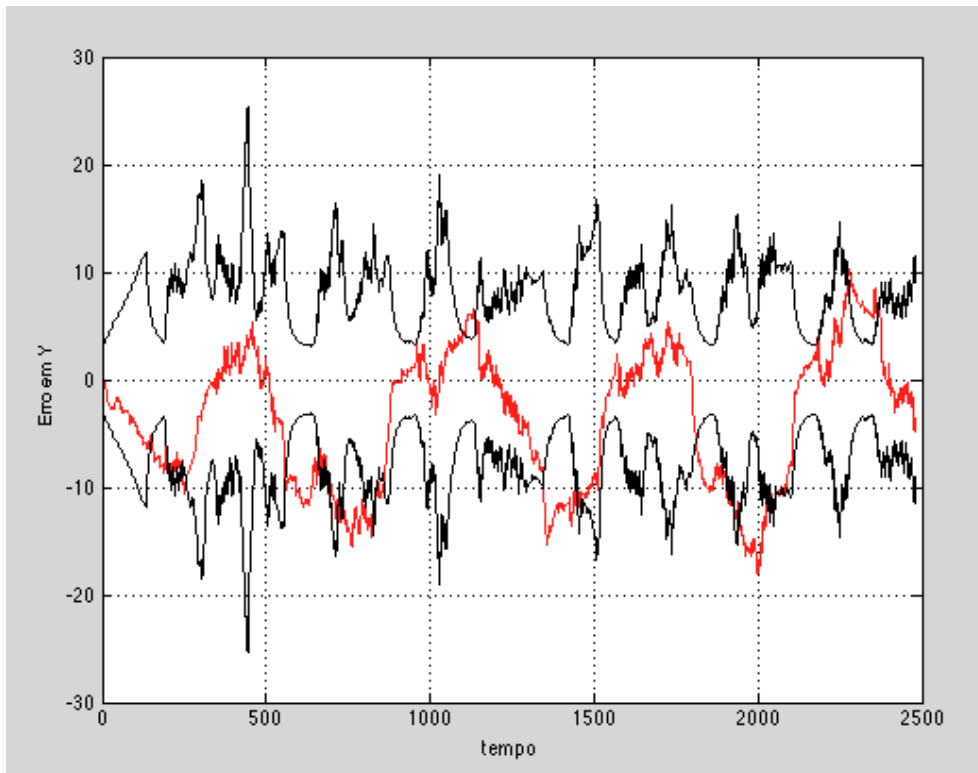


Figura 64 - Erro em Y com o intervalo de confiança de 95%.

Apesar de três retas não terem sido observadas diretamente, é possível observá-las de forma indireta no mapa de ocupação em grade, como mostrado na Figura 65 e Figura 67, entretanto as mesmas não aparecem bem definidas quando se aplica um limiar de 0,5 para as regiões ocupadas. Isso explica porque o algoritmo não foi capaz de mapear estas retas, pois os pontos extraídos nestes locais nunca foram considerados em locais ocupados.

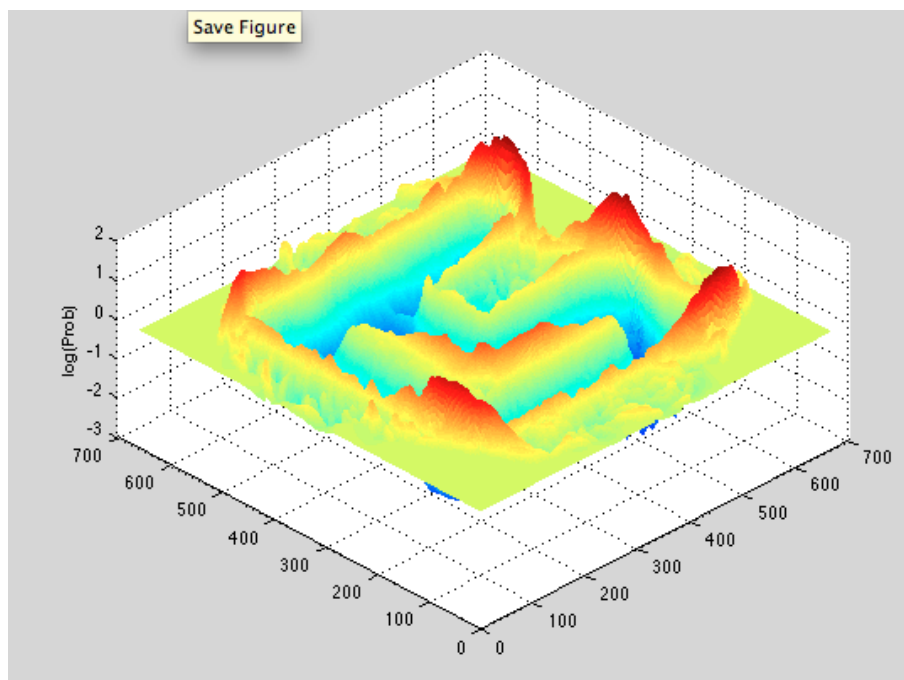


Figura 65 - Mapa de ocupação em grade para o ambiente 3.

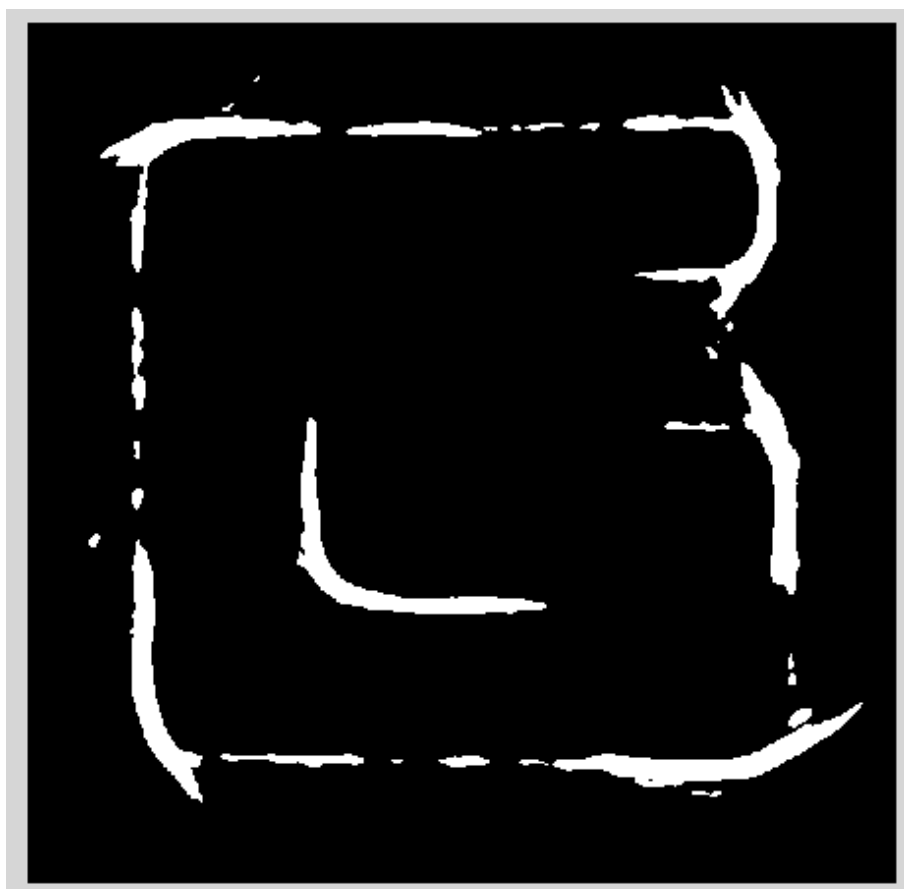


Figura 66 - Região mapeada como ocupada em branco pelo LMS-H no ambiente Escritório. Limiar superior de 0.5.

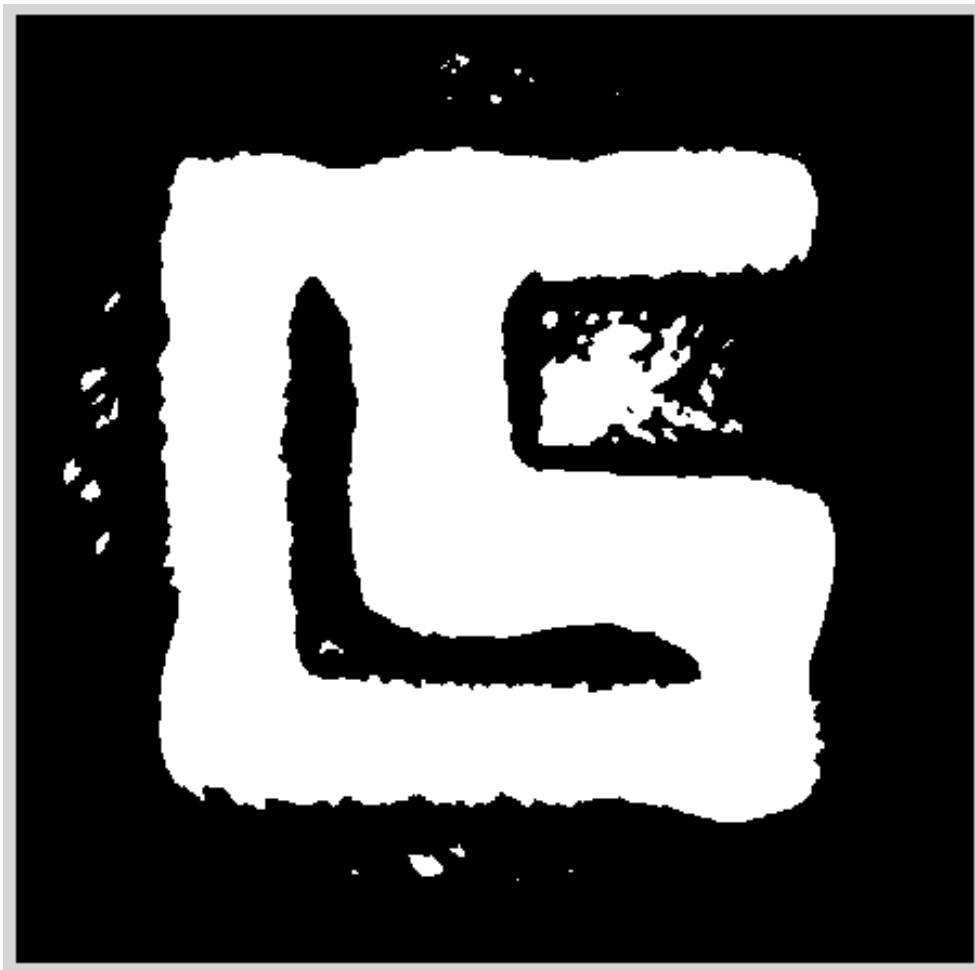


Figura 67 - Região mapeada como desocupada em branco pelo LMS-H no ambiente Escritório. Limiar inferior de -0.5.

Em termos de localização este mapa apresentou os melhores resultados, mesmo sendo o mapa teoricamente mais complexo. Na prática, todavia, o que faz com que o robô localize-se de forma eficiente é a possibilidade do mesmo detectar pelo menos duas retas não paralelas. Como este ambiente possui uma grande diversidade de retas não perpendiculares, e não possuem corredores longos, o robô é capaz de localizar-se bem.

Outro fator preponderante para a boa localização do robô neste ambiente está relacionado ao percurso realizado. Em determinada parte do percurso o robô faz uma pequena volta, visitando novamente o ambiente que acabou de passar, mas agora na direção oposta. Essa volta dá a possibilidade do robô retornar a um local recém mapeado, corrigindo o erro antes de visitar locais ainda não mapeados.

Na Figura 68, é mostrada a distância euclidiana entre o robô real e os estimados. Neste gráfico é bastante visível o crescente erro da estimativa simples, e como o erro do LMS-H permanece abaixo dos 20 cm.

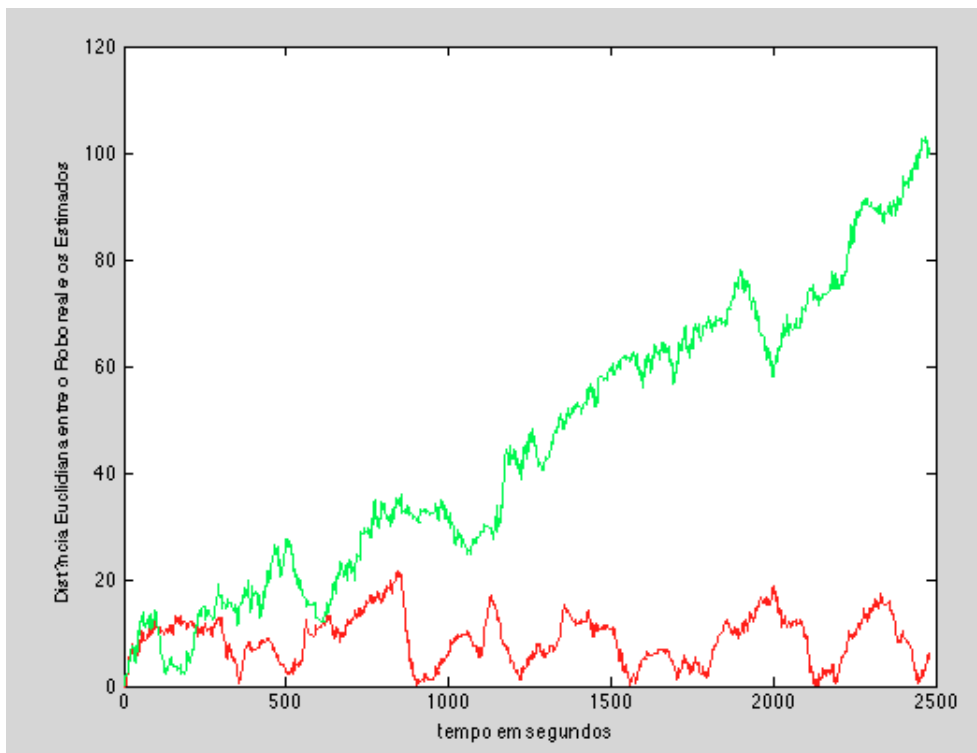


Figura 68 - Erro euclidiano do robô no ambiente Escritório.

#### 4.7

#### Ambiente 4: Ensaio em um ambiente real em “L”

Para realizar os ensaios com o robô real, o mesmo foi controlado manualmente via um link de rádio *bluetooth*. O controle manual do robô permite que se conheça a trajetória do mesmo, sem que haja um sistema de rastreamento em tempo real da sua posição. Isso nos fornece uma trajetória aproximada do percurso feito.

Na Figura 69, é mostrado o ambiente real em “L”. Os pontos em azul são as posições percorridas pelo robô na ordem em que os números aparecem, realizando a trajetória apenas uma vez.

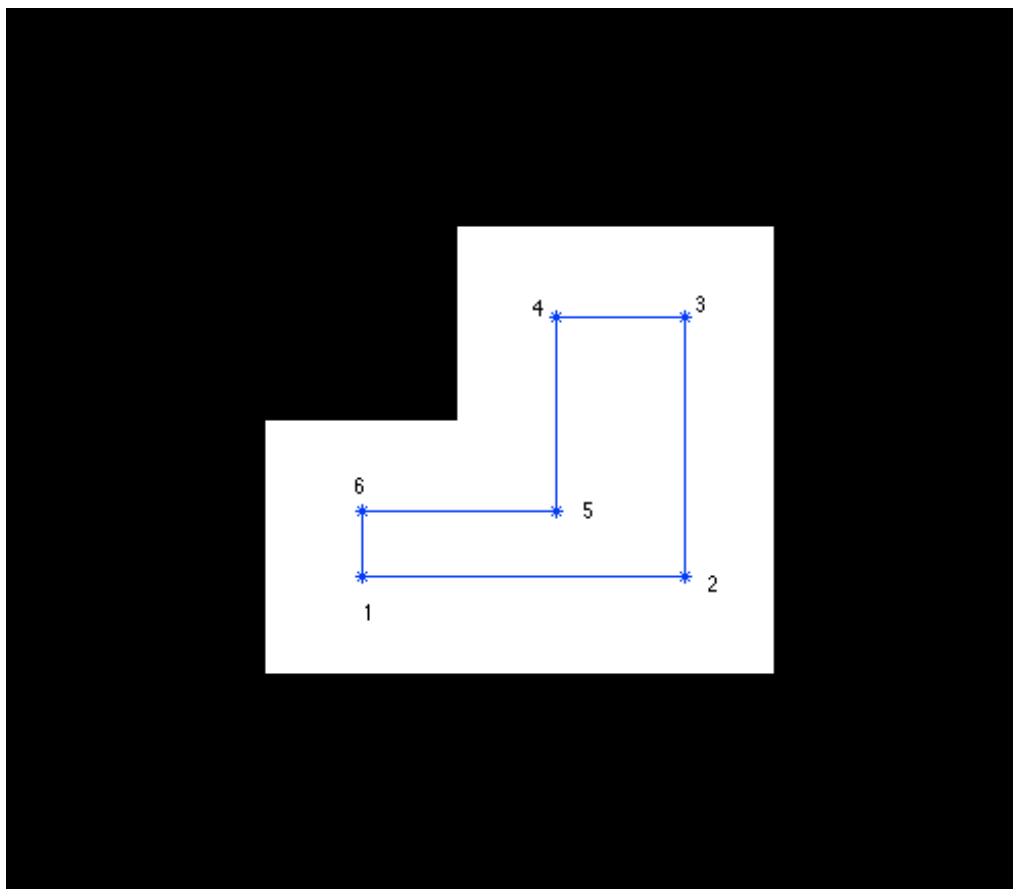


Figura 69 - Ambiente real em "L"

Na Figura 70, é possível notar que a estimação simples é capaz de fornecer uma boa estimação no início da trajetória, mas no final do trajeto seu erro já é bastante grande. Já a posição estimada via LMS-H possui um erro grande no início, e aproximando-se muito do local de onde o robô iniciou o trajeto no final da trajetória.

É possível também observar que o LMS-H foi capaz de mapear todas as paredes do ambiente, mesmo que algumas ainda com um erro considerável.

No mapa de ocupação em grade, mostrado na Figura 71, é possível observar com bastante clareza o ambiente, assim como na Figura 72.

As regiões desocupadas da Figura 73 apresentam uma grande quantidade de falhas em locais que estão desocupados. Isso ocorre por causa da grande quantidade de leituras errôneas do sonar. Ao analisar a quantidade de leituras ruins do sonar, percebe-se o porquê de tantas falhas, a Figura 74 mostra todos os pontos extraídos erroneamente das leituras dos sonares e que não foram utilizados. A grande quantidade de pontos no centro do mapa mostra a quantidade de ruído oriundo dos sonares e a causa da grande quantidade de lacunas nas região

desocupada. Uma forma de preencher estas lacunas seria revisitar o local novamente.

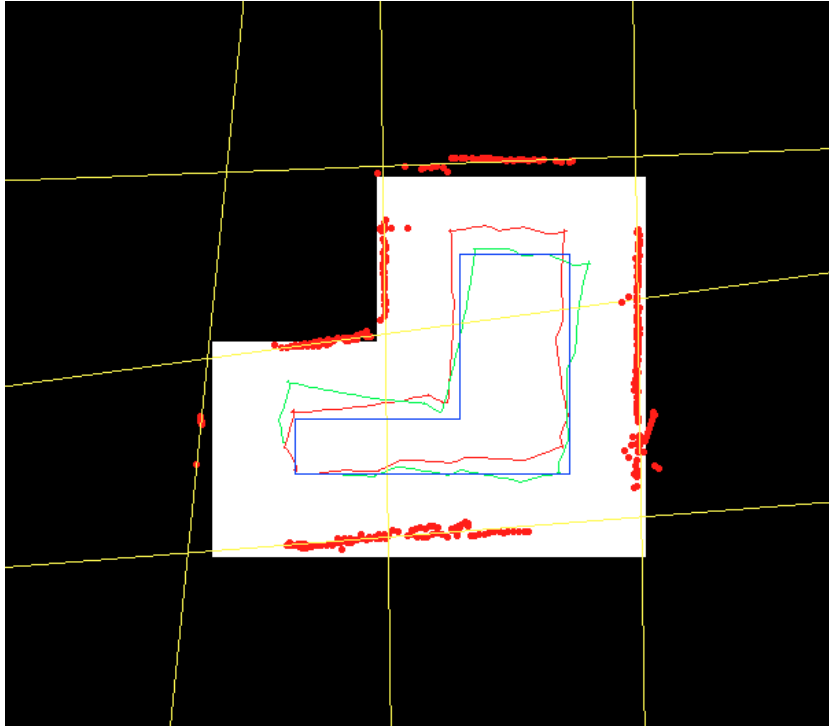


Figura 70 – Mapeamento e Trajetória no ambiente real em "L".

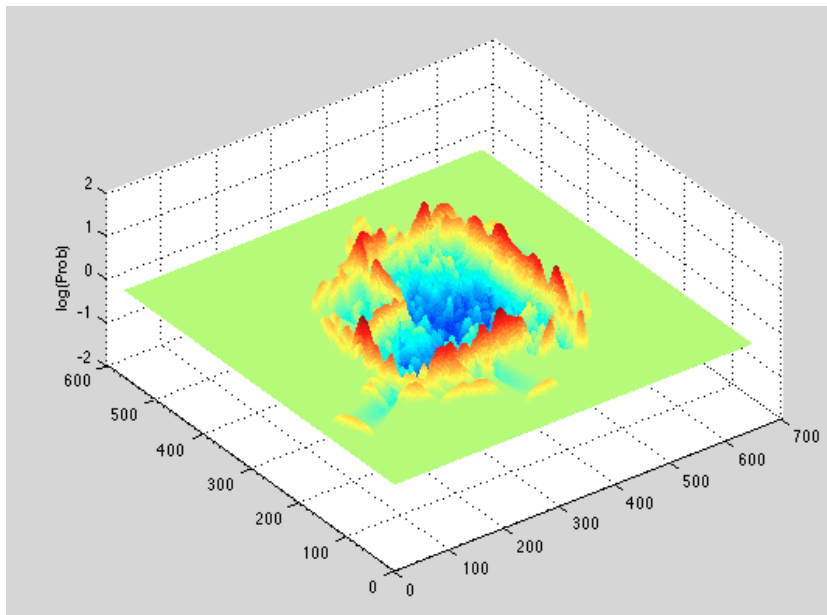


Figura 71 - Grade de ocupação para ambiente real em "L"

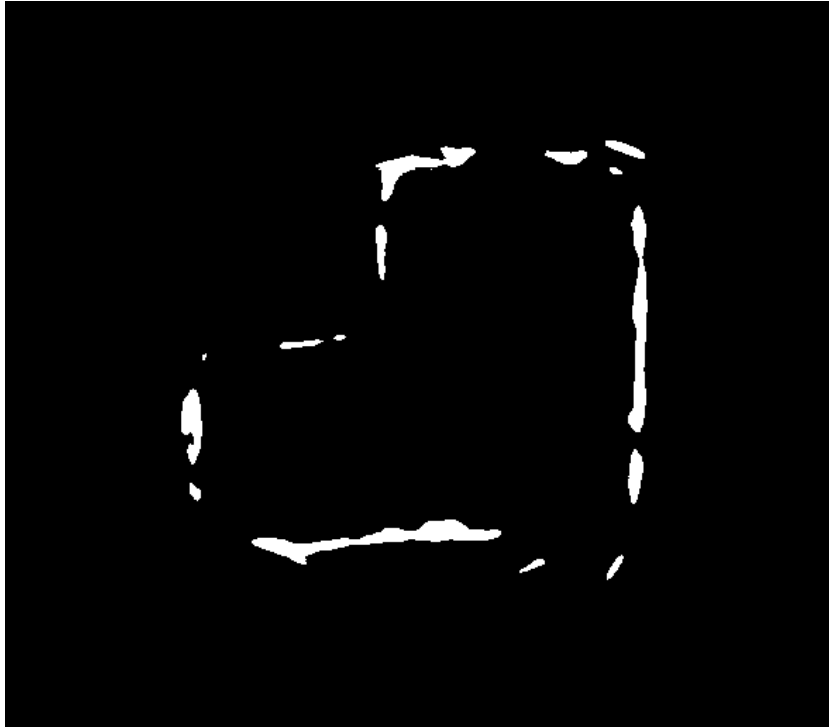


Figura 72 – Região ocupada para o ambiente real em "L". Limiar superior de 0.5.

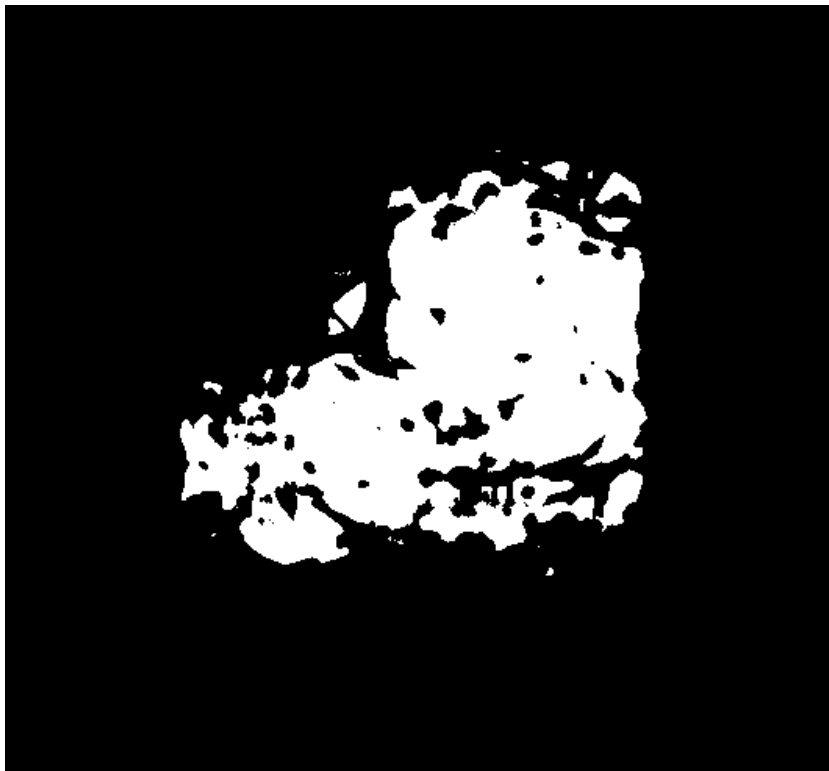


Figura 73 - Região desocupada para o ambiente real em "L". Limiar de -0.5.



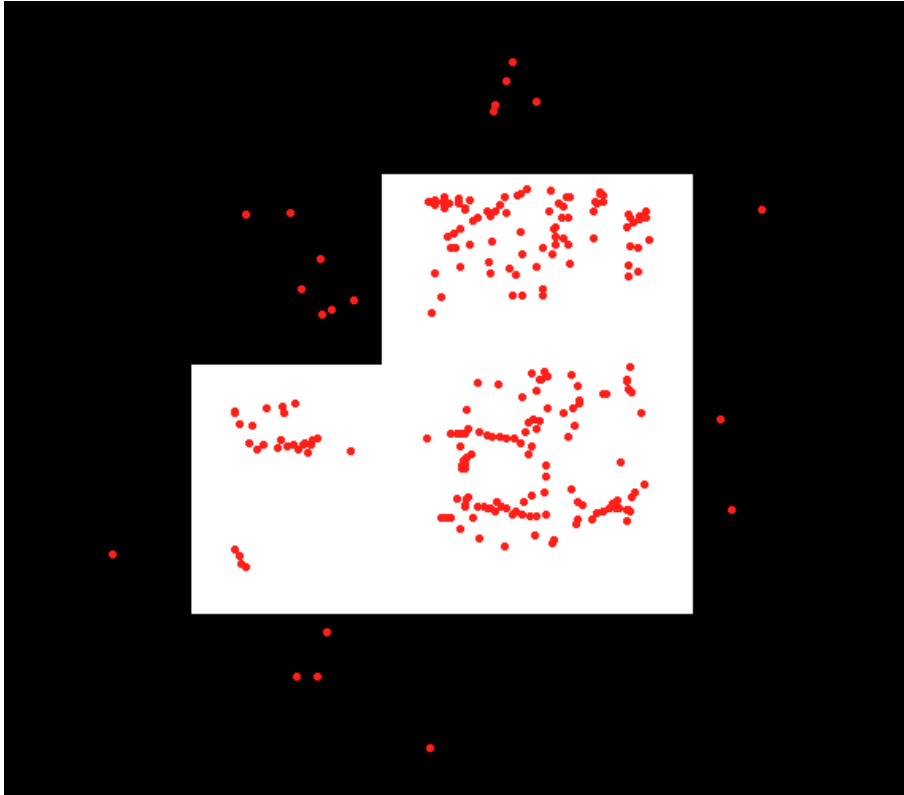


Figura 74 - Todas as leituras do sonar não utilizadas para o ambiente real em "L".

#### 4.8

##### **Ambiente 5: Corredor**

Devido a dificuldade de encontrar um ambiente real parecido ao corredor em laço fechado, foi feito um experimento em um corredor de aproximadamente nove metros de comprimento. As dificuldades enfrentadas são bem parecidas e algumas ficam bastante claras nos resultados apresentados.

Neste ambiente, o robô faz uma trajetória aproximadamente linear, iniciando no lado esquerdo para o lado direito do corredor, como mostra a reta em azul.

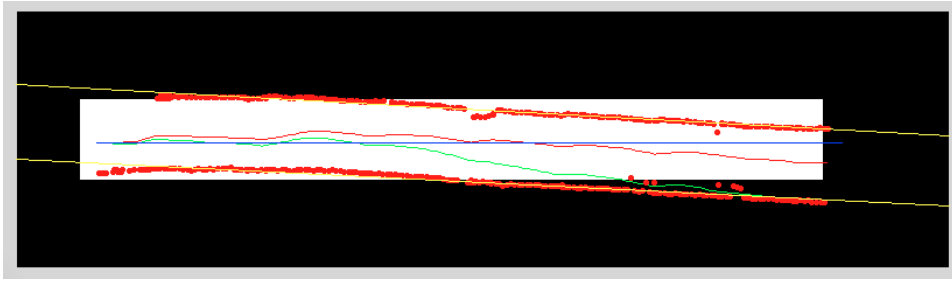


Figura 75 – Mapeamento e Trajetória para o ambiente real Corredor.

A Figura 75 mostra o resultado do mapeamento e as trajetórias das estimções via LMS-S e da estimção simples. Apesar do LMS-H não ter sido capaz de mapear exatamente o ambiente, é possível perceber uma sensível melhora na sua localização em relação à estimção simples.

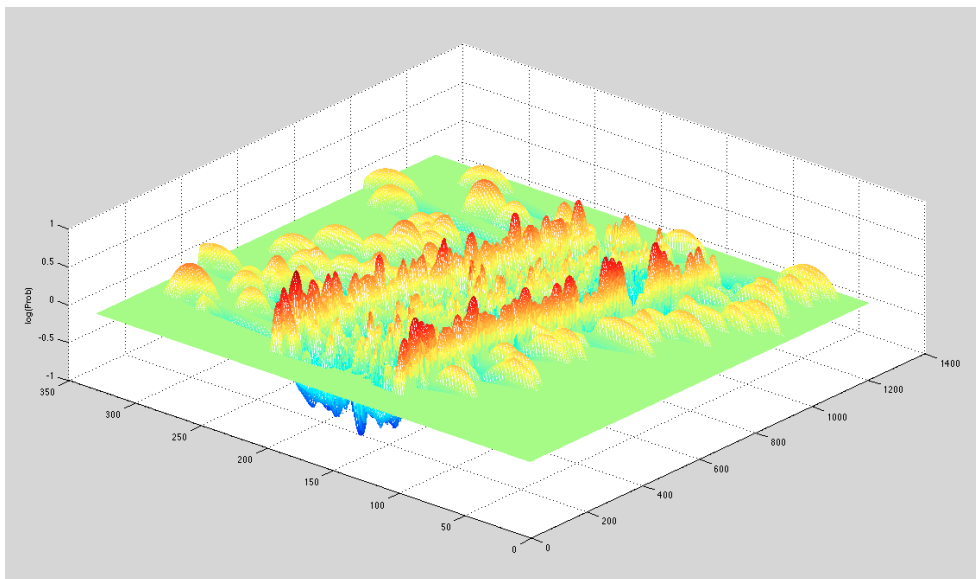


Figura 76 - Grade de ocupação para ambiente real Corredor.

É possível observar na Figura 76 a grande quantidade de leituras com o problema de *specular reflections* que um ambiente como este gera. As *specular reflection* prejudicam de forma sensível a construção do mapa de ocupação em grade, adicionando “buracos“, tanto nas áreas tidas como ocupadas, como se pode observar na Figura 77. Este problema acaba por degenerar o mapa de ocupação em grade. Entretanto, com o mapa de retas é possível distinguir bem o ambiente.



Figura 77 – Região ocupada para o ambiente real Corredor. Limiar superior de 0.5.

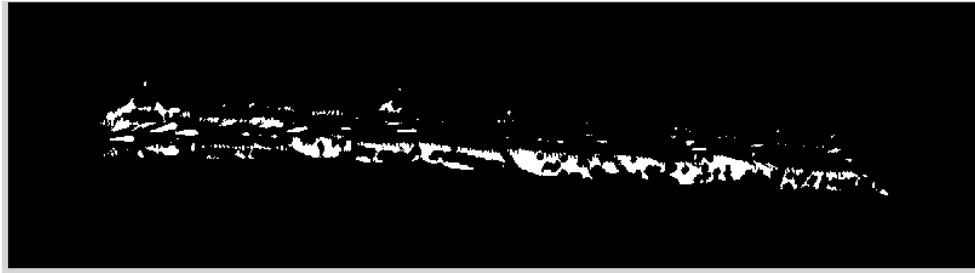


Figura 78 – Região desocupada para o ambiente real Corredor. Limiar inferior de -0.5.

#### 4.9

#### **Ambiente 6: Parte do corredor do Departamento de Engenharia Elétrica da PUC-Rio**

O sexto ambiente (e o terceiro com dados reais) é uma parte do corredor da sala dos professores do Departamento de Engenharia Elétrica da PUC-Rio. O mais interessante neste ambiente são as retas anguladas em  $45^\circ$ .

Como se pode observar na Figura 79, a trajetória realizada pelo robô inicia no ponto 1 no lado esquerdo do mapa. O robô faz uma trajetória no sentido anti-horário, retornando ao mesmo local de partida, ponto 8, no final da trajetória.

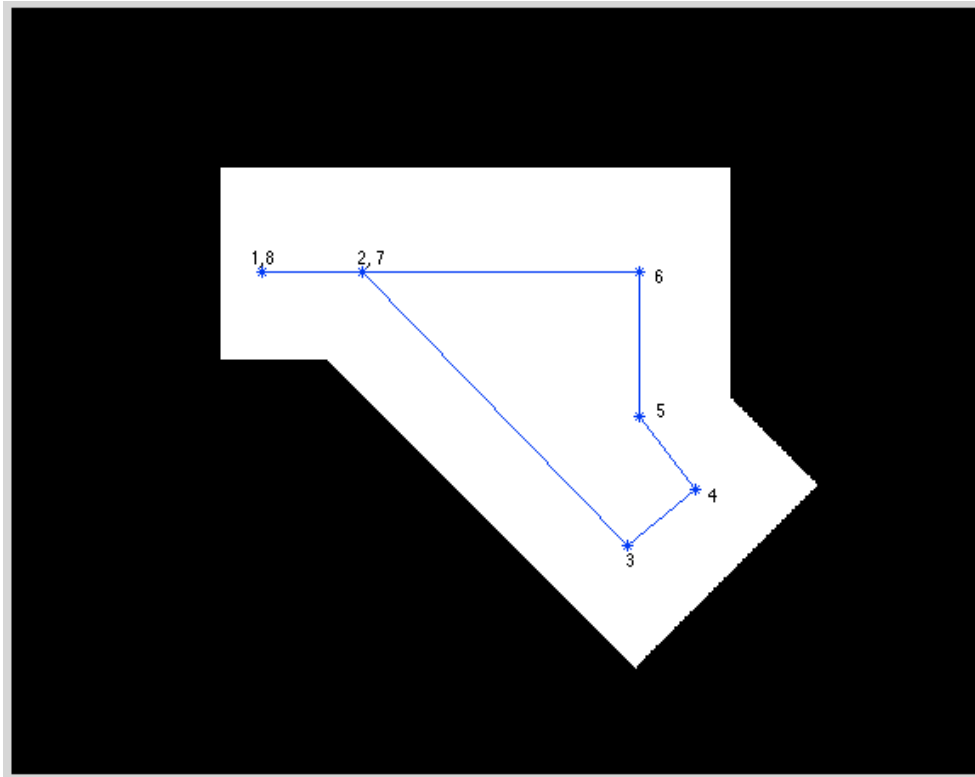


Figura 79 - Trajetória do robô no Ambiente 6

A Figura 80 mostra as retas extraídas pelo algoritmo ao longo do deslocamento do robô em amarelo. A trajetória estimada via LMS-H em vermelho, a trajetória com estimação simples em verde e a trajetória real aproximada em azul.

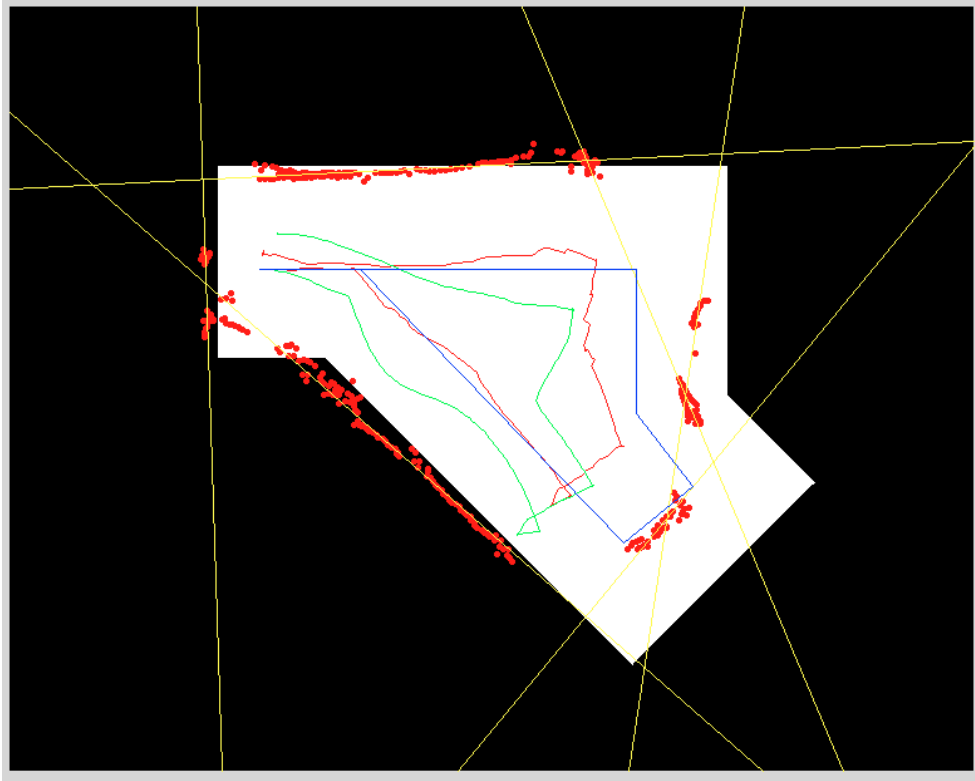


Figura 80 - Retas extraídas do Ambiente 6

A Figura 81 mostra as regiões mapeadas como ocupadas e a Figura 82 as regiões mapeadas como desocupadas.



Figura 81 - Região ocupada para o Ambiente 6. Limiar superior de 0.5

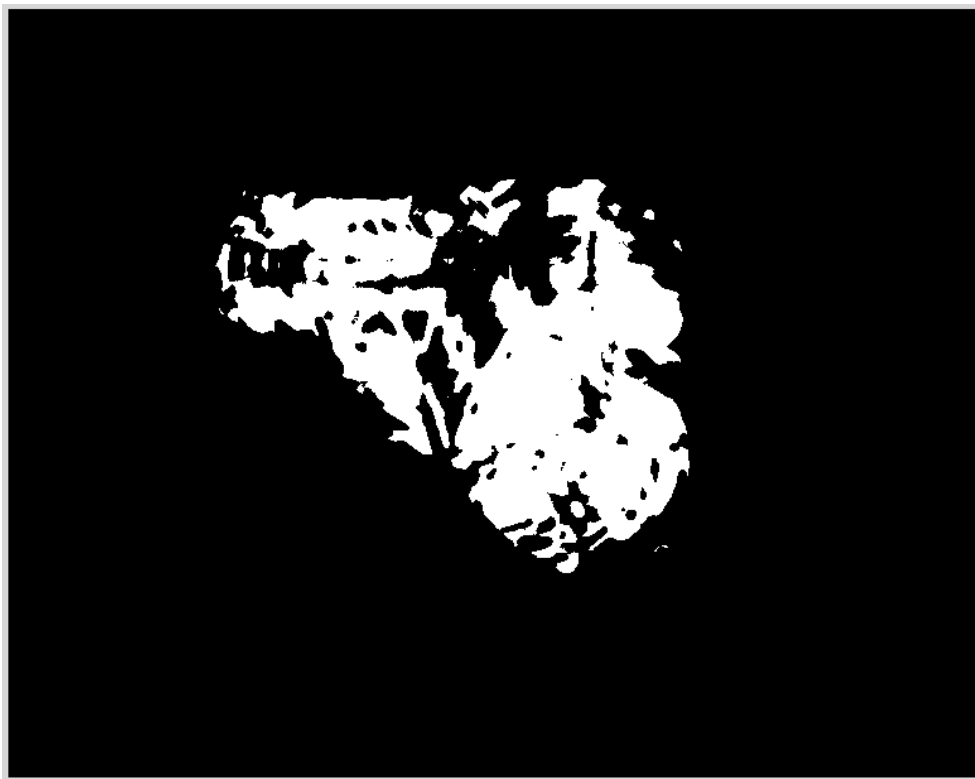


Figura 82 - Região desocupada para o Ambiente 6. Limiar inferior de -0.5

A Figura 83 mostra o mapa de ocupação em grade.

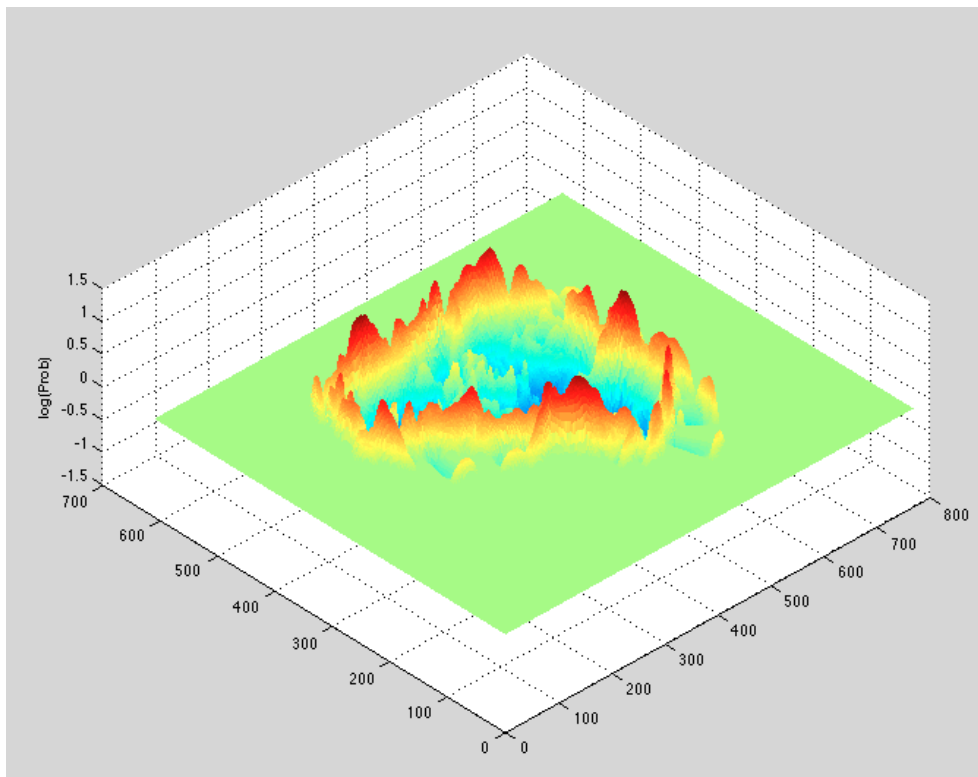


Figura 83 - Grade de ocupação para Ambiente 6.

Este ambiente possui uma característica bastante inconveniente para robôs que detectam retas, principalmente utilizando apenas sonares: ele possui retas anguladas em  $45^\circ$  ou menos. Retas pouco anguladas são difíceis de ser diferenciadas, principalmente quando o robô está próximo ao ponto de contato entre elas.

O mapeamento de duas retas pouco anguladas é ainda mais complicado. No ponto de transição de uma reta para a outra, o robô continua detectando a reta antiga erroneamente e fazendo correções na sua trajetória, e quando já está longe o suficiente para distinguir completamente ambas as retas, a sua estimativa já está bastante prejudicada. Isso acontece neste ambiente quando o robô passa pelos pontos 3, 4, 5 e 6 de sua trajetória, conseguindo retornar a trajetória real apenas quando passa a detectar a parede superior do mapa.

Neste ambiente, mais uma vez, as retas não limitadas tiveram um papel fundamental. Entre os pontos 5 e 6 da trajetória, o robô é capaz de identificar e localizar-se pela reta superior do mapa, mesmo sem antes ter visitado o ambiente naquele local. Como a reta foi mapeada durante os pontos 1 e 2 da trajetória, ela pôde ser vista novamente no ponto 6.

## **4.10 Discussão dos Resultados**

### **4.10.1**

#### **Como o ambiente interfere no Problema de Localização e Mapeamento Simultâneos**

O método mostrou-se promissor em ambientes onde o robô seja capaz de detectar pelo menos dois planos não paralelos em curtos intervalos de tempo. Ambientes com longos corredores, todavia, onde as únicas referências são duas retas paralelas durante um grande intervalo de tempo, são tipicamente problemáticos, não fornecendo informações suficientes para o robô se localizar completamente.

Qualquer método de localização que use apenas retas como referências necessita, em teoria, de pelo menos duas retas não paralelas para localizar-se completamente, como mostrado na Figura 85. A primeira dificuldade encontrada é a falta de informação da posição do robô ao longo do corredor, pois não existe nenhuma referência que diga em que lugar, ou o quanto, o robô já percorreu do corredor, a falta desta informação usualmente resulta em um mapeamento do corredor com comprimento diferente do real. No caso específico do LMS-H, o segundo problema enfrentado é o erro angular da reta com o plano real. Este erro, mesmo que pequeno, terá um efeito considerável no final do corredor, como pode ser observado na Figura 75. Este erro não é tão significativo em ambiente menores, como os ambiente 3 e 4, que além de pequenos contam com várias retas anguladas  $90^\circ$  entre si.

### **4.10.2**

#### **A bússola é realmente necessária?**

Para o método proposto, sim. Ele não irá convergir sem o uso da bússola, e ainda terá um desempenho inferior a estimação simples. Isso porque o sonar é incapaz de informar qual a angulação da reta observada com apenas uma leitura, uma vez que, apenas um ponto do obstáculo é responsável pelo retorno do eco.

Sensores capazes de detectar marcos de forma pontual, por exemplo, a visão computacional, são capazes de fornecerem a distância e o ângulo do robô ao obstáculo. Conhecendo essas duas informações, é possível localizar o robô completamente em  $X$ ,  $Y$  e  $\theta$ , como mostrado na Figura 84.



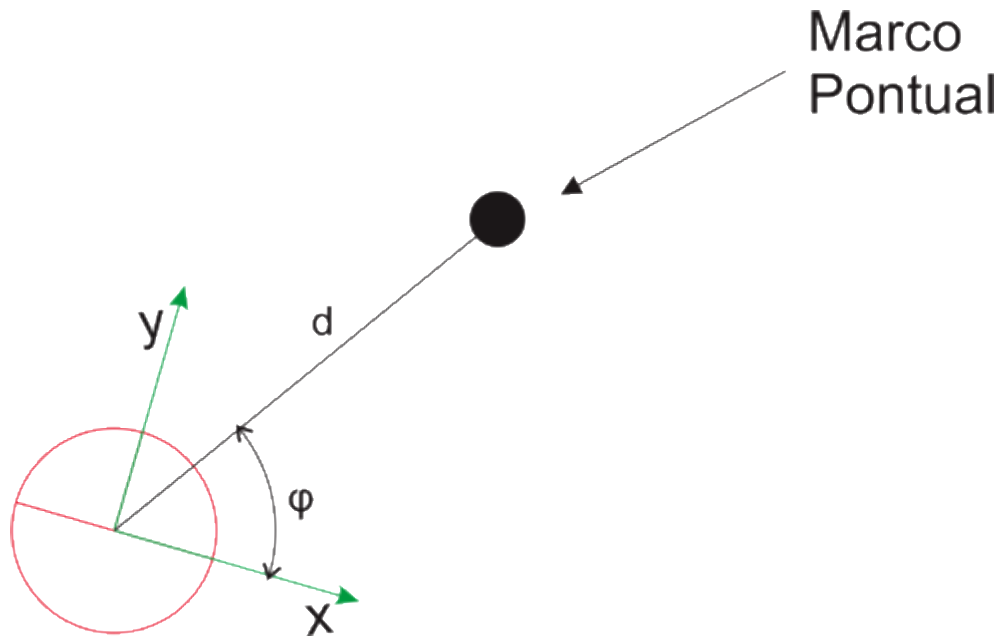


Figura 84 – Localização de um robô conhecendo a distância e o ângulo a um ponto.

Este é o mesmo princípio utilizado por (Lee & Song, 2010) onde, uma vez que uma estrutura saliente é detectada (cantos ou cones), o centro da circunferência serve como um marco pontual. Entretanto, o modelo de (Lee & Song, 2010) também está sujeito a falhas em longos corredores e locais onde não há superfícies salientes. Sendo necessária a inserção de marcos artificiais no ambiente para uma localização mais precisa.

Ao se utilizar um LRF, a cada leitura é possível saber qual o ângulo da reta em relação ao robô, dessa forma, é possível saber o ângulo e a distância à reta. Para sensores *LRF*, teoricamente, duas retas não paralelas já são o suficiente para localizar o robô completamente, como mostrado na Figura 85.

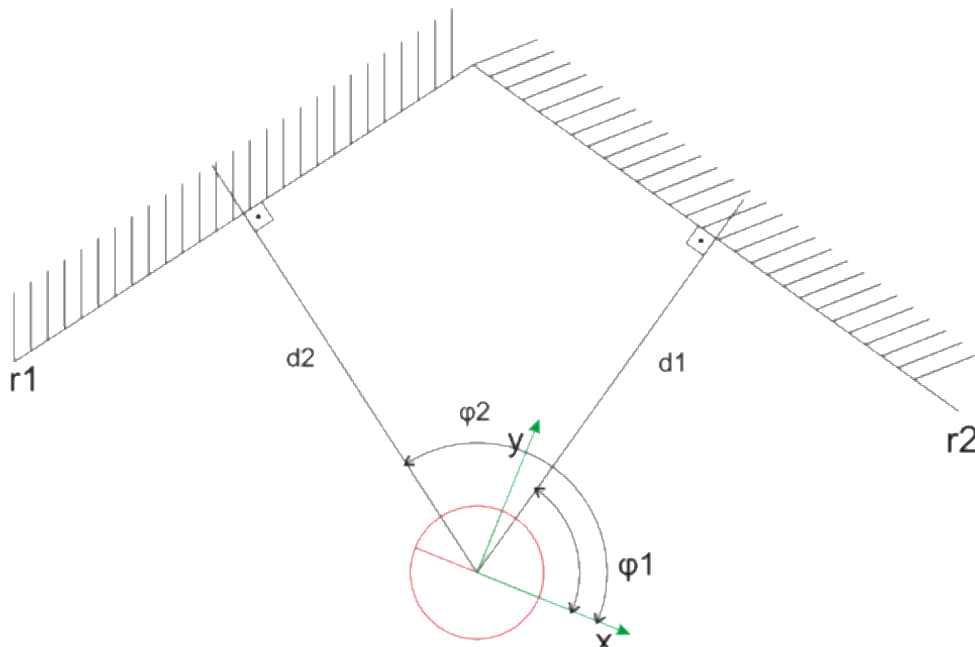


Figura 85 - Localização de um robô através de duas retas não paralelas.

#### 4.10.3

##### Vantagens e Desvantagens de manter dois tipos de mapas

O uso de um mapa de ocupação em grade para filtrar pontos *outliers* melhorou de forma significativa as retas criadas inicialmente, e foi capaz de excluir a grande maioria dos pontos gerados por *specular reflections* ou outras formas de ruído inerentes ao sonar.

Por outro lado, é computacionalmente custoso manter e calcular o mapa de ocupação em grade. Isso se agrava com o aumento da granulação do ambiente, ou seja, quanto menor for o tamanho da célula, mais custoso fica o cálculo dos pesos. Nos testes realizados foram usados células de 1 *cm* de arestas, pois essa é a resolução do sonar. Esta granulação pode ser utilizada em locais pequenos e para sonares com alcances menores. Entretanto, em locais maiores, essa granulação pode ser inviável. Uma técnica bastante utilizada, principalmente quando a capacidade computacional é muito reduzida, é calcular todos os valores possíveis anteriormente e construir uma tabela de consulta, pois usualmente é mais rápido ler os valores da memória do que calculá-los, e como a quantidade de medidas possíveis é finita, essa é uma solução bastante viável e eficiente.

Todavia, um mapa de ocupação em grade é uma das ferramentas mais úteis quando se trata de planejamento de trajetória. Mesmo que esse não seja o escopo

deste trabalho, é importante frisar que ter um mapa de ocupação em grade permite saber em quais locais o robô pode transitar. É possível ainda utilizar algoritmos que otimizem as trajetórias em locais que possuem pesos menores, evitando locais com pesos maiores. A navegação por “campo potencial” (Khatib, 1986), (Zhu, Yan, & Xing, 2006) é uma das técnicas capazes de utilizar a informação do mapa de ocupação em grade para navegar pelo ambiente desta forma. Esta técnica simula campos potenciais que direcionam o robô ao objetivo. O local do objetivo possui o “potencial” mais baixo, e os obstáculos possuem “potencial” alto, o que faz que o robô os evite, como mostrado na Figura 86. Os campos potenciais guiam o robô como uma corrente de água até o objetivo.

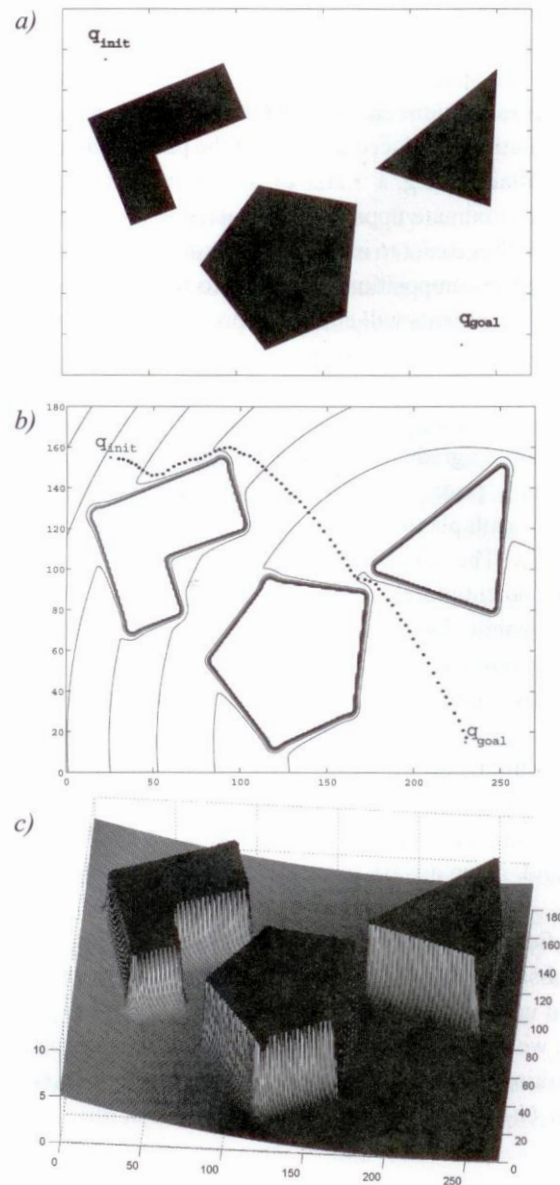


Figura 86 - Campos Potenciais para navegação (Siegwart & Nourbakhsh, 2004).

Por outro lado, localizar-se puramente utilizando o mapa de ocupação em grade é bastante complicado, e usualmente envolve comparar diversas células do mapa global com um mapa local feito pelo robô, e daí inferir o estado atual do robô através de comparações entre os dois mapas. A localização via Filtros Probabilísticos são mais eficientes e computacionalmente menos custosos, mas necessitam de marcos no mundo para serem viáveis. Por esta razão, manter um mapa de características, no caso em específico um mapa de retas, consome pouca memória e torna a correspondência das características observadas com as estimadas uma tarefa mais simples.

#### **4.10.4** **Retas limitadas ou ilimitadas?**

Em todos os trabalhos pesquisados, as retas extraídas continham um ponto inicial e um ponto final, que limitavam o segmento de reta. Todavia, existem situações (muitas delas bastante claras nos resultados apresentados) em que um plano se projeta além de um determinado ponto, onde aparentemente é seu fim, por exemplo: um móvel encostado na parede ou uma porta aberta. As retas que descrevem esses planos são colineares, pensa-se, portanto, que não há a necessidade de detectá-los novamente, uma vez que os mesmos já estão contidos no mapa. Esta abordagem retornou bons resultados para os ambientes testados, porém, sabe-se que alguns ambientes podem ser particularmente problemáticos, principalmente aqueles que possuem planos pouco inclinados uns aos outros, neste caso, o algoritmo de correspondência poderá retornar correspondências trocadas, uma vez que a diferença entre dois planos pode ser muito pequena, como mostrado na Figura 87.

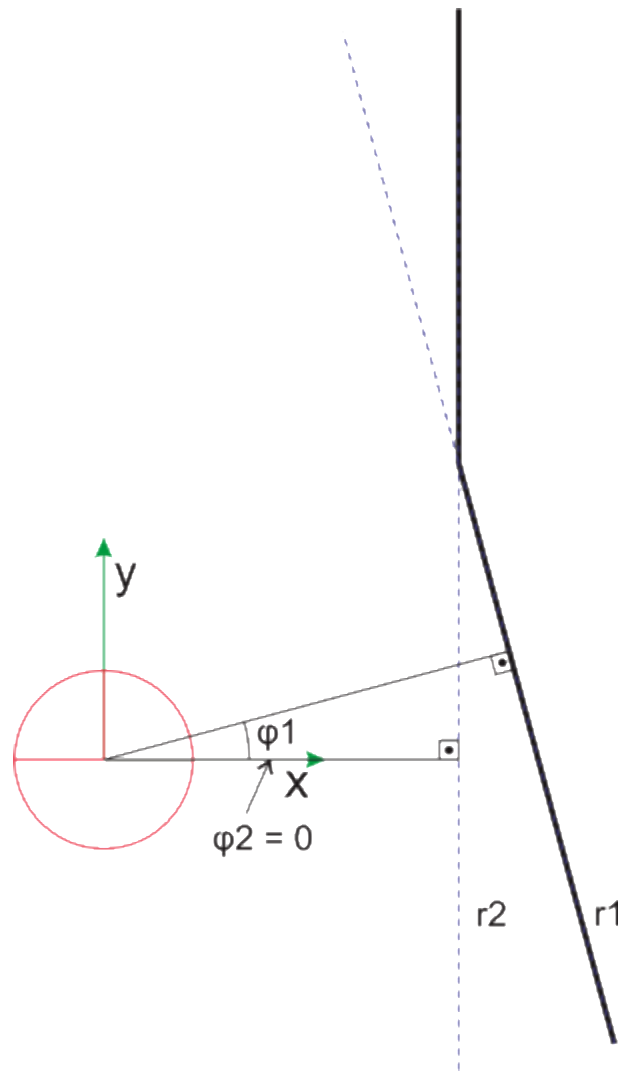


Figura 87 - Planos pouco inclinados e correspondência errada.

Este é mais um caso em que o mapa de ocupação em grade pode ser útil. Para “limitar” a visualização das retas, considerar que o sonar só poderá detectar uma reta se o ponto de contato estiver em uma célula ocupada. Esta regra simples “limita” naturalmente as retas sem a necessidade de estabelecer pontos inicial e final para as mesmas. É importante notar que esta técnica não foi implementada no algoritmo apresentado, mostrada aqui apenas como uma sugestão ao sistema.

#### 4.10.5

##### **Influência do controle e do planejamento de trajetória no mapeamento**

Apesar de muitos trabalhos, inclusive este, tratar o problema de localização e mapeamento de forma distinta ao problema de controle e planejamento de trajetória, há uma grande elo entre estes dois problemas. A forma com que o planejamento de trajetória e o controle influenciam no mapeamento foi mostrada através dos resultados dos ambientes simulados. Além disso, o algoritmo de

planejamento de trajetória deve saber tomar decisões quando o ambiente ainda está sendo mapeado e as informações disponíveis ainda são escassas. Outra função importante do sistema de planejamento de trajetória é tomar trajetórias que maximizam a detecção e atualização dos marcos do ambiente, impedindo que o robô permaneça em “pontos cegos”.

#### **4.10.6**

##### **Somente retas é o suficiente?**

Por mais que os ambientes humanos sejam formados, em sua maioria, por planos, eles contêm outras formas. Estas formas não precisam ser necessariamente representadas como elas são, uma prova disso é o trabalho de (Lee & Song, 2010), que representam cantos ou qualquer outra estrutura saliente como círculos. O que realmente é importante é possuir um marco estático e que seja de fácil detecção.

Quanto mais marcos o robô for capaz de detectar, é esperado que melhor seja seu desempenho, tanto em relação ao mapeamento quanto em relação à localização. Logo, combinar diferentes formas de detecção, utilizando um mesmo tipo de sensor, é uma das alternativas de evitar “pontos cegos”, onde o robô não possui referências suficientes para localizar-se completamente.