ROBÓTICA

MAPEAMENTO DO GRID

E

NAVEGAÇÃO POR SONAR

Prof.º: Dr. Marcelo Nicoletti Franchin

**Aluno:**

Saulo Coelho

# **INTRODUÇÃO**

Uma das áreas de estudo dos robôs móveis, é o mapeamento de ambientes que não são previamente conhecidos, também conhecida como SLAM (*simultaneous localization and mapping)*, sendo usado diversos tipos sensores (podemos citar como os mais utilizados o ultrassom e laser) para coletar informações de objetos / obstáculos do ambiente e montar o mapa com as distancias e posição.

O princípio básico do SLAM, é fazer o robô andar pelo ambiente fazendo medições e atualizando o mapa. No primeiro momento o mapa está zerado e o robô irá andar em um ambiente onde ele não possui informação alguma dos objetos que ali estão. Porém, assim que os sensores informam a possibilidade de algum objeto estar à frente do robô, o mapa é atualizado com essa distância. Entretanto essa informação do sensor poderá ser imprecisa, devido ao erro característico do sensor, que deverá de antemão ser conhecido por ensaios em laboratórios.

O paper escrito por Albert Elfes publicado em 1989 na IEEE, leva em consideração os erros provenientes de cada sensor, isso foi um marco nos estudos de missão robótica, pois ele utilizou técnicas probabilísticas para levantar o mapa do ambiente, dizendo qual a probabilidade de cada célula do grid estar ocupa ou vazia. Nesse paper ele faz várias medidas e atualiza cada célula do grid, usando o valor lido pelo sensor e o valor da probabilidade calculada anteriormente, assim as informações do ambiente são coletadas e o robô poderá saber para quais células e poderá ir ou não.

# **2.0. METODOLOGIA**

Para marcar as células que são cortadas pelo sensor e determinar qual a probabilidade de vazio ou ocupado, um programa no *software* Matlab, foi escrito. O usuário deve fornecer as coordenadas do robô, a distância retornada pelo sensor, o ângulo α de inclinação do robô com a horizontal, a distância mínima que sensor poderá ler, e o ângulo ω de abertura do cone. O programa irá retornar duas matrizes (K e M), um gráfico mostrando as linhas de atuação do sensor, uma imagem mostrando as células selecionadas, e a distribuição da probabilidade em cada célula. A matriz M mostra quais células foram cortadas pelas linhas de atuação do sensor, e a matriz K é a matriz das probabilidades.

## **2.1 Linhas de atuação do sensor**

O sensor ultrassônico, pode detectar um obstáculo emitindo uma onda sonora de alta frequência e calculando o tempo de retorno dessa onda (eco). Quando a onda é emitida um *timer* é iniciado, a onda ira se propagar pelo ar, e ao chocar com um obstáculo, uma parte da energia da onda será absorvida pelo objeto, uma outra parte será dissipada em várias direções e uma pequena parte retorna ao sensor, quando essa pequena parte é recebida no sensor, o timer é parado. Sabendo que a velocidade do som no ar é de aproximadamente 340 m/s, é possível determinar a distância percorrida pela onda, que é a mesma do robô até o objeto.

Porem um cuidado é necessário ao trabalhar com sensores ultrassônicos, pois a onda viaja em formato de um cone, (usa-se ω = 30° para a abertura do cone) e o objeto poderá estar em qualquer ponto dentro desse cone. Elfes (1989) montou uma distribuição probabilística de onda o objeto realmente está.

Sendo assim, para calcular a probabilidade de cada célula do grid estar ocupada ou não, é necessário saber quais células então dentro da abertura do cone e qual o ângulo α que o robô está fazendo com a horizontal no momento da leitura.

Para determinar quais células foram cortadas pelo sensor, foi feito um programa no *software* Matlab, e a primeira tentativa para determinar essas células, foi montar a linha superior do cone, depois a linha inferior e por último a linha que fecha do cone. Essa última linha foi aproximada para uma reta, (na pratica ela é um arco de circunferência com raio R, onde R é a distância fornecida pelo sensor). Após a montagem dessas linhas o programa percorre todos as linhas horizontais do grid e verifica quais linhas do grid eram cortadas pela linha do cone. Essa hipótese funcionou apenas para alguns casos (apenas com α grandes), pois quando do ângulo α era pequeno, algumas células do grid eram cortadas apenas nas linhas verticais, e nesse caso o programa não selecionava aquela célula.

A segunda hipótese foi então verificar, para uma determinada célula, se a linha horizontal inferior e vertical esquerda eram cortadas, ou seja, para uma célula qualquer, apenas as linhas inferior e esquerda (figura 1.1) eram testadas.

|  |
| --- |
| Figura 1.1: Linhas para uma célula que eram verificadas. |
|  |

Porem essa tentativa falhou quando a reta era crescente, pois embora a reta cortasse a linha horizontal da célula, nem sempre cortava a linha vertical esquerda.

A terceira tentativa, resolveu todas essas situações, para isso agora o programa verifica se pelo uma, das quatro linhas que definem a célula são cortadas pela reta, se for, a célula é selecionada. Para isso foi seguido o seguinte algoritmo:

1. Montar a equação das três retas que definem o cone;
2. Para cada uma das retas fazer:

2.1 – Para cada célula da matriz determinar as coordenadas máximas e mínima em x e em y e fazer:

2.1.1 – Determinar o ponto de cruzamento das retas que definem a célula com a reta que define o cone;

2.1.2 – Se ponto de cruzamento em x for maior ou igual que x mínimo e menor ou igual que x máximo, marcar essa célula com 1;

2.1.3 - Se ponto de cruzamento em y for maior ou igual que y mínimo e menor ou igual que y máximo, marcar essa célula com1.

O resulto obtido é uma matriz (M) com as células que são cortadas pela linha que forma o cone com o valor 1.

Os resultados obtidos com esse algoritmo serão mostrados no próximo capítulo.

## **2.2 Cálculo da probabilidade de a célula estar ocupada ou vazia**

O cálculo da probabilidade da célula estar vazia (Pe) é função da distância r e do *bearing* onde r é distância do ponto onde o robô está e ponto onde se quer calcular a probabilidade e o *bearing* θ é o ângulo que o robô está fazendo com a horizontal.

A distância r foi calculada como sendo a distância das coordenadas do robô até o ponto do vértice superior direito de cada célula com o valor zero que está dentro da matriz M.

A probabilidade de vazio para uma distância r é dado pela equação 2.1:

|  |  |
| --- | --- |
|  | (2.1) |

Onde:

Rmin = distância mínima a partir do qual o sensor pode começar a realizar as medições;

ε = erro do sensor.

E a probabilidade de vazio em função do ângulo θ é dada pela equação 2.2.

|  |  |
| --- | --- |
|  | (2.2) |

A probabilidade da célula estar vazia é dada por 2.3.

|  |  |
| --- | --- |
|  | (2.3) |

A probabilidade da célula estar ocupada (PO), também é função da distância r e o ângulo θ, conforme as equações 2.3 e 2.4.

|  |  |
| --- | --- |
|  | (2.4) |

|  |  |
| --- | --- |
|  | (2.5) |

A probabilidade da célula estar ocupada é dada por 2.6.

|  |  |
| --- | --- |
|  | (2.6) |

Para esses cálculos um programa no Matlab foi escrito. O algoritmo utilizado foi:

1. Determinar a linha máxima e linha mínima do cone;
2. Determinar a coluna máxima e coluna mínima do cone;
3. Zerar todas as células;
4. Para cada célula dentro do limite linha máxima e linha mínima e coluna máxima e coluna mínima, fazer:

4.1 – Se a célula em questão for 1, zerar a célula;

4.2 – Calcular a distância r;

4.3 – Se r >= a Rmin e r <= que R-erro, então calcular Pe

4.4 – Se r >= R – erro e r < = R + erro, calcular PO;

4.4 – Se a linha e coluna forem iguais a coordenado do robô, marcar essa célula com 1

As células da matriz M, que estão com valor, indicam que são células que foram cortadas pela linha que define o cone de atuação do sensor. Para essas células é atribuído o valor 0, que representa desconhecido, pois para os pontos que definem a linha de atuação a probabilidade de vazio ou ocupado é zero. Para um ponto P qualquer a uma distância r nessa linha o ângulo θ é . A figura 2.1 mostra a localização do ponto P.

|  |
| --- |
| Figura 2.1: Calculo da probabilidade para um ponto P sobre a reta de atuação do sensor. |
|  |

Para esse ponto P, temos que θ = , substituindo esse ponto na equação 2.2, resulta em , conforme a equação 2.7.

|  |  |
| --- | --- |
|  | (2.7) |

Como a probabilidade da célula estar vazia é função do ângulo e da distância r, (equação 2.3), e a probabilidade em função do ângulo é zero para esse ponto, pode-se concluir que para os pontos sobre a reta a probabilidade da célula estar vazio é zero, conforme equação 2.8.

|  |  |
| --- | --- |
|  | (2.8) |

Sendo assim, todas as células marcadas com 1 na matriz M (matriz das células cortadas pela linha do sensor), recebe o valor 0 na matriz K (matriz das probabilidades).

## **2.3 Funções usadas no programa**

Visando o aproveitamento de código, algumas funções (\*.m) foram escritas de modo que possam ser utilizadas em programas, como exemplo, para cálculo de trajetória de rota (A estrela ou o método do grid). As funções escritas são detalhadas abaixo.

### **2.3.1 Reta**

Essa função recebe como argumentos as coordenas x e y de dois pontos (P1 e P2), e retorna o coeficiente angular, coeficiente linear e indica se a reta é vertical ou não, além de desenhar o segmento de reta definido por P1 e P2 no gráfico.

A função reta é:

function [a b reta] = reta(Sx,Sy,Px,Py);

reta = 0;

if (Sx == Px)

a = 0;

b = 0;

reta = 1;

line([Px Px],[min(Py,Sy) max(Py,Sy)])

else

a = (Py - Sy) / (Px - Sx);

b = -a\*Sx+Sy;

x = linspace(min(Sx,Px),max(Sx,Px));

y = a\*x + b;

plot(x,y)

end

end

### **2.3.2 Região**

Essa função recebe como argumento um vetor linha com coordenadas em x, um vetor linha com coordenadas em y, o espaçamento do grid em x e o espaçamento do grid em y, e retorna as linhas e colunas que definem o quadrilátero onde o cone está. A figura 2.2 mostra como é esse quadrilátero para uma leitura do sensor, com o robô nas coordenadas 2 em x e em y, com R = 3, erro = 0.4, Rmin = 0.4, abertura do cone 30°e α = 45.

|  |
| --- |
| Figura 2.2: Linhas máximas e mínimas e colunas máximas e mínimas do cone. |
| Coluna máxima  Coluna mínima  Linha máxima  Linha mínima |

Essa função também limita os valores das linhas e colunas para a região dentro da matriz evitando que o programa faça referência a alguma célula que não faz parte do grid.

A função região é:

function [imin imax jmin jmax] = regiao(X,Y,gridx,gridy);

imin = 16 - ceil(max(Y) / gridy);

if imin <= 0 imin = 1

end

imax = 16 - ceil(min(Y) / gridy);

if imax >= 16 imax = 15

end

jmin = ceil(min(X) / gridx) ;

if jmin <= 0 jmin = 1

end

jmax = ceil(max(X) / gridx);

if jmax >= 26 jmax = 25

end

end

### **2.3.3 xycell**

Essa função recebe como argumento a linha e a coluna da célula e retorna as coordenadas em x e y das linhas que definem a célula.

A função xycell é:

function [xcellmin xcellmax ycellmin ycellmax] = xycell(i,j);

xcellmax = j\*0.4;

xcellmin = j\*0.4 - 0.4;

ycellmin = 6 - i\*0.4;

ycellmax = ycellmin + 0.4;

end

### **2.3.4 prob**

É a função principal do programa, que chama todas as outras. Ela recebe uma matriz com as coordenadas do robô em cada medição. Tem duas colunas, a primeira representa a coordenada em x e a segunda em y, cada linha representa coordenadas de uma medição. O segundo parâmetro é distancia retornada pelo sensor em cada medição. É um vetor coluna, onde cada coluna é referente a uma medição. O terceiro parâmetro é o ângulo α de cada medição. Assim como o parâmetro anterior, também é um vetor coluna, onde cada linha representa o ângulo de cada medida. Os três últimos parâmetros referem ao raio mínimo, o ângulo de abertura do cone (ω) e o erro da leitura do sensor.

Para obter a matriz da probabilidade de um único tiro do sensor, quando o robô estiver na posição 2 em x e em y, com o sensor retornando 3m, com α = 45°, raio mínimo de 0.4m, ω = 30 e erro igual a 0.4, deve se entrar com o seguinte comando:

prob([2 2], [3],[45],0.4,30,0.4)

A função já está preparada para receber mais de uma leitura, embora o programa ainda não contemple tal função. Mas para entrar com dois valores a função deverá ficar da seguinte forma:

prob([2 2; 2 3], [3; 2], [45; 0], 0.4, 30, 0.4)

Indicando que a primeira leitura foi feita com o robô na coordenada 2 em x e y, com R = 3 e α = 45°. A segunda leitura foi feita com o robô na posição 2 em x, 3 em y, com R = 2, e α = 0. Todas essas medições foram obtidas com raio mínimo = 0.4, ω = 30 e erro = 0.4m.

A função prob está no apêndice 1 desse trabalho.

# **3.0 RESULTADOS**

Para validar o programa, foi feita 3 simulações, alterando o valor da distância fornecida pelo sensor e o ângulo α. Todos com apenas uma leitura do sensor. Os valores negativos indicam probabilidade das células estarem ocupadas e os valores positivos indicam probabilidade das células estarem vazias.

## **3.1 Teste um**

O primeiro teste foi feito com o robô na posição x e y em 2, distância = 3, α = 0, raio mínimo = 0.4m, ω=30º e erro = 0.4. A função prob foi chamada da seguinte forma:

prob([2 2], [3],[0],0.4,30,0.4)

As linhas de atuação do cone são mostradas na figura 3.1.

|  |
| --- |
| Figura 3.1: Linhas de atuação do cone. |
|  |

As células selecionadas são mostradas na figura 3.2.

|  |
| --- |
| Figura 3.2: Células selecionadas. |
|  |

A matriz M ficou com os seguintes valores:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | C13 | C14 |
| L8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| L9 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| L10 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| L11 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| L12 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| L13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

A matriz K altera apenas as células que estão no meio do cone e com valores zerados, nesse caso apenas as células da linha 10 e 11 das colunas 10, 11, 12 e 13.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Coluna 10 | Coluna 11 | Coluna 12 | Coluna 13 |
| Linha 10 | 0.1918 | 0.0879 | -0.5763 | -0.5296 |
| Linha 11 | 0.4711 | 0.1736 | -0.75 | -0.75 |

A figura 3.3 mostra a probabilidade por cores. As cores vermelhas são números positivos e as cores azuis indicam valores negativos. Amarelo são as regiões desconhecidas.

|  |
| --- |
| Figura 3.3: Probabilidades das células estarem ocupadas e vazias. |
|  |

## **3.2 Teste dois**

O primeiro teste foi feito com o robô na posição x e y em 2, distância = 3, α = 45, raio mínimo = 0.4m, ω=30º e erro = 0.4. A função prob foi chamada da seguinte forma:

prob([2 2], [3],[45],0.4,30,0.4)

As linhas de atuação do cone são mostradas na figura 3.4.

|  |
| --- |
| Figura 3.4: Linhas de atuação do cone. |
|  |

As células selecionadas são mostradas na figura 3.5.

|  |
| --- |
| Figura 3.5: Células selecionadas. |
|  |

A matriz M ficou com os seguintes valores:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | C13 |
| L3 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| L4 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| L5 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| L06 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| L07 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| L08 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| L09 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| L10 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| L11 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A matriz K altera apenas as células que estão no meio do cone e com valores zerados, nesse caso apenas as células da linha 5, 6 e7 das colunas 8, 9, 10 e 11.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Coluna 08 | Coluna 09 | Coluna 10 | Coluna 11 |
| Linha 05 | 0 | 0 | -0.7954 | 0 |
| Linha 06 | 0 | 0.0287 | -0.8160 | -0.7954 |
| Linha 07 | 0 | 0.2831 | 0.0287 | 0 |
| Linha 08 | 0.6524 | 0 | 0 | 0 |

A figura 3.6 mostra a probabilidade por cores. As cores vermelhas são números positivos e as cores azuis indicam valores negativos. Amarelo são as regiões desconhecidas.

|  |
| --- |
| Figura 3.6: Probabilidades das células estarem ocupadas e vazias. |
|  |

## **3.3 Teste três**

O primeiro teste foi feito com o robô na posição x e y em 2, distância = 3, α = 90°, raio mínimo = 0.4m, ω=30º e erro = 0.4. A função prob foi chamada da seguinte forma:

prob([2 2], [3],[90],0.4,30,0.4)

As linhas de atuação do cone são mostradas na figura 3.7.

|  |
| --- |
| Figura 3.7: Linhas de atuação do cone. |
|  |

As células selecionadas são mostradas na figura 3.8.

|  |
| --- |
| Figura 3.8: Células selecionadas. |
|  |

A matriz M ficou com os seguintes valores:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | C3 | C4 | C5 | C6 | C7 | C8 |
| L2 | 1 | 1 | 1 | 1 | 1 | 1 |
| L3 | 1 | 1 | 0 | 0 | 1 | 1 |
| L4 | 0 | 1 | 0 | 0 | 1 | 0 |
| L05 | 0 | 1 | 0 | 0 | 1 | 0 |
| L06 | 0 | 1 | 0 | 0 | 1 | 0 |
| L07 | 0 | 1 | 1 | 1 | 1 | 0 |
| L08 | 0 | 0 | 1 | 1 | 0 | 0 |
| L09 | 0 | 0 | 1 | 1 | 0 | 0 |
| L10 | 0 | 0 | 1 | 1 | 0 | 0 |
| L11 | 0 | 0 | 1 | 1 | 0 | 0 |

A matriz K altera apenas as células que estão no meio do cone e com valores zerados, nesse caso apenas as células da linha 3, 4, 5 e 6 das colunas 5 e 6.

|  |  |  |
| --- | --- | --- |
|  | Coluna 05 | Coluna 06 |
| Linha 03 | -0.75 | -0.5296 |
| Linha 04 | -0.75 | -0.5763 |
| Linha 05 | 0.1736 | 0.0879 |
| Linha 06 | 0.4711 | 0.1918 |

A figura 3.9 mostra a probabilidade por cores. As cores vermelhas são números positivos e as cores azuis indicam valores negativos. Amarelo são as regiões desconhecidas.

|  |
| --- |
| Figura 3.9: Probabilidades das células estarem ocupadas e vazias. |
|  |

# **4.0 CONCLUSÕES**

Depois das simulações e dos gráficos gerados, pode-se concluir que as linhas de atuação do cone ficaram bem definidas entre as células do grid e as probabilidades apresentaram valores coerentes com cada situação.

Os valores para os três testes ficaram dentro do intervalo 1 e -1, sendo que números negativos indicam probabilidade da célula estar ocupada, fato esse que pode ser notado ao olhar para o final do cone, onde a probabilidade de ter um objeto é alta. Para uma linha com θ=0 pode-se perceber que o valor da probabilidade de vazio vai diminuindo até o ponto onde a probabilidade de ocupado aumenta o gráfico 4.1 mostra essa probabilidade para o teste 1, foi escolhido esse teste para fazer essa conclusão porque para a linha central do cone, a probabilidade em função do ângulo θ é sempre 1, ou seja, não influencia no cálculo da probabilidade das células estarem ocupadas ou vazias.

|  |
| --- |
| Figura 4.1: Probabilidades das células em função da distância r. |
|  |

A células que são cortadas pelas linhas são zeradas, pois a função vazia ou cheia em função do ângulo é zero para ângulos na extremidade do cone.

Como trabalho futuro, pode-se fazer mais de uma medição e sobrepor esses valores com as demais medidas, melhorando a distribuição da probabilidade.

**APENDICE**

%Trabalho apresentado para a disciplina Robótica

%professor Dr. Marcelo

%Geração das linhas de atuação de um sensor ultrassonico e

%Calculo das probabildades das celulas estarem ocupadas e vazias

%19/07/2016

%Recebe os seguintes parametros

%S = vetor das coordenadas x e y de cada disparo do sensor

%R = vetor da distancia retornada pelo sensor em cada tiro

%t = vetor da inclinação do robo com a horizontal em cada tiro

%Rmin = constante. Distancia do ponto cego do sensor

%w = consntante. Abertura do cone

%erro = erro da leitura do sensor

function [M K] = prob(S,R,t,Rmin,w,erro);

gridx = 0.4; %grid quadrado gridx = gridy

gridy = gridx;

i = 0;

j = 0;

Lin= 15;

Col= 25;

M = zeros(Lin,Col); %matriz para desenhar o cone

K = zeros(Lin,Col); %matriz das probabilidades

for m=1:length(t)

%calculos dos pontos dos segmentos da reta

%linha superior do cone

P1x = S(m,1) + cosd(t(m) + 15)\*(R(m)+ erro);

P1y = S(m,2) + sind(t(m) + 15)\*(R(m)+ erro);

%linha inferior do cone

P2x = S(m,1) + cosd(t(m) - 15)\*(R(m)+ erro);

P2y = S(m,2) + sind(t(m) - 15)\*(R(m)+ erro);

figure(1)

hold on

axis([0 10 0 6])

grid on

set(gca, 'XTick',0:0.4:10);

set(gca, 'YTick',0:0.4:6);

title('Linhas de atuação do sensor');

%reta da linha superior do cone

[ap1 bp1 reta\_vertical\_1] = reta(S(m,1),S(m,2),P1x,P1y);

%reta da linha inferior do cone

[ap2 bp2 reta\_vertical\_2] = reta(S(m,1),S(m,2),P2x,P2y);

%arco do cone aproximado para uma reta

[ap3 bp3 reta\_vertical\_3] = reta(P1x,P1y,P2x,P2y);

%monta uma matriz com os dados das retas

retas = [ap1 bp1 reta\_vertical\_1;

ap2 bp2 reta\_vertical\_2;

ap3 bp3 reta\_vertical\_3];

pontos = [S(m,1) P1x P2x;

S(m,2) P1y P2y];

%hold off

%define região do cone

[imin imax jmin jmax] = regiao([S(m,1) P1x P2x],[S(m,2) P1y P2y],gridx,gridy);

for k = 1:3

%varre as linhas do grid dentro do intervalo da reta

%calcula a probabilidade para cada uma das celulas cortadas pela

%linha do sensor

for i=imin : imax

for j=jmin : jmax

[xcellmin xcellmax ycellmin ycellmax]= xycell(i,j);

if(retas(k,3)) %reta vertical

if(pontos(1,k) <= xcellmax & pontos(1,k) >= xcellmin)

M(i,j) = 1;

end

else

%pe = Pontos de intersecção das retas

%x,y da reta horizontal min x,y reta vertical min

pe = [(ycellmin-retas(k,2))/retas(k,1) ycellmin xcellmin xcellmin\*retas(k,1) + retas(k,2);

(ycellmax-retas(k,2))/retas(k,1) ycellmax xcellmax xcellmax\*retas(k,1) + retas(k,2)];

%valida o intervalo dos pontos

for n= 1 : 2

%teste da linha horizontal

if pe(n,1) >= xcellmin & pe(n,1) <= (xcellmax) & pe(n,2) > ycellmin & pe(n,2) <= ycellmax

M(i,j) = 1;

end

%teste da linha vertical

if pe(n,3) >= xcellmin & pe(n,3) <= xcellmax & pe(n,4) > ycellmin & pe(n,4) <= ycellmax

M(i,j) = 1;

end

end

end

r = sqrt((ycellmax - S(m,2))^2+(xcellmax - S(m,1))^2);

theta = atand((ycellmax - S(m,2)) / (xcellmax - S(m,1)))-t(m);

if r >= Rmin && r < (R(m) - erro) && theta >= -w/2 && theta <= +w/2

if M(i,j) == 1

K(i,j) = 0;

else

K(i,j)= (1 -(((r-Rmin)/(R(m)-erro-Rmin))^2))\* (1- ((2\*theta)/w)^2);

end

else if r >= (R(m)-erro) && r < (R(m) + erro) && theta >= -w/2 && theta <= +w/2

if M(i,j) == 1

K(i,j) = 0;

else

K(i,j)= -1\*(1 -(((r-R(m))/(erro))^2))\* (1- ((2\*theta)/w)^2);

end

else

K(i,j) = 0;

end

end

end

end

end

figure(2)

colormap(gray);

imagesc(~M)

grid on

set(gca, 'XTick',0.5:1:25);

set(gca, 'YTick',0.5:1:15);

title('Celulas marcadas');

figure(3)

surf(K)

end