DETECÇÃO E ACOMPANHAMENTO AUTOMÁTICOS PARA SONARES

Fábio Oliveira Baptista da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Carlos Fernando Teodósio Soares

Rio de Janeiro
Outubro de 2020

DETECÇÃO E ACOMPANHAMENTO AUTOMÁTICOS PARA SONARES

Fábio Oliveira Baptista da Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Orientador: Carlos Fernando Teodósio Soares

Aprovada por: Prof. Carlos Fernando Teodósio Soares
Prof. João Baptista de Oliveira e Souza Filho
Dr. William Soares Filho

RIO DE JANEIRO, RJ – BRASIL
OUTUBRO DE 2020

Silva, Fábio Oliveira Baptista da

Detecção e Acompanhamento Automáticos Para Sonares/Fábio Oliveira Baptista da Silva. – Rio de Janeiro: UFRJ/COPPE, 2020.

XII, 89 p.: il.; 29, 7cm.

Orientador: Carlos Fernando Teodósio Soares

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2020.

Referências Bibliográficas: p. 86 – 89.

1. Sonar. 2. Cluster. 3. Tracking. I. Soares, Carlos Fernando Teodósio. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha esposa.

Agradecimentos

Dedico o desenvolvimento desse trabalho à minha esposa, por todo companheirismo ao longo desse mestrado conjunto e sem a qual eu não estaria concluindo. Agradeço aos meus pais, Denise e Marco, e a minha irmã, Fernanda, pelo apoio e suporte ao longo de toda a vida.

Agradeço ao comandante Carlos Martins, ao comandante Barreira, por permitir minha atuação na área técnica e por me incentivarem a iniciar o mestrado. Ao Orlando, por me passar um pequena parte da sua experiência, que me serviu de base para o início deste mestrado.

Agradeço ao tenente Bozzi, ao Fernando Monteiro e ao William, que me deram suporte ao desenvolvimento técnico desta dissertação, sendo companheiros e referências que levarei por toda a vida.

Agradeço ao meu orientador, por todo o suporte e compreensão ao longo dessa jornada cheia de altos e baixos, e aos amigos do IPqM e do PADS. pelo apoio ao longo da realização deste mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DETECÇÃO E ACOMPANHAMENTO AUTOMÁTICOS PARA SONARES

Fábio Oliveira Baptista da Silva

Outubro/2020

Orientador: Carlos Fernando Teodósio Soares

Programa: Engenharia Elétrica

Neste trabalho é abordada a construção de um sistema autônomo de detecção e acompanhamento de contatos, analisando cada uma das etapas necessárias.

Iniciando com a apresentação dos dados do processamento sonar, tanto passivo quanto ativo, passando pela conversão dos dados para um conjunto de pontos, chamados de detecções. Seguindo para a aplicação de técnicas para redução de dimensionalidade, como a clusterização. Além de observar os problemas da aplicação direta das técnicas de clusterização a dados de espaço-temporais, são propostas alterações nas técnicas clássicas para adequação à aplicação e melhora do desempenho dos algoritmos.

Posteriormente, cada conjunto de clusters é aplicado ao sistema de acompanhamento, que deve prospectar a presença de contatos, acompanhar a evolução temporal dos clusters de cada contato e analisar o movimento dos contatos para previsão futura de posição e velocidade.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AUTOMATIC DETECTION AND TRACKING IN SONARS

Fábio Oliveira Baptista da Silva

October/2020

Advisor: Carlos Fernando Teodósio Soares

Department: Electrical Engineering

In this work, the construction of autonomous contact detection and tracking system is addressed, analyzing each of the necessary steps.

Starting with the presentation of the data from the sonar processing, both passive and active, going through the conversion of the data to a set of points, called detections. Moving on to the application of techniques to reduce dimensionality, such as clustering. In addition to observing the problems of direct application of clustering techniques to temporal space data, changes are proposed in the classic techniques to suit the application and improve the performance of the algorithms.

Subsequently, each set of clusters is applied to the tracking system, which should prospect the presence of contacts, monitor the temporal evolution of the clusters of each contact and analyze the movement of the contacts for a future position and speed prediction.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Acústica submarina é a área que estuda a propagação das ondas sonoras na água e sua interação com o meio e as vizinhanças. Aplicações desta área fazem uso de transdutores acústico-elétricos comumente chamados de: hidrofones, que transduzem o sinal acústico em sinal elétrico; projetores, que convertem o sinal elétrico em acústico; e, simplesmente, transdutores, quando realizam ambas as funções.

Sonar, do inglês *Sound Navigation and Ranging*, é uma técnica que utiliza transdutores acústico-elétricos para a caracterização do ambiente ou para detectar, localizar e classificar ruídos de navios ou de vida marinha. Existem basicamente dois tipos de sistemas sonares: os passivos e os ativos.

Os sonares passivos recebem ondas acústicas geradas no ambiente onde os sensores estão imersos, oriundas de alguma fonte sonora. Sistemas passivos, por si só, não são capazes de estimar a distância entre os sensores e as fontes sonoras, apenas a direção, pois não sabem o instante de tempo no qual o ruído foi gerado. Existem diversas aplicações para os sonares passivos, como por exemplo: monitoramento de canais, caracterização de paisagem acústica, estudo de vida marinha, além de ser essencial para a consciência situacional de submarinos.

Sonares ativos emitem um som conhecido e identificam elementos em suas vizinhanças através do eco do sinal transmitido. Além da identificação da direção dos obstáculos, um sistema ativo consegue estimar a distância entre os transdutores e os obstáculos pelo tempo entre transmissão e recepção, desde que conhecida ou estimada a velocidade de propagação da onda acústica no meio. Existem diversas aplicações para os sonares ativos, tais como: guiagem de veículos autônomos submarinos, identificação de rompimentos em ductos submarinos, mapeamento do leito submarino, além de ser utilizado em navios de superfície para a localização de submarinos.

Ambos os tipos de sonares têm aplicações com um ou mais transdutores. Utilizar vários transdutores possibilita, aproveitando-se de sua disposição espacial, tanto transmitir ondas direcionais, quanto identificar a direção do eco ou do ruído de uma

fonte sonora. Na navegação, uma direção é, muitas vezes, chamada de marcação.

Grande parte dos sistemas sonares necessitam detectar e acompanhar a evolução temporal de contatos. Para sistemas sonares passivos, contatos são as fontes sonoras de interesse ao redor dos sensores. Já para sistemas sonares ativos, contatos são as superfícies que refletem os sinais transmitidos. A detecção e o acompanhamento desses contatos é, na maior parte das aplicações, feita de forma semi-automática auxiliando um operador a identificar e evoluir os contatos.

A construção de sistemas autônomos para a detecção e o acompanhamento de contatos expande a capacidade de sistemas de vigilância e monitoramento, por não necessitar de múltiplos operadores. Mesmo em sistemas operados, a detecção automática pode auxiliar o operador e diminuir o tempo de resposta a uma ameaça.

Existe uma literatura extensa na área de detecção e acompanhamento, mas focada em certas etapas do processo. Este trabalho surge da necessidade de se explorar estas etapas, preenchendo as lacunas existentes, bem como uni-las para construção de um sistema autônomo.

O objetivo deste trabalho é analisar a cadeia de processamento de um sistema autônomo de detecção e acompanhamento de contatos para sonares, explorando a clusterização como técnica de detecção, propondo modificações nas técnicas clássicas para melhor explorar as características dos sinais trabalhados, analisando os pontos críticos das técnicas de acompanhamento, além de abordar a inserção de análises do movimento dos contatos como um aprimoramento dos mecanismos de acompanhamento.

1.1 Organização do trabalho

O restante deste trabalho está dividido da seguinte forma. No Capítulo 2 é feita uma breve revisão de literatura, realizada em três seções. Primeiramente, uma seção sobre processamento dos sinais, a Seção 2.1, que aborda as técnicas de processamento comumente aplicadas a sistemas sonares, além de estabelecer as diferenças entre as cadeias de processamento para sonar passivo e ativo. Em seguida, uma seção sobre a detecção de contatos, a Seção 2.2, na qual são abordadas as técnicas de clusterização e sua aplicação em dados espaço-temporais. Por fim, uma seção sobre o acompanhamento das detecções ao longo do tempo, a Seção 2.3, na qual são abordadas a estratégia básica de evolução dos acompanhamentos e as técnicas para a análise do movimento dos contatos.

No Capítulo 3 é abordada a estrutura do trabalho desenvolvido, além da decomposição em etapas da cadeia de detecção e acompanhamento. A Seção 3.1 explora o simulador que servirá de base para as análises deste trabalho. A Seção 3.2 aborda a detecção por clusterização para dados de base espacial, gerados pelo simulador, os

problemas da aplicação das técnicas clássicas de clusterização a este tipo de dado, e propõe uma alteração dessas técnicas para uma adequação ao cenário de interesse deste trabalho. Na Seção 3.3 é analisada a estrutura do acompanhamento, especificando cada etapa e como estas etapas se conectam para a construção de um sistema autônomo.

No Capítulo 4 são expostos os resultados e as análises feitas neste trabalho, subdividindo as seções para sonar passivo e ativo. Na Seção 4.1 é analisado o efeito das alterações propostas na Seção 3.2 em termos de falsos positivos, falsos negativos e na estimativa de posição. Enquanto, na Seção 4.2 são comparados os resultados obtidos para as técnicas de acompanhamento analisadas.

No Capítulo 5 são conduzidas as conclusões a respeito dos resultados obtidos e realizada a sugestão de trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

2.1 Processamento para Sistema Sonar

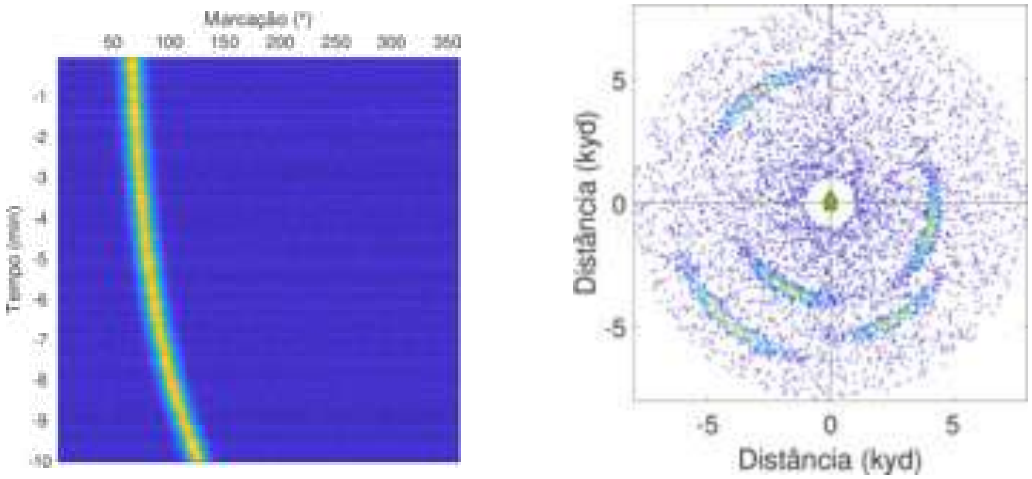
Existem dois tipos de sistemas sonares: os passivos e os ativos [1]. Embora ambos sistemas tenham como função detectar, localizar e classificar contatos, eles diferem quanto aos sinais de interesse. Enquanto os sistemas de sonares passivos estão interessados no ruído irradiado de outros navios e/ou vida marinha, os sistemas sonares ativos estão interessados no eco dos sinais transmitidos [2].

Uma das apresentações mais utilizadas em um sistema sonar passivo é um gráfico de três dimensões, conhecido como energia em *waterfall*, no qual: o eixo horizontal representa a marcação; o eixo vertical, o tempo, onde a informação do topo é a mais recente; e a cor representa a intensidade de energia de cada marcação. Como o eixo do tempo tem sua posição mais recente no topo, a cada nova janela de análise a ser apresentada, todo o gráfico é deslocado para baixo e uma nova linha é inserida no topo. Por isso, gráficos com esta distribuição temporal são conhecidos como gráficos em *waterfall*, pois caem, ao longo do tempo, como uma cachoeira [3]. A Figura 2.1(a) ilustra um gráfico de energia em *waterfall*, no qual pode-se ver um contato que se deslocou, ao longo dos últimos dez minutos, da marcação 120° para a marcação 70°.

Já em sistemas sonares ativos, uma das apresentações mais utilizadas é um gráfico de três dimensões, conhecido como geográfico, no qual: os eixos horizontal e vertical representam a distância de um ponto à posição do navio com o sonar; e a cor representa a intensidade do sinal refletido [2]. A Figura 2.1(b) ilustra um gráfico geográfico com cinco contatos nos arredores do navio com o sonar, também chamado de plataforma, representado pelo pentágono verde.

A visualização geográfica representa a energia dentro de um domínio espacial calculada durante um intervalo de tempo, muitas vezes chamado de quadro temporal. A análise de movimento dos contatos é feita pela evolução de uma nuvem de pontos

ao longo dos quadros temporais [2].



(a) Visualização passiva.

(b) Visualização ativa.

Figura 2.1: Visualizações comuns para sistemas sonar.

2.1.1 Processamentos de sinais

Sistemas sonar utilizam diversas técnicas de processamento de sinais, tais como: formação de feixe (*beamforming*), *fast fourier transform* (FFT), interpolação, normalização, filtragem, correlação, decimação, *overlap*, janelamento, entre outros [3]. Nesta seção serão discutidas a formação de feixe e a normalização, por serem as técnicas mais críticas para a compreensão deste trabalho.

2.1.1.1 Formação de feixe

Apesar de existirem aplicações de sistemas sonares com um único hidrofone, é mais comum utilizar arranjos de sensores. Com vários sensores é possível, explorando-se da conformação espacial dos seus elementos, obter um arranjo com maior ganho direcional, aumentando a relação sinal-ruído na direção de uma fonte sonora. Este processamento é conhecido como *beamforming* ou (con)formação de feixe [2].

Existem diferentes tipos de arranjos de hidrofones com diversos objetivos e aplicações. Estes arranjos podem ser divididos de acordo com a sua conformação espacial em lineares, planares ou volumétricos [4].

Dentre as diversas técnicas de formação de feixe, a técnica conhecida como atraso-e-soma (*delay-and-sum*) ilustra bem o conceito referido como filtragem espacial [5].

Conforme pode ser visto na Figura 2.2, para a geometria proposta e uma dada direção de chegada (*Direction of Arrival* - DoA), é possível calcular a distância e o respectivo intervalo de tempo (atraso) que uma frente de onda percorre para incidir em cada um dos sensores [4].

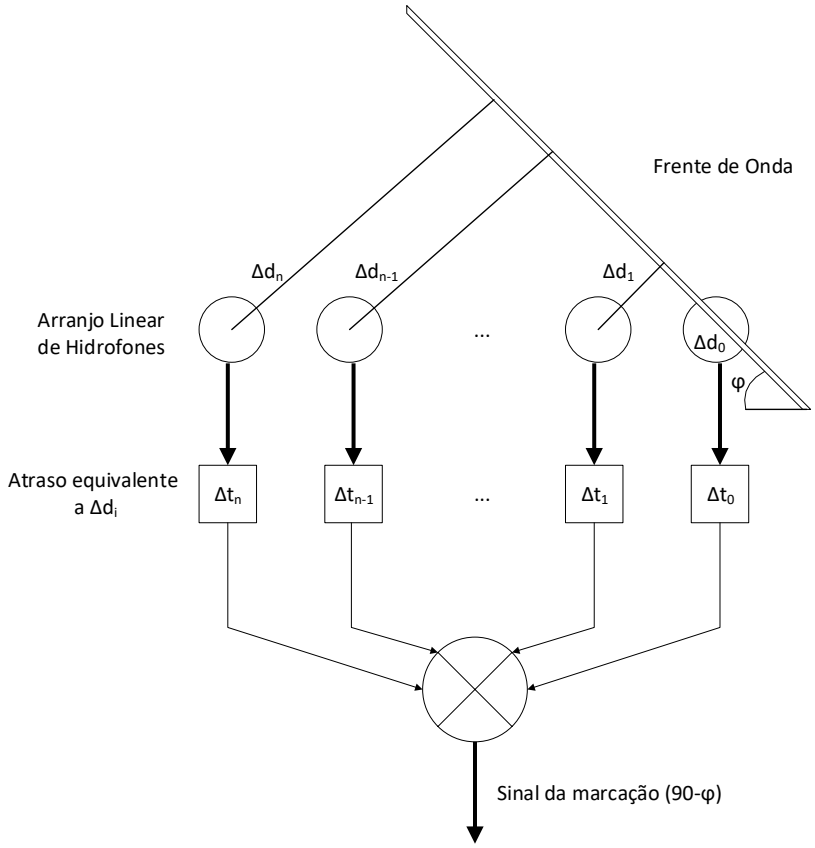


Figura 2.2: Diagrama ilustrativo do mecanismo de formação de feixe pela técnica *Delay-and-sum*.

A soma coerente dos sinais de casa sensor feita compensando-se esses atrasos, gera um ganho nos sinais vindos dessa direção, devido à interferência construtiva e uma atenuação dos sinais vindos de outras direções ou ruídos aleatórios devido à interferência destrutiva. Este efeito é conhecido como filtragem espacial. Além disso, calculando-se a matriz de atrasos para cada uma das direções de interesse, é possível prover um ganho para os sinais dessas direções, melhorando a relação sinal ruído (*Signal-to-Noise Ratio - SNR*) [4].

2.1.1.2 Normalização

Grande parte das análises dos sistemas sonares passivos e ativos são baseadas nos máximos ou picos de energia. A ênfase desses picos é um fator chave para estas análises. Em [6] é proposta a utilização da técnica *Two-Pass Split-Window* (*TPSW*) para a normalização do sinal e ênfase destes picos.

O algoritmo de *TPSW* realiza uma estimativa do ruído de fundo. A normalização por *TPSW* normaliza o sinal baseado nesta estimativa do ruído de fundo, podendo ser uma aproximação da relação sinal ruído (*SNR*) para um dado sinal [7].

A estimativa de ruído de fundo é feita em duas passagens de um filtro *split-window* com uma etapa de ceifamento intermediário [8]. A filtragem de um sinal (s) pode ser calculada como:

$$\hat{f}(k) = \frac{1}{2(N-M)} \sum_{k=-N}^{N} h_{sw}(k) s(n-k) \tag{2.1}$$

onde:

$s(i)$ = i-ésima amostra do sinal de entrada do filtro;
$\hat{f}(k)$ = k-ésima amostra da saída do filtro;
N = parâmetro inteiro positivo maior que M;
M = parâmetro inteiro positivo maior que 1;
$h_{sw}(k)$ = resposta ao impulso do filtro *split-window* discreto.

Sendo a resposta ao impulso do filtro *split-window* discreto, definida como:

$$h_{sw}(k) = \begin{cases} 0, & \|k\| < M \\ 1, & M \leqslant \|k\| < N \end{cases} \tag{2.2}$$

Uma vez filtrado, o sinal é ceifado conforme a equação:

$$g(k) = \begin{cases} \hat{f}(k), & ses(k) > \alpha \hat{f}(k) \\ s(k), & casocontrrio \end{cases} \tag{2.3}$$

onde:

$g(k)$ = k-ésima amostra do sinal ceifado;
$\hat{f}(k)$ = k-ésima amostra do sinal filtrado;
$s(k)$ = k-ésima amostra do sinal de entrada;
α = parâmetro inteiro positivo, conhecida como fator de ganho [6].

Em seguida, o sinal ceifado é filtrado novamente pelo mesmo filtro *split-window*, conforme a equação (2.1). A saída desta segunda filtragem é a saída do filtro *TPSW*, considerada como estimativa do ruído de fundo. O resultado da filtragem é altamente dependente dos parâmetros: M, N e α, que deve ser otimizados para a aplicação em foco.

A normalização por *TPSW* é uma técnica comumente empregada em sistemas sonares devido a sua boa performance na presença de ruído de fundo [7]. A aplicação do *TPSW* independentemente entre blocos temporais de dados permite que características transientes e irregulares sejam observadas [9].

2.1.2 Sonar Passivo

A cadeia de processamento tipicamente empregada em um sistema sonar passivo é ilustrada na Figura 2.3. Primeiramente, é realizada uma conformação de feixe, convertendo o sinal de cada sensor no sinal esperado de cada direção de chegada (DoA). Deste ponto, a figura ilustra quatro cadeias de processamento típicas, que serão descritas a seguir, da esquerda para a direita. Da saída do formador de feixe, extrai-se o áudio de uma direção de interesse através da seleção de um feixe. Este sinal pode ser utilizado diretamente pelo operador do sonar como áudio para análise [2].

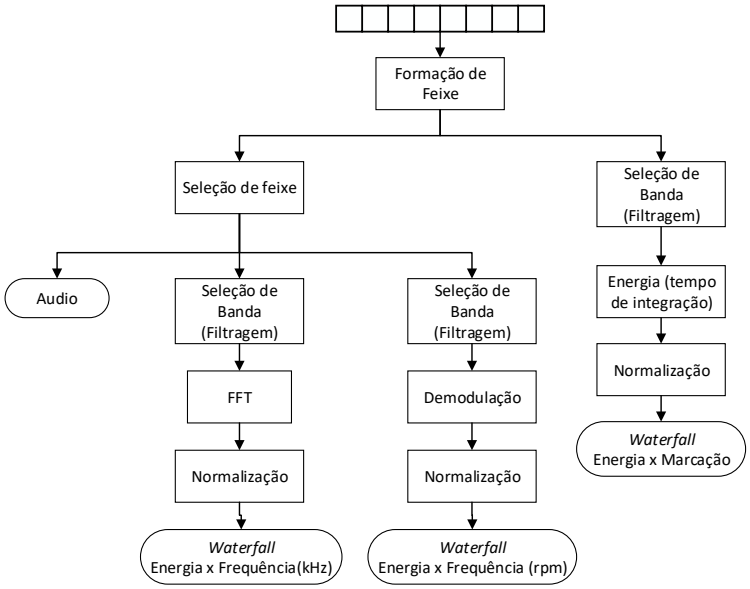


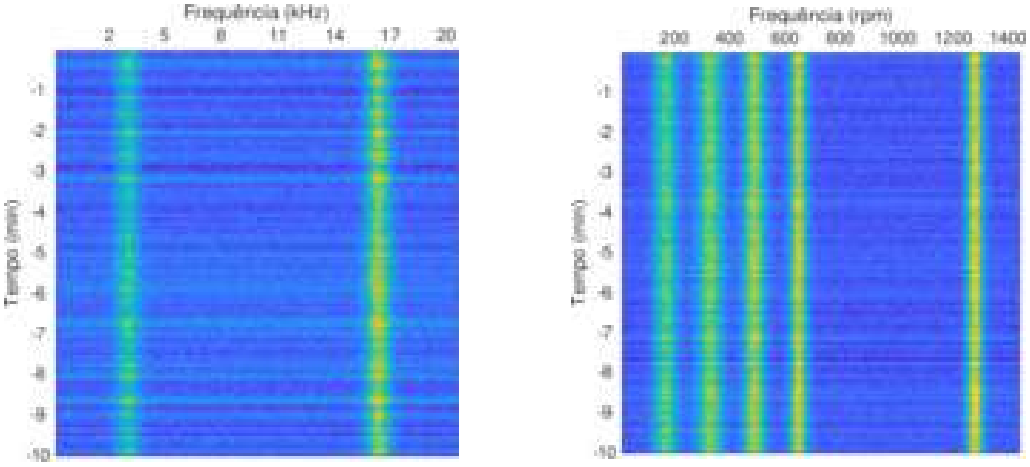
Figura 2.3: Cadeia de processamento típica para um sistema sonar passivo.

Sonares passivos são projetados para detectar ruídos irradiados acima do nível de ruído ambiente e do ruído próprio. Ruídos irradiados podem ser classificados como de banda larga ou estreita (ruído tonal). As principais fontes de ruído em navios são os ruídos de operação do maquinário e da movimentação das hélices [2].

A caracterização dos contatos de um sistema sonar passivo é feita através da extração de características do ruído, também chamada de assinatura acústica. Estas características são utilizadas para a classificação dos contatos, seja por experiência do operador ou por comparação com um conjunto de dados anteriormente caracterizado. Para o auxilio da extração destas características são utilizadas duas técnicas as análises *DEMON* (*Detection of Envelope Modulation On Noise*) e *LOFAR* (*Low Frequency Analisys and Recording*) [4].

Ruídos tonais provenientes de embarcações podem ser identificados através de análises espectrais. O *LOFAR* é uma análise espectral em banda estreita, calculada pela transformada de *Fourier* seguida de uma normalização pelo algoritmo *TPSW*

[2], conforme representado na segunda cadeia de processamento da Figura 2.3. A saída do *LOFAR* permite a visualização de linhas, que são as frequências tonais que sobressaem ao ruído. A extração destas frequências são parâmetros para a classificação de contatos. A análise *LOFAR* é exposta em um gráfico de frequência (kHz) em *waterfall* [2]. A Figura 2.4(a) exibe o resultado de um *LOFAR* para um sinal com ruídos tonais em 3 kHz e 16,5 kHz.



(a) Visualização de um *LOFAR*.

(b) Visualização de um *DEMON*.

Figura 2.4: Extração de parâmetros de áudio.

Ruídos de banda larga, como por exemplo, o ruído de cavitação da hélice, podem ser modulados pelo giro das pás. A análise *DEMON* explora a demodulação deste ruído de banda larga para se estimar a velocidade de rotação do eixo e o número de eixos e pás de uma embarcação [2]. A Figura 2.4(b) exibe o exemplo de uma análise *DEMON* para um navio com eixo girando a 180 rpm e quatro pás.

O *DEMON* é calculado filtrando o sinal numa banda de interesse, calcula-se uma demodulação e, por fim, o sinal é normalizado conforme representado na terceira cadeia de processamento da Figura 2.3 [2]. A saída deste processamento também é exibida em um gráfico de frequência (rpm) em *waterfall* [2].

Na saída do conformador de feixe, pode-se, também, calcular a energia recebida de cada direção. Para isso, o sinal é filtrado dentro de uma banda de interesse e é calculada a energia de cada direção em um dado intervalo de tempo, este chamado de tempo de integração. Posteriormente, tal energia é normalizada. A saída deste processamento é, então, exposta em um gráfico de energia em *waterfall* [2]. Esta cadeia de processamento também esta representada na Figura 2.3.

2.1.3 Sonar Ativo

A cadeia de processamento de um sistema sonar ativo é ilustrada na Figura 2.5. Primeiramente, assim como no sistema passivo, os sinais passam por uma confor-

mação de feixe, convertendo o sinal de cada sensor no sinal esperado de cada direção de chegada (DoA). Posteriormente, duas cadeias de processamento são exibidas: a primeira cadeia tem por objetivo gerar o áudio para o operador e a segunda gerar as análises de detecção dos ecos [2].

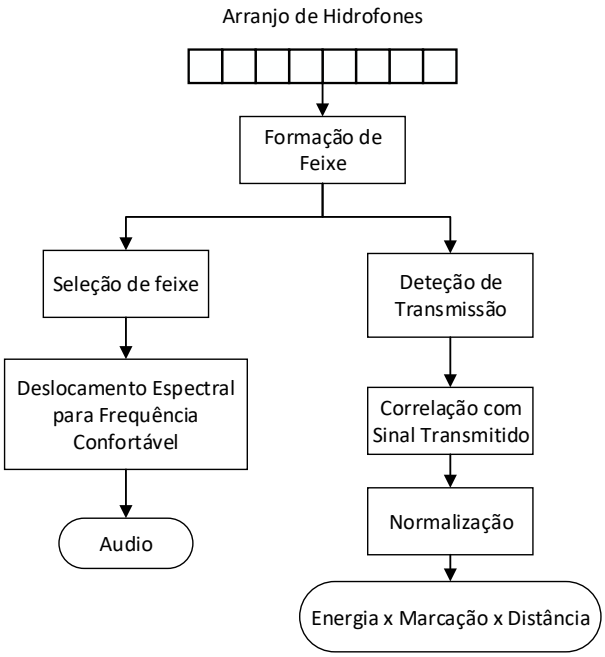


Figura 2.5: Cadeia de processamento típica para um sistema sonar ativo.

Para a obtenção do áudio de uma direção escolhida, deve-se selecionar o feixe equivalente a esta direção e, para prover um maior conforto para o operador do sonar, deslocar a frequência do áudio para uma banda mais confortável à audição humana [2].

Para a cadeia de processamento do eco, o feixe é filtrado em banda estreita na faixa do sinal transmitido e calcula-se a correlação do sinal transmitido e recebido. Após uma normalização, o dado está pronto para visualização [2].

A distância de um contato pode ser calculada por:

$$R = \frac{vt}{2}. \tag{2.4}$$

onde:

R = distância entre plataforma e contato;
v = velocidade de propagação da onda acústica na água;
t = intervalo de tempo entre transmissão e eco.

Vale destacar que a estimativa de distância de um eco é calculada considerando que a onda emitida deve propagar até o contato e, após ser refletida, propagar de volta

ao emissor, por isso a estimativa de distância leva em conta metade do tempo de propagação medido [2].

O eco do sinal transmitido por um sonar ativo é recebido com um deslocamento espectral, devido ao efeito *doppler*. O movimento relativo entre o navio transmissor (plataforma) e o navio reflexor (contato) gera deslocamentos espectrais com valores da ordem de 1,33 Hz a cada m/s de velocidade relativa [2].

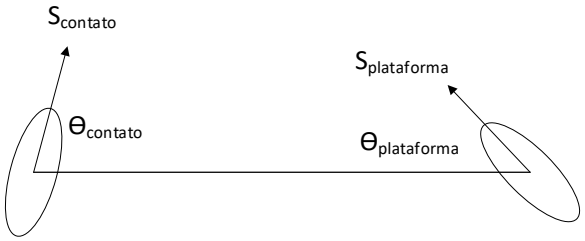


Figura 2.6: Efeito *doppler* ocasionado pelo movimento entre plataforma e contato.

A Figura 2.6 ilustra o movimento da plataforma e de um contato. O movimento da plataforma e do contato geram desvios de frequências que são combinados no sinal retornado. O desvio gerado pelo movimento do navio i pode ser calculado com base na equação:

$$\Delta f_i = \frac{2S_i cos(\theta_i)}{f/v} \tag{2.5}$$

onde:

Δf_i = desvio de frequência causado pelo navio i;
S_i = velocidade do navio i;
θ_i = ângulo entre rumo do navio i e a reta entre a plataforma e o contato;
f = frequência transmitida;
v = velocidade de propagação da onda acústica na água.

O desvio de frequência total será igual a:

$$\Delta f = 2\Delta f_{plataforma} + \Delta f_{contato} \tag{2.6}$$

onde:

Δf = desvio de frequência do eco;
$\Delta f_{plataforma}$ = desvio de frequência causado pela plataforma;
$\Delta f_{contato}$ = desvio de frequência causado pelo contato,

o movimento do navio plataforma causa doppler durante a emissão e a recepção, por isso, o efeito é resultante é dobrado.

Costuma-se utilizar a técnica conhecida como *Own Doppler Nullification* (*ODN*), que consiste em compensar o desvio de frequência causado pelo movimento conhecido da plataforma, para obter uma estimativa da velocidade radial do contato pelo desvio de frequência do sinal recebido [2].

2.2 Detecção

2.2.1 Algoritmos de Clusterização

Algoritmos de clusterização são um conjunto de técnicas de aprendizado não-supervisionado amplamente utilizadas para a análise exploratória de dados [10], visando compreender sua estrutura ou para a redução de dimensionalidade [11]. Estas técnicas dividem os dados em grupos, conhecidos como clusters, por sua similaridade no espaço de dados de entrada, também chamados de espaço de características ou *feature space* [10].

O objetivo das técnicas de clusterização é calcular grupos (clusters) que maximizem a similaridade dos dados dentro de um mesmo cluster (intra-cluster) e minimizem similaridade entre diferentes clusters (inter-clusters), gerando assim grupos de dados maximamente semelhantes entre si e minimamente semelhantes aos demais grupos [12]. As técnicas de clusterização podem ser classificadas sobre vários aspectos [13].

Quanto à arquitetura as técnicas são classificadas em: *homogeneous*, onde todos os dados têm as mesmas *features*; e *heterogeneous*, onde cada dado tem as suas próprias *features* [13].

Quanto a implementação as técnicas podem ser agrupadas em diversos grupos, tais como: hierárquicos, particionais e baseados em densidade [13]. Métodos hierárquicos são divididos em dois tipos, de acordo com a abordagem: divisivos e aglomerativos. Enquanto a abordagem divisiva, inicia a análise com todos os dados em um único cluster e vai subdividindo os clusters a abordagem aglomerativa, inicia a análise com cada dado pertencendo a um único cluster e vai unindo os clusters [13].

Métodos particionais utilizam representantes dos cluster no espaço dos dados e são divididos em: centróide, onde o representante do cluster é o centro de gravidade dos elementos do cluster [13]; e medóide, no qual os representantes dividem o espaço por proximidade, sendo todos os pontos mais próximos de um dado representante que dos demais considerado um ponto daquele cluster [10].

Métodos baseados em densidade classificam os dados dependendo da sua conectividade, contorno e região. Cada elemento é classificado como: ponto de núcleo (*core point*), ponto de borda (*border point*) ou ponto de ruído (*noise point*). Todo

elemento que tenha mais vizinhos que um dado limiar, dentro de uma hiperesfera de semelhança, é considerado um ponto de núcleo. Todo elemento que não é um ponto de núcleo, mas faz parte da vizinhança de um ponto de núcleo é um ponto de borda. Os demais são pontos de ruído [10].

2.2.1.1 *Mean shift Algorithm*

O algoritmo *mean shift*, proposto em [14] e generalizado em [15], é uma técnica de clusterização baseada em busca gradiente ascendente [15]. Na qual o conjunto de pontos (S) é evoluído, iterativamente, para o *sample mean* cálculado para cada elemento (s) do conjunto (S). O conjunto (S) possui n pontos ($s_1, s_2 ... s_n$) independentes e identicamente distribuídos no hiperespaço das *features*, sendo também chamado de malha de busca [14].

O calculo do *sample mean* pode ser feito pela equação:

$$m(s) = \frac{\sum_{x \in X} K(x - s)x}{\sum_{x \in X} K(x - s)} \tag{2.7}$$

onde

$m(s)$ = *sample mean* de s;
X = conjunto de dados a serem clusterizados;
x = um dados pertecente a X;
K = uma função, denominada *kernel*.

Existem diversos *kernels* que podem ser aplicados ao algoritmo, o mais simples é o *flat kernel* [15], determinado pela equação:

$$K(y) = \begin{cases} 0 & \text{if } \|y\| > \gamma \\ 1 & \text{otherwise} \end{cases} \tag{2.8}$$

onde

$K(y)$ = resultado do *kernel* para uma entrada y;
γ = *bandwidth*.

O *flat kernel*, aplicado ao algoritmo, atua como uma selecão dos pontos dentro de uma hiperesfera, centrada no ponto s, de hiper-raio γ [16]. A diferença $m(s) - s$ é conhecida como *mean shift* e dá nome à técnica [15]. Ao final do algoritmo, cada ponto de convergência, local para onde os elementos de S convergiram, é considerado o centro de um cluster. Os elementos pertencentes a X que estejam dentro do *bandwidth* do centro de um cluster são os elementos daquele cluster [16].

O Algoritmo 1 descreve a implementação do *mean shift*, adaptado de [16]:

Algoritmo 1: Implementação do algoritmo *mean shift* .

do

  $\Delta_{max} = 0$; ▷ Maior *mean sample* da iteração

  foreach $s \in S$ **do**

    $s_{next} = m(s)$; ▷ Calcula o *sample mean* de s

    $\Delta_{max} = \max(\Delta_{max}, \|s - m(s)\|)$; ▷ Atualiza maior *mean sample*

    $s = s_{next}$; ▷ Atualiza s para $m(s)$

while $\Delta_{max} \ncong 0$;

O critério de convergência a ser alcançado pode ser representado pelo maior *mean sample* (Δ_{max}), obtido para todos os elementos do conjunto S, atingir valor arbitrariamente próximo de zero. Conforme a malha de busca migra, assumindo a posição do *sample mean*, o maior *mean shift* diminui. A convergência do algoritmo é comprovada para o *flat kernel* [15].

O *mean shift algorithm* é muito utilizado por ser um algoritmo intuitivo, básico e ter uma alta capacidade exploratória dos dados [15].

2.2.1.2 *DBSCAN*

O *Density-Based Spatial Clustering of Applications with Noise* (*DBSCAN*) considera a densidade (q) de um ponto x como sendo o número de elementos que cercam x dentro de uma hiperesfera de raio ϵ centrado em x no espaço das *features*, ou seja, o número de elementos cuja dissimilaridade a x seja menor do que um fator ϵ [17].

Conforme descrito na Seção 2.2.1, os pontos são classificados como:

ponto de núcleo: densidade maior ou igual a q.

ponto de borda: densidade menor que q e vizinho de um ponto de núcleo.

ponto de ruído: densidade menor que q e não vizinho de um ponto de núcleo.

A Figura 2.7 ilustra estes conceitos aplicados a um conjunto de dados de exemplo. Em preto, os pontos de núcleo, em azul os pontos de borda e em verde os pontos de ruído, ϵ está ilustrado pelos círculos centrados nos pontos e o parâmetro q é três.

Note que dentro dos círculos centrados nos pontos pretos temos ao menos q outros pontos. Já nos círculos centrados nos pontos azuis, temos menos que q pontos, mas os pontos azuis estão dentro dos círculos de algum ponto preto. Enquanto isso, os pontos verdes não possuem q pontos nos seus círculos, nem fazem parte dos círculos de pontos pretos. Nesse caso teria-se um único cluster no *dataset* composto pelos pontos pretos e azuis.

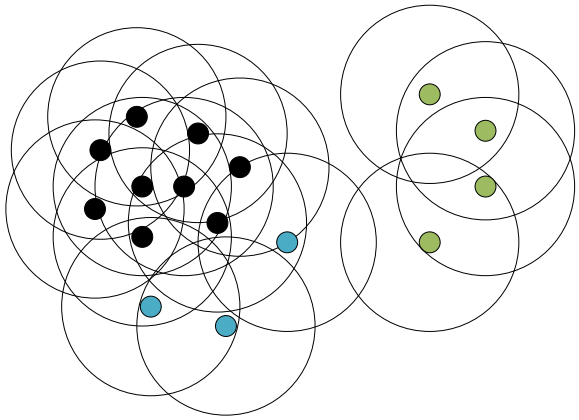


Figura 2.7: Classificação de pontos pelo *DBSCAN*. Em preto, os pontos de núcleo; em azul, os pontos de borda; e em verde, os pontos de ruído.

O *DBSCAN* é altamente sensível à alteração dos parâmetros q e ϵ. Entretanto, é insensível à ordem de apresentação dos elementos, fazendo com que um cluster possa ser selecionado a partir de qualquer um de seus elementos. Esta propriedade é conhecida como densamente conectado e implica que todos os elementos de um cluster estão conectados a todos os outros elementos do cluster através de uma sucessão de pontos menos distantes entre si que ϵ. Esse conceito está ilustrados na Figura 2.8, onde o ponto A e o ponto B estão densamente conectado, pois existe um caminho de ponto a ponto do conjunto de dados com distâncias menores que ϵ que os une [17].

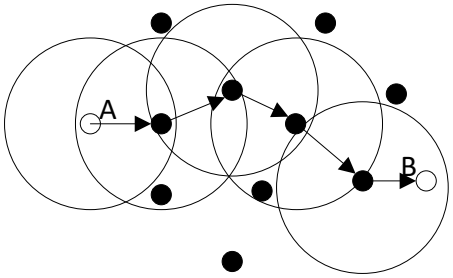


Figura 2.8: densamente conectado.

O algoritmo de *DBSCAN* pode ser descrito por um *loop* executado até que todos os pontos do conjunto de dados tenham sido analisados. Cada iteração se inicia pela seleção arbitrária de um elemento x pertencente ao conjunto de dados X. Depois, seleciona-se o subconjunto de X cujos pontos sejam densamente conectados a x, todos esses pontos são agrupados como um possível cluster e removidos do conjunto

de dados de analise. Se dentro desse conjunto de pontos existir pelo menos um ponto de núcleo esse conjunto é considerado um cluster, caso contrário todos os pontos são considerados ruídos.

2.2.2 Clusterização no Espaço Tempo

Clusterização espaço-temporal é a identificação de clusters móveis em um conjunto de dados ao longo do tempo. Clusterização espaço-temporal apresenta problemas únicos para a clusterização. O mais crítico deles é a intersecção de clusters. Quando os elementos se movem ao longo do tempo, um cluster pode se sobrepor a outro por algum tempo, tornando dificil associar os clusters antes da sobreposição àqueles depois da sobreposição. Este processo é conhecido como *merge and split* [18].

Técnicas de clusterização tradicionais funcionam bem com dados estáticos que contém clusters bem separáveis, tendendo a unir clusters quando eles se interceptam. Conjunto de dados temporais contém clusters que, em alguns momentos, são bem separados, mas em outros se sobrepõem. É interessante que algoritmos de clusterização espaço-temporais sejam capazes de lidar com pequenos períodos de sobreposição [18].

Muitas técnicas de clusterização se baseiam no cálculo de similaridade ou dissimilaridade. A forma mais comum de calcular essa similaridade é um cálculo de distância ponderada no espaço dos dados, como a distância de *Mahalanobis* [11]. Para aplicações espaciais, costuma-se utilizar o caso particular da distância Euclidiana [18].

Em [19] é proposto um método para detecção de cluster baseado na ideia de *Moving Micro-Cluster*, uma extensão do conceito de *micro-cluster* do algoritmo de *BIRCH* [20].

Os algoritmos de clusterização espaço-temporal aplicam técnicas tradicionais de clusterização em um quadro temporal, criando um conjunto de *micro-clusters*. A posição e o movimento dos *micro-clusters* podem ser utilizados para predizer quando os *micro-clusters* irão sofrer *merge and split* [18].

2.3 Acompanhamento

2.3.1 *Competitive Learning*

Realizar a evolução temporal dos clusters gerados na etapa de detecção exige técnicas capazes de realizar o processamento *online*, que não necessitem de todos os dados para serem utilizados, que possam receber os dados um a um (ou bloco a bloco) e que realizem o processamento conforme os dados sejam entregues pelos sensores [11].

Competitive Learning são as técnicas que se utilizam um conjunto de unidades (células) para representar os dados. Cada célula compete por cada novo dado de entrada. A célula vencedora é atualizada e as demais mantidas. Estas técnicas também são conhecidas por *winner-take-all* [11].

A estratégia de *Competitive Learning* pode ser utilizada para realizar uma clusterização *online* e tem as seguintes vantagens: exige menos memória, reduz a demanda computacional e têm capacidade de se adaptar automaticamente às variações temporais dos dados [11].

2.3.2 *Adaptative Resonance Theory* (*ART*)

Existem diversas técnicas para, clusterização *online*, tais como: *online k-means*, *adaptative resonance theory* (*ART*) e *self organized map (*SOM*)* [11]. Para a aplicação deste trabalho, são necessárias técnicas do tipo incremental, nas quais as células são adicionadas conforme a necessidade, pois o número de células não é conhecido a priori. Cada célula, neste caso, será a representação de um contato. Nesta seção será explicado o algoritmo *ART*, que servirá de base para o estrutura do acompanhamento deste projeto.

O algoritmo *ART* é uma técnica incremental, na qual cada célula representa um ponto no espaço de dados. Baseia-se nos conceitos de: raio de vigilância, p, que é a região de atuação da célula; e o fator de atualização, η, que é um número entre 0 e 1 que define como a posição da célula é atualizada [11].

O algoritmo calcula a menor distância entre o dado de entrada, x, e cada célula do conjunto C, $c_0, c_1, ...c_k$ por:

$$\phi_i = \underset{c \in \mathcal{C}}{\operatorname{argmin}} ||c_i - x||_2 \tag{2.9}$$

onde:

ϕ_i = menor distância entre x e as células existentes;
c_i = posição da i-ésima célula;
C = conjunto de células;
x = dado corrente,

e atualiza o conjunto seguindo:

$$\begin{cases} c_{k+1} = x, & se\, \phi_i > p \\ c_i = c_i + \eta(x - c_i), & casocontrrio \end{cases} \tag{2.10}$$

onde

c_i = posição da i-ésima célula;
k = número de células existentes;
x = dado corrente;
ϕ_i = menor distância entre x e as células existentes;
p = raio de vigilância;
η = fator de atualização,

ou seja, caso ϕ_i seja menor ou igual a p, a célula c_i é atualizada pela combinação de c_i e x segundo o fator de atualização. Quando o fator de atualização é 0, a nova posição de c_i é igual à posição original, não sofrendo atualização. Quando o η é 1, a nova posição de c_i é a posição do novo dado x. Para valores intermediários, a posição será uma interpolação ponderada da posição original e da posição x. Caso ϕ_i seja maior que p, nenhuma das células atuais pode competir pelo ponto x. Portanto, uma nova célula c_{k+1} é criada na posição de x.

A Figura 2.9 exemplifica a evolução temporal do algoritmo *ART* após a entrada de dois novos dados. No momento inicial, quadro superior esquerdo, tem-se três células representadas por pontos pretos e seus respectivos raios de vigilância, representados por círculos centrados nesses pontos. O ponto em verde representa um dado que está sendo inserido na rede neste instante. As distâncias entre as células existentes e o novo ponto estão destacadas. Como a menor das distâncias é maior que o raio de vigilância da célula mais próxima, conforme descrito na Equação (2.10), uma nova célula é criada, resultando no segundo quadro superior direito.

O terceiro quadro, do meio à esquerda, ilustra a inserção do segundo dado, representado em verde. Novamente, as distâncias são destacadas. Como a menor distância entre as células e o novo ponto é menor que o raio de vigilância, então, a célula mais próxima é declarada vencedora e sua posição será atualizada conforme a Equação (2.10). O quarto quadro, do meio à direita, ilustra esta atualização, em termos do fator de atualização (η). Se o fator for 0, a nova posição seria a posição original da célula. Se for 1, a célula se move para a nova posição. Para qualquer fator intermediário, a célula se deslocará na reta que une ambos os pontos. O quinto quadro ilustra a posição final da rede, após a entrada destes dois novos pontos.

2.3.3 Análise de Movimento dos Contatos

Conforme discutido na Seção 2.3.2, o acompanhamento deste trabalho será feito com base na lógica do algoritmo *ART*. Realizar a atualização da posição dos acompanhamentos através da interpolação da posição atual da célula e da nova medição faz muito sentido para um espaço de *features* genérico. Entretanto, para uma aplicação espaço-temporal, pode-se analisar o movimento dos contatos para se estimar a posição futura das células com base no seu histórico de movimento. Com isso, pode-se

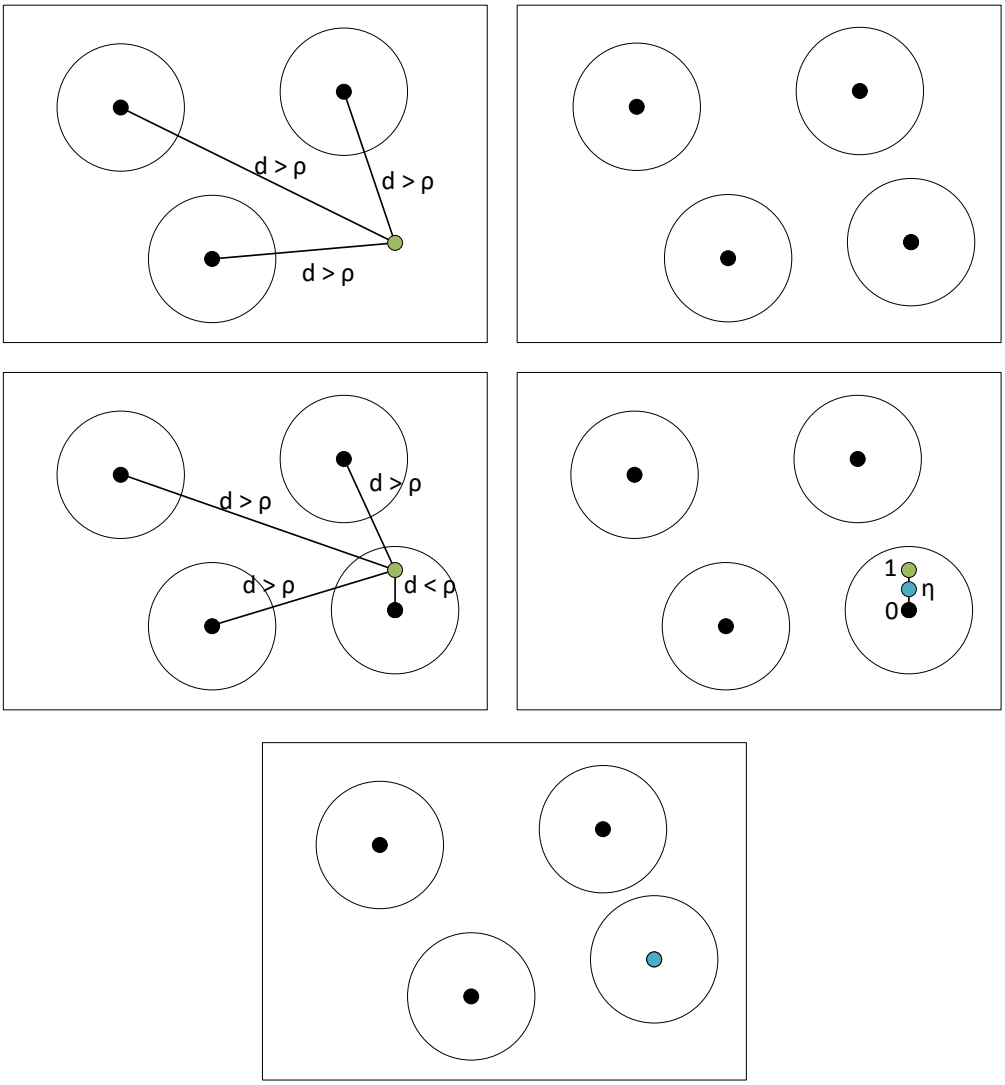


Figura 2.9: Evolução do algoritmo ART para um exemplo arbitrário.

modificar o algoritmo para calcular as distância e os raios de vigilância com base na estimativa de posição e não na posição atual da célula.

Filtros preditivos, como α-β, podem ser utilizados para obter estimativas das variáveis de um modelo de dinâmica do sistema [21]. Um modelo de dinâmica do sistema para um movimento uniforme pode ser definido pelas equações:

$$x_{n+1} = x_n + T\dot{x}_n \tag{2.11a}$$

$$\dot{x}_{n+1} = \dot{x}_n \tag{2.11b}$$

onde

x_{n+1} = estimativa da posição de tempo discreto $n+1$;
x_n = posição no instante em n;
T = intervalo de tempo entre as amostras n e $n+1$;
$\dot{x}_n$ = derivada das posições (velocidades) em n;
$\dot{x}_{n+1}$ = estimativa da derivada das posições em $n+1$.

O modelo de dinâmica do sistema permite que sejam feitas previsões de um estado futuro com base em estimativas de posição e velocidade. Os filtros preditivos realizam o cálculo e a atualização das estimativas de posição e de velocidade através da observação da posição ao longo do tempo. É possível, também, estender o modelo de dinâmica do sistema para movimento uniformemente variado ou movimentos não uniformes [21].

2.3.4 Filtro α-β

Filtro α-β, ou seu análogo g-h, são filtros preditivos que corrigem suas estimativas de posição e velocidade através de medições da posição. As equações de atualização do filtro podem ser descritas com base nas equações a seguir, adaptadas de [21]:

$$x_{n,n} = x_{n,n-1} + \alpha(y_n - x_{n,n-1}) \tag{2.12a}$$

$$\dot{x}_{n,n} = \dot{x}_{n,n-1} + \beta(\frac{y_n - x_{n,n-1}}{T}) \tag{2.12b}$$

onde

$x_{n,n}$ = estimativa da posição no instante n realizada no instante n, ou seja, realizada após a medição da posição em n;
$x_{n,n-1}$ = estimativa da posição no instante n realizada no instante $n-1$ através da Equação (2.11a);
α = fator de atualização da posição;
y_n = medição da posição no instante n;
$\dot{x}_{n,n}$ = estimativa da velocidade no instante n feita no instante n, ou seja, feita após a medição da posição em n;
$\dot{x}_{n,n-1}$ = estimativa da velocidade no instante n feita no instante $n-1$ através da equação (2.11b);
β = fator de atualização da velocidade;
T = intervalo de tempo entre as amostras n e $n-1$.

O filtro α-β, portanto, é atualizado conforme o Algoritmo 2, a cada instante de tempo n, no qual uma nova medição de posição y_n é recebida.

Vale ressaltar que o caso particular onde $\alpha = \eta$, $\beta = 0$ e $\dot{x}_0 = 0$, teremos o caso de atualização de posição do algoritmo ART, no qual a estimativa de posição futura

Algoritmo 2: Atualização de α-β .

Input: $y_n, x_{n-1,n-1}, \dot{x}_{n-1,n-1}, T$

$x_{n,n-1} = x_{n-1,n-1} + T\dot{x}_{n-1,n-1}$; ▷ `equação (2.11a)`

$\dot{x}_{n,n-1} = \dot{x}_{n-1,n-1}$; ▷ `equação (2.11b)`

$x_{n,n} = x_{n,n-1} + \alpha(y_n - x_{n,n-1})$; ▷ `equação (2.12a)`

$\dot{x}_{n,n} = \dot{x}_{n,n-1} + \beta(\frac{y_n - x_{n,n-1}}{T})$; ▷ `equação (2.12b)`

é a posição atual e a atualização da posição atual é a uma interpolação da posição anterior e a atual. Além disso, o fator β tem função análoga ao fator de atualização, só que aplicada ao cálculo de velocidade e não de posição.

O filtro α-β com parâmetros fixos é capaz de acompanhar, sem erro, movimentos do tipo uniforme e com erro fixo movimentos uniformemente variados, assumindo que as medições de posição tenham erros de média nula [21].

Até este ponto, foi demonstrado como evoluir um filtro α-β dadas as estimativas feitas em um dado momento $n-1$ para um momento n após a realização de uma nova medição y_n. Entretanto, falta descrever o processo de inicialização do filtro, ou seja, de como fazer as estimativas iniciais de posição, x_{n_0,n_0} e velocidade, $\dot{x}_{n_0,n_0}$.

O filtro α-β precisa ser inicializado com erro das estimativas iniciais suficientemente pequeno para garantir a estabilidade do filtro, o que vai variar de acordo com o cenário [21].

Nesta dissertação, propõe-se iniciar um acompanhamento baseado no algoritmo *ART* por não depender de estimativas iniciais. Após n medições, pode-se validar o acompanhamento e, utilizando-se os n primeiros pontos, realizar uma estimativa da próxima posição e, consequentemente, uma estimativa da velocidade atual.

Capítulo 3

Projeto

3.1 Cenário

3.1.1 Simulador

O projeto e as análises deste trabalho foram feitas com base em um simulador de contatos desenvolvido para esta dissertação. No simulador, cada navio é representado pelas seguintes variáveis de estado: a posição no espaço cartesiano x e y (yd); a direção de navegação, comumente chamada de rumo θ (°); e a velocidade v (yd/ciclo).

A evolução temporal dos navios (dinâmica) é controlada através das entradas: guinada, que é a velocidade angular do rumo, ω; e aceleração linear a. A evolução das variáveis de estado pelas entradas se dá através das seguintes equações:

$$x_n = x_{n-1} + \Delta t \; v_{n-1} \; cos(\theta_{n-1}) \tag{3.1a}$$

$$y_n = y_{n-1} + \Delta t \; v_{n-1} \; sin(\theta_{n-1}) \tag{3.1b}$$

$$\theta_n = \theta_{n-1} + \Delta t \; \omega_n \tag{3.1c}$$

$$v_n = v_{n-1} + \Delta t \; a_n \tag{3.1d}$$

onde:

x_n = posição no eixo horizontal no n-ésimo passo de simulação;
Δt = intervalo de tempo entre as amostras $n-1$ e n, ou seja, um ciclo de simulação;
v_n = velocidade no n-ésimo passo de simulação;
θ_n = rumo no n-ésimo passo de simulação;
ω_n = guinada no n-ésimo passo de simulação;
a_n = aceleração linear no n-ésimo passo de simulação.

Todos os cenários são iniciados com todos os navios em posições, rumos e velocidades aleatórias. Buscando-se simular um cenário de operação real de navios, ao longo da simulação, aleatoriamente, podem ser sorteados novos rumos e velocidades desejados para cada navio. Ao serem sorteados novos rumos e velocidades, são realizadas as guinadas e as acelerações necessárias para atingi-los e essas guinadas e acelerações são aplicadas as entradas dos modelos daquele cada navio.

O simulador define uma região para todos os navios permanecerem durante a simulação, quando um navio se aproxima de uma fronteira em rumo de saída é forçado um novo rumo de interesse oposto ao rumo atual do navio, mantendo-o dentro da região de interesse. Foram consideradas como região de interesse neste trabalho círculos de raio: 6 kyd, para os contatos; e 2 kyd, para o navio plataforma, de forma a evitar distâncias maiores que 8 kyd, por ser essa a distância analisada na parte ativa das simulações.

As análises ativas e passivas foram feitas de forma independente, portanto o simulador gera para as análises: de detecção uma malha de pontos; e de acompanhamento um conjunto de detecções simuladas. Com isso as etapas podem ser analisadas separadamente evitando que erros da detecção interfiram na análise de acompanhamento.

O simulador gera pontos para detecção e detecções simuladas, com base no cálculo das coordenadas, marcações e distâncias, relativas à posição atual da plataforma, a fim de representar os dados no mesmo espaço que os dados reais do sonar. As coordenadas são sempre calculadas de forma absoluta, para facilitar o acompanhamento dos dados e evitar que as guinadas da plataforma influenciem bruscamente na dinâmica dos contatos. As marcações serão referenciadas na direção do eixo x positivo, analogamente ao que é feito para o norte verdadeiro na navegação. O cálculo da marcação e da distância de um contato c no instante de tempo n é feito pela equação:

$$d(c,n) = \sqrt{(x_{c,n} - x_{p,n})^2 + (y_{c,n} - y_{p,n})^2} \tag{3.2a}$$

$$\Theta(c,n) = arctan(\frac{y_{c,n} - y_{p,n}}{x_{c,n} - x_{p,n}}) \tag{3.2b}$$

onde

$d(c,n)$ = distância entre o contato c e a plataforma p no instante de tempo n;
$x_{c,n}$ = posição x do contato c no instante de tempo n;
$y_{c,n}$ = posição y do contato c no instante de tempo n;
$x_{p,n}$ = posição x da plataforma no instante de tempo n;
$y_{p,n}$ = posição y da plataforma no instante de tempo n;
$\Theta(c,n)$ = marcação do contato c no instante de tempo n.

Para análises de detecção foram gerados dados a cada 100 ms de dinâmica de movimento. No caso do sonar passivo, uma curva com o valor de energia para cada marcação de 0° a 360° com passo de 1°. Já no caso do sonar ativo, uma superfície com o valor de energia para cada marcação 0° a 360° com passo de 1° e distância de 0 kyd a 8 kyd com passo de 10 yd.

Essa curva ou superfície, entrada da etapa de detecção, é considerada simulação da saída do processamento discutido na seção 2.1 e equivale a uma curva de ruído gaussiano somada a energia de cada contato. Cada contato é representado por uma pico gaussiano com um dado desvio padrão, este desvio é aleatório e característico de cada contato. No caso do sonar passivo, um desvio padrão entre 3° e 10°, já no caso do sonar ativo, uma gaussiana bidimensional com desvio padrão de marcação entre 10° e 30° e desvio padrão de distância entre 100 yd e 500 yd. Todos os contatos tem o mesmo valor de pico das curvas gaussianas e a relação sinal-ruído é controlada pelo ruído de fundo.

Para análises de acompanhamento foram gerados dados a cada 30 s de dinâmica de movimento. Foram gerados pontos equivalentes a saída da etapa de detecção, portanto, um conjunto de pontos com *features*: no caso do sonar passivo, marcação e energia; no caso do sonar ativo, marcação, distância e energia. O simulador foi projetado com os parâmetros:

Falsos negativos: Probabilidade de não gerar o ponto equivalente a um contato, sendo avaliada para cada contato a cada geração de forma independente.

Falsos positivos: Número de pontos gerados a cada ciclo de simulação, além dos pontos dos contatos.

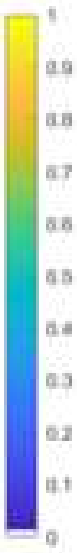
3.1.2 Tipos de dados e visualizações

As detecções, para os sistemas passivos, são um conjunto de pontos com as *features*: marcação, em graus (°); e energia, em decibéis (dB). Essas detecções serão visualizadas como ilustrado na Figura 3.1(a), onde o eixo horizontal representa a marcação, enquanto a energia é representada pela intensidade de cor segundo a escala de cor da Figura 3.1(b). A Figura 3.1(a) exemplifica um cenário com três contatos nas marcações: 70°, 120°, e 315°.

Para visualização de clusters sobre gráficos de detecção, serão utilizadas visualizações como a da Figura 3.2, no qual, cada cluster é representado por um circulo vermelho sobreposto à visualização das detecções.

As detecções para sonar passivo podem ser visualizadas com histórico, conforme descrito na Seção 2.1, em um gráfico de marcação x tempo x energia em *waterfall*. A Figura 3.3 ilustra esta modalidade de visualização em um cenário com três contatos cujas marcações se modificam ao longo do tempo.

(a) Visualização passiva de detecções.



(b) Escala de cor normalizada.

Figura 3.2: Visualização passiva de detecções e clusters.

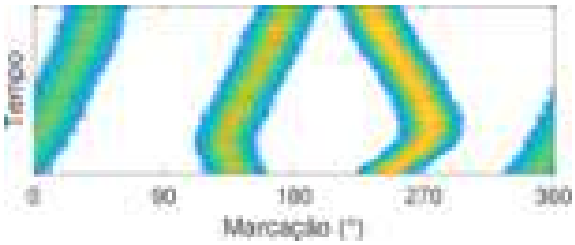


Figura 3.3: Visualização passiva de detecções em *waterfall*.

Para a visualização dos acompanhamentos sobre gráficos de energia em *waterfall*, serão utilizados gráficos como o da Figura 3.4, nos quais cada acompanhamento é representado por uma linha sobreposta à visualização das detecções, sendo o círculo a posição atual daquele acompanhamento.

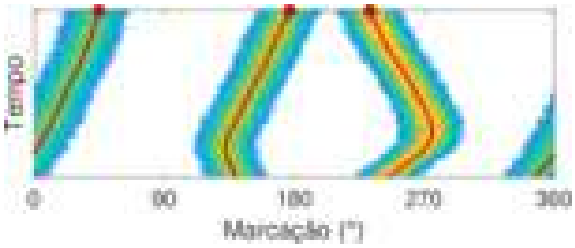


Figura 3.4: Visualização passiva de detecções e clusters em *waterfall*.

As detecções para sistemas ativos são um conjunto de pontos com *features*: marcação, em graus (°); distância, em jardas (yd); e energia, em decibéis (dB). As detecções serão convertidas para as coordenadas retangulares antes de serem clusterizados, pois, na maioria dos casos, as técnicas de clusterização aplicadas ao sistema de coordenadas cartesianas têm melhor performance do que quando aplicadas ao sistema de coordenadas polares [22].

Estas detecções convertidas serão visualizadas como na Figura 3.5, onde os eixos cartesianos representam a distância em jardas (yd) e a energia é representada pela mesma escala de cor do passivo, conforme ilustrado na Figura 3.1(b). No exemplo da Figura 3.5, tem-se um cenário com cinco contatos ao redor da plataforma, vale destacar que dois destes contatos tem uma sobreposição nas suas regiões de influência.

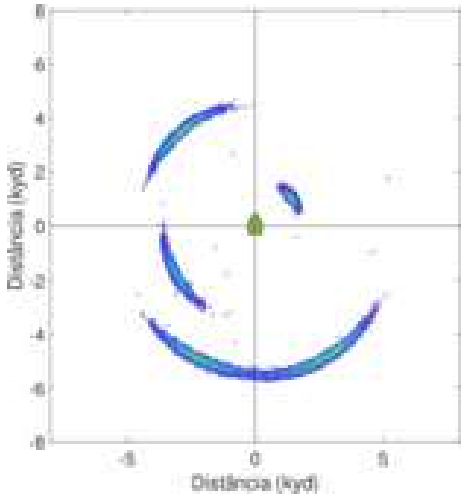


Figura 3.5: Visualização ativa de detecções.

Para a visualização de clusters sobre gráficos de detecção, serão utilizados gráficos como o da Figura 3.6, nos quais, cada cluster é representado por um circulo sobreposto à visualização das detecções.

Em um sistema sonar ativo real existe uma região próxima ao navio, na qual não é possível realizar detecções. Isso ocorre porque o sonar realiza a transmissão com os mesmos transdutores que a recepção, por isso, durante a transmissão e o transiente que se segue o sinal amostrado dos transdutores não permite um cálculo real de detecção. Para simular este efeito, as detecções deste projeto só serão geradas em distâncias maiores que 1 kyd da plataforma. Essa distância está sobre-estimada, mas não impacta significativamente nos resultados das simulações. A Figura 3.7 ilustra esse efeito. Com um limiar de detecção baixo, é fácil notar o circulo de 1 kyd de raio, centrado na plataforma, no qual não são geradas detecções.

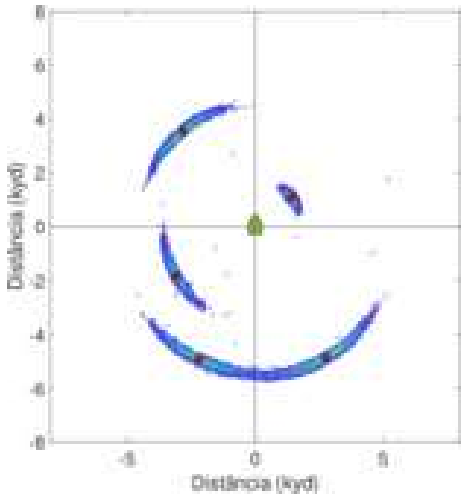


Figura 3.6: Visualização ativa de detecções e clusters.

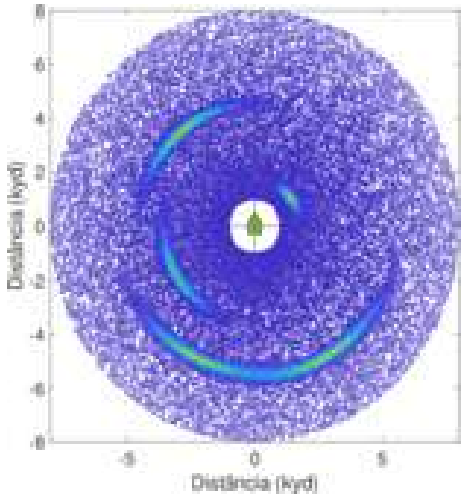


Figura 3.7: Visualização ativa de detecções.

3.2 Detecção

A detecção é a etapa em que um quadro temporal é convertido em clusters. A entrada de dados para a detecção é um mapa de marcação por energia, para sonar passivo; ou, no caso do sonar ativo, distância por distância por energia (coordenadas cartesianas após a conversão das coordenadas polares geradas pelo simulador). Esse mapa precisa ser convertido na matriz de pontos de interesse (detecções). Idealmente, apenas um ponto para cada contato.

Esse processo é feito pelo ceifamento do mapa para valores de energia acima do limiar de detecção (λ). Entretanto, isso gera um conjunto de pontos ao redor da posição dos contatos e é necessário aplicar uma técnica de identificação de grupos de detecções que representem o mesmo contato e reduza essas detecções iniciais para se aproximar da detecção ideal.

Para a aplicação em foco neste trabalho, precisa-se de algoritmos de clusterização

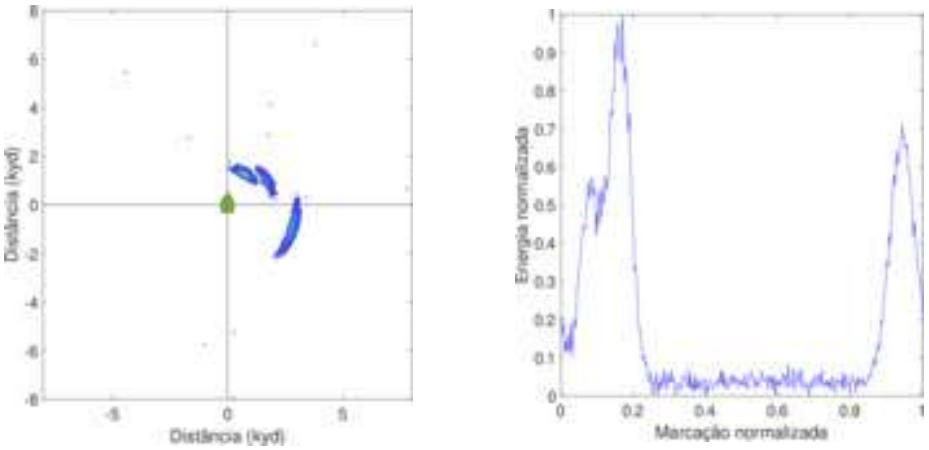
que não necessitem do estabelecimento do número de clusters a *priori*. Portanto, foram utilizadas as técnicas de *Mean-Shift Clustering* e *DBSCAN* por sua grande capacidade exploratória, por não assumirem uma conformação espacial para os clusters a *priori*, por serem simples e por serem usuais.

3.2.1 Problemas da Clusterização

As técnicas de clusterização difundidas na literatura calculam as similaridades com base em distâncias no espaço dos dados, no espaço de *features*. Na aplicação em estudo, tem-se um conjunto de dimensões espaciais e uma dimensão de energia. Tem-se a princípio duas opções para aplicar as técnicas de clusterização clássicas a estes dados: aplicar uma clusterização desprezando a dimensão de energia ou utilizar a dimensão de energia junto das dimensões espaciais.

Utilizar a energia no cálculo da similaridade gera dois problemas. Primeiramente, o peso da *feature* energia em relação às dimensões espaciais, o que implicará em mais um parâmetro para ajuste e otimização da clusterização. Em segundo lugar, um problema mais grave, utilizar a energia na conta da similaridade equivale a uma distorção no espaço, o que faz pontos de energia similares parecerem mais próximos do que estão, o que pode gerar clusters não contíguos espacialmente.

Pode-se ilustrar esse efeito com o cenário simulado na Figura 3.8, para o qual têm-se três contatos nas seguintes posições (distância∠marcação): 2 kyd∠30°, 1,5 kyd∠60° e 3 kyd∠340°. A Figura 3.8(a) mostra a visão geográfica do cenário, enquanto a Figura 3.8(b) mostra o gráfico de energia por marcação, ambos normalizados entre zero e um, do menor ao maior valor de cada *feature* dentre as detecções.



(a) Visualização geográfica do cenário. (b) Energia por marcação normalizadas.

Figura 3.8: Cenário ilustrativo, energia no cálculo da similaridade.

A Figura 3.9 mostra os passos intermediários da evolução da malha de busca do algoritmo de *mean shift*, representando cada um dos pontos da malha S por um

círculo vermelho sobreposto ao gráfico de energia normalizada. Pode-se ver que a distorção de espaço faz com que a região de proximidade entre os dois picos mais a esquerda atraí o movimento da malha de busca do algoritmo, deslocando o ponto de convergência da marcação do pico.

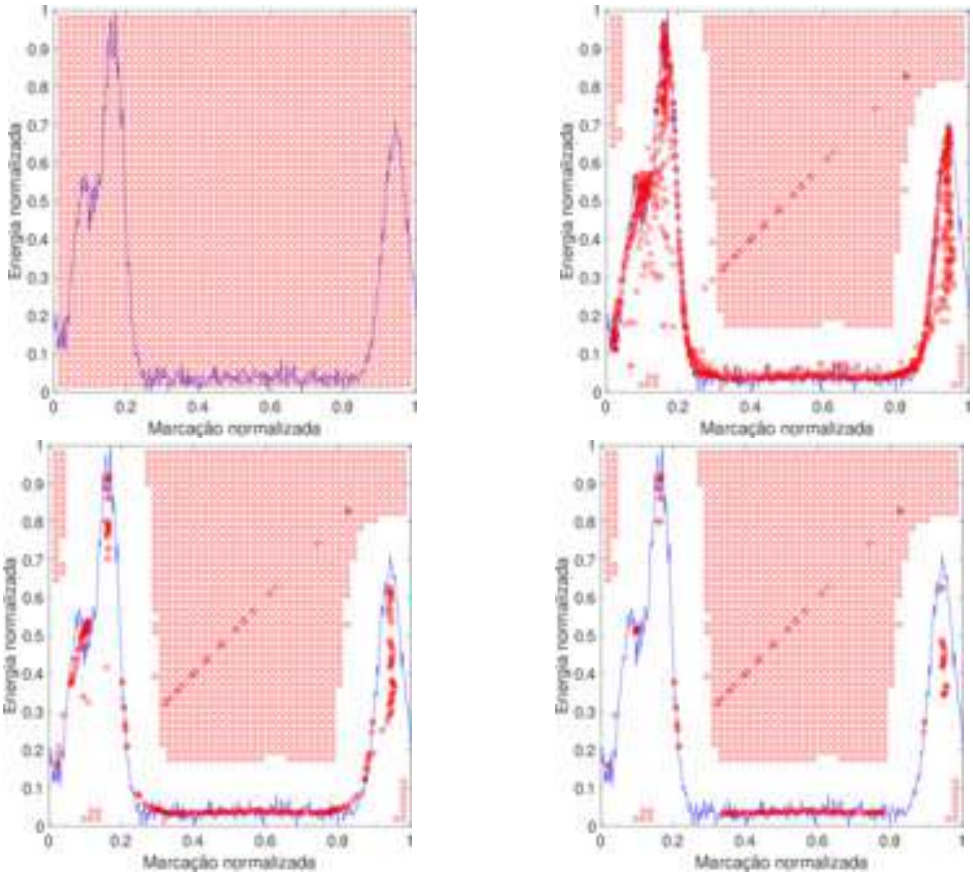


Figura 3.9: Evolução do algoritmo de *Mean-Shift*.

A Figura 3.10 ilustra a posição de convergência da malha, destacadas como um circulo verde onde o cluster gerado na região intermediária dos picos não é contíguo espacialmente, dado que estão dentro do círculo partes de ambos os picos, mas não a região do vale. Na aplicação em análise, clusters não contíguos não fazem sentido físico, por isso, essa opção não será utilizada nessa dissertação.

A outra opção é utilizar as técnicas de clusterização ignorando a dimensão de energia, com isso tem-se outro tipo de problema. A clusterização, apenas com as *features* espaciais, vai realizar a busca da região com maior densidade de pontos. Entretanto, os dados são gerados a partir de uma matriz homogênea de pontos. Para o sistema passivo, um ponto é produzido a cada 1°. Para o sistema ativo, um ponto a cada 1° e a cada 10 yd. Nem todos estes pontos viram detecções, apenas aqueles acima do limiar de detecção (λ). Para um mesmo λ, a quantidade de pontos dependerá da relação sinal-ruído. Em certas configurações, as regiões dos picos ficam com uma disposição de pontos quase homogênea, o que pode atrapalhar a migração

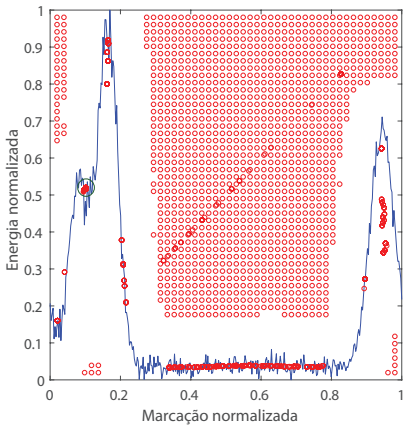
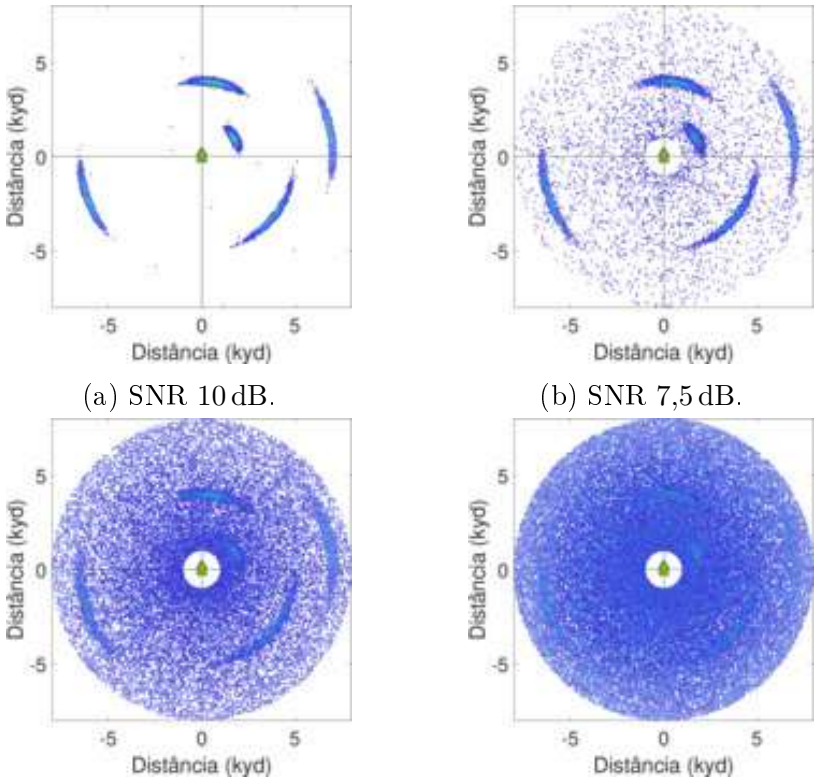


Figura 3.10: Cluster sem continuidade espacial.

do algoritmo.

Para ilustrar, podemos utilizar o cenário simulado da Figura 3.11, no qual temos cinco contatos gerados para o mesmo λ, mas com diferentes SNR: 2,5 dB, 5 dB, 7,5 dB e 10 dB.



(a) SNR 10 dB. (b) SNR 7,5 dB.

(c) SNR 5 dB. (d) SNR 2,5 dB.

Figura 3.11: Cenários ilustrativos, variação da SNR.

O limiar de detecção não é fixo e, tipicamente, é controlado pelo operador do sonar. O balanço do SNR e do λ define indiretamente a quantidade de pontos que se tornam detecções. Os algoritmos de clusterização se valem da densidade dos pontos

para a localização dos centros.

Para certos balanços de *SNR* e do λ, a nuvem de pontos terá densidade como função da distância da plataforma, sendo maior quanto mais próximo do centro, devido à conversão das coordenadas polares para retangulares, como ilustra a Figura 3.12.

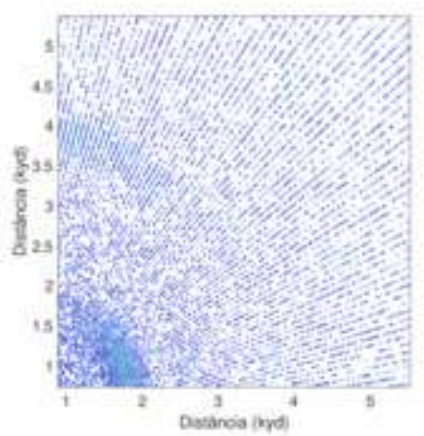


Figura 3.12: Densidade de pontos com SNR baixo.

Com isso, em cenários mais ruidosos, os algoritmos tradicionais de clusterização podem não conseguir localizar os picos. A Figura 3.13 ilustra a clusterização do cenário mostrado na Figura 3.11, com o algoritmo de *mean shift*.

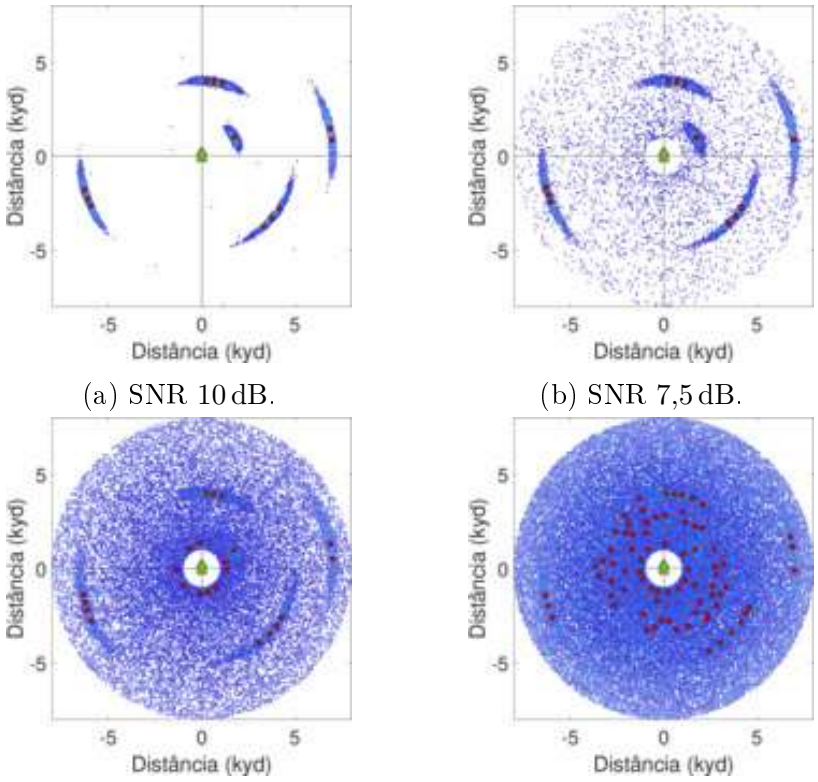
Nessa figura, pode-se ver que o algoritmo *mean shift* tem dificuldades de migrar dentro da nuvem de pontos de um contato, gerando mais de um ponto de convergência para o mesmo contato.

Algoritmos como o *DBSCAN*, que são focados em localizar clusters contíguos independente da sua distribuição, são ainda mais susceptíveis ao aumento da densidade da nuvem de pontos, por vezes, reconhecendo todos os pontos como um único cluster. A Figura 3.14 ilustra a clusterização do cenário da Figura 3.11 com o algoritmo *DBSCAN*.

Pode-se ver que, com valores maiores de *SNR*, o algoritmo consegue isolar corretamente os clusters. Conforme a relação sinal-ruído caí e o número de detecções aumenta, mais as detecções vizinhas aos contatos são incluídas nos clusters. Para o cenário com *SNR* menor que 5 dB, o algoritmo já clusteriza todas as detecções como um único cluster, gerando um cluster praticamente na posição da plataforma.

3.2.2 Modificação proposta

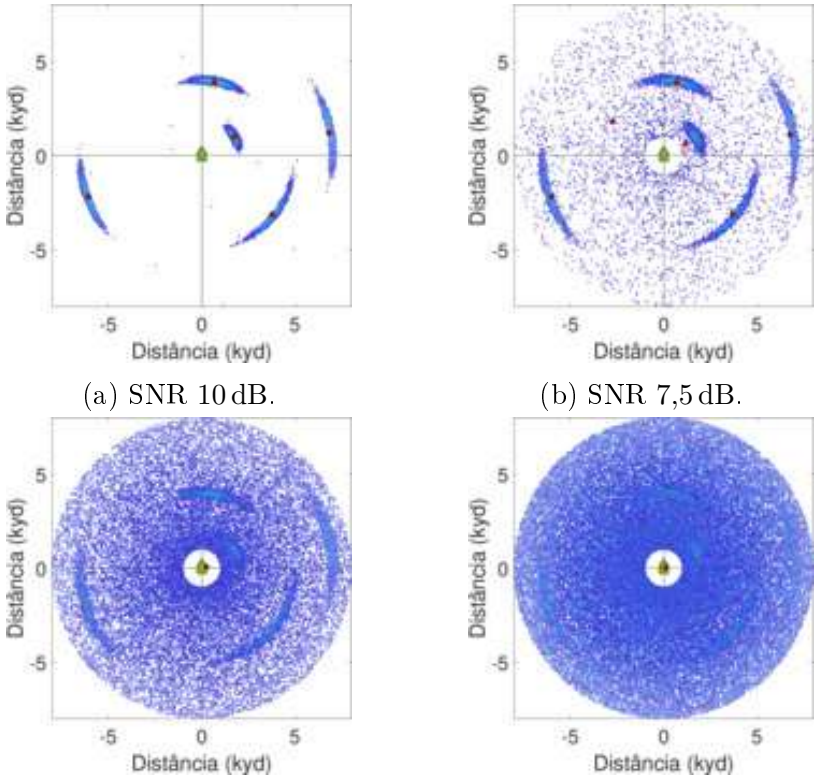
Na aplicação deste trabalho, as técnicas clássicas de clusterização levam aos problemas discutidos na Seção 3.2.1. Por isso, neste trabalho será discutida uma terceira abordagem ao problema do tratamento da energia na clusterização.



(a) SNR 10 dB. (b) SNR 7,5 dB.

(c) SNR 5 dB. (d) SNR 2,5 dB.

Figura 3.13: Efeito da quantidade de detecções no *Mean-Shift Clustering*.



(a) SNR 10 dB. (b) SNR 7,5 dB.

(c) SNR 5 dB. (d) SNR 2,5 dB.

Figura 3.14: Efeito da quantidade de detecções no *DBSCAN*.

A proposta é alterar os algoritmos, levando em conta a energia em momentos de exploração e desconsiderando a energia em momentos de cálculo das vizinhanças. Essa modificação deve ser pensada para cada algoritmo independentemente. Como neste trabalho as técnicas em análise são a *mean shift* e a *DBSCAN*, será proposta a modificação para estas duas técnicas, outras técnicas podem ser igualmente modificadas em trabalhos futuros.

A principal influência que a energia deve ter nos algoritmos de clusterização é influenciar a posição dos centroides e não das vizinhanças, de forma a não gerar clusters descontíguos, mas explorar a região em busca dos picos de energia.

3.2.2.1 *Mean-Shift Clustering*

Conforme explicado na Seção 2.2.1.1 a migração da malha de busca (S) é feita pelo cálculo da média dos pontos ponderados por um *kernel*. No caso do *flat kernel*, isto é equivalente a duas etapas: a verificação da vizinhança, separando os pontos internos à hiperesfera de raio γ, e a atualização da posição para o centroide destes pontos (*sample mean*).

Seguindo o objetivo da modificação proposta na Seção 3.2.2, manteve-se a verificação da vizinhança apenas nas *features* espaciais e a atualização da posição do centroide ponderando as *features* espaciais pela *feature* energia, o que pode ser descrito pela modificação da Equação (2.8) para:

$$m(s) = \frac{\sum_{x \in X} \hat{K}(x, s) p(x)}{\sum_{x \in X} \hat{K}(x, s)} \tag{3.3}$$

onde:

s = ponto da malha de busca, com apenas *features* espaciais;
$m(s)$ = *sample mean* de s;
X = conjunto de dados a serem clusterizados;
x = um dados pertecente a X, com *features* espaciais e energia;
$\hat{K}$ = *kernel proposto* em função de x e s;
$p(x)$ = *features* espaciais de x,

e aplicação do *kernel*:

$$\hat{K}(x, s) = \begin{cases} 0 & \text{if } \|p(x) - s\| > \gamma \\ l(x) & \text{otherwise} \end{cases} \tag{3.4}$$

onde:

$\hat{K}(x, s)$ = *kernel* para uma entrada x e s;
s = ponto com *features* espaciais;
x = ponto com *features* espaciais e energia;
$p(x)$ = *features* espaciais de x;
$l(x)$ = *feature* energia de x;
γ = *bandwidth*.

O cálculo do *sample mean*, combinado com a aplicação do kernel, pode ser reinterpretado para uma única equação de modo a facilitar a compreensão da modificação proposta. Para isso, pode-se considerar que o *flat kernel* realiza a seleção de um subconjunto Y_s dos dados X, ou seja:

$$Y_s \subset X, \forall x \in X \mid \|p(x) - s\| \leqslant \gamma \tag{3.5}$$

sendo

s = ponto com *features* espaciais;
Y_s = subconjunto de X próximos ao ponto s;
x = ponto com *features* espaciais e energia;
X = conjunto de dados a serem clusterizados;
$p(x)$ = *features* espaciais de x;
γ = *bandwidth*.

Logo, Y_s é o subconjunto de X, para todo x pertencente a X, cuja distância entre s e as *features* espaciais de x seja menor ou igual a γ.

O *sample mean* para um ponto s dado o conjunto Y no algoritmo tradicional pode ser calculado como:

$$m(s) = \frac{\sum_{y \in Y_s} p(y)}{\sum_{y \in Y_s} 1} \tag{3.6}$$

onde

s = ponto com *features* espaciais;
$m(s)$ = *sample mean* de s;
Y_s = subconjunto de X próximos ao ponto s;
y = ponto pertencentes ao conjunto Y_s;
$p(y)$ = *features* espaciais de y.

Já o *sample mean* para um ponto s dado o conjunto Y no algoritmo modificado é calculado como:

$$m(s) = \frac{\sum_{y \in Y_s} p(y)l(y)}{\sum_{y \in Y_s} l(y)} \tag{3.7}$$

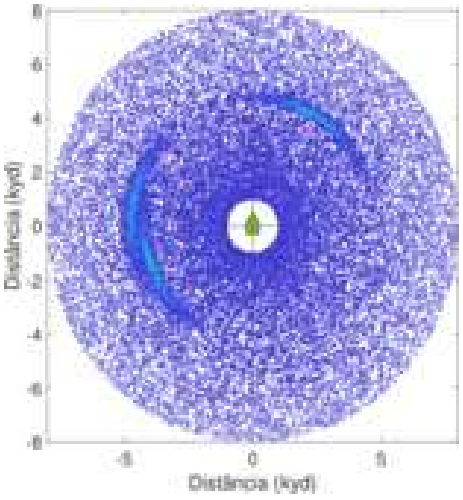
onde

$$
\begin{aligned}
s &= \text{ponto com } \textit{features} \text{ espaciais;} \\
m(s) &= \textit{sample mean} \text{ de } s; \\
Y_s &= \text{subconjunto de } X \text{ próximos ao ponto } s; \\
y &= \text{ponto pertencentes ao conjunto } Y_s; \\
p(y) &= \textit{features} \text{ espaciais de } y; \\
l(y) &= \textit{feature} \text{ energia de } y,
\end{aligned}
$$

deixando explícito que a *feature* energia atua ponderando as *features* espaciais. Assim, a migração da malha de pontos S será atraída para os picos de energia. Na prática, pode ser necessário modificar a *feature* energia para aumentar o efeito da energia na migração e no cálculo do centroide. Por exemplo, substituindo $l(y)$ por $l(y)^2$ na equação.

Pode-se utilizar o cenário da Figura 3.15 para exemplificar o efeito desta modificação, no qual temos três contatos nas posições: 4,8 kyd$\angle$60°, 4,75 kyd$\angle$170° e 4,25 kyd$\angle$200°. A Figura 3.15(a) exibe o cenário na visualização passiva, enquanto a Figura 3.15(b) exibe a visualização ativa. Neste cenário, as dispersões dos dois primeiros contatos acabam se sobrepondo em ambas as visualizações.

(a) Visualização passiva do cenário.

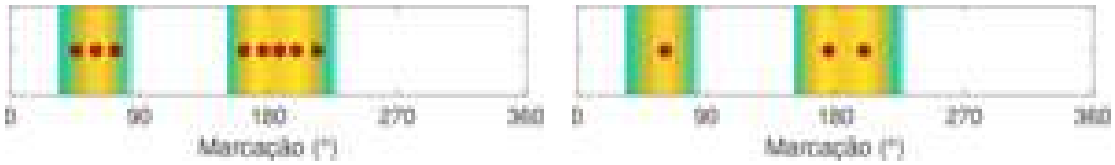


(b) Visualização ativa do cenário.

Figura 3.15: Cenário de análise da modificação no algoritmo *mean shift*.

A Figura 3.16 mostra, na visualização passiva, o resultado da aplicação dos algoritmos *mean shift* tradicional e modificado no cenário da Figura 3.15. A Figura 3.16(a) exibe o resultado da clusterização do algoritmo tradicional e a Figura 3.16(b) do algoritmo modificado.

Com o ruído e o limiar de detecção utilizados, a nuvem de pontos que contém os dois primeiros contatos é bem homogênea na região dos contatos. O algoritmo tradicional não consegue migrar a malha dentro dessas regiões, criando vários clus-

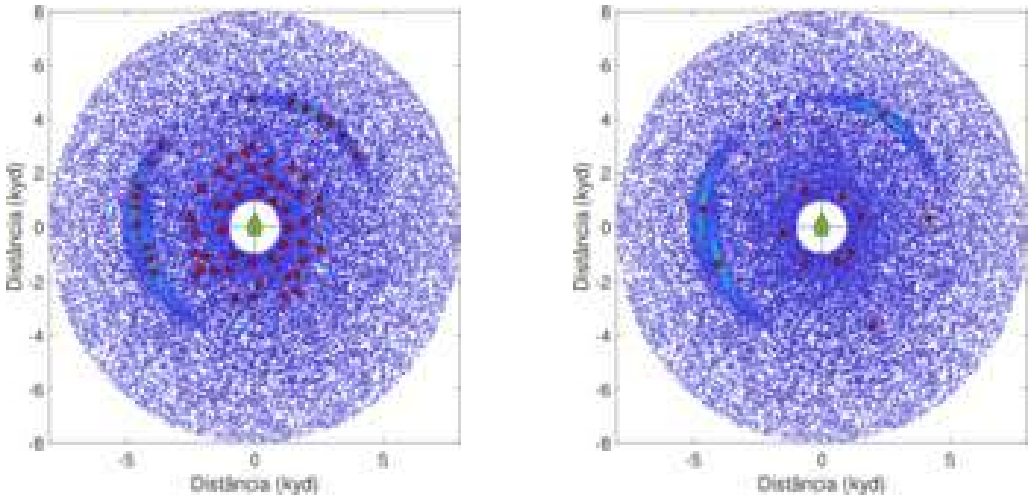


(a) Algoritmo tradicional. (b) Algoritmo modificado.

Figura 3.16: Análise da modificação no algoritmo *mean shift*, visualização passiva.

ters próximos. Já o algoritmo modificado consegue migrar utilizando a informação de energia, apesar do cálculo de vizinhança ignorar esta *feature*.

Ainda neste cenário, efeito similar pode ser visto também na visualização ativa. A Figura 3.17 exibe na visualização ativa o resultado da aplicação dos algoritmos *mean shift* tradicional e modificado para o cenário da Figura 3.15. A Figura 3.17(a) exibe o resultado da clusterização do algoritmo tradicional e a Figura 3.17(b) do algoritmo modificado.



(a) Algoritmo tradicional. (b) Algoritmo modificado.

Figura 3.17: Análise da modificação *mean shift*, visualização ativa.

Pode-se ver que o mesmo efeito do passivo acontece com o algoritmo tradicional no ativo. As regiões de maior concentração de pontos têm distribuições quase homogêneas e o algoritmo tradicional não consegue migrar dentro dessas regiões, criando vários clusters próximos ao longo das dispersões.

Já o algoritmo modificado consegue migrar utilizando a informação de energia. Além disso, como os pontos são gerados em coordenadas polares e utilizados com coordenadas cartesianas, ocorre uma concentração natural dos pontos para distâncias menores. Como consequência, são geradas falsas concentrações abaixo de 3 kyd, aumentando a probabilidade de falsos positivos.

3.2.2.2 *DBSCAN*

Conforme explicado na Seção 2.2.1.2, a busca do algoritmo *DBSCAN* objetiva a localização de clusters pela não interrupção de vizinhanças, ou seja, o cluster se extende enquanto houver vizinhos válidos para quaisquer pontos internos ao cluster.

Para inserir a modificação proposta na Seção 3.2.2, é necessária uma modificação mais significativa do que a da Seção 3.2.2.1. O *DBSCAN* procura vizinhança a partir de pontos aleatórios e só acaba quando não há mais vizinhos válidos, explorando o conceito de *density connected*. Por isso, ele pode ser inicializado de qualquer ponto, sem interferir no resultado final da clusterização.

O Algoritmo 3 descreve a modificação proposta para o *DBSCAN*. Propõe-se, em um primeiro momento, ordenar a lista de dados X em ordem decrescente de energia X_s e utilizar esse conjunto. Iniciar uma variável auxiliar C com o tamanho do número de dados de X que representa um mapa associativo dos clusters. Se o valor de $C(i)$ é n, o ponto $X(i)$ pertence ao n-ésimo cluster.

Algoritmo 3: Modificação proposta para o *DBSCAN* .

```
Input: X_s                          ▷ conjunto de dados ordenados por energia
n = 0 ;                                 ▷ Inicia um contador de clusters
C = zeros(size(X_s)) ;                      ▷ C identificador de cluster
for i = 1 to size(X_s) do
    if C(i) = 0 then
        n = n + 1 ;
        C(i) = n ;
    for j = i + 1 to size(X_s) do
        if ‖p(X_s(j)) − p(X_s(i))‖ ⩽ ϵ then
            if C(j) ≠ 0 then
                C(j) = C(i) ;
                break
```

Visitando todos os pontos do conjunto de dados X_s, conforme a ordenação, se o ponto em análise ainda não foi clusterizado ($C(i) = 0$), ele inicia um novo cluster e é registrado como parte deste cluster. Caso contrário, para cada um dos dados ainda não visitados, se a distância dentro do espaço das *features* espaciais for menor que ϵ e o dado ainda não estiver sido clusterizado ($C(j) \neq 0$), o dado atual é inserido no mesmo cluster que o dado em análise no *loop* principal $C(i)$.

Dessa forma em ordem de energia os pontos irão criando clusters e inserindo os vizinhos neles sem analisar os vizinhos dos vizinhos diretamente. Permitindo que outros picos gerem novos clusters mesmo que sejam densamente conectados a um pico maior.

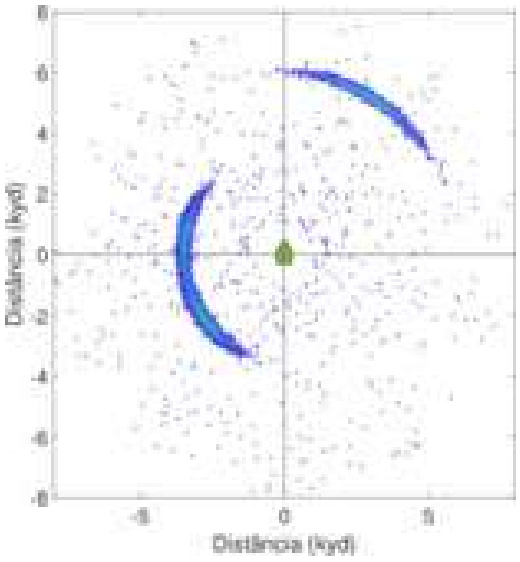
Ao final, apenas os clusters que contenham ao menos um ponto de núcleo são

validados.

Essa modificação é bem mais significativa, trocando a busca ampla por uma busca controlada por energia. Atua como se a clusterização fosse feita para o nível mais alto de energia e, posteriormente, como se o nível de energia fosse diminuindo e os novos pontos criassem novos clusters ou entrassem em clusters já existentes.

Para ilustrar o efeito desta modificação, pode-se utilizar o cenário da Figura 3.18, no qual temos três contatos nas posições: 3,5 kyd∠170°, 3,5 kyd∠220° e 6 kyd∠60°. A Figura 3.18(a) exibe o cenário na visualização passiva, enquanto, a Figura 3.18(b) na visualização ativa. Neste cenário, as dispersões dos dois primeiros contatos acabam se sobrepondo em ambas as visualizações.

(a) Visualização passiva do cenário.



(b) Visualização ativa do cenário.

Figura 3.18: Cenário de análise da modificação no *DBSCAN*.

A Figura 3.19 mostra, na visualização passiva, o resultado da aplicação dos algoritmos *DBSCAN* tradicional e modificado ao cenário da Figura 3.18. A Figura 3.19(a) exibe o resultado da clusterização do algoritmo tradicional e a Figura 3.19(b) do algoritmo modificado.

(a) Algoritmo tradicional.

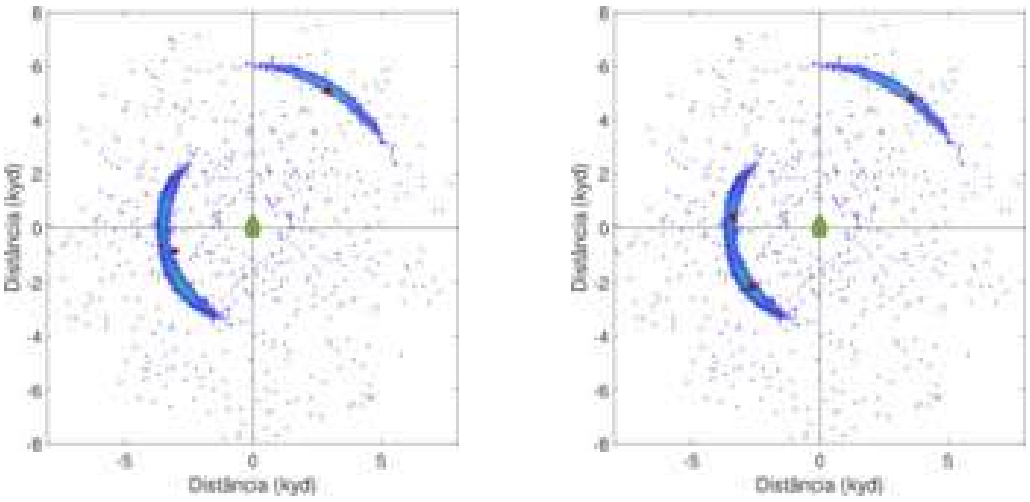
(b) Algoritmo modificado.

Figura 3.19: Análise da modificação no *DBSCAN*, visualização passiva.

Com o ruído e o limiar de detecção utilizado, as nuvens de pontos dos dois primeiros contatos se sobrepõem na região inferior dos picos. O algoritmo tradicional faz o que se espera dele e une toda a nuvem em um único cluster. Já o algoritmo

modificado consegue identificar e inicializar dois clusters, pois inicia a clusterização pelos picos de energia.

Ainda neste cenário, efeito similar pode ser percebido também na visualização ativa. A Figura 3.20 exibe, na visualização ativa, o resultado da aplicação dos algoritmos *DBSCAN* tradicional e modificado ao cenário da Figura 3.18. A Figura 3.20(a) exibe o resultado da clusterização do algoritmo tradicional e a Figura 3.20(b) do algoritmo modificado.



(a) Algoritmo tradicional. (b) Algoritmo modificado.

Figura 3.20: Análise da modificação no *DBSCAN* visualização ativa.

Pode-se perceber que o mesmo efeito da análise passiva acontece com a análise ativa para o algoritmo tradicional. Quando as regiões de influência dos clusters se sobrepõem, o algoritmo tradicional cria um cluster único. Já o algoritmo modificado consegue partir dos picos e identificar a existência dos dois clusters em ambos os cenários.

3.3 Acompanhamento

O acompanhamento apresentado nesta dissertação está dividido em três etapas:

Atribuição: Etapa na qual se associa cada cluster aos acompanhamentos existentes.

Atualização: Etapa na qual os acompanhamentos têm suas posições atualizadas baseadas nos clusters atribuídos na etapa anterior.

Prospecção: Etapa na qual os clusters não associados aos acompanhamentos podem inicializar novos acompanhamentos.

Os acompanhamentos para sonar passivo e ativo têm características e enfrentam problemas diferentes. Em sonares passivos, o maior desafio para o acompanhamento é o frequente cruzamento de contatos. Como a única informação que se tem é a marcação, acabam ocorrendo frequentes *merges and splits*.

Já em sonares ativos o maior problema são os falsos alarmes. A detecção pode gerar clusters espúrios e, em sistemas autônomos, diversos acompanhamentos falsos podem ser inicializados.

Nesta dissertação, serão analisados e comparados dois tipos de acompanhamentos, conforme descritos na Seção 2.3 e, daqui em diante, denominados simplesmente como ART e α-β.

3.3.1 Atribuição

A atribuição se inicia estimando a posição atual dos contatos, com base no tipo do acompanhamento, e associando, dentre os clusters gerados na etapa de detecção, aquele que equivale à nova medição de cada contato. Essa associação é, muitas vezes, não trivial, como por exemplo: podem haver diversos clusters na região da posição estimada de um contato; podem não haver clusters na região da posição estimada de um contato; ou pode ocorrer uma disputa de um mesmo cluster entre dois contatos, em casos de aproximação.

Esta etapa é pouco discutida na literatura de acompanhamento, onde geralmente o foco é dado na etapa de atualização e estimativa de movimento. Entretanto, esta etapa é crítica, pois a associação errada de um cluster pode fazer o acompanhamento se perder; inverter, no caso de dois contatos sofrerem *merge and split*; ou deteriorar as estimativas de posição e velocidade dos acompanhamentos.

Uma estratégia básica de atribuição é utilizar um cálculo de vizinhança similar ao da etapa de clusterização. Para isso, deve-se calcular a similaridade entre a posição estimada de cada acompanhamento e cada um dos clusters do quadro temporal atual. Essa similaridade pode ser calculada como uma distância no espaço de *features* dimensionais; uma distância no espaço de *features* completo; ou outro cálculo baseado nas informações extraídas dos clusters, como por exemplo: o número de detecções que compõem aquele cluster e a *feature* energia destes clusters.

Conforme descrito na Seção 2.3.2, é necessário definir um raio de vigilância p como região de busca de clusters para os acompanhamentos centrados na estimativa de posição dos acompanhamentos atuais, seguindo as equações (2.11a) e (2.11b). Nessa dissertação, será utilizado apenas o cálculo de distância entre as *features* espaciais, sendo calculadas todas as distâncias entre acompanhamentos e clusters, realizando as associações com prioridade para as menores distâncias, não permitindo atribuir mais de um cluster a um mesmo acompanhamento e nem um mesmo clusters

a mais de um acompanhamento.

3.3.2 Atualização

Na atualização, as novas medições atribuídas são utilizadas para atualizar as posições dos acompanhamentos. Como a medição e a estimativa serão ponderadas para atualizar a posição do contato, a atualização vai depender da técnica de acompanhamento utilizada, conforme discutido na Seção 2.3.

A cada nova etapa de atribuição, cada uma das técnicas utilizará a posição estimada do contato e do cluster atribuído. Vale ressaltar que se pode fazer uso a equação:

$$x_{n+k} = x_n + kT\dot{x}_n \tag{3.8}$$

onde

x_{n+k} = estimativa da posição no instante de tempo $n+k$ feito no instante n;
x_n = ultima posição atualizada, no instante de tempo n;
k = número de quadros temporais desde a última atualização do acompanhamento;
T = intervalo de tempo entre quadros temporais;
$\dot{x_n}$ = derivada das *features* espaciais,

como uma variante da Equação (2.11a) para a estimativa da posição, dado que a última atribuição foi há k ciclos. A equação (**??**) não precisa ser modificada, por se tratar de uma modelagem de movimento uniforme.

De forma similar, a atualização da posição e velocidade dos contatos, podendo-se utilizar as equações:

$$x_{n,n} = x_{n,n-k} + \alpha(y_n - x_{n,n-k}) \tag{3.9a}$$

$$\dot{x}_{n,n} = \dot{x}_{n,n-k} + \frac{\beta}{k}(\frac{y_n - x_{n,n-k}}{T}) \tag{3.9b}$$

onde

$x_{n,n}$ = estimativa da posição no instante n feita no instante n, ou seja, feita após a medição da posição em n;

$x_{n,n-k}$ = estimativa da posição no instante n feita no instante $n-k$ através da equação (3.8);

α = fator de atualização da posição;

y_n = medição da posição no instante n;

$\dot{x}_{n,n}$ = estimativa da velocidade no instante n feita no instante n, ou seja, feita após a medição da posição em n;

$\dot{x}_{n,n-k}$ = estimativa da velocidade no instante n feita no instante $n-1$ através da equação (2.11b);

β = fator de atualização da velocidade;

T = intervalo de tempo entre quadros temporais.

como variação das equações (2.12a) e (2.12b) para atualização da posição e velocidade, dado que a última atribuição foi feita a k ciclos.

3.3.3 Prospecção

Na prospecção, os clusters não associados a um contato durante a etapa de atribuição devem ser analisados sob a perspectiva de criação de novos contatos. Esta etapa pode não existir em um sistema onde a detecção seja orientada por um operador. Em sistemas autônomos existem dois modos básicos de acompanhamento: o manual e o automático.

No acompanhamento manual, um novo contato só é criado por um operador e, uma vez criado, o sistema acompanha e atualiza a posição deste contato, seguindo as duas etapas anteriores: atribuição e atualização. Já no acompanhamento automático, existe a etapa adicional de prospecção que iniciará acompanhamentos de forma automática.

Existem duas opções para o acompanhamento automático: os contatos criados automaticamente são indicados e, após a validação do operador, iniciados no sistema; ou a validação é realizada de forma automática por algum critério previamente estabelecido.

Nesta dissertação, esta etapa será feita com o enfoque puramente automático, tanto para a criação de acompanhamento, quanto para a validação dos mesmos. Cada cluster não associado na etapa de atribuição poderá criar um novo possível contato.

Cada possível contato criado será iniciado como um acompanhamento com atualização baseada na rede ART com fator de atualização alto, devido a alta adaptabilidade desta estratégia. Inicializar um acompanhamento mais complexo como o α-β exige uma estimativa inicial de movimento apropriada, o que não pode ser

feito corretamente em um primeiro momento. Uma vez que o possível contato seja acompanhado por n passos de simulação, ele pode ser validado pelo sistema como um acompanhamento válido. Nesta dissertação, o contato será validado se, durante esses n passos, o acompanhamento tiver recebido $\frac{n}{2}$ atribuições, outras formas de validação possam ser estudadas futuramente. Após esses n passos de simulação, é possível criar o acompanhamento definitivo ou excluir o acompanhamento do possível contato. No caso da inicialização de um acompanhamento α-β pode-se utilizar as detecções atribuídas ao acompanhamento ART para realizar a necessária estimativa inicial do movimento do contato.

A etapa de prospecção desta dissertação pode ser dividida em duas etapas. Num primeiro momento retira-se do conjunto de clusters C todas aqueles que estiverem dentro da região de influência dos contatos existentes. Para isso faz-se um *loop* analisando cada contato e identificando o subconjunto de clusters próximos a posição do contatos no espaço de *features*, removendo este subconjunto do conjunto C em análise.

O subconjunto dos clusters que restaram são ordenados em ordem decrescente de energia. O *loop* seguinte é executado até o subconjunto ordenado C_s se tornar um conjunto vazio, $\varnothing$. A cada passo deste *loop*, inicializa-se um novo possível contato na posição do cluster de maior energia do subconjunto, $C_s(0)$. Depois, retiram-se do conjunto todos os clusters que estejam próximos ao novo contato criado. Com isso, garante-se que não serão inicializados possíveis contatos dentro das regiões de influência de outros contatos ou possíveis contatos.

Além disso na prospecção pode-se analisar a perda de um contato, ou seja, o que fazer com um acompanhamento que não recebe mais detecções. Novamente tem-se diversas opções de identificar contatos perdidos e mais simples seria manter o acompanhamento com o movimento estimado por algumas detecções e não havendo atribuições num período o contato é automaticamente excluído. Nesse dissertação, o simulador não foi projetado para simular a perda de contatos por distância ou condições ambientais de propagação e portanto esta etapa não sera analisada. Mas vale destacar que para um sistema autônomo que opere continuamente essa etapa é vital para manter contatos coerentes sem deixar diversos contatos perdidos andando sem terem detecções atribuídas.

Capítulo 4

Resultados

Nesta seção, são analisados os efeitos da alteração proposta para a detecção sonar e as características das técnicas de acompanhamento de forma simulada. O simulador utilizado gera um conjunto de pontos servindo de entrada para as técnicas de detecção e um conjunto de clusters servindo de entrada para as técnicas de acompanhamento.

Os cenários analisados são inicializados de forma aleatória, tanto no que tange a localização dos contatos, quanto relativamente ao rumo e a velocidade. Para as análises da detecção na seção 4.1, os cenários são estáticos, e cada nova entrada é oriunda de um novo cenário aleatório. Já para o acompanhamento na seção 4.2, cada cenário é evoluído no tempo, e são aleatoriamente escolhidas alterações de rumo e velocidade dos navios, de forma a tornar o movimento não-uniforme, conforme descrito na Seção 3.1.1.

Vale destacar que em um cenário real, variações de velocidade e de rumo em meios navais é lenta, não havendo variações abruptas ou degraus. As simulações foram feitas com uma dinâmica de movimento acelerada para reduzir o número de ciclos de simulação necessários para a observação de alterações significativas das coordenadas dos navios.

Todas as análises comparativas feitas nessa seção expõem as diferentes técnicas às mesmas entradas, comparando-as de forma isenta. Dentro dos cenários sorteados, pode haver cenários nos quais a correta detecção ou acompanhamento não seja possível, por exemplo, se dois contatos estiverem muito próximos e suas nuvens de pontos estiverem completamente sobrepostas. Esses cenários são possíveis de ocorrer em cenários reais e a capacidade dos algoritmos em lidar com tais casos é comparada de forma implícita quando não explicita.

Ao longo de toda a seção, são feitos testes estatísticos para a comparação das técnicas duas a duas. É utilizado o teste pareado de *Wilcoxon Signed-ranks* [23] com a hipótese nula de que os resultados das duas técnicas, quando expostas a mesma entrada, gerem resultados cuja distribuição combinada tenham mediana zero. Em

termos práticos, a rejeição da hipótese, representada por 1, indica com confiabilidade de 95% que os resultados possuem uma mesma média, ou seja, que as técnicas obtêm resultados estatisticamente significativos, enquanto a não rejeição da hipótese nula, representada por 0, indica que não é possível rejeitar a hipótese nula com essa confiabilidade, portanto, os resultados podem ou não vir da mesma distribuição. Foi escolhido fazer análises comparativas com o teste de *Wilcoxon Signed-ranks*, por ser um método não-paramétrico, ou seja, que não exige o resultado dos algoritmos provenham de uma distribuição conhecida a priori, pois nem todos os resultados têm distribuição gaussiana [24].

4.1 Detecção

Neste capítulo serão comparadas as técnicas de clusterização tradicionais (desconsiderando a energia) e as técnicas modificadas neste trabalho, primeiramente, para sonar passivo e, subsequentemente, para sonar ativo. As técnicas serão chamadas por simplicidade daqui para frente apenas de: *mean shift* tradicional, *mean shift* modificado, *DBSCAN* tradicional e *DBSCAN* modificado.

A entrada de dados para a detecção é um mapa de marcação por energia, para o sonar passivo; ou, no caso do sonar ativo, distância por distância por energia (utilizando coordenadas cartesianas após a conversão das coordenadas polares geradas pelo simulador). Esse mapa precisa ser convertido em uma matriz de pontos de interesse (detecções). Esse processo é feito pelo ceifamento do mapa para valores de energia acima do limiar de detecção (λ), portanto, só os pontos acima desse limiar serão considerados para a clusterização.

Dois fatores são críticos para análise dos algoritmos de clusterização: a *SNR* e o λ, pois ambos influenciam na quantidade e na densidade dos pontos gerados na etapa de detecção. As Tabelas 4.1 e 4.2 ilustram o efeito da variação destes parâmetros para o mesmo cenário passivo e ativo, respectivamente.

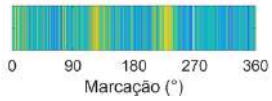
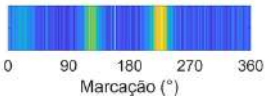
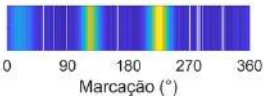
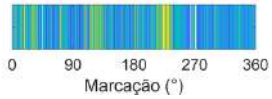
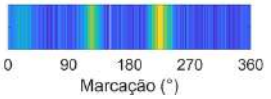
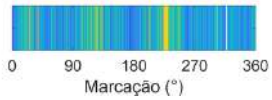
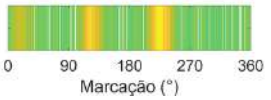
$\lambda \backslash SNR$	3 dB	6 dB	9 dB
3			
6			
9			

Tabela 4.1: Cenário ilustrativo de variação de *SNR* e λ para sonar passivo.

Para um mesmo λ, quanto maior for a *SNR*, menor é o número de pontos gerados e mais a nuvem de pontos fica focada nos picos. Por outro lado, quanto menor a *SNR*, maior é a quantidade de pontos espúrios e a probabilidade de geração de clusters indesejados. De forma análoga, para uma mesma *SNR*, quanto menor for o λ, mais pontos espúrios serão detectados e, quanto maior o λ, mais os pontos ficam concentrados nos picos.

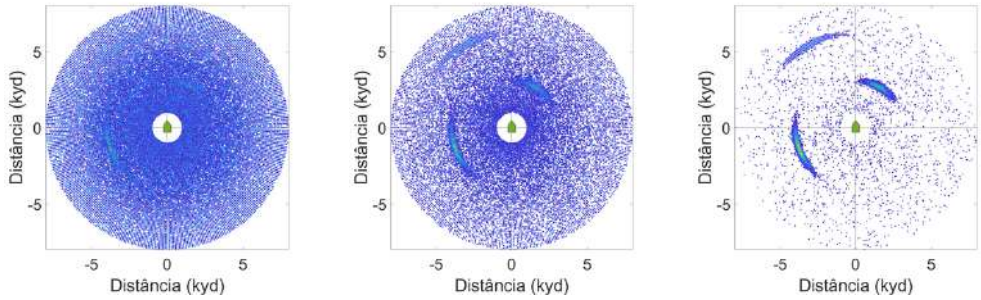
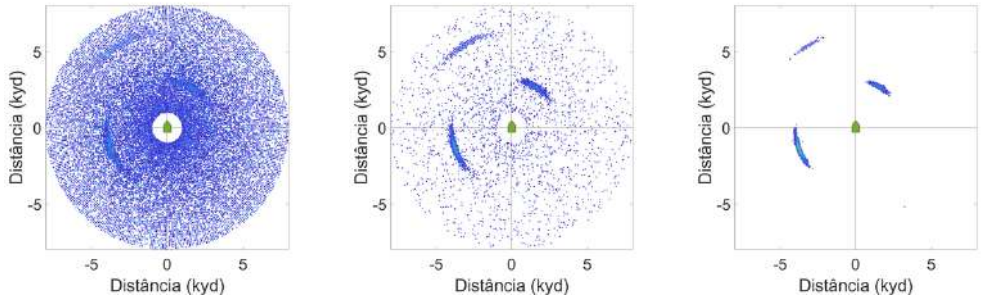
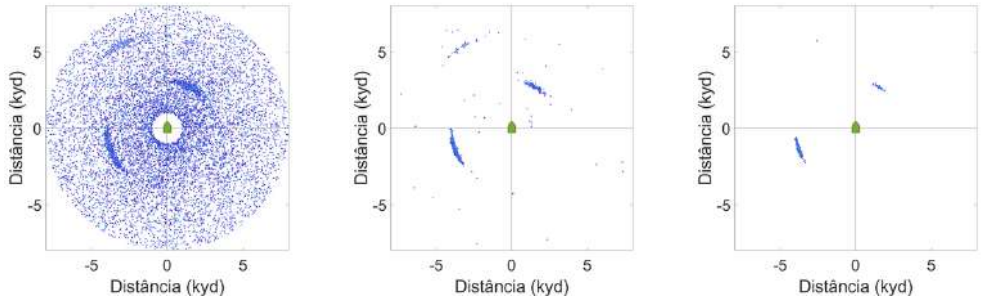
$\lambda \backslash SNR$	3 dB	6 dB	9 dB
3			
6			
9			

Tabela 4.2: Cenário ilustrativo da variação de *SNR* e λ para sonar ativo.

O balanço entre *SNR*, do ambiente, e λ, controlado pelo operador, é uma tarefa difícil operação, geralmente entregue ao controle do operador do sonar. Reduzir o número de pontos e focar nos sinais mais fortes facilita as etapas de detecção e acompanhamento, reduzindo o número de falsos alarmes. Entretanto, essa redução mascara contatos com menor energia e aumenta a probabilidade de falsos negativos. Esse efeito pode ser visualizado na Tabela 4.2, onde, no cenário com maior *SNR* e λ, o contato mais distante não gera detecções.

A operação típica do operador sonar é, inicialmente, aumentar o número de pontos enquanto se está em etapa exploratória, em busca de detecção de contatos. E,

uma vez encontrados contatos e caracterizado o cenário, reduzir o número de pontos progressivamente, mantendo os contatos e refinando tanto o acompanhamento quanto a visualização dos contatos.

O balanço dinâmico entre o λ em relação a *SNR* pode auxiliar sistemas automáticos, mas está fora do escopo dessa dissertação. Aqui será analisado o efeito da variação da *SNR* e do λ na performance dos algoritmos, como uma forma de avaliar a robustez das técnicas e os efeitos da modificação proposta para a clusterização.

4.1.1 Sonar passivo

Para o sonar passivo, foram feitas simulações de quinhentos cenários, cada um com três contatos. Cada cenário foi simulado com todas as combinações de *SNR* e λ de valores entre 2 e 10 com passo de $0,5$.

Escolheu-se analisar cenários com três contatos, porque ser um cenário realista de um número comum de contatos para operação simultânea em sonar passivo. Além disso, como só se tem a marcação, com um número maior de contatos é provável que existam muitas sobreposições que deteriorariam os resultados. O número de cenários analisados foi arbitrado de forma a obter resultados estatisticamente relevantes nos testes comparativos, sem inviabilizar o tempo total de simulação.

A Figura 4.1 exemplifica o resultado das técnicas de clusterização para um mesmo cenário para diferentes valores de *SNR* e λ. Na qual, cada linha apresenta o resultado da clusterização por uma das técnicas, de cima para baixo: *mean shift* tradicional, *mean shift* modificado, *DBSCAN* tradicional e *DBSCAN* modificad e cada coluna um diferente balanço de *SNR* e λ, ilustrando o resultado qualitativo da aplicação dos algoritmos.

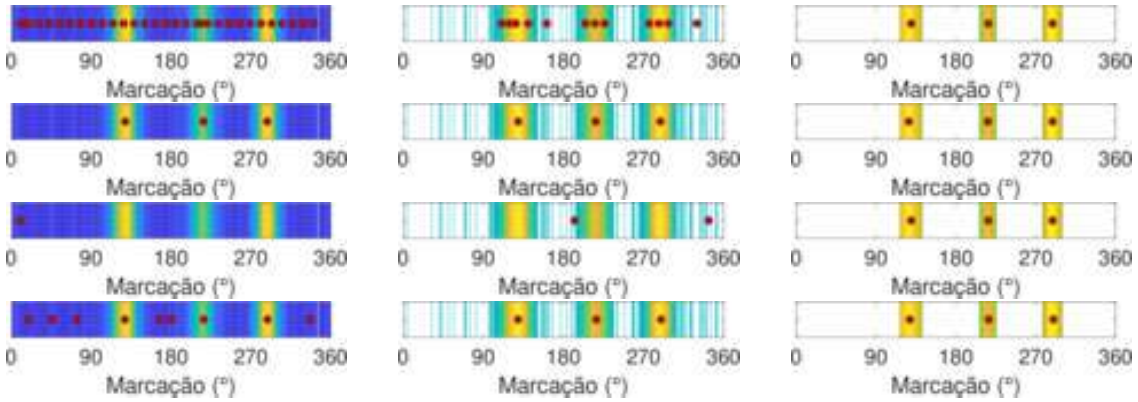


Figura 4.1: Cenário ilustrativo do resultado da clusterização, onde em cada linha foi aplicada uma técnica. De cima para baixo: *mean shift* tradicional, *mean shift* modificado, *DBSCAN* tradicional e *DBSCAN* modificado. Com *SNR* em 5 dB e o λ variando da direita para a esquerda de 2.5, ,5 e 7.5.

Pode-se observar que o *mean shift* tradicional tende a gerar mais falsos positivos,

além de ser altamente dependente do número de detecções. O *DBSCAN* tradicional não consegue lidar com cenários com maior concentração de pontos, unindo os picos aos espúrios vizinhos e perdendo a localização efetiva dos picos. Já as técnicas modificadas apresentam maior capacidade de encontrar os picos, mesmo em cenários com alta concentração de pontos. Além disso, observa-se que, para o cenário onde o balanço dos parâmetros consegue isolar bem os picos e gerar poucas detecções espúrias, todas as técnicas alcançam resultados similares.

4.1.1.1 Falsos negativos

Nesta seção, são analisados os resultados para falsos negativos, contatos que não geraram clusters dentro de uma janela de busca de 20°.

A Tabela 4.10 apresenta o resultado do teste estatístico *Wilcoxon Signed-ranks* pareado entre o algoritmo *mean shift* tradicional e o modificado.

Conforme descrito no início da Seção 4, os 1's da tabela representam a rejeição da hipótese nula, indicando com confiabilidade de 95% que os resultados das técnicas não vêm da mesma distribuição, ou seja, as distribuições dos resultados das técnicas para as mesmas entradas são estatisticamente diferentes, enquanto a não rejeição de hipótese nula, representado por 0, indica que não é possível rejeitar a hipótese nula com essa confiabilidade. Portanto, os resultados podem ou não vir da mesma distribuição.

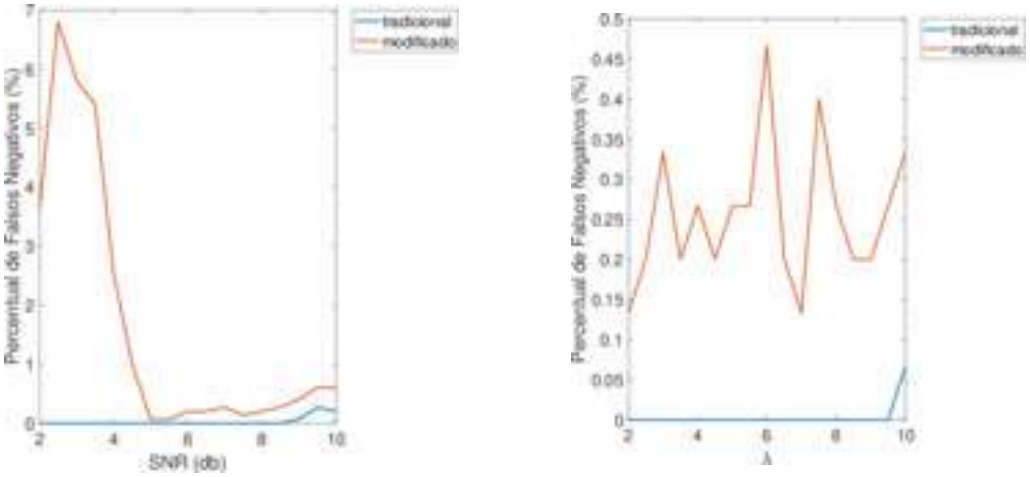
Tabela 4.3: Comparação falsos negativos para *mean shift*.

		SNR																
		2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	2,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	3	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	3,5	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	4	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0
	4,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0
	5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	5,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
λ	6	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0
	6,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	7	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	7,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	8	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	8,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	9	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	9,5	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
	10	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0

Para o algoritmo de *mean shift*, a modificação proposta leva a alterações estatis-

ticamente relevantes quanto à geração de falsos negativos, predominantemente para valores de *SNR* menores que 4,5 dB.

Os resultados das simulações para o *mean shift* tradicional e modificado também podem ser vistos como na Figura 4.2. A Figura 4.2(a) mostra o percentual de falsos negativos em função da *SNR* para um valor fixo de λ. Enquanto a Figura 4.2(b) mostra o percentual de falsos negativos em função do λ para um valor fixo de *SNR*.



(a) Para λ fixo.

(b) Para SNR fixo.

Figura 4.2: Falsos negativos para *mean shift*.

Combinando o resultado do teste estatístico com o da figura para parâmetros fixos, pode-se observar que para valores de *SNR* maiores que 4,5 dB, o resultado dos algoritmos tradicionais e modificados são próximos e a hipótese nula é majoritariamente não rejeitada. O número de falsos negativos do algoritmo tradicional é menor que o do algoritmo modificado em todos os cenários, mas não é estatisticamente comprovada a diferença de resultados.

Já para valores de *SNR* menores que 4,5 dB, o algoritmo modificado apresenta uma piora nos resultados, distanciando-se daqueles obtidos com o algoritmo tradicional, passando a rejeitar a hipótese nula majoritariamente. Portanto, o *mean shift* modificado demonstra maiores dificuldade ao lidar com *SNR* menores, tendo de forma geral um número de falsos negativos menor de 7% para todos os cenários.

É possível entender o porquê do algoritmo tradicional apresentar um percentual de falsos negativos muito baixo observando o resultado do *mean shift* tradicional para um dos casos do cenário de exemplo da Figura 4.1, destacado na Figura 4.3.

Pode-se observar que o *mean shift* tradicional gera um conjunto de clusters distribuídos ao longo das regiões dos contatos. Com isso, em cenários de grande sobreposição dos contatos, nos quais o algoritmo modificado identifique dois contatos como um único, o algoritmo tradicional gerará múltiplos clusters ao longo da nuvem combinada dos contatos. A análise considerará que o algoritmo modificado, identificando dois contatos como um único, cometeu um falso negativo, enquanto que

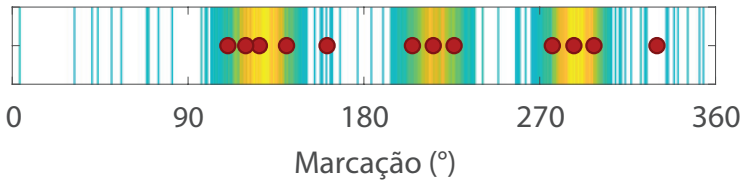


Figura 4.3: Resultado da clusterização passiva para *mean shift* tradicional.

para o algoritmo tradicional, a análise considerará a correta identificação de ambos os contatos, pois haverá um cluster para cada contato dentro das respectivas janelas de busca.

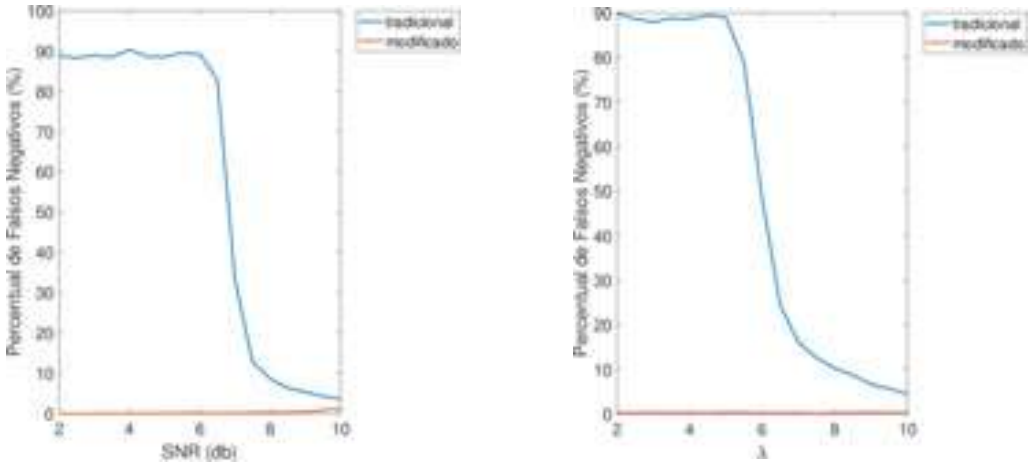
Entretanto, a criação de múltiplos clusters irá impactar o número de falsos positivos, que será analisada na Seção 4.1.1.2. Na prática, gerar uma nuvem de clusters na região dos contatos dificulta a criação de acompanhamentos e a identificação real do contato não será uma solução vantajosa.

Para o algoritmo de *DBSCAN*, conforme pode ser visto na Tabela 4.4, a modificação proposta gera alterações estatisticamente relevantes em quase todas as combinações de *SNR* e λ.

Tabela 4.4: Comparação falsos negativos para *DBSCAN*.

		SNR																
		2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	5,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
λ	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	6,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	7,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	8,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	9,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Os resultados das simulações para o *DBSCAN* tradicional e o modificado, também podem ser vistos na Figura 4.4. A Figura 4.4(a) mostra o percentual de falsos negativos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.4(b) mostra o percentual de falsos negativos em função do λ para um valor fixo de *SNR*.



(a) Para λ fixo. (b) Para SNR fixo.

Figura 4.4: Falsos negativos para *DBSCAN*.

Combinando o resultado do teste estatístico com o da figura para parâmetros fixos, pode-se observar que o algoritmo modificado apresenta um menor número de falsos negativos para quase todas as combinações de *SNR* e λ analisadas. Além disso, essa diferença é maior para a região onde os parâmetros são baixos, sendo reduzida conforme ambos os parâmetros aumentam de valor.

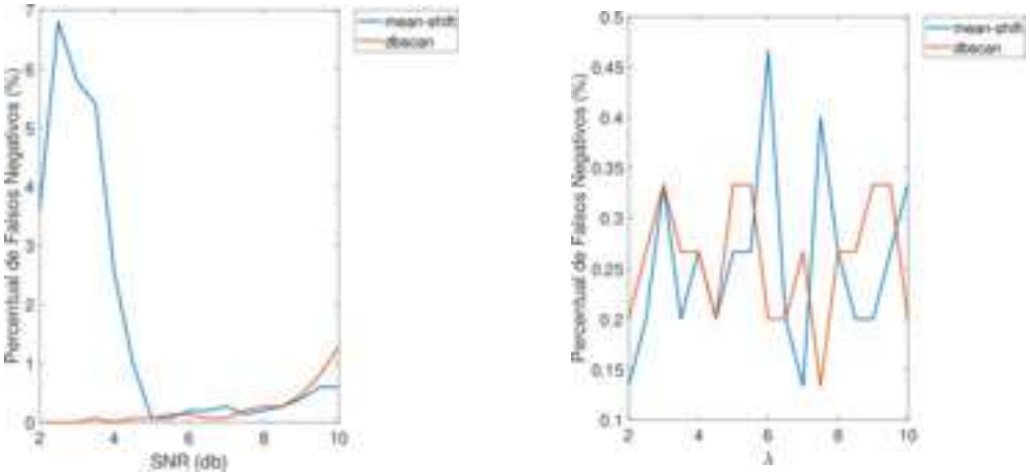
A Tabela 4.5 mostra o resultado estatístico da comparação entre o *mean shift* e *DBSCAN*, ambos modificados.

Tabela 4.5: Comparação falsos negativos para algoritmos modificados.

		SNR																
		2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	2,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	3	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	3,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	4	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	4,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	5,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
λ	6	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	6,5	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	7	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	7,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	8	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	8,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
	9	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1
	9,5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1
	10	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1

Os resultados das simulações para os algoritmos modificados também podem

ser vistos na Figura 4.5. A Figura 4.5(a) mostra o percentual de falsos negativos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.5(b) mostra o percentual de falsos negativos em função do λ para um valor fixo de *SNR*.



(a) Para λ fixo.

(b) Para SNR fixo.

Figura 4.5: Falsos negativos para algoritmos modificados.

Combinando o resultado do teste estatístico com o da figura para parâmetros fixos, pode-se observar que o algoritmo *mean shift* apresenta maior número de falsos negativos para valores de *SNR* menores que 4,5 dB, demostrando uma maior incapacidade para lidar com *SNR* baixos. Vale ressaltar que o percentual de falsos negativos ainda é inferior a 7%, não sendo um resultado que inviabilize o uso da técnica.

Na região onde ambos os parâmetros são altos, o resultado passa a ser favorável ao *mean shift*, demostrando que o *DBSCAN* tem uma maior propensão a perda de contatos quando a nuvem de pontos diminui. Para valores intermediário dos parâmetros, os resultados não são suficientemente diferentes para se rejeitar a hipótese nula.

4.1.1.2 Falso Positivos

Foram também computados os dados de falso positivos, ou seja, de clusters gerados que não representam a posição real de um contato, com os quais foram feitos testes estatísticos do tipo *Wilcoxon Signed-ranks* pareados, com o objetivo de verificar se as modificações propostas gerariam alterações estatisticamente significativas.

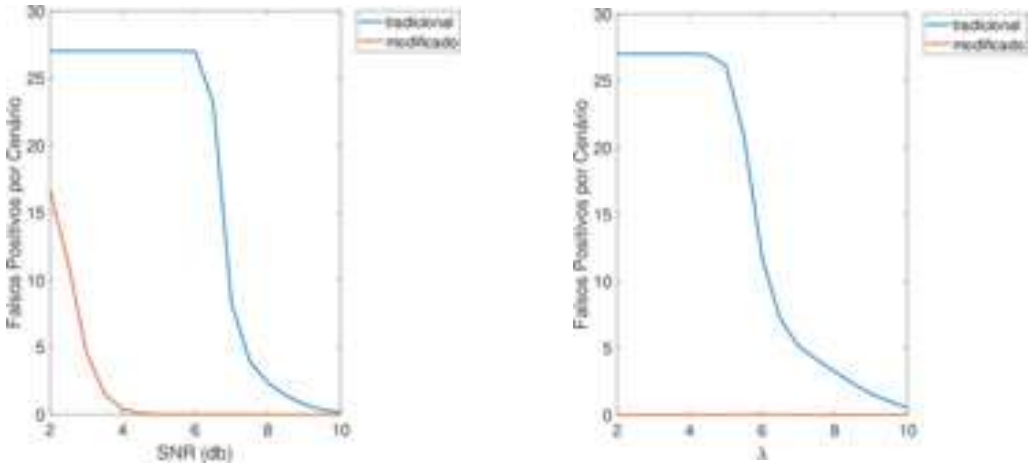
A Tabela 4.6 exibe o resultado do teste estatístico *Wilcoxon Signed-ranks* pareado entre os algoritmos *mean shift* tradicional e modificado.

Para o algoritmo de *mean shift*, a modificação proposta gerou alterações estatisticamente relevantes quanto à geração de falsos positivo, para todas as combinações dos parâmetros.

Tabela 4.6: Comparação falsos positivos para *mean shift*.

		SNR																
		2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	5,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
λ	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	6,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	7,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	8,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	9,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Os resultados das simulações para o *mean shift* tradicional e modificado, também podem ser vistos na Figura 4.6. A Figura 4.6(a) mostra o número de falsos positivos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.6(b) mostra o número de falsos positivos em função do λ para um valor fixo de *SNR*.



(a) Para λ fixo. (b) Para SNR fixo.

Figura 4.6: Falsos positivos para *mean shift*.

Combinando o resultado do teste estatístico com o da figura para parâmetros fixos, pode-se observar que a alteração proposta reduz a geração de falsos positivos para todos os valores de parâmetros analisados e que, em todos os cenários, os resultados são estatisticamente diferentes.

O algoritmo tradicional, independentemente do balanço de *SNR* e λ, tem dificuldade de migrar dentro da nuvem de detecções e acaba gerando múltiplos clusters na região de cada contato. Além disso, fora dos picos dos contatos, o algoritmo tradicional também gera um número maior de falsos positivos. O número de falsos positivos diminui conforme os parâmetros aumentam, mas não chega a atingir os resultados do algoritmo modificado dentro dos cenários simulados.

A Tabela 4.7 exibe o resultado do teste estatístico *Wilcoxon Signed-ranks* pareado entre o algoritmo *DBSCAN* tradicional e o modificado.

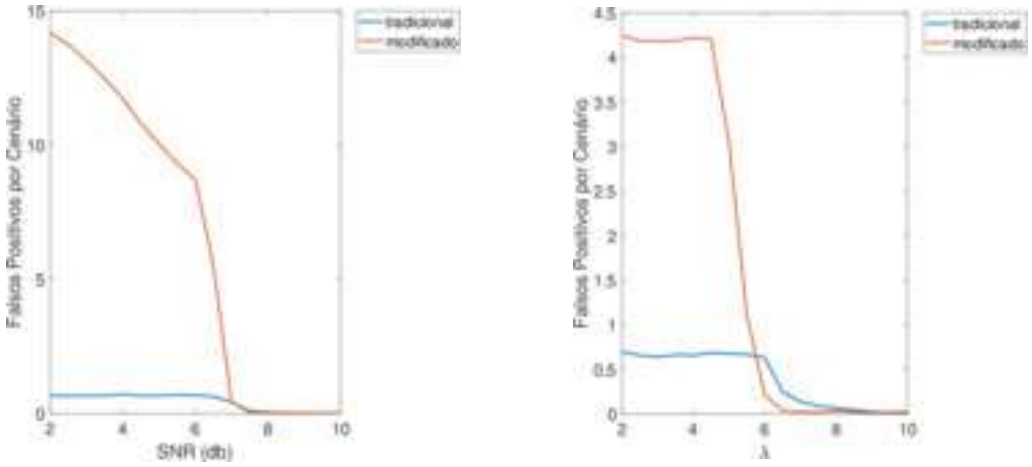
Tabela 4.7: Comparação falsos positivos para *DBSCAN*.

		SNR																
		2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	5,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
λ	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	6,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	7,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	8	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0
	8,5	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
	9	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
	9,5	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	10	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Para o algoritmo de *DBSCAN*, a modificação proposta gera alterações estatisticamente relevantes quanto à geração de falsos positivos para a maior parte das combinações dos parâmetros. Apenas na região onde ambos os parâmetros são altos (maiores que $6,5$), ocorre a não rejeição da hipótese nula, predominando conforme o aumento dos parâmetros.

Os resultados das simulações para o *DBSCAN* tradicional e o modificado também podem ser vistos na Figura 4.7. A Figura 4.7(a) mostra o número de falsos positivos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.7(b) mostra o número de falsos positivos em função do λ para um valor fixo de *SNR*.

Combinando o resultado do teste estatístico com o da figura para parâmetros fixos, pode-se observar que, para valores altos de ambos os parâmetros, o resultado dos algoritmos se tornam próximos, sendo a região da tabela de teste estatístico onde a hipótese nula é, frequentemente, não rejeitada. Isto ocorre porque com altos



(a) Para λ fixo.

(b) Para SNR fixo.

Figura 4.7: Falsos positivos para *DBSCAN*.

valores de *SNR*, os picos são mais bem definidos, permitindo que a escolha de um λ alto elimine quase complemente pontos fora das regiões dos picos, tornando os cenários mais simples para a clusterização. No cenário de exemplo da Figura 4.1, pode-se observar este fenômeno na última coluna.

De forma geral, para valores intermediários e baixos de um ou de ambos os parâmetros, a modificação proposta gera alterações estatisticamente relevantes quanto à redução da geração de falsos positivos.

A Tabela 4.8 exibe o resultado do teste estatístico *Wilcoxon Signed-ranks* pareado entre os algoritmos modificados.

Para os algoritmos modificados, observa-se que para valores de *SNR* menores que 8 dB, a hipótese nula é sempre rejeitada, já para valores maiores, a rejeição da hipótese nula se concentra onde o λ é menor.

Os resultados das simulações para os algoritmos modificados também podem ser vistos na Figura 4.8. A Figura 4.8(a) mostra o número de falsos positivos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.8(b) mostra o número de falsos positivos em função do λ para um valor fixo de *SNR*.

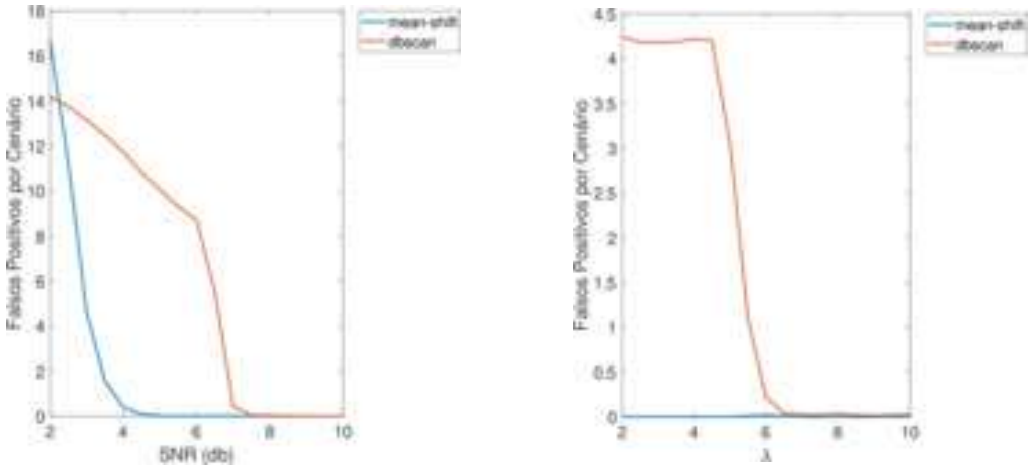
Combinando o resultado do teste estatístico com o da figura para parâmetros fixos, pode-se observar que o resultado do algoritmo *mean shift* é, para quase todos os balanços dos parâmetros, melhor que o do *DBSCAN*. Para valores muito baixos ou muito altos de ambos os parâmetros, os resultados se aproximam e por vezes se invertem.

4.1.1.3 Estimativa de posição

Para todos os cenários simulados, foram avaliados os erros de marcação dos clusters em relação às marcações reais dos contatos. Os erros foram comparados indepen-

Tabela 4.8: Comparação falsos positivos para algoritmos modificados.

		SNR																
		2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	2,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	3,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	4,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
	5,5	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
λ	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
	6,5	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
	7	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0
	7,5	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0
	8	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
	8,5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
	9	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
	9,5	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0



(a) Para λ fixo.

(b) Para SNR fixo.

Figura 4.8: Falsos positivos para algoritmos modificados.

dentemente do valor de *SNR* e λ, sendo analisados de forma global para todos os verdadeiros positivos dos cenários. O valor de média ($\bar{b}$) e desvio padrão (δ_b) dos erros de marcação para cada técnica estão dispostas na Tabela 4.9.

Para a análise dos erros de marcação, vale lembrar que nesta dissertação a menor unidade de marcação dos dados de entrada é de 1°.

Quanto ao erro médio de marcação, todas as técnicas obtiveram resultado inferior à menor unidade da entrada, portanto, razoavelmente próximo de zero.

Observa-se, também, que as alterações propostas reduziram o desvio padrão dos

Tabela 4.9: Média e desvio padrão dos erros de marcação.

	$b(^\circ)$	$\delta_b(^\circ)$
mean shift	$-0,1665$	$3,3628$
mean shift modificado	$0,2184$	$2,1448$
DBSCAN	$0,0439$	$6,8371$
DBSCAN modificado	$-0,0050$	$2,5521$

erros de marcação para ambas as técnicas, obtendo um valor próximo entre as duas técnicas modificadas. Além disso, a *DBSCAN* tradicional apresenta o maior erro, da ordem de duas vezes o erro do *mean shift* tradicional.

Vale ressaltar que todos os resultados têm valores de erro de marcação da mesma ordem de grandeza, poucas unidades de graus, não representando óbices quanto à utilização dessas técnicas para a aplicação em foco.

4.1.2 Sonar ativo

Para o sonar ativo, foram feitas simulações de cem cenários, cada um com cinco contatos. Cada cenário foi simulado com todas as combinações de *SNR* e λ de valores entre 4 e 10 com passo de $0,5$. Não foram utilizados valores de *SNR* e λ menores, porque a quantidade de pontos gerados aumenta, aumentando também o tempo de execução dos algoritmos, inviabilizando as simulações paramétricas desta dissertação.

Foi escolhido analisar cenários com cinco contatos, porque dentro do cenário real é um número comum de contatos para operação simultânea do sonar ativo. Trabalhando com marcação e distância, é possível simular mais contatos que no passivo sem tantas sobreposições. Ainda assim, utilizar ainda mais contatos aumentaria a probabilidade de cenários com muitas proximidades de contatos que deteriorariam os resultados. O número de cenários analisados foi arbitrado de forma a obter resultados estatisticamente relevantes nos testes, sem inviabilizar o tempo total de simulação.

A Figura 4.9 exemplifica o resultado das técnicas de clusterização para um mesmo cenário para diferentes valores de *SNR* e λ. Cada linha apresenta o resultado da clusterização por uma das técnicas. De cima para baixo: *mean shift*, *mean shift* modificado, *DBSCAN* e *DBSCAN* modificado. Cada coluna apresenta um diferente balanço de *SNR* e λ, ilustrando o resultado qualitativo dos algoritmos, que será analisado quantitativamente ao longo desta seção.

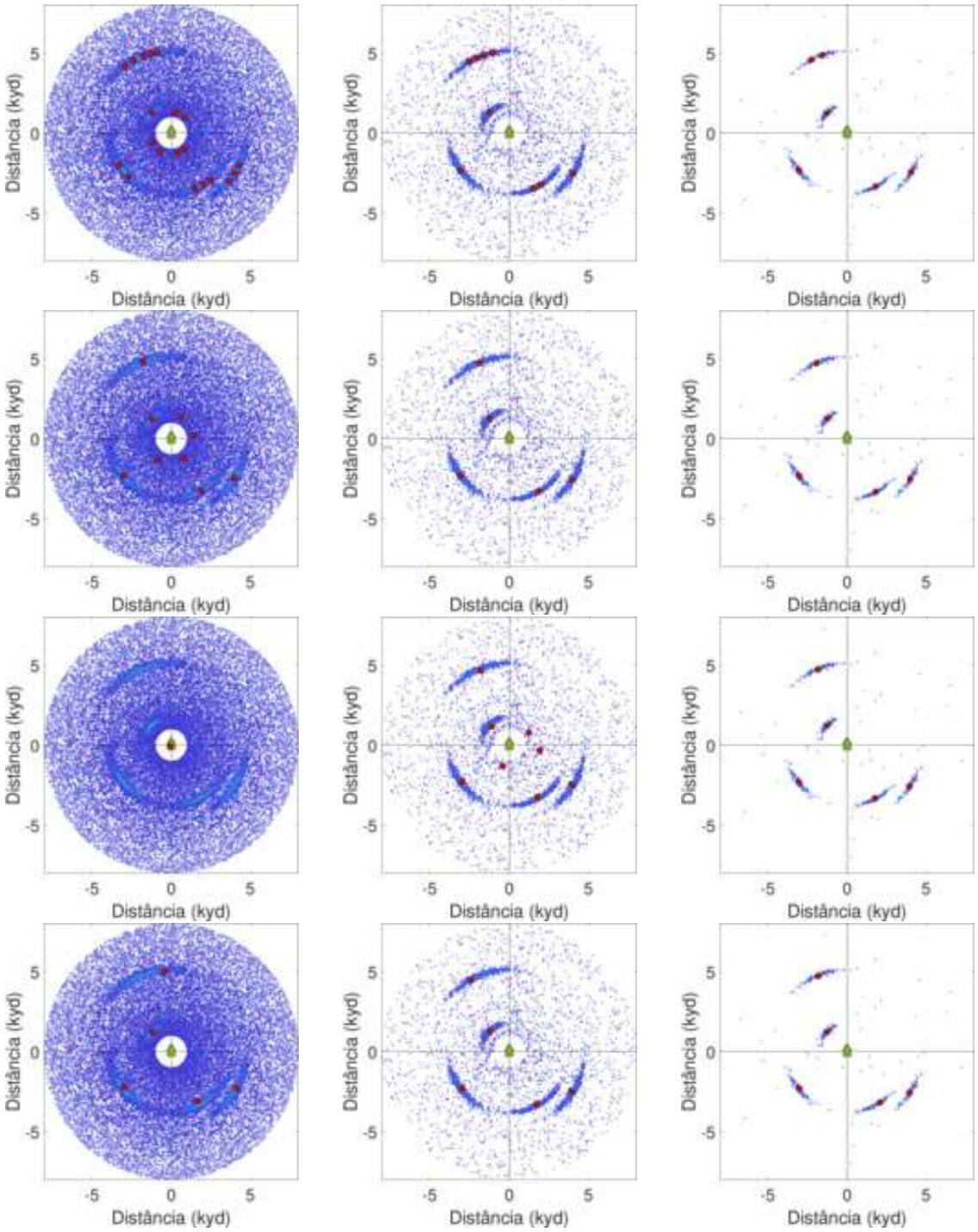


Figura 4.9: Cenário ilustrativo resultado da clusterização ativa. Cada linha um algoritmo de cima para baixo: *Mean-Shift*, *Mean-Shift* modificado, *DBSCAN* e *DBSCAN* modificado. Cada coluna um diferente balanço de *SNR* e λ.

4.1.2.1 Falsos negativos

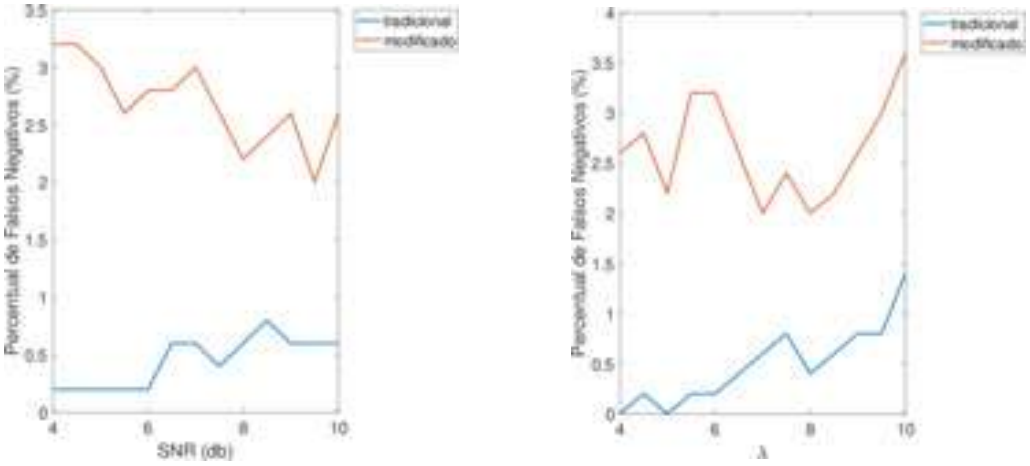
Nesta seção, são analisados os resultados para falsos negativos, contatos que não geraram clusters dentro de uma janela de busca de 800 yd. A Tabela 4.10 exibe o resultado do teste estatístico *Wilcoxon Signed-ranks* pareado entre o algoritmo *mean shift* tradicional e modificado.

Tabela 4.10: Comparação falsos negativos para *mean shift*.

		SNR												
		4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	4	1	1	1	1	1	1	1	1	1	1	1	1	1
	4,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1	1
	5,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	1	1	1
	6,5	1	1	1	1	1	1	1	1	1	1	1	1	1
λ	7	1	1	1	1	1	1	1	1	1	1	0	1	1
	7,5	1	1	1	1	1	1	1	1	1	0	1	1	0
	8	1	1	1	1	1	1	1	1	1	1	0	0	1
	8,5	1	1	1	1	1	1	1	1	1	1	1	0	1
	9	1	1	1	1	1	1	1	1	1	0	1	1	1
	9,5	1	1	1	1	1	1	1	1	1	1	0	0	0
	10	1	1	1	0	1	1	1	0	1	1	1	0	1

Para o algoritmo de *mean shift*, conforme pode ser visto na Tabela 4.10, a modificação proposta gera alterações estatisticamente relevantes quanto à geração de falsos negativos, na maior parte dos pares de *SNR* e λ. Apenas na região onde ambos os parâmetros são altos, concentram-se os testes onde a hipótese não nula é rejeitada.

Os resultados das simulações para o *mean shift* tradicional e modificado, também podem ser vistos na Figura 4.10. A Figura 4.10(a) mostra o percentual de falsos negativos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.10(b) mostra o percentual de falsos negativos em função do λ para um valor fixo de *SNR*.



(a) Para λ fixo. (b) Para SNR fixo.

Figura 4.10: Falsos negativos para *mean shift*.

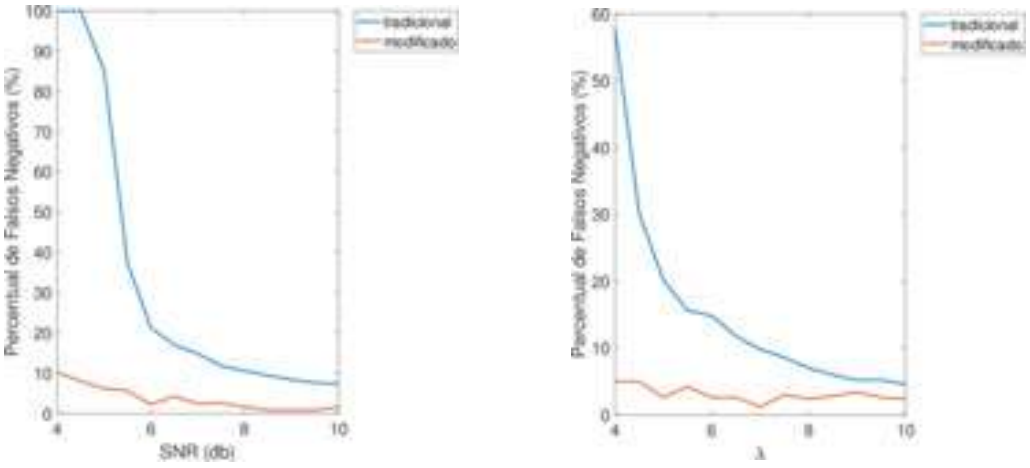
Em praticamente todos os cenários, o algoritmo tradicional gera menos falsos negativos que o algoritmo modificado. Vale destacar que o percentual de falsos ne-

gativos é muito baixo em ambas as técnicas sendo inferior a 4%. Conforme explicado na Seção 4.1.1.1, o baixíssimo índice de falsos negativos do *mean shift* tradicional é um artefato causado pela criação de múltiplos clusters na região de cada contato. O mesmo fenômeno que ocorre para o cenário passivo.

Para o *DBSCAN*, conforme pode ser visto na Tabela 4.11, os resultados rejeitam a hipótese nula para a maior parte dos valores de *SNR* e λ assim como os resultados do *mean shift*. Entretanto, neste caso, o resultado é favorável à técnica modificada, conforme pode-se ver na Figura 4.11. A Figura 4.11(a) mostra o percentual de falsos negativos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.11(b) mostra o percentual de falsos negativos em função do λ para um valor fixo de *SNR*.

Tabela 4.11: Comparação falsos negativos para *DBSCAN*.

		SNR												
		4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	4	1	1	1	1	1	1	1	1	1	1	1	1	1
	4,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1	1
	5,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	1	1	1
	6,5	1	1	1	1	1	1	1	1	1	1	1	1	1
λ	7	1	1	1	1	1	1	1	1	1	1	1	1	1
	7,5	1	1	1	1	1	1	1	1	1	1	1	1	0
	8	1	1	1	1	1	1	1	1	1	0	1	0	0
	8,5	1	1	1	1	1	1	1	0	0	0	0	0	0
	9	1	1	1	1	1	0	1	0	0	0	0	0	0
	9,5	1	1	1	1	1	1	0	0	0	0	0	0	0
	10	1	1	1	0	0	0	0	0	0	0	0	0	0



(a) Para λ fixo.

(b) Para SNR fixo.

Figura 4.11: Falsos negativos para *DBSCAN*.

Nas duas comparações feitas entre as técnicas tradicionais e as modificadas,

observa-se que para valores altos de ambos os parâmetros, os resultados dos algoritmos se tornam próximos, sendo a região das tabelas de teste estatístico onde a hipótese nula é mais frequentemente não rejeitada. Isso ocorre porque, com altos valores de *SNR*, os picos são mais bem definidos, permitindo que a escolha de um λ alto elimine quase complemente pontos fora das regiões dos picos, tornando assim tais cenários mais simples para a etapa de clusterização. No cenário de exemplo da Figura 4.9, pode-se observar este fenômeno na última coluna.

De forma geral, para valores intermediários e baixos de um ou ambos os parâmetros, a modificação proposta gera alterações estatisticamente relevantes nos resultados. No caso do *mean shift*, o algoritmo tradicional tem um menor índice médio de falsos negativos, enquanto que para o *DBSCAN*, o algoritmo modificado tem o índice menor.

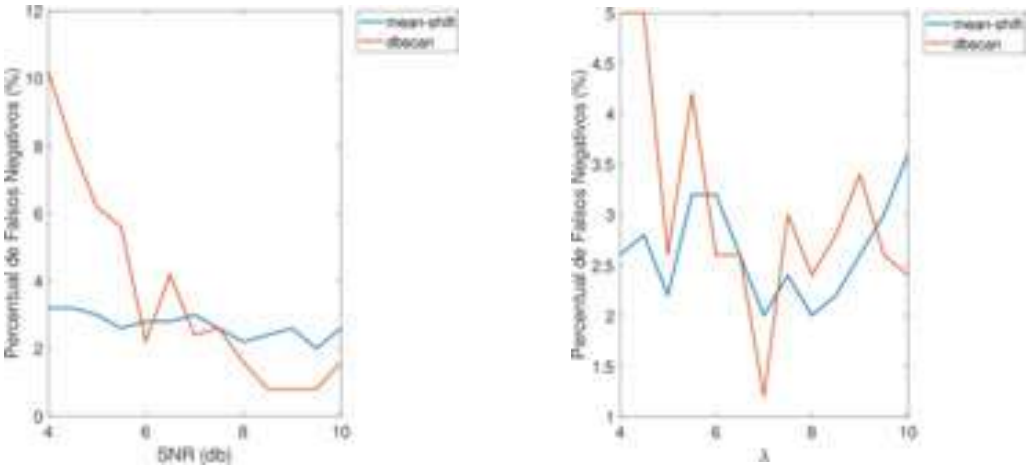
A Tabela 4.12 mostra o resultado estatístico da comparação entre o *mean shift* e *DBSCAN* modificados. Pode-se ver que a rejeição da hipótese nula se concentra na região onde os valores de *SNR* e λ são baixos.

Tabela 4.12: Comparação falsos negativos técnicas modificadas.

		SNR												
		4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	4	1	1	1	1	1	1	1	0	0	0	0	0	0
	4,5	1	1	1	1	1	1	1	1	0	0	0	0	0
	5	1	1	1	1	1	1	1	0	0	0	0	0	0
	5,5	1	1	1	1	1	1	0	0	0	0	0	0	0
	6	1	1	1	1	1	0	0	0	0	0	0	0	1
	6,5	1	1	1	1	0	0	0	0	0	0	1	0	0
λ	7	1	1	1	0	0	0	0	0	0	0	0	1	0
	7,5	1	1	1	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0
	8,5	1	0	0	0	0	0	0	0	0	0	0	0	0
	9	1	0	0	0	0	0	0	0	0	0	0	0	0
	9,5	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	1	0	0	0	0	0	0	0	0	0	0

A Figura 4.12 exibe uma comparação dos algoritmos modificados. A Figura 4.12(a) mostra o percentual de falsos negativos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.12(b) mostra o percentual de falsos negativos em função do λ para um valor fixo de *SNR*.

Considerando o resultado do teste estatístico e das figuras para parâmetros fixos, pode-se ver que, para a região onde ambos os parâmetros são baixos, o *mean shift* apresenta um percentual de falsos negativos menor que o *DBSCAN*. Para os outros balanços de *SNR* e λ, os resultados são próximos e não estatisticamente conclusivos.



(a) Para λ fixo. (b) Para SNR fixo.

Figura 4.12: Falsos negativos para algoritmos modificados.

4.1.2.2 Falso Positivos

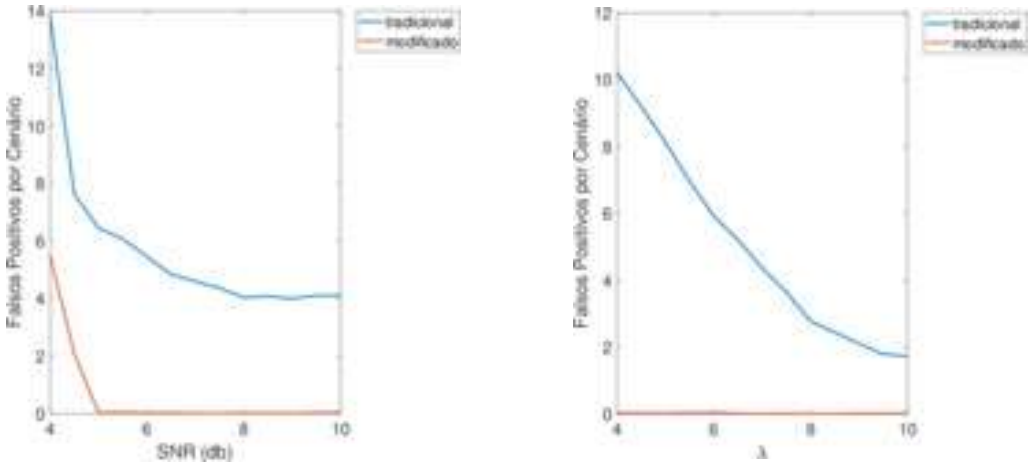
Nesta seção, são analisados os resultados para falso positivos, clusters gerados que não representem a posição real de um contato. A Tabela 4.13 exibe o resultado do teste estatístico *Wilcoxon Signed-ranks* pareado entre o algoritmo *mean shift* tradicional e modificado.

Tabela 4.13: Comparação falsos positivos para *mean shift*.

		SNR												
		4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	4	1	1	1	1	1	1	1	1	1	1	1	1	1
	4,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1	1
	5,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	1	1	1
	6,5	1	1	1	1	1	1	1	1	1	1	1	1	1
λ	7	1	1	1	1	1	1	1	1	1	1	1	1	1
	7,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	1	1	1	1	1
	8,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	9	1	1	1	1	1	1	1	1	1	1	1	1	1
	9,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	1	1	1	1	1	1	1	1	1

Para o algoritmo de *mean shift*, a modificação proposta gera alterações estatisticamente relevantes em todas as análises, independentemente dos parâmetros.

Os resultados das simulações para o *mean shift* tradicional e modificado, também podem ser vistos na Figura 4.13. A Figura 4.13(a) mostra o número de falsos positivos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.13(b) mostra o número de falsos positivos em função do λ para um valor fixo de *SNR*.



(a) Para λ fixo. (b) Para SNR fixo.

Figura 4.13: Falsos positivos para *mean shift*.

Combinando a análise do teste estatístico e da figura para parâmetros fixos, pode-se ver que o algoritmo modificado gera menos falsos positivos e que este resultado é estatisticamente relevante. O cenário de exemplo da Figura 4.9 demostra que, além da geração de múltiplos clusters por contato, o algoritmo tradicional também gera um maior número de clusters em regiões espúrias.

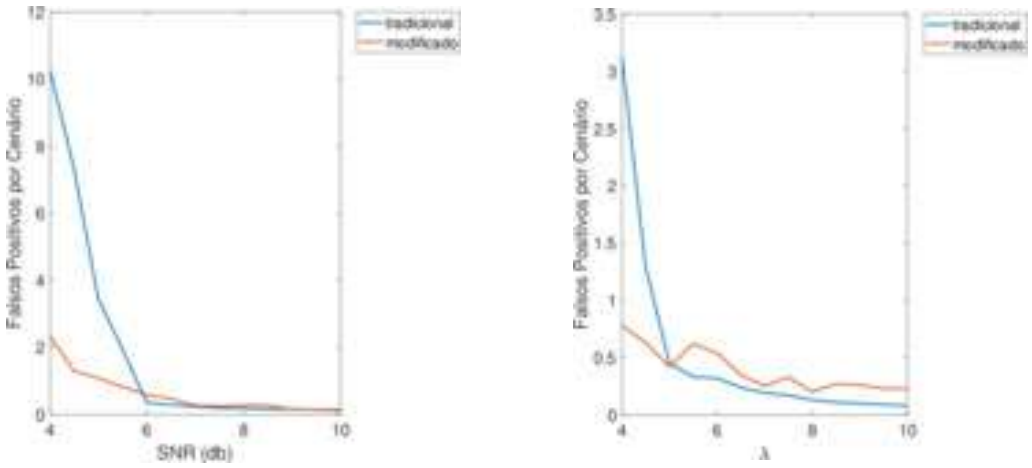
A Tabela 4.14 exibe o resultado da comparação do algoritmo de *DBSCAN* tradicional e modificado. A modificação proposta gera alterações estatisticamente relevantes quanto à geração de falsos positivos, predominantemente, na região de *SNR* abaixo de 8 dB, sendo menos provável a rejeição da hipótese nula para valores de λ intermediários.

Tabela 4.14: Comparação falsos positivos para *DBSCAN*.

		SNR												
		4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	4	1	1	1	1	0	1	1	1	1	0	0	0	0
	4,5	1	1	1	0	1	1	1	1	0	0	0	0	0
	5	1	1	1	1	1	1	1	0	0	0	0	0	1
	5,5	1	1	1	1	1	1	0	1	0	0	0	0	0
	6	1	1	1	1	1	1	1	1	0	0	0	0	0
	6,5	1	1	1	1	0	0	0	0	0	0	1	0	0
λ	7	1	1	1	1	1	1	0	0	0	0	0	0	0
	7,5	1	1	1	1	1	0	1	1	1	0	0	0	0
	8	1	1	0	1	1	1	1	0	0	0	0	0	0
	8,5	1	1	1	1	1	1	0	1	0	0	0	0	0
	9	1	0	1	1	1	1	1	1	0	1	0	0	0
	9,5	1	1	1	1	1	1	1	1	0	0	0	0	0
	10	0	1	1	1	1	1	1	1	1	0	0	0	0

Os resultados das simulações para o *DBSCAN* tradicional e modificado também

podem ser vistos na Figura 4.14. A Figura 4.14(a) mostra o número de falsos positivos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.14(b) mostra o número de falsos positivos em função do λ para um valor fixo de *SNR*.



(a) Para λ fixo.

(b) Para SNR fixo.

Figura 4.14: Falsos positivos para *DBSCAN*.

Combinando o resultado do teste estatístico com o da figura para parâmetros fixos, pode-se observar que a alteração proposta reduz a geração de falsos positivos na região de *SNR* abaixo de 8 dB. Para as demais, o número de falsos positivos é próximo entre as duas técnicas e a hipótese nula é majoritariamente não rejeitada.

Em termos de geração de falsos positivos para a análise ativa, a modificação proposta, de forma geral, reduziu significativamente o número menor de falsos positivos gerados para ambas as técnicas.

A Tabela 4.15 exibe o resultado da comparação do algoritmo *mean shift* e *DBSCAN* modificados. A maioria das combinações de *SNR* e λ apresentam diferenças estatisticamente relevantes.

Os resultados das comparações para o *mean shift* e o *DBSCAN* modificados também podem ser vistos na Figura 4.15. A Figura 4.15(a) mostra o número de falsos positivos em função da *SNR* para um valor fixo de λ, enquanto a Figura 4.15(b) mostra o número de falsos positivos em função do λ para um valor fixo de *SNR*.

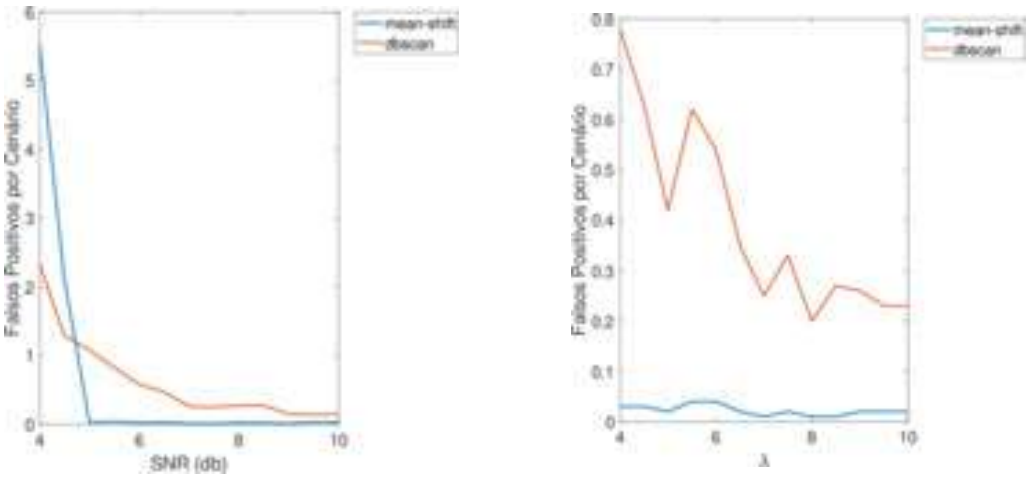
Na comparação entre as técnicas modificadas, para a maior parte das combinações dos parâmetros, o algoritmo *DBSCAN* gera um maior número de falsos positivos.

4.1.2.3 Estimativa de posição

Para todos os cenários simulados, foram avaliados os erros de marcação e distância dos clusters em relação às marcações e distâncias reais dos contatos. Os erros foram

Tabela 4.15: Comparação falsos positivos para técnicas modificadas.

		SNR												
		4	4,5	5	5,5	6	6,5	7	7,5	8	8,5	9	9,5	10
	4	1	1	1	1	1	1	1	1	1	1	1	1	1
	4,5	1	1	1	1	0	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1	0
	5,5	1	1	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	1	1	1
	6,5	1	1	1	1	1	1	1	1	1	1	1	1	1
λ	7	1	1	1	1	1	1	1	1	1	1	1	1	1
	7,5	1	1	1	1	1	1	1	1	1	1	1	1	0
	8	1	1	1	1	1	1	1	1	1	1	1	1	0
	8,5	0	1	1	1	1	1	1	1	1	1	1	1	0
	9	1	1	1	1	1	1	1	1	1	1	1	0	0
	9,5	1	1	1	1	1	1	1	1	1	1	1	1	0
	10	1	1	1	1	1	1	1	1	1	1	1	0	1



(a) Para λ fixo.

(b) Para SNR fixo.

Figura 4.15: Falsos positivos para algoritmos modificados.

comparados, independentemente do valor de *SNR* e λ, sendo analisados de forma global para todos os cenários. O valor de média e desvio padrão dos erros de marcação e distância para cada técnica estão dispostos na Tabela 4.16, sendo: $\bar{b}$, média do erro de marcação; σ_b, desvio padrão dos erros de marcação; $\bar{r}$, média do erro de distância; e σ_r, desvio padrão dos erros de marcação.

Tabela 4.16: Distribuições de marcação e distância.

	$\bar{b}(^\circ)$	$\sigma_b(^\circ)$	$\bar{r}$(yd)	σ_r(yd)
mean shift	0,017	2,123	1,319	17,158
mean shift modificado	0,019	1,445	-9,445	22,069
DBSCAN	0,015	1,773	-45,236	50,753
DBSCAN modificado	0,022	2,804	-42,672	50,245

Para a análise dos erros de marcação e distância, vale lembrar que nesta dissertação a menor unidade dos dados de entrada é de: 10 yd de distância; e 1° de marcação.

Os resultados de marcação indicam erro com média próxima de zero e os desvios padrões da ordem de poucas unidades de grau, estando praticamente todos os algoritmos dentro de uma mesma faixa de erro. Vale destacar que no *DBSCAN* a modificação aumentou a variância, enquanto que no *mean shift* a modificação reduziu a variância.

Já os resultados de distância apontam maiores discrepâncias entre as técnicas. O *mean shift* tradicional gera média e variância bem inferiores às demais técnicas. Entretanto, a média do *mean shift* modificado pode ser considerada dentro da mesma ordem de grandeza que a do algoritmo tradicional, dado que ambas as médias têm módulo menor que 10 yd.

O *DBSCAN* em ambos os casos gera um desvio da média, um *offset*, da ordem de quarenta e poucas jardas. Esse *offset* acontece pelo formato do cluster. Enquanto o *mean shift* busca o pico na migração da malha, o *DBSCAN* isola todos os elementos do cluster e, uma vez isolado, calcula a posição da detecção pelo ponderamento dos elementos do cluster.

O cluster do *DBSCAN* é uma seção de círculo, o que gera um erro na estimativa de distância mas não afeta a marcação. Os *DBSCANs* apresentam desvios padrões muito similares, mas com valores maiores que os algoritmos *mean shift*. A Figura 4.16 ilustra um cluster de exemplo, onde o ponto em preto é a verdadeira posição do contato, e o ponto em vermelho é o centro do cluster, calculado pela ponderação dos pontos.

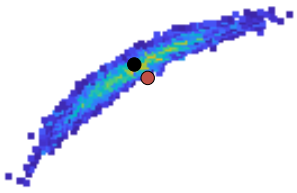


Figura 4.16: *DBSCAN*, *offset* na estimativa de distância. Em preto, a posição real; em vermelho, a posição do cluster.

Pode-se modificar o cálculo da posição dos clusters, ponderando-se as detecções em coordenadas polares ao invés de cartesianas. Esse *offset* pode ser um artefato de realização da ponderação em coordenadas cartesianas.

A modificação no *mean shift*, aumentou o desvio padrão do erro de distância,

mas manteve-se as diferenças inferiores a menor unidade de distância. A alteração no *DBSCAN* também não altera o desvio padrão do erro de distância significativamente. Além disso, pode-se ver que o desvio padrão do erro de distância da técnica *mean shift* é menor que a do *DBSCAN*.

Na prática da utilização de sonar ativo, a precisão da informação de marcação é muito mais crítica do que a informação de distância, sendo erros da ordem de poucos graus e algumas dezenas de jardas encontrados em todas as técnicas dentro dos limites aceitáveis de operação.

4.2 Acompanhamento

Neste capítulo são comparadas as técnicas de acompanhamento a partir de dados simulados, como se saídos da etapa de detecção, conforme explicado na Seção 3.1.1, primeiramente para o sonar passivo e subsequentemente para o sonar ativo. As técnicas de acompanhamento explicadas na Seção 2.3 serão denominadas, por simplicidade como: *ART* e α-β.

A entrada de dados para o acompanhamento é um conjunto de pontos com as *features* específicas para a análise passiva e ativa. Esses dados passam pelas etapas de atribuição e atualização para análise dos algoritmos descritos. A etapa de prospecção será discutida posteriormente em separado, pois é independente das técnicas e não interfere no desempenho delas.

4.2.1 Sonar passivo

4.2.1.1 Ajuste de Parâmetros

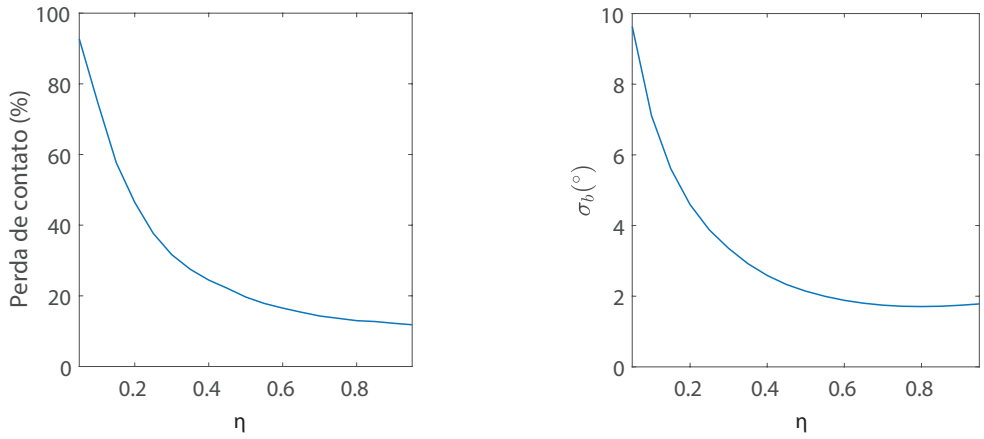
Nesta seção, são analisados os efeitos dos parâmetros de atualização das técnicas de acompanhamento. Para o *ART*, tais fatores correspondem ao fator de atualização, η, nos valores de $0,05$ a $0,95$, com passo de $0,05$, e para o α-β, os parâmetros α e β nos valores de $0,05$ a $0,95$ com passo de $0,05$ cada um. Os valores 0's e 1's não foram simulados por serem os casos extremos: acompanhamento com parâmetros 0 se mantém estático na posição inicial; acompanhamentos com parâmetros 1's seguem fielmente a última posição medida, não filtrando o ruído de medição.

Para analisar o efeito destes parâmetros nos acompanhamentos, foram simulados cem cenários, cada um deles com duzentos passos de atualização de posição. Além disso, foi variado o número de falsos negativos de 0% a 50%, de 10% em 10%, e o desvio padrão do erro de marcação dos clusters de entrada de $0°$ a $5°$, de $0,5°$ em $0,5°$. Em cada cenário, foi criado apenas um contato e inicializado um acompanhamento na verdadeira posição do contato e, no caso do α-β, com a verdadeira informação

de velocidade. Análises sobre como realizar as estimativas iniciais de posição e velocidade são abordadas na Seção 4.2.1.4.

O resultado compilado para o acompanhamento ART pode ser visto na Figura 4.17. A Figura 4.17(a) exibe o percentual de perda de contatos em função de η. Pode-se ver que, quanto maior for o η, menor é a quantidade de contatos perdidos.

Quanto maior for o η, mais plástico o acompanhamento será, e mais próxima a estimativa será da posição medida. Quanto menor for o η, mais estável o acompanhamento será, ficando mais imune às oscilações das medições, resultando numa maior inércia para a modificação de sua posição. Esse fenômeno é conhecido como dilema estabilidade-plasticidade [11].



(a) Perda de contato. (b) Marcação.

Figura 4.17: Efeito do fator de atualização.

A Figura 4.17(b) exibe o desvio padrão do erro de marcação do acompanhamento em função do η. O menor valor do gráfico ocorre com η em $0,8$, mas os valores praticamente se estabilizam para um η superior a $0,6$.

Já para o acompanhamento α-β, para facilitar a visualização, o resultado foi apresentado como na Figura 4.18, onde o eixo y representa os valores de α, o eixo x representa os valores de β, e o número de perdas é representado pelo mesmo mapa de cores da Figura 4.19.

Pode-se verificar que, em termos de perda de contatos, existem duas regiões em azul escuro, onde a perda é menor. Uma com β de $0,5$ e outra com β entre $0,2$ e $0,25$.

Fazendo uso do mesmo tipo de visualização, a Figura 4.20(a) exibe o resultado para o erro de marcação, enquanto a Figura 4.20(b) exibe o resultado para o erro de velocidade angular.

Quanto ao erro de estimativa de marcação, a região de β baixo e de α alto apresenta os melhores resultados. Quanto ao erro de estimativa de velocidade angular, os

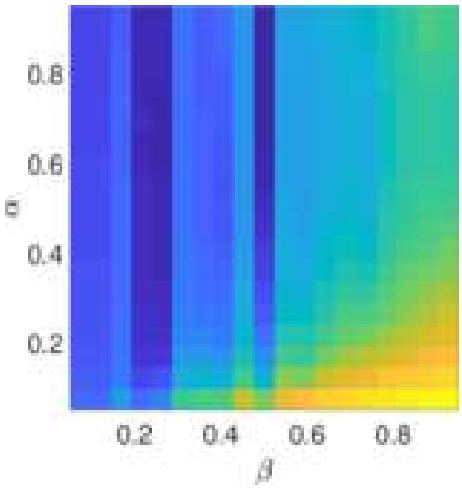


Figura 4.18: Perda de contatos α-β.

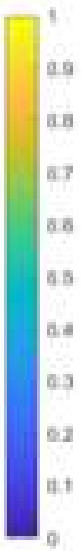


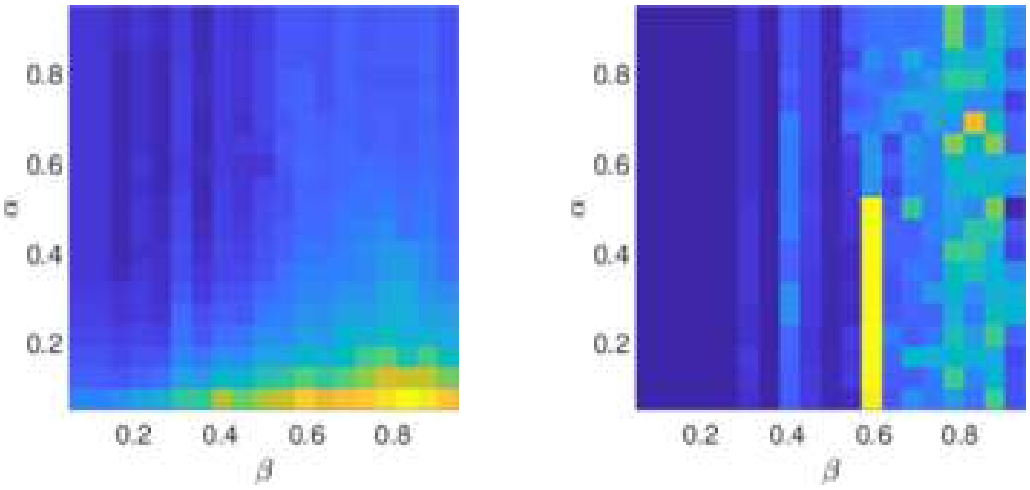
Figura 4.19: Mapa de cor, análise dos parâmetros α e β.

resultados apresentam-se muitas vezes independentes do valor de α, sendo melhores para valores mais baixos de β, em especial para β menor que $0,3$.

Analisando o resultado para sonar passivo no restante da dissertação, para fins de simplicidade, foram utilizados os valores da Tabela 4.17.

Tabela 4.17: Parâmetros escolhidos.

		Valores
Parâmetros	η	0,8
	α	0,7
	β	0,25



(a) Marcação. (b) Velocidade angular.

Figura 4.20: Efeito dos parâmetros α e β.

4.2.1.2 *Merge and Split*

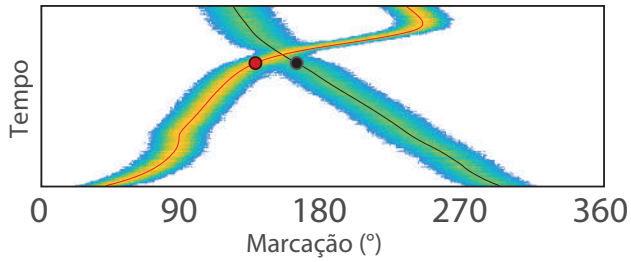
Nesta seção será analisada a capacidade dos algoritmos de acompanhamento em tratar o fenômeno de *merge and split*. Existem dois tipos de cenários onde ocorre o *merge and split*: quando ocorre um cruzamento de contatos próximos ou quando os contatos se aproximam e se afastam sem cruzar. Foram sorteados cenários com dois contatos e estes cenários foram simulados duzentos passos, só sendo registrados quando os contatos se aproximam a distâncias menores que 5°, onde as nuvens de detecções se sobreporiam.

Foram simulados cem cenários para cada um deles, variando-se o número de falsos negativos de 0% a 40%, de 10% em 10%; o desvio padrão do erro de marcação dos dados de entrada de 0° a 5°, de 1° em 1°.

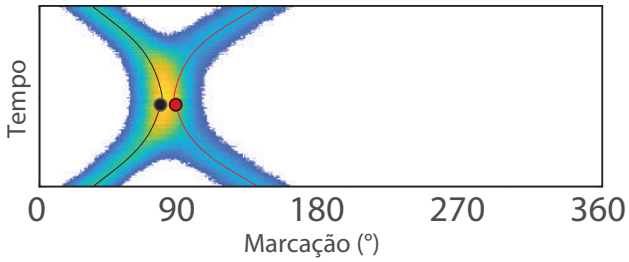
Os dois fenômenos são ilustrados para o cenário passivo na Figura 4.21, onde ocorre o *merge and split* entre o contato em vermelho e o contato em preto. Os pontos destacados representam o ponto de máxima aproximação dos contatos. A Figura 4.21(a) representa um cruzamento; enquanto a Figura 4.21(b), uma aproximação e afastamento.

A Tabela 4.18 mostra o percentual de inversão de contatos para o acompanhamento ART em cenários de *merge and split*, parametrizada pelo número de falsos negativos 0% a 40% na entrada (linhas da tabela) e pelo desvio padrão de marcação dos dados de entrada de 0° a 5° (colunas da tabela). O valor médio de inversões é $49,23\%$, e, para o acompanhamento ART, a inversão é quase independente dos parâmetros analisados.

A Tabela 4.19 mostra o resultado análogo ao da Tabela 4.18 para o acompanhamento α-β. O valor médio de inversões é $19,35\%$, e a quantidade de inversões aumenta com o aumento dos erros de marcação, mas são quase independentes do



(a) Exemplo de cenário com cruzamento.



(b) Exemplo de cenário com aproximação.

Figura 4.21: Cenários de *merge and split* passivos.

Tabela 4.18: Percentual de inversão de contatos *ART*.

		σ_b					
		0°	1°	2°	3°	4°	5°
falsos negativos	0%	60,0%	52,5%	45,5%	50,0%	42,0%	47,0%
	10%	59,5%	52,5%	50,5%	52,0%	44,0%	49,5%
	20%	54,5%	54,0%	53,5%	46,0%	46,5%	48,5%
	30%	53,0%	49,0%	46,5%	43,5%	49,0%	45,0%
	40%	48,0%	47,0%	46,0%	47,5%	47,0%	47,5%

número de falsos negativos.

Tabela 4.19: Percentual de inversão de contatos, α-β.

		σ_b					
		0°	1°	2°	3°	4°	5°
falsos negativos	0%	5,0%	19,5%	28,0%	26,5%	28,5%	28,5%
	10%	4,0%	17,0%	19,5%	23,5%	23,0%	33,5%
	20%	4,5%	13,0%	23,0%	23,5%	24,0%	27,0%
	30%	4,0%	15,5%	21,0%	22,0%	22,0%	24,5%
	40%	4,5%	15,0%	16,5%	20,5%	21,0%	22,5%

Dentro dos cenários simulados, a estimativa de movimento feita pelo filtro α-β reduz, significativamente, o número de inversões em cruzamento de contatos. Quanto menor o erro da etapa de detecção, maior é a probabilidade do acompanhamento com a estimativa de movimento resolver corretamente um caso de *merge and split*. Ambas as técnicas se mostraram bem imunes ao número de falsos negativos.

4.2.1.3 Prospecção

A prospecção, como descrita na Seção 3.3.3, consiste na identificação automática de novos contatos. Para obter e testar a identificação automática de contatos, foram realizadas simulações de cem cenários com dois contatos verdadeiros. Foi variado o número de falsos positivos por passo de simulação de 0 de 20, de 1 em 1, e a quantidade de passos de simulação de 2 a 10, de 1 em 1.

Os termos falso positivos e falso alarmes são tipicamente sinônimos, mas por questão de clareza, foi convencionado para esta seção que o número de falso positivos é o número de clusters gerados fora das posições dos contatos simulados, enquanto o número de falso alarmes é o número de acompanhamentos inicializados fora das posições reais dos contatos.

A Figura **??** exibe o resultado do número médio de falsos alarmes para os cenários simulados em função do número de passos. Quanto maior o número de passos de simulação levados em conta antes de validar um possível acompanhamento, menor é o número de falsos alarmes. Isso acontece porque os falsos positivos de entrada têm posição e distribuição aleatórias. Então, quanto maior for a quantidade de passos analisados, menor será a probabilidade de geração consistente de falsos positivos na região de um falso contato.

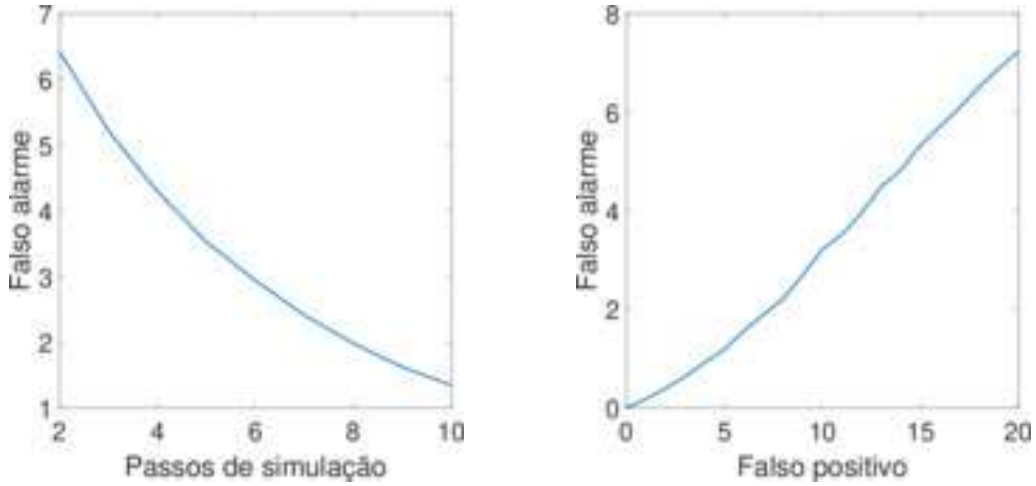


Figura 4.22: Falsos alarmes para cenários passivos.

A Figura **??** exibe o resultado do número médio de falso alarme para os cenários simulados em função do número de falsos positivos de entrada. Quanto maior for o número de falsos positivos na entrada, maior será o número de falsos alarmes.

4.2.1.4 Estimativas iniciais

Uma vez que o acompanhamento do possível contato seja feito e validado, a sequência natural é inicializar o acompanhamento α-β a partir da estimativa de marcação ($\tilde{b}$)

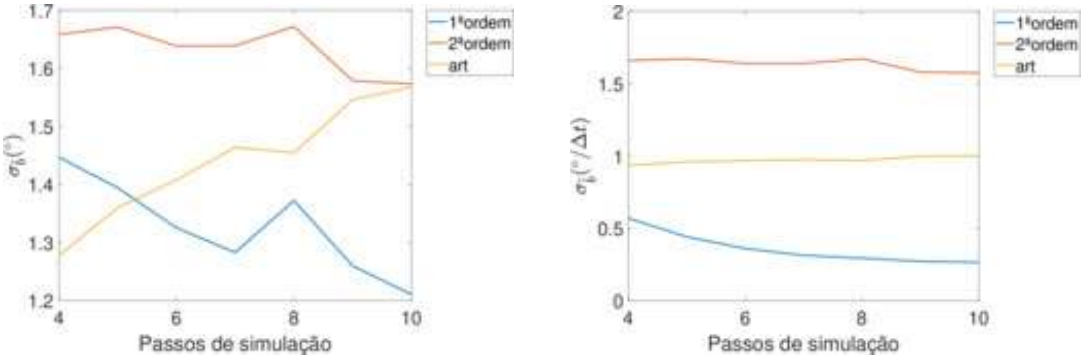
e da variação de marcação ($\tilde{\dot{b}}$) iniciais. Nesta seção, serão comparados três métodos para a realização dessas estimativas.

O primeiro método consiste no uso direto do algoritmo ART, considerando, como estimativa de marcação, a marcação atual do acompanhamento, e como estimativa de variação de marcação, a diferença entre a marcação atual e a marcação do passo anterior de simulação.

O segundo e o terceiro método são duas regressões lineares, de primeira e segunda ordem calculadas a partir das detecções atribuídas ao acompanhamento. As regressões serão feitas para as marcações existentes, sendo consideradas como estimativa de marcação inicial do contato a posição extrapolada pela regressão para o passo atual de simulação e a estimativa de variação da marcação a diferença entre a posição extrapolada pela regressão para próximo passo de simulação e o atual.

Para análise, foram simulados cem cenários com três contatos, variando-se o número de falsos positivos por passo de simulação entre 0 e 10, de 1 em 1, a quantidade de passos de simulação entre 4 e 10, de 1 em 1, e o desvio padrão do erro de marcação da detecção de entrada (σ_b) de 0° a 5°, de 0,5° em 0,5°.

A Figura 4.23(a) exibe o resultado do desvio padrão da estimativa de marcação ($\sigma_{\tilde{b}}$) em função da quantidade de passos na simulação. Pode-se ver que para valores maiores do que cinco passos de simulação, a regressão de primeira ordem obtém o melhor resultado e, para valores menores, o algoritmo ART obtém o melhor resultado.



(a) Estimativa de marcação. (b) Estimativa da variação de marcação.

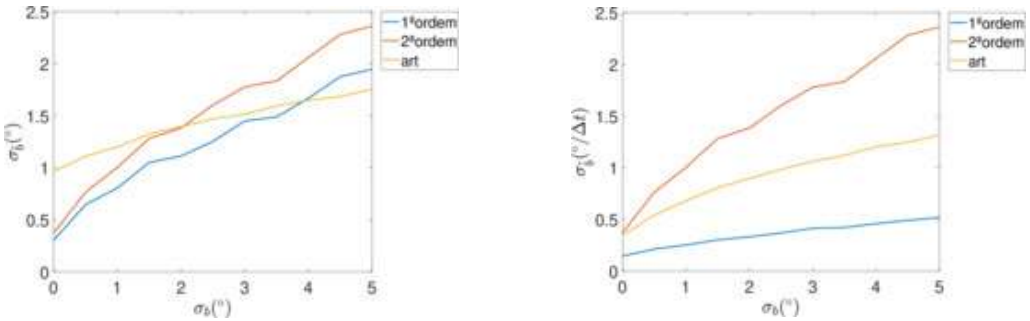
Figura 4.23: Estimativas em função da quantidade de passos.

A Figura 4.23(b) exibe o resultado do desvio padrão da estimativa de variação de marcação ($\sigma_{\tilde{\dot{b}}}$) em função da quantidade de passos na simulação. Em todos os cenários simulados, a regressão de primeira ordem obtém os melhores resultados, enquanto a regressão de segunda ordem o pior resultado.

A Figura 4.24(a) exibe o resultado do desvio padrão da estimativa de marcação ($\sigma_{\tilde{b}}$) em função de σ_b. Pode-se ver que o resultado do algoritmo ART apresenta os melhores resultados para valores acima de 4°. Entretanto, para valores de σ_b

menores que 4°, o resultado da regressão de primeira ordem é melhor.

A Figura 4.24(b) exibe o resultado do desvio padrão da estimativa de variação de marcação ($\sigma_{\tilde{\dot{b}}}$) em função de σ_b. O resultado indica que, para os cenários simulados, a regressão de primeira ordem obtém os melhores resultados, enquanto a regressão de segunda ordem o pior resultado.



(a) Estimativa de marcação. (b) Estimativa da variação de marcação.

Figura 4.24: Estimativas em função do erro de entrada.

De forma geral, nos cenários simulados, as estimativas de marcação apresentam resultados todos dentro da mesma faixa de dispersão, com a regressão de primeira ordem com resultados melhores para a maior parte dos cenários. Já para a estimativa de variação de marcação o resultado da regressão de primeira ordem é melhor para todos os cenários analisados.

4.2.2 Sonar ativo

4.2.2.1 Ajuste de Parâmetros

Nesta seção, são analisados os efeitos básicos dos parâmetros de atualização das técnicas de acompanhamento, para o ART, o fator de atualização η nos valores de $0,05$ a $0,95$, de $0,05$ em $0,05$, e para o α-β, os parâmetros α e β nos valores de $0,05$ a $0,95$, de $0,05$ em $0,05$ cada um.

Para analisar o efeito desses parâmetros nos acompanhamentos, foram simulados cem cenários cada um deles, com duzentos passos de atualização de posição, variando o número de falsos negativos de 0% a 50%, de 10% em 10%; o desvio padrão de posição de $0\,\text{yd}\angle 0°$ a $100\,\text{yd}\angle 5°$, de $20\,\text{yd}\angle 1°$ em $20\,\text{yd}\angle 1°$.

Em cada cenário, foi criado apenas um contato e inicializado um acompanhamento na verdadeira posição do contato e, no caso do α-β, com a verdadeira informação de rumo e velocidade como estimativa inicial. Análises sobre como realizar esta estimativa inicial de posição, rumo e velocidade serão abordados na Seção 4.2.2.4.

O resultado compilado para o acompanhamento ART pode ser visto nas Figuras 4.25 e 4.26. A Figura 4.25 exibe o resultado de percentual de perda de contato em

função de η. Pode-se ver que, para valores acima de $0,5$, o algoritmo atinge quase a estabilidade, com valor ótimo em $0,7$.

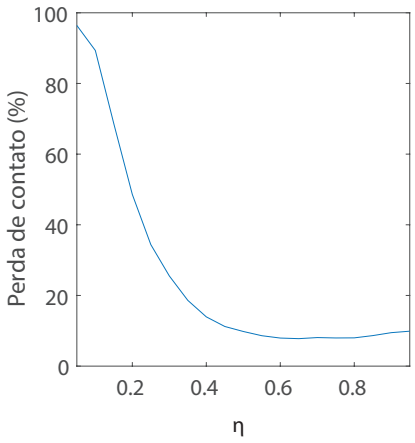
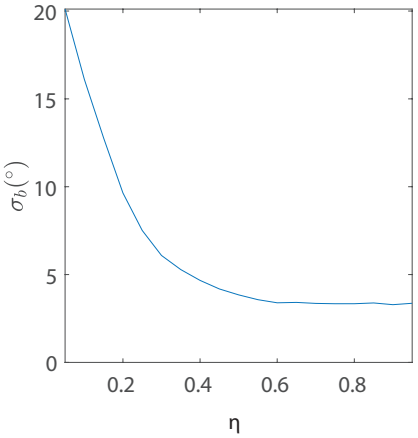
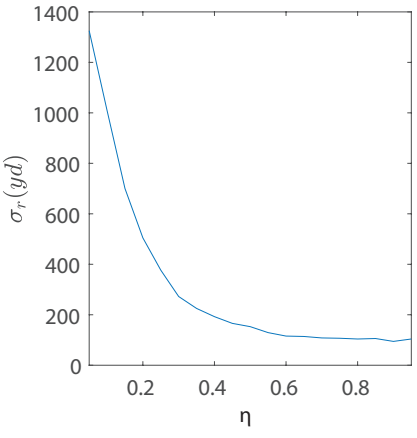


Figura 4.25: Perda de contato.

A Figura 4.26(a) exibe o desvio padrão do erro de estimativa da marcação do acompanhamento em função do η. Já a Figura 4.26(b) exibe o desvio padrão do erro de estimativa da distância do acompanhamento em função do η. O mesmo efeito da Figura 4.25 pode ser visto tanto no erro de marcação, quanto no erro de distância. O resultado estabiliza para valores de η a partir de $0,5$, entretanto, em ambos os casos o ponto mínimo é alcançado em $0,5$.



(a) Marcação.



(b) Distância.

Figura 4.26: Efeito do fator de atualização.

Para o acompanhamento α-β, para facilitar a visualização, os resultados são apresentados em mapa de cor, como na Seção 4.2.1.1. A Figura 4.27 exibe o percentual de perda de contatos, onde o eixo y representa os valores de α, o eixo x representa os valores de β, e o número de perdas é representado pelo mesmo mapa de cor da Figura 4.19. Pode-se perceber que, em termos de perda de contatos, a região de

menor perda de contatos ocorre para valores baixos de β e valores intermediários de α.

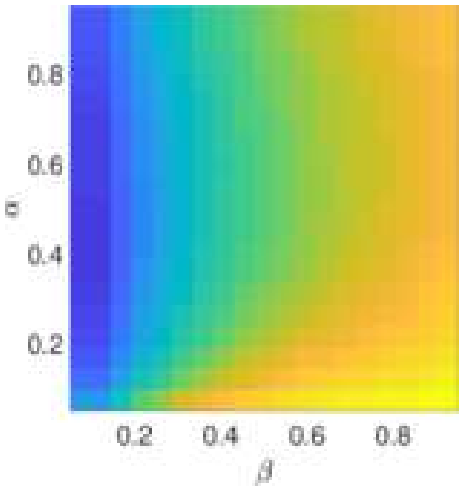


Figura 4.27: Perda de contatos α-β.

No mesmo tipo de visualização, pode-se ver o resultado para o erro de estimativa de distância, na Figura 4.28(a); o erro de estimativa de marcação, na Figura 4.28(b); o erro de estimativa de velocidade, na Figura 4.28(c); e o erro de estimativa de rumo, na Figura 4.28(d).

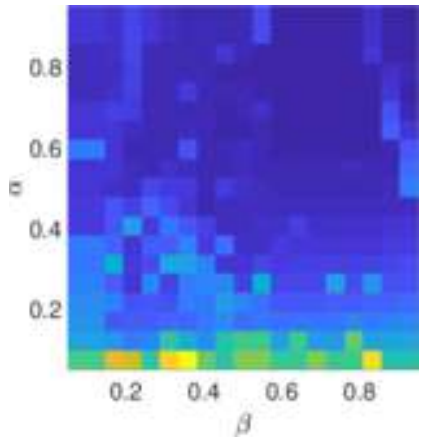
O resultado para o erro de estimativa de distância tem ponto mínimo na região de β acima de $0,5$ e α acima de $0,6$, mas os resultados são próximos para grande parte dos valores, exceto quando um dos parâmetros é inferior a $0,3$. O resultado do erro de estimativa de marcação tem mínimo na região de β abaixo de $0,2$ e α entre entre $0,3$ e $0,5$.

O resultado do erro de estimativa de velocidade exibe um gradiente quase constante, tendo o melhor resultado quanto maior α e menor β. O resultado do erro de estimativa de rumo tem mínimo para β abaixo de $0,2$ e α próximo de $0,4$.

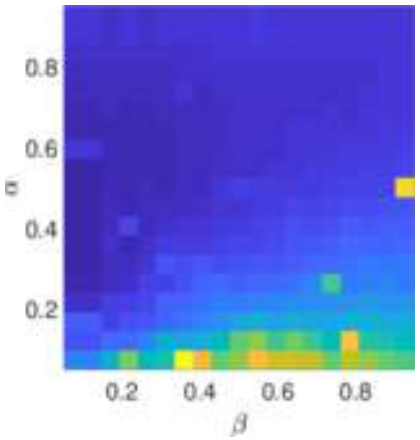
Analisando o resultado para sonar ativo e visando minimizar os erros de estimativas de marcação, rumo, velocidade e perda de contatos, abdicando um pouco da estimativa de distância, por ser menos crítica para as aplicações de sonares ativos, serão utilizados os valores da Tabela 4.20, no restante desta dissertação.

Tabela 4.20: Parâmetros escolhidos.

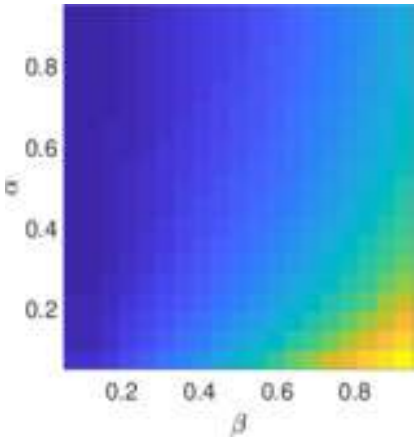
		Valores
Parâmetros	η	0,5
	α	0,5
	β	0,1



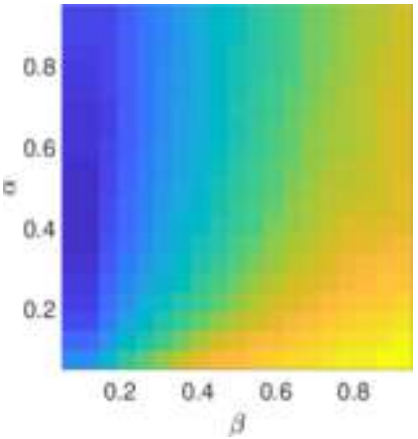
(a) Erro de estimativa de distância.



(b) Erro de estimativa de marcação.



(c) Erro de estimativa de velocidade.



(d) Erro de estimativa de rumo.

Figura 4.28: Efeito dos parâmetros α e β.

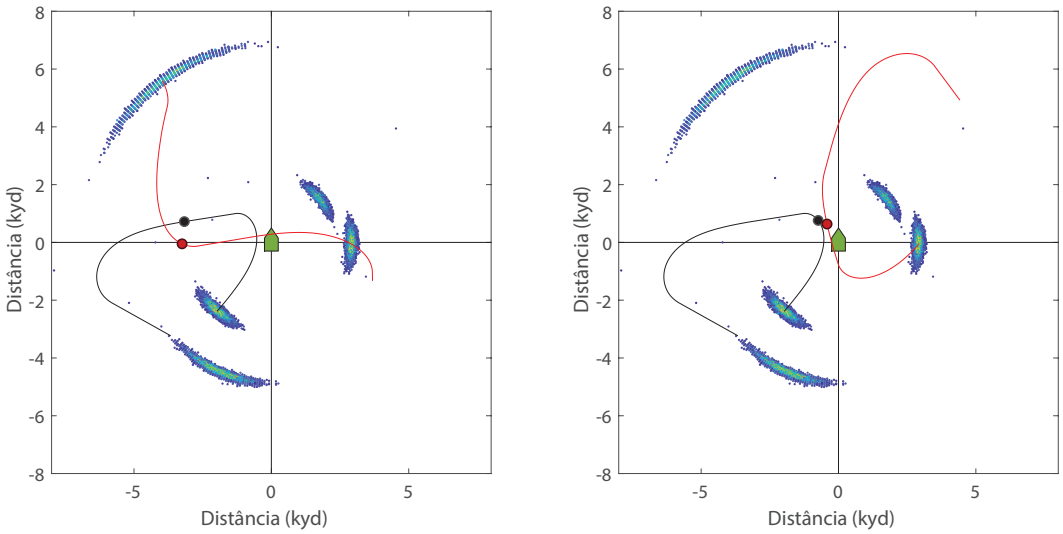
4.2.2.2 *Merge and Split*

Nesta seção, é analisada a capacidade dos algoritmos de acompanhamento em tratar o fenômeno de *merge and split*. Existem dois tipos de cenários onde ocorre o *merge and split*: quando ocorre um cruzamento de contatos próximos ou quando os contatos se aproximam e se afastam sem se cruzar. Foram sorteados cenários com cinco contatos e estes cenários foram evoluídos por duzentos passos, só sendo registrados quando dois contatos se aproximam a distâncias menores que 800 yd, onde as nuvens de detecções se sobreporiam.

Foram simulados cem cenários, cada um deles variando o número de falsos negativos de 0% a 40%, de 10% em 10%; o desvio padrão da posição de 0 yd$\angle 0°$ a 100 yd$\angle 5°$, de 20 yd$\angle 1°$ em 20 yd$\angle 1°$.

Os dois fenômenos são ilustradas para o cenário ativo na Figura 4.29, onde ocorre o *merge and split* entre o contato em vermelho e o contato em preto, com suas detecções representadas na sua posição final. Os pontos destacados representam o ponto de máxima aproximação dos contatos, na Figura 4.29(a) exibindo um cenário de cruzamento, enquanto a Figura 4.29(b) exibe o cenário de aproximação e

afastamento.



(a) Exemplo de cenário com cruzamento. (b) Exemplo de cenário com aproximação.

Figura 4.29: Cenários de *merge and split* ativos.

A Tabela 4.21 mostra o percentual de inversão de contatos para o acompanhamento ART em cenários de *merge and split*, parametrizada pelo número de falsos negativos 0% a 40% na entrada (linhas da tabela), e do desvio padrão de posição (σ_p) de $0\,\text{yd}\angle 0°$ a $100\,\text{yd}\angle 5°$ (colunas da tabela). O valor médio de inversões é $47,22\%$ e, para o acompanhamento ART, não parece existir uma correlação clara dos resultados e dos parâmetros analisados.

Tabela 4.21: Percentual de inversão de contatos, ART.

		$\sigma_p(\text{yd}\angle°)$					
		$0\angle 0$	$20\angle 1$	$40\angle 2$	$60\angle 3$	$80\angle 4$	$100\angle 5$
falsos negativos	0	35%	$38,5\%$	39%	$44,5\%$	52%	$47,5\%$
	10%	$40,5\%$	47%	$40,5\%$	49%	$52,5\%$	$52,5\%$
	20%	$43,5\%$	44%	51%	$54,5\%$	$46,5\%$	45%
	30%	$51,5\%$	51%	$48,5\%$	50%	53%	$51,5\%$
	40%	$50,5\%$	50%	$52,5\%$	$51,5\%$	$48,5\%$	$39,5\%$

A Tabela 4.22 mostra o resultado análogo ao da Tabela 4.21 para o acompanhamento α-β. O valor médio de inversões é $5,17\%$.

Pode-se verificar que, dentro dos cenários analisados, a estimativa de movimento feita pelo filtro α-β reduz significativamente o número de inversões em cruzamento de contatos.

Vale ressaltar que os resultados de inversão do acompanhamento, tanto para sonar ativo quanto para passivo, a técnica ART obtém resultado próximo a 50%, indicando que a inversão é praticamente aleatória, o que é coerente com o fato de

Tabela 4.22: Percentual de inversão de contatos, α-β.

		σ_p(yd$\angle^\circ$)					
		$0\angle0$	$20\angle1$	$40\angle2$	$60\angle3$	$80\angle4$	$100\angle5$
falsos negativos	0	0%	9%	20,5%	12,5%	5%	4,5%
	10%	0%	14%	18%	4,5%	3,5%	1,5%
	20%	0%	13,5%	13,5%	4,5%	1%	0%
	30%	0%	11,5%	7,5%	3%	1,5%	0,5%
	40%	1,5%	11%	4,5%	1,5%	1%	1%

não haver nenhuma estimativa de movimento, cabendo ao acaso inverter ou não o acompanhamento.

Já para α-β, o resultado médio para sonar ativo é da ordem de 5%, enquanto o sonar passivo é da ordem de 18%, coerente com a estimativa de movimento, que é mais próxima ao real no sonar ativo, porque enquanto a estimativa de movimento do sonar ativo é feita em duas dimensões equivalente ao movimento real do contato, a estimativa de movimento do sonar passivo é feita com base em uma projeção do movimento real.

4.2.2.3 Prospecção

A prospecção, como descrita na Seção 3.3.3, consiste na identificação automática de novos contatos. Para obter e testar a identificação automática de contatos, foram realizadas simulações de cem cenários com três contatos verdadeiros para cada parâmetro analisado, variou-se a quantidade de falsos positivos por passo de simulação de 0 de 100, de 2 em 2, e a quantidade de passos de simulação de 2 a 10, de 1 em 1.

Os termos falsos positivos e falsos alarmes são tipicamente sinônimos, mas por questão de clareza, foi convencionado para essa seção, assim como na Seção 4.2.1.3, que o número de falsos positivos é o número de clusters gerados fora das posições dos contatos simulados, enquanto o número de falsos alarmes é o número de acompanhamentos inicializados fora das posições reais dos contatos.

A Figura **??** exibe o resultado do número médio de falso alarme para os cenários simulados em função do número de passos. Quanto maior o número de passos de simulação levados em conta antes de se validar um possível acompanhamento, menor é o número de falsos alarmes. Isto acontece, porque os falsos positivos de entrada têm posição e distribuição aleatória. Então, quanto maior for a quantidade de passos analisados, menor será a probabilidade de geração consistente de falsos positivos na região de um falso contato.

A Figura **??** exibe o resultado do número médio de falsos alarmes para os cenários simulados em função do número de falsos positivos de entrada. Quanto maior for o número de falsos positivos na entrada, maior será o número de falsos alarmes.

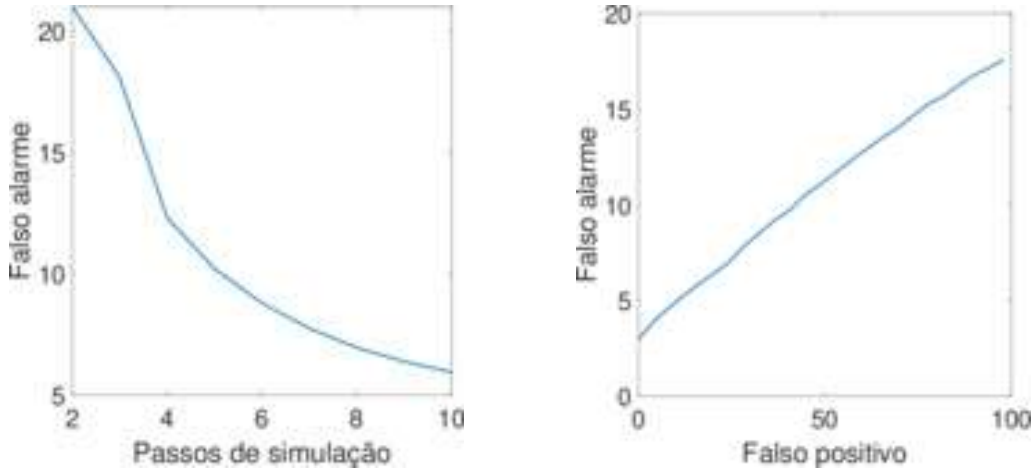


Figura 4.30: Falsos alarmes para cenários ativos.

4.2.2.4 Estimativas iniciais

Uma vez que o acompanhamento do possível contato seja feito e validado, a sequência natural é inicializar o acompanhamento α-β a partir da estimativa de marcação ($\tilde{b}$), de distância ($\tilde{r}$), de rumo ($\tilde{\theta}$) e de velocidade ($\tilde{v}$). Nesta seção é analisado o erro dessas estimativas pelos métodos ART e regressões de primeira e segunda ordem, assim como foi feito na Seção 4.2.1.4 para o sistema passivo.

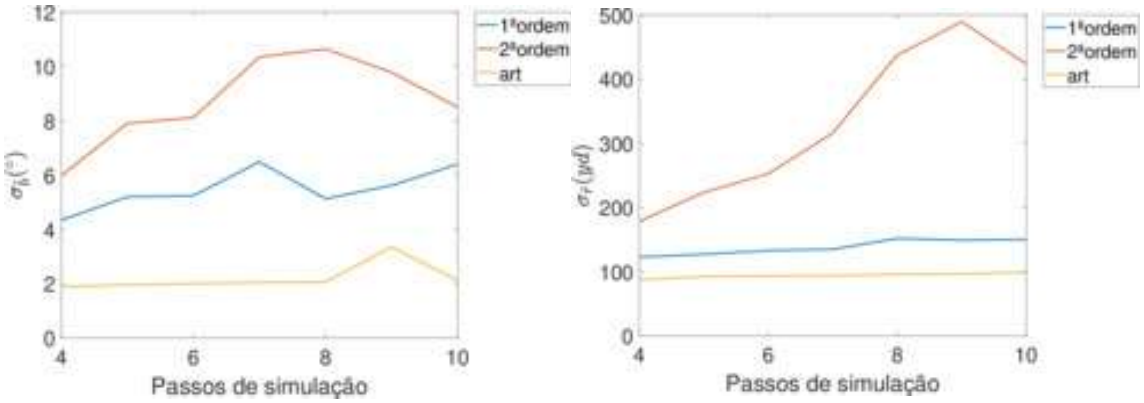
Para análise, foram simulados cem cenários com três contatos para cada parâmetro analisado. Foram variados o número de falsos positivos por passo de simulação de 0 de 10, de 1 em 1; a quantidade de passos de simulação de 4 a 10, de 1 em 1; e o erro de posição das detecções de entrada com a desvio padrão (σ_p) de $0\,\text{yd}\angle 0°$ a $100\,\text{yd}\angle 5°$, de $10\,\text{yd}\angle 0{,}5°$ em $10\,\text{yd}\angle 0{,}5°$.

A Figura 4.31(a) exibe o resultado do desvio padrão da estimativa de marcação ($\sigma_{\tilde{b}}$) em função da quantidade de passos na simulação. Pode-se ver que, nos cenários analisados, o algoritmo ART apresenta o melhor resultado, enquanto que na regressão de segunda ordem, o pior resultado independente do número de passos.

A Figura 4.31(b) exibe o resultado do desvio padrão na estimativa de distância ($\sigma_{\tilde{r}}$) em função da quantidade de passos na simulação, com o resultado equivalente ao erro de estimativa de marcação.

A Figura 4.32(a) exibe o resultado do desvio padrão da estimativa de rumo ($\sigma_{\tilde{\theta}}$) em função da quantidade de passos na simulação. O resultado indica que, para os cenários simulados, a regressão de primeira ordem apresenta melhor resultado, independente do número de passos. O mesmo vale para a estimativa de velocidade ($\sigma_{\tilde{v}}$), conforme ilustra a Figura 4.32(b).

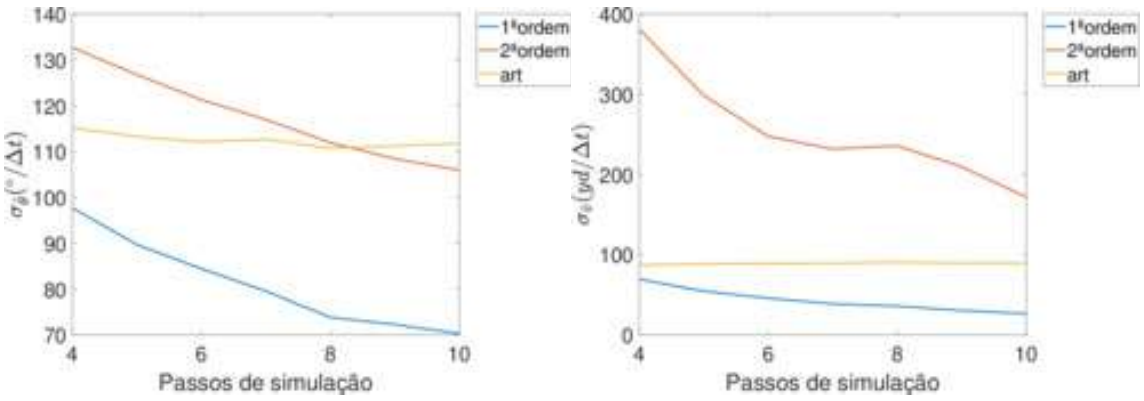
A Figura 4.33(a) exibe o resultado do desvio padrão da estimativa de marcação ($\sigma_{\tilde{p}}$) em função de σ_b. Pode-se ver que o resultado do algoritmo de ART apresenta resultados menos sensíveis aos erros da etapa de detecção, obtendo melhores resul-



(a) Estimativa de marcação. (b) Estimativa de distância.

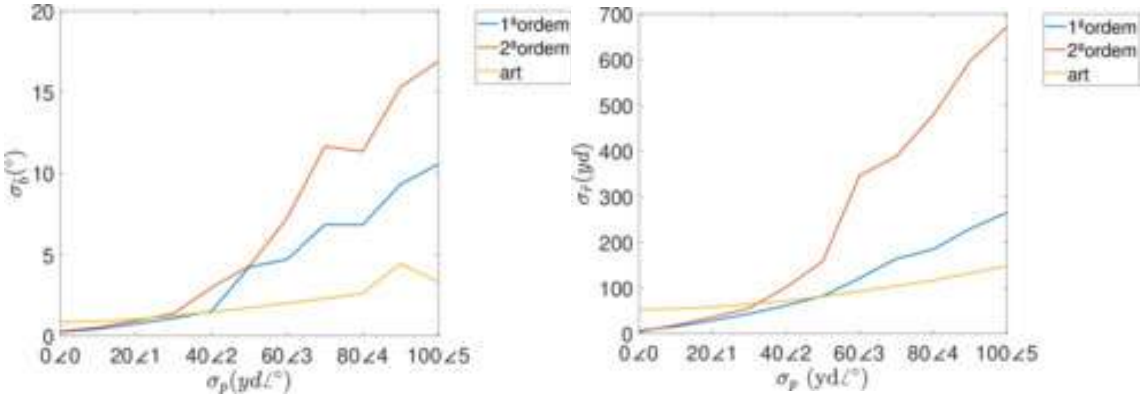
Figura 4.31: Estimativas de posição em função da quantidade de passos.

tados na média. Entretanto, para valores de σ_p menores que 40 yd$\angle 2°$, o resultado da regressão de primeira ordem é melhor.



(a) Estimativa de rumo. (b) Estimativa de velocidade.

Figura 4.32: Estimativas de rumo e velocidade em função da quantidade de passos.



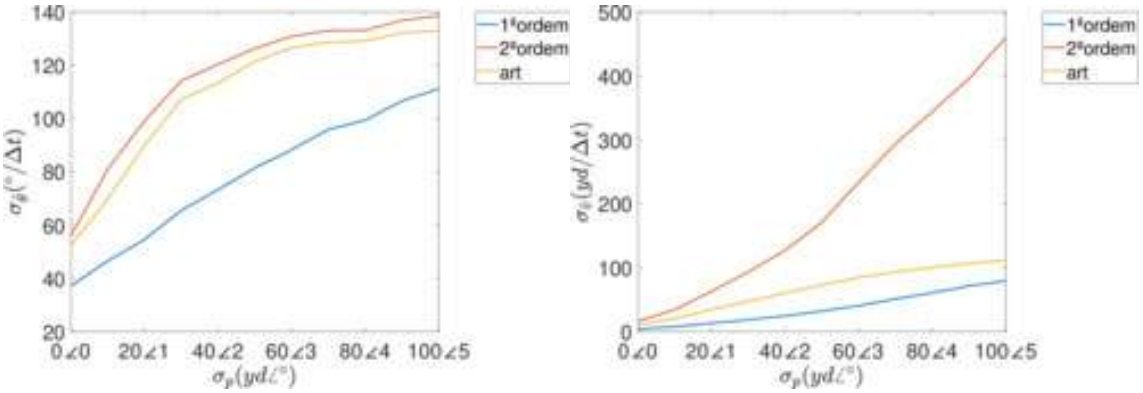
(a) Estimativa de marcação. (b) Estimativa de distância.

Figura 4.33: Estimativas de posição em função do erro de entrada.

Efeito similar pode ser visto na Figura 4.33(b), para o resultado do desvio padrão da estimativa de distância ($\sigma_{\tilde{r}}$) em função de σ_p. Para valores de σ_p menores que

50 yd∠2,5°, a regressão de primeira ordem apresenta os melhores resultados e, para valores maiores, o algoritmo de ART apresenta os melhores resultados.

A Figura 4.34(a) exibe o resultado do desvio padrão da estimativa de rumo ($\sigma_{\tilde{\theta}}$) em função de σ_p. Pode-se ver que o resultado da regressão de primeira ordem apresenta resultados melhores ao longo de toda a faixa analisada e a regressão de segunda ordem, os piores resultados. O mesmo vale para a estimativa de velocidade ($\sigma_{\tilde{v}}$), conforme ilustra a Figura 4.34(b).



(a) Estimativa de rumo.

(b) Estimativa de velocidade.

Figura 4.34: Estimativas de rumo e velocidade em função do erro de entrada.

Nos cenários simulados, as estimativas de marcação e de distância para os valores mais baixos dos erros de posição da etapa de detecção, a regressão de primeira ordem apresenta melhores resultados. Já para valores maiores, o algoritmo ART apresenta o melhor resultado. Para a estimativa de rumo e velocidade, os resultados da regressão de primeira ordem são melhores que as demais técnicas independente dos parâmetros analisados.

Os resultados desta seção são similares aos da Seção 4.2.1.3, indicando que as melhores estimativas de posição se dividem entre as técnicas ART e de regressão de primeira ordem, enquanto, para as estimativas de velocidade, a regressão de primeira ordem sempre apresenta melhores resultados, dentro dos cenários analisados.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1 Conclusões

Neste trabalho foi abordada a construção de um sistema autônomo de detecção e acompanhamento de contatos, analisando-se cada uma das etapas bem como, os impactos quanto a escolha de parâmetros críticos para cada uma delas.

Tudo se inicia com a apresentação dos dados do processamento sonar, em conjunto de dados de *features*: marcação e energia para sonar passivo; ou marcação, distância e energia, para sonar ativo.

A primeira etapa da detecção é a conversão dos dados de entrada em uma etapa de detecção por limiar (λ) para um conjunto de pontos no espaço de coordenadas das *features*. A escolha do λ é crítica para a performance dos algoritmos de clusterização, pois impacta diretamente no número de pontos, na concentração e na sua dispersão.

Em uma segunda etapa, os pontos coordenados detectados são aplicados a alguma técnica de clusterização para a criação dos clusters e redução do número de detecções. As técnicas modificadas se mostraram menos sensíveis ao número de pontos e aos valores de *SNR* e λ, além de exibir um melhor desempenho para valores de *SNR* baixas, sendo mais adequados à aplicação real, onde o controle do λ é complicado e os contatos reais estão susceptíveis a *SNR* baixas.

Para os cenários passivos analisados, ambos os algoritmos modificados apresentaram performance similar. Enquanto o *mean shift* gera uma quantidade menor de falsos positivos, o *DBSCAN* gera menos falsos negativos, principalmente para *SNR* baixos. Quanto à estimativa de posição, os resultados também são semelhantes entre as técnicas modificadas. De forma prática, ambas as técnicas podem ser aplicadas, onde a técnica mais adequada vai depender da aplicação em foco. Vale adicionar que não foi analisado o tempo de execução dos algoritmos, que é crítico para o caso de uma aplicação em tempo real.

Para sonares ativos, o desempenho da técnica *mean shift* modificada foi, de forma

geral, superior ao da técnica *DBSCAN* modificada, sendo a mais indicada para a aplicação. Em termos de falsos negativos, sendo superior na região de parâmetros baixos e com desempenho próximo nas demais. Já para falsos positivos, a *mean shift* modificada foi superior em quase todos os cenários, rejeitando a hipótese nula para quase todos os pares de parâmetros analisados. Além disso, o *mean shift* apresenta menores médias e desvio padrões, tanto em erro de marcação, quanto em erro de distância.

Cada conjunto de detecções é aplicada ao sistema de acompanhamento, passando pelas três etapas: atribuição, atualização e prospecção. A atribuição dos clusters foi feita por maior proximidade entre os clusters e as posições estimadas dos contatos.

A atualização das posições dos contatos, após a atribuição, é feita e comparada entre o clássico algoritmo *ART* e uma modificação do sistema de atualização do algoritmo, para inserção da estimativa de posição e velocidade por um filtro α-β. Os resultados deste trabalho indicam a importância da análise de movimento para a solução do problema de *merge and split*, apresentando o acompanhamento α-β uma redução significativa das inversões nos mesmos cenários quando comparado ao acompanhamento *ART*, tanto para o caso ativo quanto passivo.

A prospecção de novos contatos é uma etapa essencial em sistemas autônomos e para criação automática de contatos. Para essa etapa, pouco discutida na literatura, foi proposto um sistema de criação e validação dos possíveis contatos. Os resultados desta etapa são simples dentro das análises, quanto maior é o número de passos para análise, menor será a quantidade de falsos alarmes. Entretanto, maior o tempo entre o surgimento de um contato e a detecção do mesmo. Na prática, esse intervalo mínimo para a detecção pode ser crítico para certas aplicações, compensando a criação de falsos alarmes, como, por exemplo, para a detecção de torpedos. Em outros casos, esse intervalo não é tão crítico e pode-se abdicar desse tempo de resposta visando gerar um menor número de falsos alarmes.

Por fim, ainda foram analisadas três formas de realizar as estimativas iniciais das posições e velocidades dos contatos após a validação feita na etapa de prospecção. As estimativas iniciais de posição têm resultados divididos entre o algoritmo *ART* e uma regressão linear de primeira ordem. Já para estimativas de velocidade, a regressão linear de primeira ordem obteve os melhores resultados em todos os cenários analisados, tanto para o passivo, quanto para o ativo.

5.2 Trabalhos Futuros

Existem diversas partes da construção de sistemas autônomos de detecção e acompanhamento que carecem de maiores análises, sendo um tema muito amplo e abrangente. Inicialmente, é necessária a validação do sistema completo com dados reais

para validação das análises feitas neste trabalho. Para isso, seria necessária a gravação de um cenário controlado com informações precisas da posição da plataforma e do(s) contato(s) conhecido(s). Vale ressaltar que seria indicada a gravação de posição por GPS (*Global Positioning System*), dado que a gravação com AIS (*Automatic Identification System*) tem atrasos e imprecisões que inviabilizam validações precisas das estimativas de marcação e distância, mas poderiam ser utilizadas para uma análise preliminar.

Para a etapa de detecção, pode-se analisar e comparar as técnicas de clusterização modificadas propostas neste trabalho com outras técnicas de clusterização modificando-as seguindo a lógica de separação das *features* espaciais, como por exemplo: *Distributed Combining Algorithm* (*DCA*) [25], *Ordering Point To Identify Clustering Structure* (*OPTICS*) [26], *fuzzy c- means* [27] e *Distributed Model based Clustering* (*DMC*) [28]. Além disso, pode-se comparar com outras técnicas de detecção de pico.

Em termos de acompanhamento, na etapa de atribuição, este trabalho se restringiu à atribuição por proximidade espacial. Poderia-se analisar outras alternativas como considerar a *feature* energia, média dos pontos na região, entre outras.

Na etapa de atualização, pode-se analisar outras técnicas de análise de movimento dos contatos como filtro de Kalman, principalmente para os sistemas sonares passivos. Poderia-se tentar estimar a distância, por métodos como Ekelund [30], porque enriquecendo-se a estimativa de movimento dos contatos, espera-se uma melhora nos resultados de tratamento de *merge and split*.

Para a etapa de prospecção, pode-se estudar outras formas de validação de um possível contato para redução de falsos alarmes, por exemplo, utilizar as informações intermediárias de velocidade e não considerar contatos com alterações bruscas. Além disso, pode-se estudar como combinar o resultado do algoritmo *ART* ao da regressão de primeira ordem para melhor estimar a posição e a velocidade iniciais dos acompanhamentos.

Referências Bibliográficas

[1] HODGES, R. P. *Underwater acoustics: analysis, design, and performance of sonar*. Chichester, West Sussex [England] ; Hoboken, NJ, J. Wiley, 2010. ISBN: 978-0-470-68875-5.

[2] WAITE, A. D. *Sonar for practising engineers*. 3rd ed ed. Chichester, Wiley, 2002. ISBN: 978-0-471-49750-9. OCLC: ocm47271347.

[3] NIELSEN, R. O. *Sonar signal processing*. The Artech House acoustics library. Boston, Artech House, 1991. ISBN: 978-0-89006-453-5.

[4] BOZZI, F. D. A. *Conformação de Feixe em Sonar Passivo para um Arranjo Cilíndrico de Hidrofones*. Tese de Mestrado, Universidade Federal do Rio de Janeiro, Brasil, 2016.

[5] TREES, H. L. V. *Detection, Estimation, and Modulation Theory: Part IV*. Hoboken, John Wiley & Sons, 2005. ISBN: 978-0-471-46383-2. Disponível em: <`http://public.eblib.com/choice/publicfullrecord.aspx?p=221339`>. OCLC: 475926535.

[6] CHEN, C.-H., LEE, J.-D., LIN, M.-C. "Classification of Underwater Signals Using Neural Networks", *Tamkang Journal of Science and Engineering*, v. 3, 06 2000.

[7] CARBONE, C. P., KAY, S. M. "A Novel Normalization Algorithm Based on the Three-Dimensional Minimum Variance Spectral Estimator", *IEEE Transactions on Aerospace and Electronic Systems*, v. 48, n. 1, pp. 430–448, jan. 2012. ISSN: 0018-9251. doi: 10.1109/TAES.2012.6129646. Disponível em: <`http://ieeexplore.ieee.org/document/6129646/`>.

[8] ESQUEF, P. A., BISCAINHO, L., VÄLIMÄKI, V., et al. "Removal of Long Pulses from Audio Signals Using Two-pass Split-Window Filtering". p. 9, 05 2002.

[9] LAMPERT, T. A., O'KEEFE, S. E. "A survey of spectrogram track detection algorithms", *Applied Acoustics*, v. 71, n. 2, pp. 87–100, fev. 2010. ISSN:

0003682X. doi: 10.1016/j.apacoust.2009.08.007. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0003682X09001959>.

[10] GARIMA, GULATI, H., SINGH, P. "Clustering Techniques in Data Mining: A Comparison". p. 6, 03 2015.

[11] ALPAYDIN, E. *Introduction to machine learning*. Adaptive computation and machine learning. Third edition ed. Cambridge, Massachusetts, The MIT Press, 2014. ISBN: 978-0-262-02818-9.

[12] NIKHARE, N. B., PRASAD, P. S. "A review on inter-cluster and intra-cluster similarity using bisected fuzzy C-mean technique via outward statistical testing". In: *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 215–217, Coimbatore, jan. 2018. IEEE. ISBN: 978-1-5386-0807-4. doi: 10.1109/ICISC.2018.8399066. Disponível em: <https://ieeexplore.ieee.org/document/8399066/>.

[13] SINGH, D., GOSAIN, A. "A Comparative Analysis of Distributed Clustering Algorithms: A Survey". In: *2013 International Symposium on Computational and Business Intelligence*, pp. 165–169, New Delhi, India, ago. 2013. IEEE. ISBN: 978-0-7695-5066-4. doi: 10.1109/ISCBI.2013.40. Disponível em: <http://ieeexplore.ieee.org/document/6724345/>.

[14] FUKUNAGA, K., HOSTETLER, L. "The estimation of the gradient of a density function, with applications in pattern recognition", *IEEE Transactions on Information Theory*, v. 21, n. 1, pp. 32–40, jan. 1975. ISSN: 0018-9448. doi: 10.1109/TIT.1975.1055330. Disponível em: <http://ieeexplore.ieee.org/document/1055330/>.

[15] YIZONG CHENG. "Mean shift, mode seeking, and clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 17, n. 8, pp. 790–799, ago. 1995. ISSN: 01628828. doi: 10.1109/34.400568. Disponível em: <http://ieeexplore.ieee.org/document/400568/>.

[16] DERPANIS, K. G. "Mean Shift Clustering". 2005.

[17] THEODORIDIS, S., KOUTROUMBAS, K. *Pattern recognition*. 4. ed ed. Amsterdam, Elsevier Acad. Press, 2009. ISBN: 978-1-59749-272-0. OCLC: 550588366.

[18] ROSSWOG, J., GHOSE, K. "Detecting and Tracking Spatio-temporal Clusters with Adaptive History Filtering". In: *2008 IEEE International Conference on Data Mining Workshops*, pp. 448–457, Pisa, Italy, dez. 2008. IEEE. doi:

10.1109/ICDMW.2008.93. Disponível em: <http://ieeexplore.ieee.org/document/4733968/>.

[19] LI, Y., HAN, J., YANG, J. "Clustering moving objects". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pp. 617–622, Seattle, WA, USA, ago. 2004. Association for Computing Machinery. ISBN: 978-1-58113-888-7. doi: 10.1145/1014052.1014129. Disponível em: <https://doi.org/10.1145/1014052.1014129>.

[20] ZHANG, T., RAMAKRISHNAN, R., LIVNY, M. "BIRCH: An Efficient Data Clustering Method for Very Large Databases". 1996.

[21] BROOKNER, E. *Tracking and Kalman filtering made easy*. New York, Wiley, 1998. ISBN: 978-0-471-18407-2.

[22] JOSHI, M. R., PATIL, Y. S. "Analysis of change in coordinate system on clustering". In: *2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, pp. 1–7, Bangalore, India, mar. 2016. IEEE. ISBN: 978-1-5090-1936-6. doi: 10.1109/ICCTAC.2016.7567339. Disponível em: <http://ieeexplore.ieee.org/document/7567339/>.

[23] WILCOXON, F. "Individual Comparisons by Ranking Methods", *Biometrics Bulletin*, v. 1, n. 6, pp. 80, dez. 1945. ISSN: 00994987. doi: 10.2307/3001968. Disponível em: <https://www.jstor.org/stable/10.2307/3001968?origin=crossref>.

[24] DEMSAR, J. "Statistical Comparisons of Classifiers over Multiple Data Sets", *Journal of Machine Learning Research*, v. 7, pp. 1–30, 01 2006.

[25] MORE, P., HALL, L. "Scalable clustering: a distributed approach". In: *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, v. 1, pp. 143–148, Budapest, Hungary, 2004. IEEE. ISBN: 978-0-7803-8353-1. doi: 10.1109/FUZZY.2004.1375705. Disponível em: <http://ieeexplore.ieee.org/document/1375705/>.

[26] GHANEM, S., KECHADI, T., TARI, A. K. "New approach for distributed clustering". In: *Proceedings 2011 IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services*, pp. 60–65, Fuzhou, China, jun. 2011. IEEE. ISBN: 978-1-4244-8352-5. doi: 10.1109/ICSDM.2011.5969005. Disponível em: <http://ieeexplore.ieee.org/document/5969005/>.

[27] DUNN, J. C. "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics*, v. 3, n. 3, pp. 32–57, jan. 1973. ISSN: 0022-0280. doi: 10.1080/01969727308546046. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/01969727308546046>.

[28] MERUGU, S., GHOSH, J. "A privacy-sensitive approach to distributed clustering", *Pattern Recognition Letters*, v. 26, n. 4, pp. 399–410, mar. 2005. ISSN: 01678655. doi: 10.1016/j.patrec.2004.08.003. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0167865504001801>.

[29] PEARSON, K. "LIII. *On lines and planes of closest fit to systems of points in space*", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, v. 2, n. 11, pp. 559–572, nov. 1901. ISSN: 1941-5982, 1941-5990. doi: 10.1080/14786440109462720. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/14786440109462720>.

[30] WAGNER, D. H., MYLANDER, W. C., SANDERS, T. J. "Target Motion Analysis". ago. 2019. Disponível em: <https://en.wikipedia.org/w/index.php?title=Target_Motion_Analysis&oldid=912663348>. Page Version ID: 912663348.