MARINHA DO BRASIL
DIRETORIA DE ENSINO DA MARINHA
CENTRO DE INSTRUÇÃO ALMIRANTE WANDENKOLK

CURSO DE APERFEIÇOAMENTO AVANÇADO EM
SEGURANÇA DA INFORMAÇÃO E COMUNICAÇÕES

1º Ten WALMOR CRISTINO LEITE JUNIOR

ACIONAMENTO DE ATAQUES CIBER-ELETRÔNICOS EM SISTEMAS RADAR NO AMBIENTE MARÍTIMO

Rio de Janeiro
2020

1º Ten WALMOR CRISTINO LEITE JUNIOR

ACIONAMENTO DE ATAQUES CIBER-ELETRÔNICOS EM SISTEMAS RADAR NO AMBIENTE MARÍTIMO

Monografia apresentada ao Centro de Instrução Almirante Wandenkolk como requisito parcial à conclusão do Curso de Aperfeiçoamento Avançado em Segurança da Informação e Comunicações.

Orientadores:
Alan Oliveira de Sá, D. Sc.
Carlos Vinício Rodríguez Ron, D. Sc.

CIAW
Rio de Janeiro
2020

Leite Junior, Walmor Cristino.
Acionamento de ataques ciber-eletrônicos em sistemas radar no ambiente marítimo / Walmor Cristino Leite Junior. — Rio de Janeiro, 2020.
71f.: il.

Orientador: Prof. Dr. Alan Oliveira de Sá;
Prof. Dr. Carlos Vinício Rodríguez Ron.

Monografia (Curso de Aperfeiçoamento Avançado de Segurança da Informação e Comunicações) — Centro de Instrução Almirante Wandenkolk, Rio de Janeiro, 2020.

1. Guerra cibernética. 2. Guerra ciber-eletrônica. 3. Ambiente marítimo. I. Centro de Instrução Almirante Wandenkolk. II. Título.

1º Ten WALMOR CRISTINO LEITE JUNIOR

ACIONAMENTO DE ATAQUES CIBER-ELETRÔNICOS EM SISTEMAS RADAR NO AMBIENTE MARÍTIMO

Monografia apresentada ao Centro de Instrução Almirante Wandenkolk como requisito parcial à conclusão do Curso de Aperfeiçoamento Avançado em Segurança da Informação e Comunicações.

Aprovada em ___ de _____ de 2020

Banca Examinadora:

_____

CMG (RM1-EN) Gian Karlo Huback Macedo de Almeida – CIAW

_____

Alan Oliveira de Sá, D. Sc. – CIAW

_____

Carlos Vinício Rodríguez Ron, D. Sc. – PUC Rio

CIAW
Rio de Janeiro
2020

Dedico este trabalho à Fragata Liberal, meu primeiro navio, e a todos aqueles que tiveram a honra de bradar “Nosso Barco, Nossa Alma”.

AGRADECIMENTOS

Agradeço aos oficiais e praças com os quais tive o privilégio de conviver a bordo da Fragata Liberal. Todos contribuíram sobremaneira para o meu aprimoramento profissional e pessoal. Seus ensinamentos e exemplos fazem com que eu busque diariamente a mesma excelência que pude observar.

Aos meus orientadores, Prof. Dr. Alan Oliveira de Sá e Prof. Dr. Carlos Vinício Rodríguez Ron, pelos conhecimentos transmitidos e pelo ambiente cordial que sempre fizeram questão de manter.

Ao coordenador do Aperfeiçoamento Avançado em Segurança da Informação e Comunicações, CMG (EN-RM1) Gian Karlo Huback Macedo de Almeida, pela dedicação que demonstrou ao longo de todo o curso. Saiba que seu trabalho nunca passou desapercebido.

Por fim, agradeço a minha companheira, Ingrid Araújo, por suportar ao meu lado todas as provações características da profissão. Todas as conquistas que alcancei até o momento são fruto de nossa parceria.

“Somebody just used a new weapon, and this weapon will not be put back in the box.”

(General Michael Hayden)

ACIONAMENTO DE ATAQUES CIBER-ELETRÔNICOS EM SISTEMAS RADAR NO AMBIENTE MARÍTIMO

RESUMO

A guerra cibernética, que busca explorar e manipular informações digitais, torna-se cada vez mais relevante no contexto internacional. Grandes ataques cibernéticos foram documentados e os estudos gerados para o seu entendimento possibilitaram o desenvolvimento de outros ainda mais poderosos e com grande potencial para alcançar objetivos estratégicos. Dentre esses ataques encontram-se os malware, códigos maliciosos que objetivam interferir no funcionamento de sistemas. O Stuxnet, exemplo de malware, supostamente produzido por Estados, foi capaz de atrasar em alguns anos o programa nuclear de um país. A operação militar Orchard, ataque da força aérea israelense em território sírio, demonstrou a possibilidade de emprego da guerra eletrônica, que compreende ações que utilizam o espectro eletromagnético, em coordenação com a guerra cibernética. Essa integração apresenta-se como uma nova tendência na guerra moderna, a guerra ciber-eletrônica, com novas possibilidades e ameaças. Nesse contexto, é importante que a Marinha do Brasil tenha conhecimento de como essa nova dimensão da guerra pode afetar as operações militares de caráter naval. Assim, foram realizadas simulações em ambientes computacionais, voltados para o meio naval, que emulam funcionalidades de um sistema radar genérico de navegação marítima. Utilizando as linguagens de programação Python e o *software* MATLAB, testam-se hipóteses de emprego dessa nova dimensão no ambiente marítimo. Então, são utilizadas técnicas de processamento de sinais, e de imagens, para demonstrar a importância estratégica da dimensão ciber-eletrônica.

Palavras-chave: Guerra cibernética. Guerra ciber-eletrônica. Ambiente marítimo.

LISTA DE FIGURAS

LISTAS DE SIGLAS E ABREVIATURAS

C-Ap-A	Curso de Aperfeiçoamento Avançado
CIAW	Centro de Instrução Almirante Wandenkolk
DLL	*Dynamic Link Library*
GC	Guerra Cibernética
GCE	Guerra Ciber-Eletrônica
GCR	Guerra Centrada em Redes
GPS	*Global Positioning System*
MAE	Medida de Ataque Eletrônico
MB	Marinha do Brasil
MATLAB	*software* de computação numérica da empresa MathWorks
PCC	Coeficiente de Correlação de Pearson
PPI	*Plan Position Indicator*
RADAR	*Radio Detection and Ranging*
SAD	*Sum of Absolute Differences*
SIC	Segurança da Informação e Comunicações
SO	Sistema Operacional
SSD	*Sum of Squared Differences*

SUMÁRIO

1 INTRODUÇÃO

Vive-se em uma era de crescente dependência de sistemas computadorizados (MENDONÇA, 2014). A automação industrial e diversos sistemas de controle e sensoriamento tornam-se cada vez mais dependentes de complexos módulos eletrônicos, conectados em redes de computadores, que gerenciam grande parte dos processos de forma autônoma. Porém, as vulnerabilidades desses sistemas têm sido exploradas de diversas maneiras. Parcharidis (2018) identifica e realiza simulações de ataques cibernéticos maliciosos que foram desenvolvidos para prejudicar o funcionamento e, em casos recentes, afetar o mundo físico através da manipulação de atuadores controlados eletronicamente. Essas simulações fornecem informações técnicas fundamentais para o entendimento da dinâmica dos ataques.

Esses atos hostis demonstraram grande capacidade de afetar instalações sensíveis, transformando-se em meios eficazes para alcançar objetivos estratégicos. Segundo Sanger (2018), uma usina nuclear iraniana teve seu funcionamento prejudicado em virtude de um ataque cibernético baseado no vírus Stuxnet e o serviço de distribuição de energia elétrica ucraniano sofreu um mau funcionamento provocado por um vírus denominado KILLDISK, ambos os ataques supostamente conduzidos por Estados.

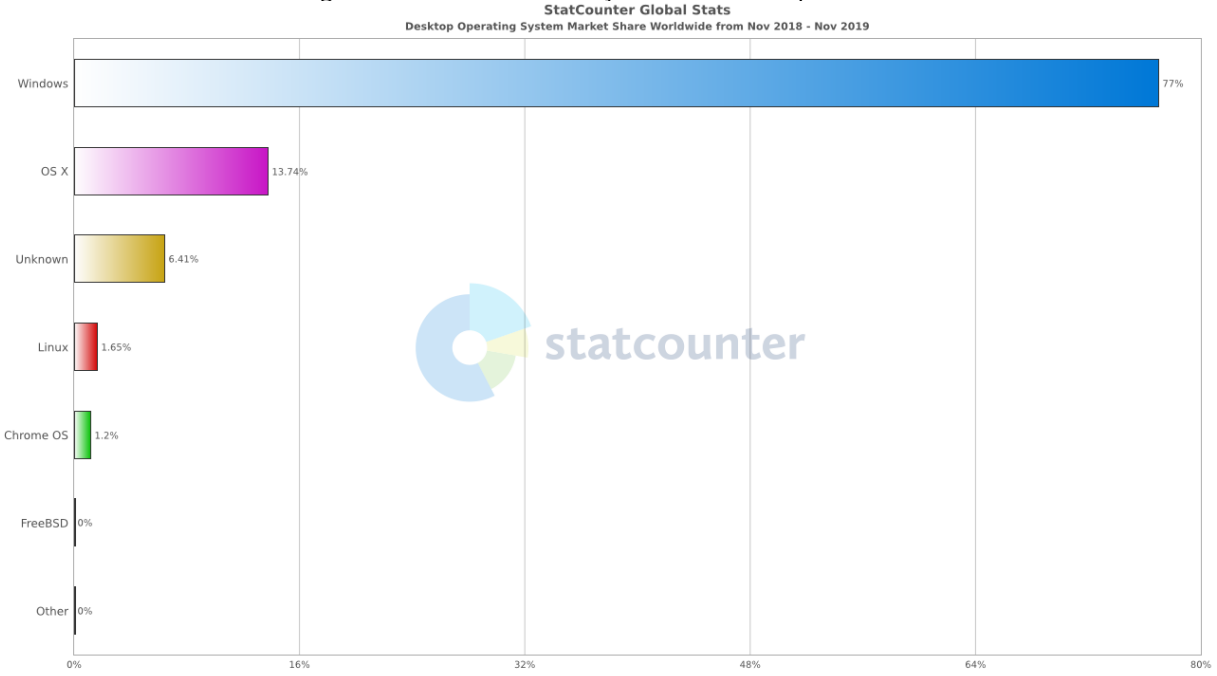
A conversão de sinais analógicos para digitais possibilitou a ampliação da capacidade dos sensores que utilizam o espectro eletromagnético. Esse processo permite o transporte de dados analógicos para linguagem de máquina e promove a integração com sistemas computacionais (FALLEIRO, 2015). Assim, o processamento de dados digitais torna-se cada vez mais presente em sistemas eletrônicos de sensoriamento. A exposição desses sensores ao espectro eletromagnético os torna vulneráveis a Guerra Eletrônica (GE). Com a integração a sistemas cibernéticos, esses ataques passam a alcançar não apenas os componentes eletrônicos, mas também componentes cibernéticos (SÁ, MACHADO e ALMEIDA, 2019).

Qualquer sistema baseado na integração entre sensores eletromagnéticos e sistemas computacionais está vulnerável a ataques que se encontram na interseção entre a eletrônica e a cibernética, também conhecidos como ataques ciber-eletrônicos. O meio militar emprega extensivamente tecnologias desse tipo e ao analisar, por exemplo, uma plataforma naval moderna percebe-se que diversos sistemas estão vulneráveis a esses ataques (SÁ, MACHADO e ALMEIDA, 2019). Esse trabalho se propõe a testar possibilidades de emprego de ataques ciber-eletrônicos, em especial contra sistemas radar da navegação através de plataformas de teste computacionais geradas em linguagens de programação em alto nível.

1.1 Apresentação do Problema

Sistemas cibernéticos estão cada vez mais presentes em meios navais, e essa tendência também se aplica aos meios da Marinha do Brasil (MB) (SÁ, MACHADO e ALMEIDA, 2019). A MB adota o conceito de Guerra Centrada em Redes (GCR), que pode ser entendido como a utilização integrada, através de redes de computadores, das diversas tecnologias que compõem uma unidade de modo a converter as capacidades dos sistemas em poder de combate (BRASIL, 2007). Também deve-se comentar que a Guerra Cibernética (GC) é definida como o conjunto de ações destinadas a explorar e manipular informações digitais, reconhecendo que para alcançar tal objetivo utilizam-se sistemas de informação e redes de computadores (Brasil, 2007).

Figura 1.1 - Índice de utilização de Sistemas Operacionais



Fonte: Statcounter, 2020

Um sistema computacional está baseado na integração entre os componentes físicos, *hardware*, componentes lógicos, *software* e usuário. Para que essa dinâmica se dê de maneira satisfatória é necessário que o *software* tenha uma boa interface com o *hardware*. Isso não é algo simples, um mesmo *software* deve ser capaz de interagir com diversos tipos de *hardware* diferentes. Por isso criou-se um elemento intermediário, os Sistemas Operacionais (SO). Esse elemento funciona como um tradutor entre *hardware* e *software*, facilitando a integração entre eles (TANENBAUM, 2009).

Machado e Maia (2013a) definem SO como um conjunto de rotinas e procedimentos para realizar a gestão dos recursos computacionais oferecidos pelo *hardware* e elementos de interface com o usuário. Statcounter (2020) disponibiliza uma série de estatísticas envolvendo sistemas de informação. Entre elas verifica-se que atualmente o SO *Windows* segue como o mais empregado mundialmente com uma taxa de utilização de 77% em desktops, de acordo com a Figura 1.1. Essa tendência faz com que muitos sistemas navais, militarizados ou não, sejam desenvolvidos para operar nesse SO.

Falliere, Murchu e Chien (2011), técnicos da empresa de segurança digital *Symantec*, produziram um dossiê sobre o vírus de computador Stuxnet. Esse documento descreve o funcionamento dessa arma digital e atesta que ela explora uma série de vulnerabilidades do SO *Windows*. Seu refinamento levanta questões sobre o seu desenvolvimento, não seria possível desenvolver uma arma tão elaborada sem um amplo suporte técnico associado a um material humano com elevada capacitação. Segundo Sanger (2018), ele teria sido criado por instituições Estatais para alcançar objetivos estratégicos a nível internacional.

Nesse contexto percebe-se que a ameaça cibernética a sistemas atualmente utilizados na MB é real. Atores estatais, ou não, podem enveredar esforços para explorar vulnerabilidades para obter vantagens estratégicas em uma determinada situação tática, podendo colocar um meio ou mesmo toda uma força naval em situação de risco. O escopo desse trabalho é a realização e análise de testes que sirvam como prova da possibilidade de emprego de ações ofensivas que utilizam conceitos de Guerra Ciber-Eletrônica (GCE), interseção entre GE e GC, no ambiente marítimo.

1.2 Justificativa e Relevância

A Estratégia Nacional de Defesa identifica a segurança cibernética como um dos pontos fundamentais para a garantia dos interesses nacionais brasileiros (BRASIL, 2012). Desta forma, torna-se válido produzir uma análise sobre os possíveis impactos de ações de guerra cibernética em um meio naval.

A pesquisa proporcionará um melhor entendimento das capacidades, vulnerabilidades e ameaças do ambiente cibernético e como seus conceitos podem ser aplicados às operações navais. Servirá como material para assessoria aos tomadores de decisão em questões afetas ao assunto e contribuirá para o aprimoramento da capacitação cibernética no campo militar, ponto destacado na Estratégia Nacional de Defesa (BRASIL, 2012).

Nesse cenário, é conveniente produzir conhecimento prático sobre o assunto. Os recursos computacionais disponíveis permitem a criação de ambientes virtuais para testes. Dessa forma, a utilização desses ambientes em simulações da ataques ciber-eletrônicos possibilita o levantamento de dados relevantes para posterior aprofundamento em trabalhos futuros.

1.3 Objetivos

A presente pesquisa objetiva fomentar o pensamento estratégico no que tange a questões envolvendo a aplicação da GC e GE no contexto das operações navais. Também busca propor um campo de estudo a ser explorado em próximas edições do Curso de Aperfeiçoamento Avançado (C-Ap-A) ministrado pelo Centro de Instrução Almirante Wandenkolk (CIAW).

1.3.1 Objetivo Geral

Essa pesquisa tem como objetivo realizar uma análise a respeito da aplicabilidade de conceitos e técnicas de GCE no ambiente marítimo. Com esse objetivo, será aplicada uma pesquisa exploratória que, para Gil (2002), tem como finalidade propor questões mais precisas ou apresentar novas possibilidades para trabalhos futuros. Para esse fim, foram realizados testes e simulações em plataformas digitais para avaliar hipóteses de emprego da GCE contra um meio naval.

1.3.2 Objetivos Específicos

Criar plataformas digitais, utilizando linguagens de programação de alto nível, para testar hipóteses de emprego de ataques ciber-eletrônicos no ambiente marítimo. O alvo a ser estudado será um sistema radar genérico utilizado para navegação marítima. Serão empregadas técnicas de processamento de sinais e imagens.

1.4 Etapas do Trabalho

O presente trabalho encontra-se organizado em seis capítulos, descritos a seguir:

O capítulo 1, INTRODUÇÃO, apresenta o problema abordado no trabalho e promove uma breve contextualização.

O capítulo 2, REFERENCIAL TEÓRICO, apresenta conceitos e definições necessários para o bom entendimento deste trabalho. São comentados o atual contexto da GC e as novas tendências nessa dimensão da guerra. Posteriormente, são apresentados conceitos básicos de processamento de imagens, em especial da técnica de *template matching*. Aborda-se a teoria de um sistema radar genérico, do processamento de sinais, e sua integração com sistemas computacionais.

O capítulo 3, METODOLOGIA, apresenta a classificação da pesquisa realizada, o método para obtenção de dados e suas limitações.

O capítulo 4, DESCRIÇÃO DO AMBIENTE DE SIMULAÇÃO, descreve as características das plataformas digitais utilizadas para a realização dos experimentos, que objetivam testar hipóteses de emprego da GCE contra sistemas radar empregados em meios navais.

O capítulo 5, ANÁLISE DOS RESULTADOS, aborda os resultados obtidos nos experimentos, que testam técnicas de processamento de imagens e processamento de sinais.

Finalmente, o capítulo 6, CONCLUSÃO, completa esta monografia, apresentando as conclusões e considerações finais referentes ao trabalho em questão, assim como sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Para o melhor entendimento dos resultados, este capítulo apresentará alguns conceitos e definições utilizados ao longo do trabalho. Também serão apresentados trabalhos relacionados e fundamentos técnicos atinentes à presente pesquisa de forma a familiarizar o leitor com o campo de estudo explorado.

2.1 Conceitos e Definições

Nessa seção serão apresentados fundamentos de tecnologia da informação e de vertentes da guerra, necessários para a melhor compreensão do presente trabalho:

a) Assinatura Digital

Para Stallings (2015), trata-se de um conjunto de operações computacionais que possibilitam que um usuário comprove a autenticidade e integridade de um conjunto de dados. Esses procedimentos podem ser aplicados a arquivos que contém apenas informações para leitura ou arquivos executáveis;

b) Sistema Operacional

Tanenbaum (2009) define SO como a interface entre *hardware* e *software.* Essa interface estabelece procedimentos que permitem gerenciar os recursos físicos e lógicos do sistema como um todo;

c) Dynamic Link Library

Machado e Maia (2013b) descrevem esse elemento do sistema *Windows* como uma biblioteca que funciona como interface entre uma determinada aplicação e o SO. As *Dynamic Link Library* (DLL) são formadas por conjuntos de procedimentos genéricos, organizados previamente, que podem ser utilizados por aplicações fazendo com que os códigos de desenvolvimento dessas aplicações possam ser simplificados;

d) Python

Kumar e Panda (2019) descrevem Python como uma linguagem de programação de alto nível, criada em 1991, com alta versatilidade e simplicidade. Ao longo dos anos essa linguagem foi ganhando cada vez mais popularidade sendo hoje uma das linguagens mais utilizadas no mundo. Possui a capacidade de trabalhar utilizando bibliotecas e pacotes que simplificam a programação de forma semelhante as DLL's do *Windows;*

e) MATLAB

Trata-se de um *software* para computação numérica, dotado de uma linguagem de programação de alto nível própria (MATHWORKS, 2019). Possui soluções, chamadas *toolbox*, para processamento de dados voltadas para diversas aplicações, como o processamento de sinais e simulações computacionais de dispositivos e sistemas eletrônicos. É possível simular o funcionamento de um sistema *Radio Detection and Ranging* (RADAR) como será apresentado posteriormente;

f) Malware

Compreende *software*, ou parte dele, que tenha como objetivo comprometer qualquer requisito básico de SIC (REGAN, 2019). Busca explorar vulnerabilidades dos sistemas de informação para alcançar determinados objetivos;

g) Device Driver

Também chamados apenas de driver, conforme Machado e Maia (2013a) é um código que faz a interface do SO com dispositivos de *hardware*. É importante perceber que devido a sua importância esse elemento possui alta importância nos SO e possuem privilégios para executar comandos sem o controle direto do SO;

h) Spoofing

Técnica utilizada para enganar um receptor. Gera-se um sinal semelhante ao que o alvo espera receber, mas com informações falsas com objetivos maliciosos (MENDONÇA, 2014);

i) Guerra Eletrônica

A GE é a utilização militar da energia eletromagnética para determinar, explorar, reduzir e prevenir o uso hostil do espectro eletromagnético pelo inimigo garantindo, também, o livre uso desta energia pelas forças amigas (SCHLEHER, 1994);

j) Guerra Cibernética

É definida como ações destinadas a explorar e manipular informações digitais, reconhecendo que para alcançar tal objetivo utilizam-se sistemas de informação e redes de computadores (BRASIL, 2007);

k) Guerra Cinética

Define-se com a dimensão da guerra que está situada no domínio do mundo físico e envolve a aplicação de forças reais, não virtuais (SÁ, MACHADO e ALMEIDA, 2019);

l) Medida de Ataque Eletrônico

Compreende ações que objetivam degradar a capacidade inimiga de utilização do espectro eletromagnético. Medida de Ataque Eletrônico (MAE), Também se refere a extração de informações estratégicas a partir das emissões eletromagnéticas realizadas pelo oponente, como a identificação do equipamento emissor ou sua localização (BRASIL, 2003).

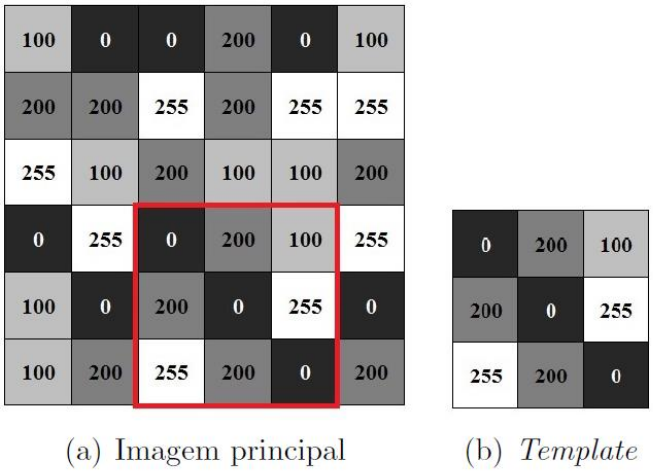
2.1.1 Segurança da Informação e Comunicações

Compreende o conjunto de ações que objetivam garantir a disponibilidade, integridade e confidencialidade de dados e informações (BRASIL, 2019). Para a MB, Segurança da Informação e Comunicações (SIC) envolve as dimensões física, lógica, de tráfego e criptológica. Não se trata apenas de um conjunto de procedimentos, envolve recursos computacionais, *hardware, software,* e instruções normativas.

> Requisitos Básicos de SIC
> a) Disponibilidade - capacidade da informação digital estar disponível para alguém autorizado a acessá-la no momento próprio.
> b) Integridade - capacidade da informação digital somente ser modificada por alguém autorizado;
> c) Confidencialidade - capacidade da informação digital somente ser acessada por alguém autorizado;
> d) Autenticidade - capacidade da origem da informação digital ser aquela identificada. (BRASIL, 2019, p. 9-1).

2.1.2 *Template Matching*

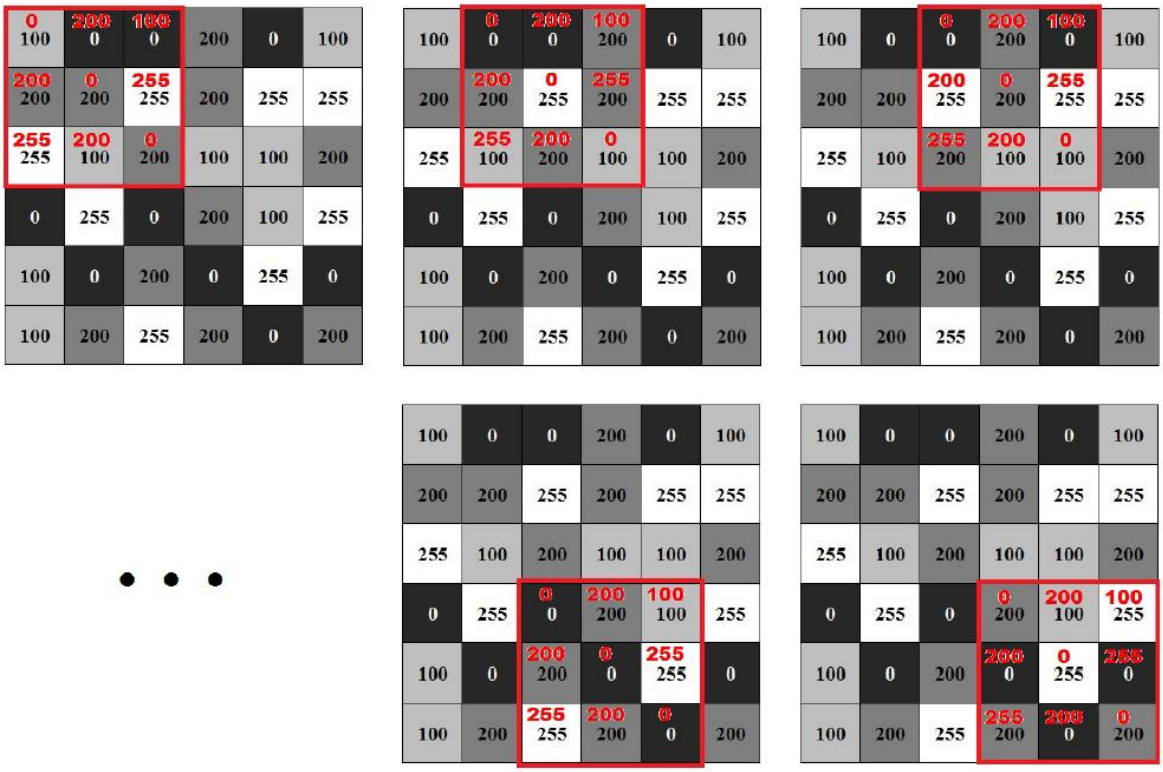
Figura 2.1 - Matrizes que representam imagens em preto e branco



Fonte: Tavares, 2016, p.26

Trata-se um conjunto de procedimentos utilizados em processamento de imagens para encontrar similaridades entre dois elementos (TAVARES, 2016). De uma maneira geral é estabelecido um *Template*, ou elemento que se deseja buscar em uma Imagem Principal. A imagem que será analisada é então dividida em quadrantes relativos aos seus pixels, conforme a Figura 2.2, e então a busca é iniciada. É estabelecido um algoritmo matemático que gera um índice de similaridade para cada interação do processo de busca e, caso esse índice seja superior a um mínimo definido pelo usuário, possibilita a detecção do *Template* dentro da Imagem Principal. Essa operação demanda um alto custo computacional proporcional aos tamanhos das imagens envolvidas. Por outro lado, fornece alto grau de eficácia em rastreamento (TAVARES, 2016).

Figura 2.2 - Ilustração do processo de *template matching*



Fonte: Tavares, 2016, p.27

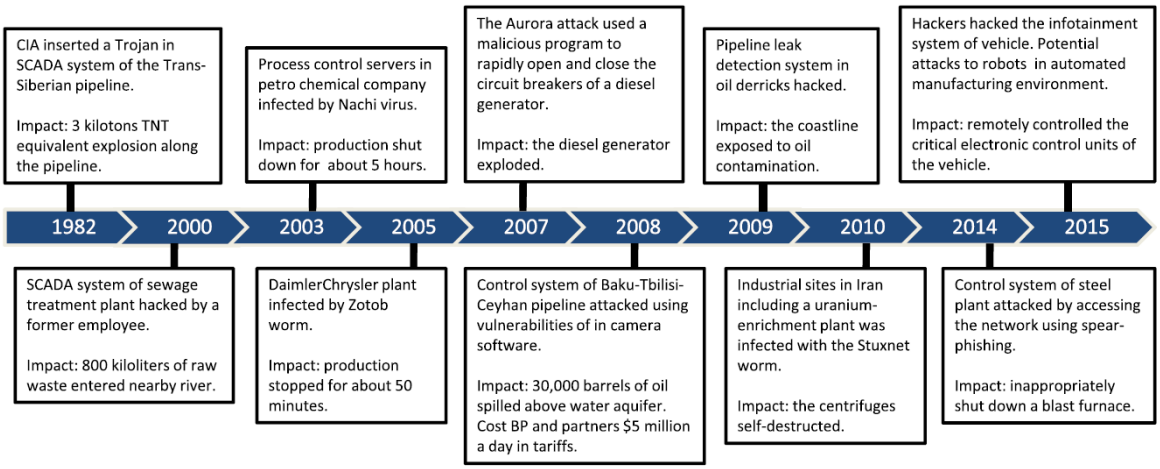
2.2 Trabalhos Relacionados

Nessa seção serão apresentados trabalhos que estão diretamente relacionados com a proposta desta pesquisa, fornecendo informações que possibilitam o melhor entendimento do problema apresentado.

2.2.1 Guerra Cibernética

A utilização da dimensão cibernética como meio para a realização de ataques em busca de objetivos estratégicos não é uma novidade. Conforme a Figura 2.3, ataques de grande vulto tem ocorrido desde a década de 80. Com o passar do tempo a complexidade dessas ações foi incrementada e atualmente possui alto nível de sofisticação. Vale ressaltar que cada um desses eventos está relacionado a um impacto relevante e com tremendo potencial estratégico. Assim, é importante realizar uma breve revisão de alguns pontos nessa linha do tempo (MCLAUGHLIN *et al.*, 2016).

Figura 2.3 - Linha do tempo dos principais ataques cibernéticos e seus impactos



Fonte: McLaughlin *et al*., 2016

O *malware* Aurora, testado em 2007 pelo Laboratório Nacional de Idaho, é um exemplo de como um código malicioso pode causar impactos na dimensão física. O alvo desse código foi uma planta elétrica baseada em um gerador a diesel controlado eletronicamente. Após obter acesso a rede de controle, o atacante implantou o malware. Os atuadores eletrônicos passaram a receber comandos maliciosos fazendo com que o sistema entrasse em colapso, chegando a causar a explosão do gerador (MCLAUGHLIN *et al*., 2016).

Em 2008 vulnerabilidades de *softwares* de controle de câmeras sem fio foram exploradas para que atacantes conseguissem acesso a rede de controle de um oleoduto turco. Após a obtenção do acesso necessário, comandos maliciosos foram emitidos para os elementos responsáveis pelo controle de pressão dos dutos, fazendo com que o funcionamento ocorresse fora dos limites de segurança. Como consequência houve uma explosão responsável pelo derramamento de trinta mil barris de petróleo (MCLAUGHLIN *et al*., 2016).

O advento do *malware* Stuxnet, em 2010, merece especial destaque nesse contexto. Segundo Zetter (2015), o malware alcança um nível de complexidade surpreendente. Seu refinamento levanta questões sobre o seu desenvolvimento, não seria possível desenvolver uma arma tão elaborada sem um amplo suporte técnico associado a um material humano com elevada capacitação. Sanger (2018) afirma que ele teria sido criado por instituições Estatais para alcançar objetivos estratégicos no ambiente internacional. A vítima de maior vulto do Stuxnet foi uma usina nuclear iraniana que teve seu funcionamento prejudicado, atrasando o programa nuclear desse país em alguns anos.

Falliere, Murchu e Chien (2011) identificam um arquivo DLL entre os elementos centrais desse *malware*, que contém uma biblioteca de comandos própria, conforme a Figura

2.4. De uma forma geral, esse *malware* insere uma DLL maliciosa no SO *Windows* e executa de maneira discreta os comandos necessários para o seu funcionamento. Para passar desapercebido, o Stuxnet utiliza comandos com nomes que não existem no sistema original, o que costumam causar falha. A DLL maliciosa monitora esses nomes e os executa mesmo que o sistema original não os reconheça.

Figura 2.4 - Tabela de funções da DLL maliciosa (Stuxnet)

DLL Exports

Export #	Function
1	Infect connected removable drives, starts RPC server
2	Hooks APIs for Step 7 project file infections
4	Calls the removal routine (export 18)
5	Verifies if the threat is installed correctly
6	Verifies version information
7	Calls Export 6
9	Updates itself from infected Step 7 projects
10	Updates itself from infected Step 7 projects
14	Step 7 project file infection routine
15	Initial entry point
16	Main installation
17	Replaces Step 7 DLL
18	Uninstalls Stuxnet
19	Infects removable drives
22	Network propagation routines
24	Check Internet connection
27	RPC Server
28	Command and control routine
29	Command and control routine
31	Updates itself from infected Step 7 projects
32	Same as 1

Fronte: Falliere, Murchu e Chien, 2011, p.12

Além disso, o *malware* tem a capacidade de se propagar através de dispositivos móveis (FALLIERE, MURCHU e CHIEN, 2011). Assim, torna-se capaz de alcançar redes isoladas da internet. Essa capacidade é obtida através da utilização de um driver adulterado. Esse driver, apesar de modificado por um agente alheio ao fabricante, possui assinatura digital legítima do desenvolvedor original. Isso faz com que o sistema reconheça o driver como autêntico e integro, fazendo com que o sistema o execute sem qualquer alarme. O driver, de comunicação com dispositivos móveis, acaba permitindo que o malware seja copiado para qualquer dispositivo que seja inserido na máquina infectada. Esse dispositivo, por sua vez, ao ser inserido em uma máquina livre do *malware* transporta a DLL maliciosa para a nova máquina.

Para o presente estudo é necessário compreender que o Stuxnet demonstrou a possibilidade de exploração das vulnerabilidades do sistema SO *Windows* para executar um

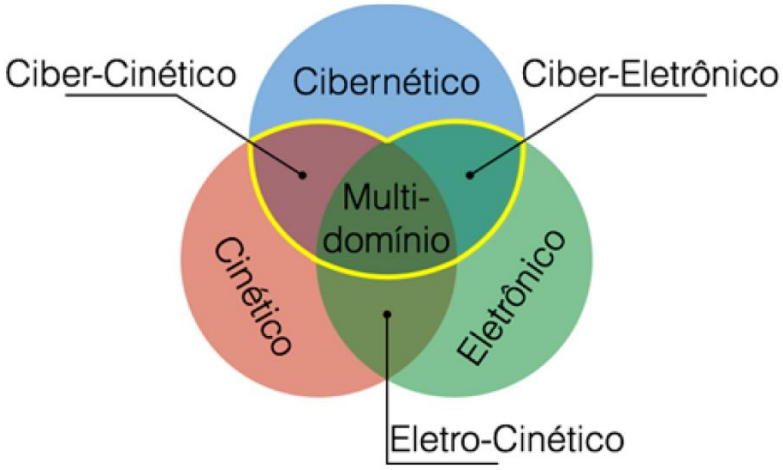
malware de maneira furtiva. Também é capaz de utilizar um driver malicioso, com assinatura digital forjada, fazendo com que funções sensíveis ao SO sejam executadas de forma maliciosa. Por fim, é importante entender que atores estatais podem estar envolvidos do desenvolvimento e operacionalização dessa arma cibernética.

Percebe-se uma notável evolução na complexidade dos ataques cibernéticos com o passar do tempo. O Stuxnet apresentou um novo nível de sofisticação, onde toda a ação é automatizada. Ele também apresenta uma modularidade em relação a suas funções, pode-se analisar com clareza seus blocos de funcionamento (FALLIERE, MURCHU e CHIEN, 2011). A dinâmica de infecção de dispositivos móveis, obtenção de privilégios de execução no sistema e execução furtiva podem ser aplicadas a diversos outros objetivos. Hoje diversos estudos estão disponíveis para todos aqueles que desejarem estudá-lo, inclusive para indivíduos com intenções maliciosas. Os mesmos conceitos podem ser explorados para atingir outras plataformas, inclusive sistemas utilizados em meios navais que utilizam sistemas com as mesmas vulnerabilidades.

2.2.2 O Encontro das guerras cibernética, eletrônica e cinética

Sá, Machado e Almeida (2019), dissertam sobre o fenômeno da integração entre as dimensões cibernética, eletrônica e cinética em operações de guerra. A aplicação desses conceitos separadamente tem sido usual em ações militares modernas, entretanto percebe-se que existe uma tendência de que esses ambientes sejam integrados de forma que ações em um deles cause efeitos nos demais.

Figura 2.5 - Domínios de influência/impacto em ações militares



Fonte: Sá, Machado e Almeida, 2019

A Figura 2.5 apresenta os conceitos de campos ciber-cinéticos, ciber-eletrônico, eletro-cinético de multidomínio. Os autores descrevem a classificação ciber-cinética como ações que objetivam causar efeitos no mundo físico através de ataques à sistemas computacionais. O campo ciber-eletrônico é aquele onde ações de guerra eletrônica buscam atingir sistemas computacionais, corresponderia a uma nova etapa evolutiva de ataques eletrônicos. “Nesse caso, o espectro eletromagnético é utilizado pelo atacante para enviar um fluxo de dados ao processador do sistema alvo de forma a manipular seu processo computacional, comprometendo assim o seu funcionamento.” (SÁ, MACHADO e ALMEIDA, 2019). Já o campo eletro-cinético estabelece uma relação entre o mundo físico e o espectro eletromagnético, de acordo com os autores, um exemplo de aplicação seria a utilização de minas magnéticas. Nessas minas, a detonação, efeito físico, é realizada através da detecção do campo eletromagnético gerado por estruturas metálicas. Como observa-se nessa figura, o campo multidomínio se refere a ações que apresentam características de todos os campos.

Como comentado anteriormente, um sistema radar opera de forma a emitir sinais e receber os sinais refletidos por alvos. Essas informações são então convertidas e, em sistemas modernos, gravadas em uma memória para tratamento e apresentação através de sistemas computacionais. Dessa forma os dados eletromagnéticos são convertidos para linguagem de máquina e lidos por um *software* que dará prosseguimento ao seu processamento (SÁ, MACHADO e ALMEIDA, 2019). Caso um atacante consiga manipular as informações eletromagnéticas é possível inserir informações maliciosas nessa memória causando prejuízo ao *software* utilizado, ação caracterizada como ciber-eletrônica. Por isso os autores identificam sistemas radar, de uso militar ou civil, como alvos potenciais para ataques ciber-eletrônicos.

2.2.3 Guerra Ciber-Eletrônica

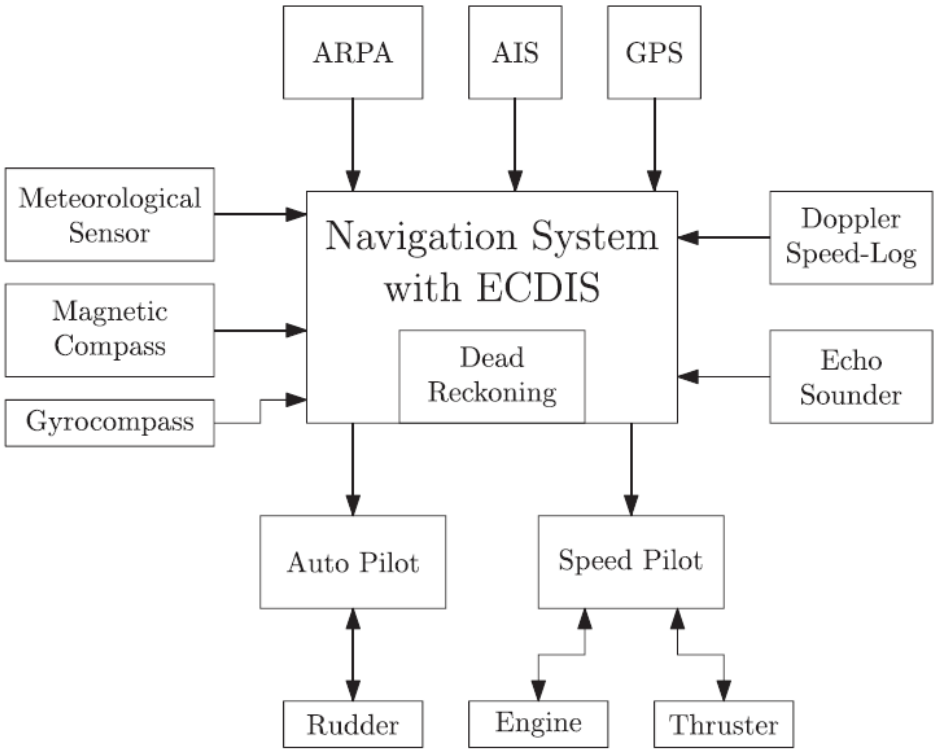
Conforme comentado anteriormente, a conversão analógico-digital possibilita a integração entre sensores eletromagnéticos e sistemas computacionais (FALLEIRO, 2015). Esse fato promove um grande incremento na capacidade dos sistemas que utilizam essa tecnologia, pois as informações obtidas por eles podem ser tratadas e apresentadas de forma otimizada. Porém essa tecnologia abre as portas dos sistemas computacionais envolvidos para ataques conduzidos através do espectro eletromagnético e captados por sensores.

Sá, Machado e Almeida (2019) apresentam a operação Orchard, ação militar conduzida pelo Estado de Israel em 2007 contra alvos estratégicos sírios. Um conjunto de aeronaves israelenses realizaram um bombardeio em território sírio sem que fossem detectados pelos radares aéreos inimigos. Esse fato levantou diversos questionamentos sobre as técnicas que poderiam ser utilizadas para atingir tamanho nível de descrição.

Supostamente houve a utilização de uma MAE em coordenação com um *malware* previamente instalado nos sistemas sírios. Adee (2008) chama esse tipo de mecanismo de *Kill Switch*. O ataque eletrônico teria sido empregado para transmitir um sinal eletromagnético que após recepção, e conversão para linguagem de máquina, foi capaz de ativar o gatilho digital do *malware* alojado no sistema para que a apresentação da PPI fosse adulterada (SÁ, MACHADO e ALMEIDA, 2019). Assim os operadores não foram capazes de detectar as aeronaves antes da execução dos ataques.

Bhatti e Humphreys (2017) avaliam a hipótese de ataque ciber-eletrônicos a sistemas de navegação *Global Positioning System* (GPS). Através da MAE correta é possível privar um alvo do acesso à informação GPS, além disso, com o *spoofing* do sinal é possível inserir informações de posição adulteradas no sistema alvo. Porém, a simples transmissão de uma posição falsa, ou de uma MAE que degrade o sinal, pode chamar a atenção de uma tripulação atenta. Em navios modernos os sistemas de auxílio a navegação estão cada vez mais integrados. Os sistemas GPS, piloto automático, controle de velocidade e outros, se comunicam com um sistema integrado de navegação, *Eletronic Chart Display and Information System* (ECDIS), de acordo com a Figura 2.6. Assim, fica ainda mais complexo realizar um ataque através do sistema GPS sem gerar inconsistências com outro sistema.

Figura 2.6 - Diagrama de blocos de um sistema integrado de navegação



Fonte: Bhatti e Humphreys, 2017

Foram testados algoritmos, que apresentaram desempenho satisfatório, capazes de inserir dados e de ludibriar os demais sistemas integrados pela ECDIS através do *spoofing* do sinal GPS. De acordo com Bhatti e Humphreys (2017) é possível reduzir significativamente as chances de que um ataque desse tipo seja notado pela tripulação, pois todos os dados continuarão aparentemente coerentes.

De acordo com Lockheed Martin (2020), grande empresa do mercado internacional de defesa, o ambiente ciber-eletrônico representa o próximo nível evolutivo da guerra moderna. A condução de operações nesse contexto permitirá alcançar objetivos sem a necessidade de utilização de armamentos tradicionais. Assim, para acompanhar a evolução da arte da guerra é preciso desenvolver conhecimentos relacionados a esse ambiente.

2.3 Fundamentação técnica

Nessa seção serão apresentados os fundamentos técnicos utilizados para a construção dos ataques propostos nos experimentos apresentados nessa pesquisa.

2.3.1 Processamento de Imagens e rastreamento de padrões

Tavares (2016) descreve o reconhecimento de padrões como um dos ramos de maior relevo para a inteligência artificial, esse campo busca fazer com que máquinas sejam capazes de realizar associações próximas a de um ser humano. O reconhecimento de objetos possibilita a identificação de padrões, utilização de biometria para identificação, Reconhecimento facial e diversas outras aplicações como defesa e vigilância. A identificação de objetos em vídeos e imagens tem sido cada vez mais exploradas em sistemas digitais. O autor também identifica e define a segmentação, modelagem do plano de fundo, detecção de pontos e aprendizado supervisionado como métodos relevantes.

De uma forma sucinta, Conforme Tavares (2016), a segmentação realiza a divisão da imagem principal em partes menores utilizando algoritmos que aumentam as chances de que uma destas partes menores compreenda o objeto de interesse. A modelagem de plano de fundo está baseada na utilização de algoritmos que possibilitam identificar, em uma imagem principal, o plano de fundo, assim é possível perceber a silhueta de objetos a partir da sua fronteira com o plano de fundo. Já a detecção de pontos se empenha na detecção de pontos notáveis em objetos, é possível mapear um determinado objeto em uma disposição de pontos para que sejam buscados em uma imagem principal. Por fim, a aprendizagem supervisionada está relacionada a criação de um banco de informações que contenha dados de um determinado objeto de interesse nos mais diversos ângulos e situações, de forma que o sistema de detecção seja capaz de aumentar sua chance de detecção conforme a alimentação de seu banco de informações.

Tendo esses métodos a nossa disposição pode-se aplicá-los em técnicas de rastreamento. É possível classificar as técnicas em três categorias: rastreamento baseado em pontos, rastreamento baseado em *kernel* e rastreamento baseado em silhuetas (TAVARES, 2016). Percebe-se que a primeira técnica busca a identificação do objeto de interesse através de um determinado padrão de pontos, utilizando o método de detecção de pontos. O rastreamento baseado em *kernel* objetiva a identificação do objeto através de regiões notáveis, divide-se a imagem principal em segmentos buscando a identificação do objeto de interesse. O *template matching*, ou casamento por modelo, é um exemplo de como essa técnica pode ser aplicada. Já o rastreamento por silhueta está focado na identificação do objeto alvo pela localização de sua silhueta apenas. Na figura a seguir é possível visualizar uma aplicação de cada técnica.

Figura 2.7 - Exemplos de aplicação das técnicas para localização da câmera

(a) Por pontos (b) Por *kernel* (c) Por silhueta

Fonte: Tavares, 2016, p.23

Conforme comentado anteriormente, o *template matching* utiliza uma imagem modelo que contém o objeto que se deseja localizar e divide a imagem principal em segmentos de tamanhos iguais ao do modelo. Esses segmentos são comparados com o modelo um a um até que o grau de semelhança seja maior que o mínimo estabelecido pelo usuário. É possível observar essa mecânica na Figura 2.1 e na Figura 2.2 Esse grau de semelhança é estabelecido através da comparação de valores de intensidade em cada pixel. Tavares (2016) explana três das técnicas utilizadas para o estabelecimento desse coeficiente de semelhança: *Sum of Absolute Differences* (SAD), *Sum of Squared Differences* (SSD) e correlação cruzada.

O SAD realiza o somatório das diferenças entre as intensidades do pixel do *template* e seu equivalente em um determinado segmento da imagem principal. Nesse caso, quanto mais próximo de zero, maior a semelhança entre as imagens. A equação a seguir ilustra a obtenção do somatório, onde p_i é a intensidade do pixel no *template* e a_i é a intensidade do pixel na imagem principal:

$$SAD = \sum_{i=1}^{N} |(p_i - a_i)| \quad (1)$$

Já no SSD, os valores das diferenças são elevados ao quadrado antes da soma. Apesar disso, a relação de semelhança continua sendo maior conforme o valor do somatório se aproxima de zero.

$$SSD = \sum_{i=1}^{N}(p_i - a_i)^2 \quad (2)$$

Por fim, o cálculo da correlação cruzada é realizado utilizando a multiplicação das intensidades dos pixels. Nesse caso os valores de saída podem ter grandes dimensões, o que dificulta a análise.

$$corr = \sum_{i=1}^{N}(p_i)(a_i) \quad (3)$$

Para resolver o problema da grande amplitude de valores foi implementado um método de normalização que possibilita utilizar a correlação entre pixels dentro de um intervalo definido de [-1;1]. De acordo com Miranda (apud TAVARES, 2014) a expressão também é conhecida como Coeficiente de Correlação de Pearson (PCC).

$$PCC = \frac{\sum_{i=1}^{N}(p_i - \overline{p})(a_i - \overline{a})}{\sqrt{\sum_{i=1}^{N}(p_i - \overline{p})^2}\sqrt{\sum_{i=1}^{N}(a_i - \overline{a})^2}} \quad (4)$$

Onde $\bar{p}$ e $\bar{a}$ são as médias das intensidades dos pixels no *template* e no segmento da imagem principal, respectivamente.

Também é importante comentar que há uma série de fatores aleatórios que podem influenciar a detecção de objetos (TAVARES, 2016). É possível que a existência de ruído, por exemplo, altere as propriedades da imagem de forma a gerar falsos positivos ou falsos negativos. Da mesma forma, variações de iluminação ou coloração também podem afetar a detecção, assim é preciso assumir que independentemente do método utilizado ainda haverá chance de falha na detecção.

Em relação ao escopo dessa pesquisa, pode-se perceber que a detecção de imagens apresenta potencial para integração com códigos maliciosos. De acordo com Tavares (2016), esse tipo de tecnologia pode ser empregado para guiamento de misseis em direção a imagem

do alvo. Esse mesmo conceito poderia ser aplicado a GC, fazendo com que um código malicioso seja ativado ao detectar um determinado padrão.

2.3.2 Sistema Radar

A publicação da MB GE-204 versa sobre processamento de sinais e sistemas radar (Brasil, 2016). Conforme essa publicação, os sistemas radar apresentam informações em um monitor apropriado para que um operador possa interpretar os dados fornecidos pelo sistema. Também de acordo com Brasil (2016), os alvos podem ser detectados através da propriedade de reflexão de ondas eletromagnéticas. Assim, utilizando um transmissor em uma determinada posição pode-se emitir um sinal com propriedades definidas pelo projeto do sistema e, caso haja um objeto em seu caminho, receber um sinal refletido. De acordo com o intervalo de tempo entre a transmissão do pulso e a recepção do sinal refletido é possível calcular a distância do objeto ao transmissor com a equação radar.

Equação do radar

Densidade de potência à distância r: $W_a = \frac{P_e G}{4\pi r^2}$

Potência isotrópica equivalente reflectida pelo alvo: $P_a = W_a \, \sigma$ → área efectiva de eco (área equivalente de eco) — *radar cross section*

Densidade de potência na antena receptora: $W = \frac{P_a}{4\pi r^2} = \frac{P_e G}{4\pi r^2} \frac{\sigma}{4\pi r^2}$

Potência recebida na antena: $P_r = W \, A_0$ → área de captura da antena receptora $A_0 = G\frac{\lambda^2}{4\pi}$

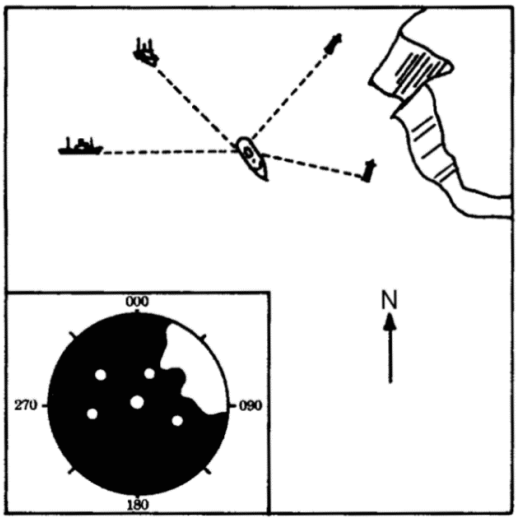
Potência recebida na antena: $P_r = \frac{P_e G^2 \lambda^2 \sigma}{(4\pi)^3 r^4} = \frac{P_e A_0^2 \sigma}{4\pi \lambda^2 r^4}$

equação do radar (1) — Alcance máximo: $r_{max} = \left[\frac{P_e G^2 \lambda^2 \sigma}{(4\pi)^3 P_{r\,min}}\right]^{\frac{1}{4}} = \left(\frac{P_e A_0^2 \sigma}{4\pi \lambda^2 P_{r\,min}}\right)^{\frac{1}{4}}$ (5)

limiar de detecção

Para que a exibição seja mais inteligível para o operador é necessário que haja, também, a informação sobre a direção, ou marcação, em relação ao transmissor. Brasil (2016) descreve que essa informação pode ser obtida de acordo com a formatação do feixe emitido, que pode ser modelado fisicamente, através do formato das antenas, ou eletronicamente, através da utilização de arranjos de transmissores. Dependendo do formato adotado para o refletor, obtemos padrões de radiação diferentes. No caso de antenas parabólicas retangulares ou cilíndricas, temos a formação de um feixe vertical largo e outro horizontal estreito, que possibilita a discriminação da marcação do alvo. Refletores parabólicos quadrados ou circulares, causam um padrão de radiação estreito tanto vertical quanto horizontalmente, possibilitando a discriminação da marcação e da altitude do alvo.

Figura 2.8 - Representação PPI

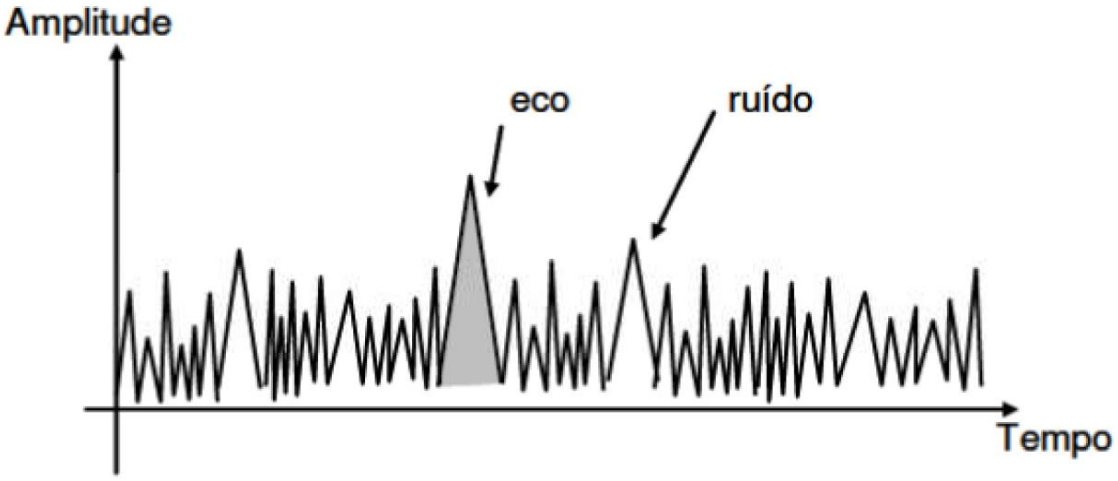


Fonte: Miguens, 1996, p.419

Esses dados são então convertidos para a forma gráfica de forma que o operador possa interpretá-los da melhor maneira possível. Grande parte dos radares de navegação utiliza a apresentação *Plan Position Indicator* (PPI), representada na Figura 2.8 (MIGUENS, 1996). O sistema também pode agregar outros mecanismos para incrementar o fornecimento de informações como cálculos de efeito doppler para oferecer a velocidade dos alvos.

Conforme análise apresentada por Brasil (2016), o processamento de sinais envolvido para a geração dessa imagem é bem complexo. Os sinais presentes do ambiente geram pequenas distorções, chamadas de ruído, que interferem na recepção do sinal refletido como verifica-se na figura abaixo.

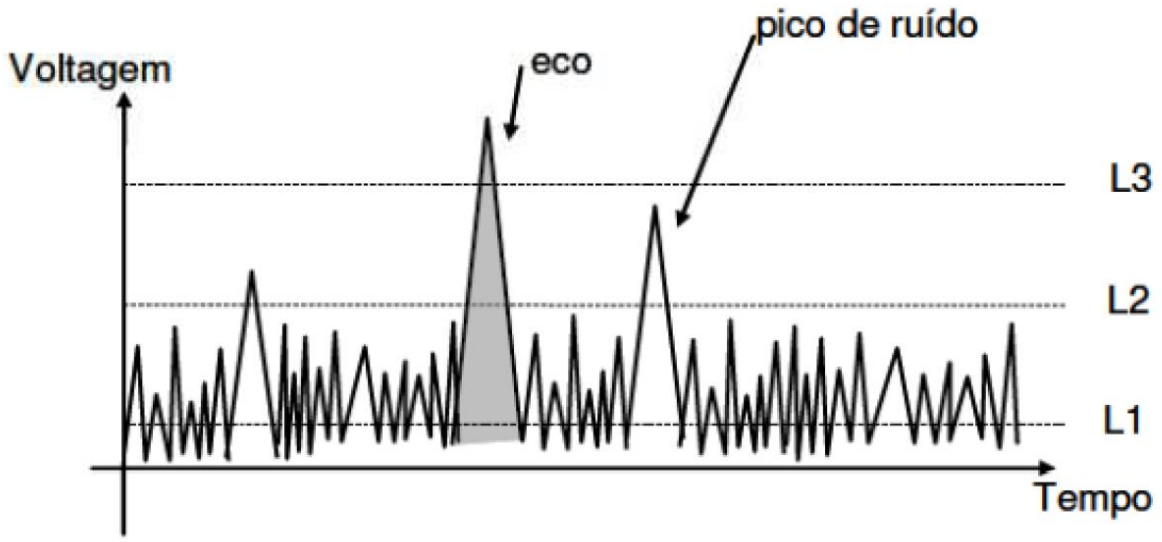
Figura 2.9 - Sinal refletido em ambiente com ruído



Fronte: Brasil, 2016, p.37

Para evitar a produção de imagens falsas na PPI, foi criado o método de estabelecimento de um nível mínimo para que um sinal recebido pelo detector seja considerado um sinal refletido, esse nível mínimo é conhecido como *threshold level* (BRASIL, 2016). Esse valor pode ser alterado tanto para mais quanto para menos de acordo com a presença de ruídos no ambiente. Na medida em que aumentamos o *threshold*, diminuímos a probabilidade de detectar um alvo falso, falso positivo, mas diminuímos a probabilidade de detecção, podendo gerar um falso negativo. Na medida em que diminuímos o *threshold* aumentamos a probabilidade de detecção, mas aumentamos a probabilidade de falso positivo, Figura 2.10.

Figura 2.10 - Estabelecimento de níveis de threshold



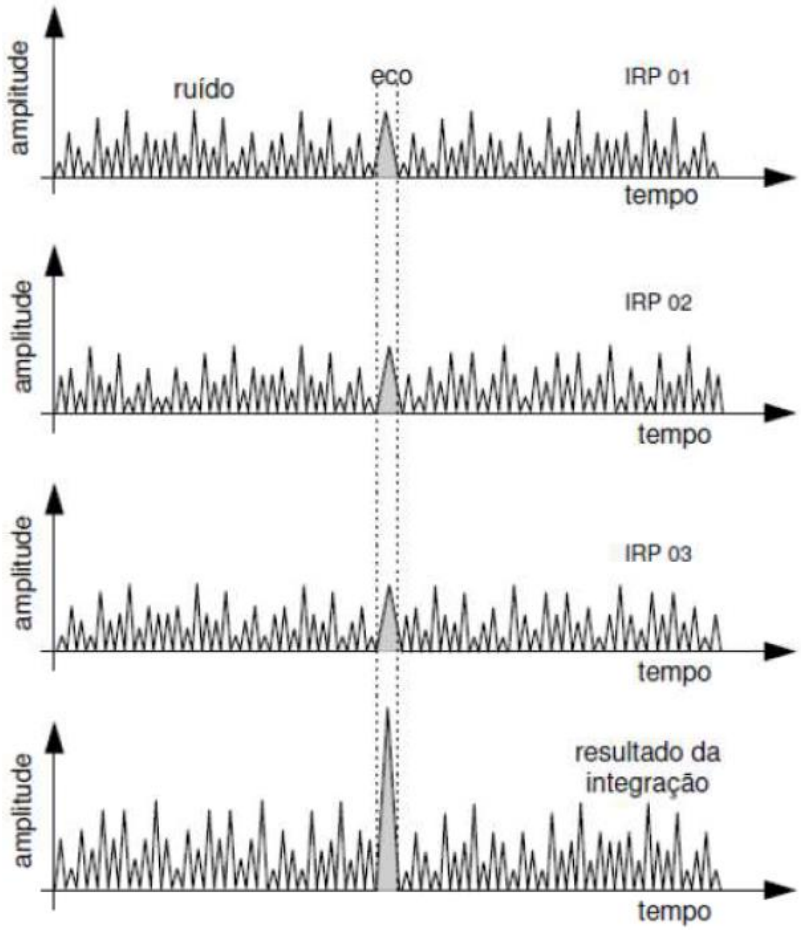
Fronte: Brasil, 2016, p.38

> **a)** Para um *threshold level* muito baixo (L1), o eco, ainda que bem-faço, será detectado, mas muitos picos de ruído serão confundidos com sinais úteis. Uma probabilidade de detecção de 100%, neste caso, é ilusória, pois o sinal estará mergulhado no ruído;
> **b)** Elevando-se este nível (L2), elimina-se boa parte do ruído, porém, o sinal de eco tem de ter potência suficiente para chegar até lá; e
> **c)** No nível mais elevado (L3) dificilmente ocorrerá um falso alarme, porém, existe o risco de não detectar nem o sinal de eco, a não ser que o mesmo seja muito forte. (GE-204, 2016, p.38).

Além desse mecanismo é possível utilizar a integração de pulsos, Figura 2.11, que realiza a integração dos valores recebidos em um determinado número de pulsos transmitidos (BRASIL, 2016). Caso exista um alvo, o sinal refletido ocorrerá em todos os pulsos. O ruído, devido ao seu caráter aleatório, possui uma probabilidade muito baixa de ocorrer várias vezes

em uma mesma posição. Assim, o sinal refletido tende a permanecer após a integração enquanto o ruído tende a ser filtrado.

Figura 2.11 - Integração de pulsos



Fonte: Brasil, 2016, p.49

Por fim, é importante comentar que, conforme Sá, Machado e Almeida (2019), em muitos sistemas modernos, operações de processamento e apresentação ficam a cargo de sistemas computacionais baseados em *software*. Assim, existe a comunicação entre o *hardware*, que executa a transmissão e recepção de sinais, e um *software*, em execução em um SO, que efetua tratamento e apresentação de dados. Como interface entre *hardware* e *software* há uma memória, onde o *hardware* grava as informações recebidas para posterior processamento pelo *software*.

3 METODOLOGIA

A pesquisa tem como objetivo realizar uma análise a respeito da aplicabilidade da GCE no ambiente marítimo. Para isso, foram utilizadas plataformas experimentais em sistemas computacionais. Neste capítulo serão apresentadas as classificações da pesquisa realizada no presente trabalho e os métodos de coleta e tratamento de dados empregados.

3.1 Classificação da Pesquisa

De acordo com os enquadramentos possíveis, pode-se classificar a pesquisa conforme os seguintes quesitos:

3.1.1 Quanto aos fins

Será utilizada uma pesquisa exploratória que, para Gil (2002), tem como finalidade propor questões mais precisas ou apresentar novas possibilidades para trabalhos futuros. A pesquisa pode, também, ser classificada como descritiva. Segundo Gil (2002), esse tipo de pesquisa tem por objetivo a descrição das características de determinado fenômeno e se utiliza de técnicas padronizadas de coleta de dados.

3.1.2 Quanto aos meios

Os conceitos sugeridos serão aplicados em uma pesquisa experimental em laboratório, que, conforme Gil (2002), se refere a simulações realizadas em condições controladas, geralmente em local circunscrito, com possibilidade de reprodução. Serão utilizadas plataformas computacionais em simulações, com fatores controlados.

3.2 Limitações do Método

Em decorrência da utilização de simulações em ambientes controlados, os resultados não são influenciados por todas as variáveis possíveis em um ambiente real. Dentre essas variáveis encontram-se: ruídos, influência de nuvens e estado do mar.

3.3 Coleta e Tratamento de Dados

Os dados foram coletados através de simulações em ambiente controlado utilizando a linguagem de programação Python e o sistema computacional MATLAB. Será realizada uma análise quantitativa, objetivando o levantamento de estatísticas, e uma abordagem qualitativa para avaliar a eficácia dos experimentos. Os procedimentos adotados serão mais bem descritos nos capítulos seguintes.

4 DESCRIÇÃO DO AMBIENTE DE SIMULAÇÃO

A presente pesquisa analisará a aplicabilidade de conceitos de GCE no ambiente marítimo. Em um primeiro momento será apresentado o cenário para a aplicação do processamento de imagens e posteriormente o cenário para utilização de conceitos de processamento de sinais como parte integrante de um ataque ciber-eletrônico.

4.1 Premissas

Nos experimentos, considera-se que os ataques estão sendo realizados a um sistema radar genérico utilizado em uma plataforma marítima. Esse sistema radar realiza a detecção eletromagnética e realiza a gravação dos dados em linguagem binária em uma memória para posterior tratamento e apresentação por um *software* executado no SO *Windows* 10. O computador que executa o sistema está infectado por um *malware* que opera em modo discreto, sem executar nenhuma ação, até que seja detectado um determinado sinal de ativação. O malware permanece observando o sistema para detectar o sinal pré-programado e, então, acionar seus procedimentos para prejudicar o funcionamento do sistema.

4.2 Especificações da plataforma de testes

As simulações foram realizadas em um sistema computacional com processor Intel i7 com capacidade de 2.5 Ghz, 8GB de memória RAM DDR3 e 250 GB de disco rígido em estado sólido. O SO utilizado foi o Microsoft Windows 10 Home 64 bits. Essas informações se encontram resumidas no quadro a seguir:

Quadro 1 - Especificações da plataforma de testes

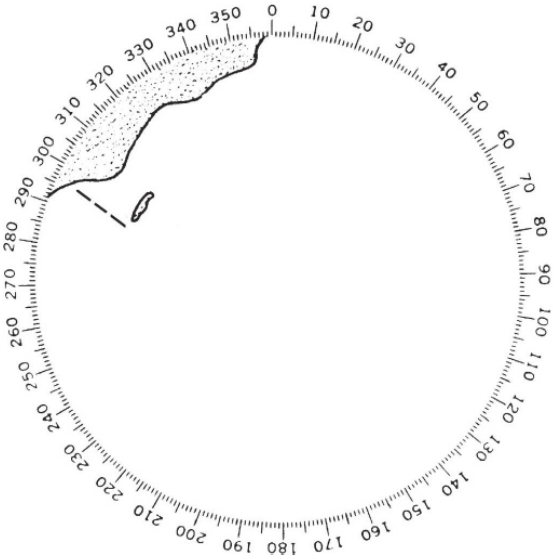
Especificações do Sistema	
Sistema Operacional	Microsoft Windows 10 Home 64 bit
CPU	Intel i7-6500U 2.5 Ghz
RAM	8 GB DDR3
Disco Rígido em Estado Sólido	250 GB

Fonte: Elaborado pelo autor

4.3 Ativação baseada no processamento de imagens

Como comentado anteriormente, considera-se que o sistema alvo está infectado com um malware ativado após a detecção de um determinado sinal recebido a partir de uma MAE. Nesse primeiro caso vamos considerar que o sinal de ativação é processado e apresentado como uma imagem exibida na PPI, ou tela de apresentação do radar. Sabemos que um sistema radar apresenta ao operador os sinais refletidos recebidos através de uma imagem pela PPI. Entretanto, é possível a um terceiro provocar a geração de imagens que não estão relacionadas a sinais refletidos, essa técnica também pode ser utilizada como auxílio à navegação, conforme a Figura 4.1 (MIGUES, 1996).

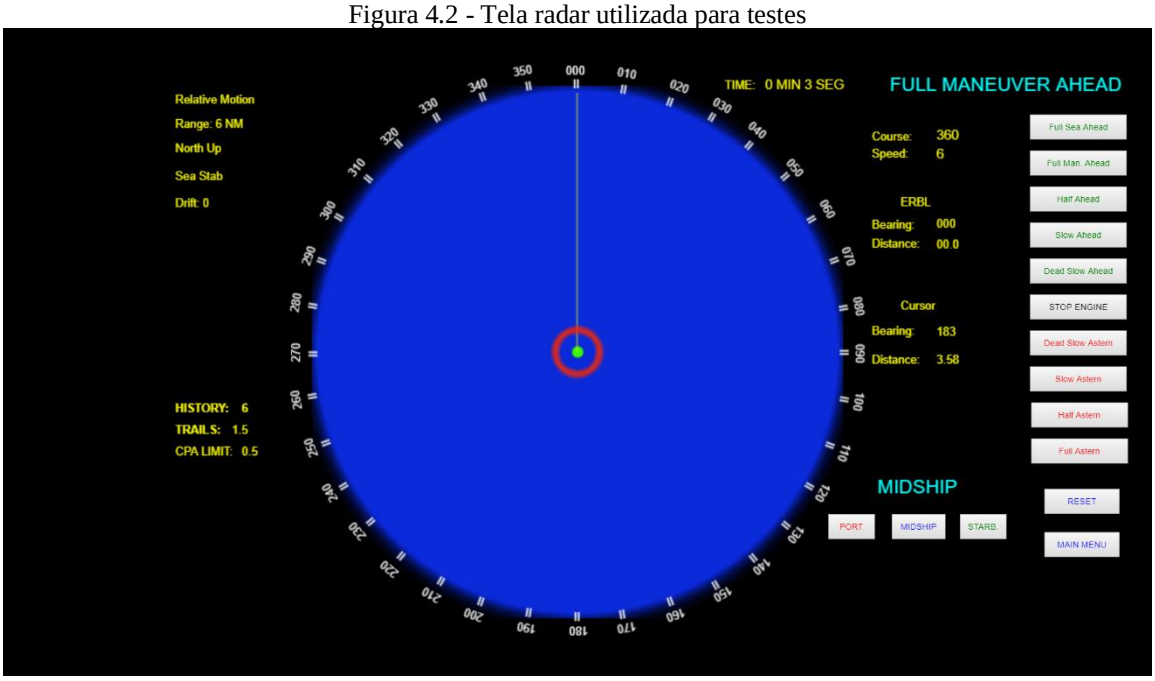
Figura 4.1 - Geração de imagens na PPI como auxílio à navegação



Fonte: Miguens, 1996, p. 438

Sabendo disso, é possível que o *malware* instalado execute uma busca constante na imagem exibida no monitor até que se encontre um determinado padrão que servirá para provocar a ativação de uma ação maliciosa. Para testar essa hipótese, foi utilizada a apresentação de uma PPI simulada com o *software CINEMATIC RADAR SIMULATOR 2.0*, como demonstrado na Figura 4.2. Além disso foram gerados 30 cenários fictícios, representando situações reais onde uma plataforma naval poderia se encontrar para testar a eficácia do método de busca empregado, de acordo com o Apêndice A.

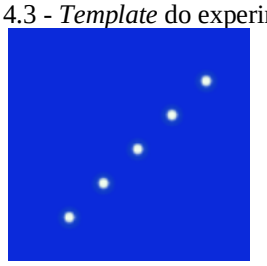
Figura 4.2 - Tela radar utilizada para testes



Fonte: Covelli, 2017

Para implementar o método de busca, a linguagem Python foi utilizada. No algoritmo foi estabelecido uma busca, utilizando a técnica de *template matching*, em uma imagem teste. Considera-se que essa imagem teste é uma situação instantânea do monitor do radar e foi captada pela *malware*. O *template* utilizado para a busca pode ser visualizado na imagem a seguir:

Figura 4.3 - *Template* do experimento



Fonte: Elaborada pelo autor

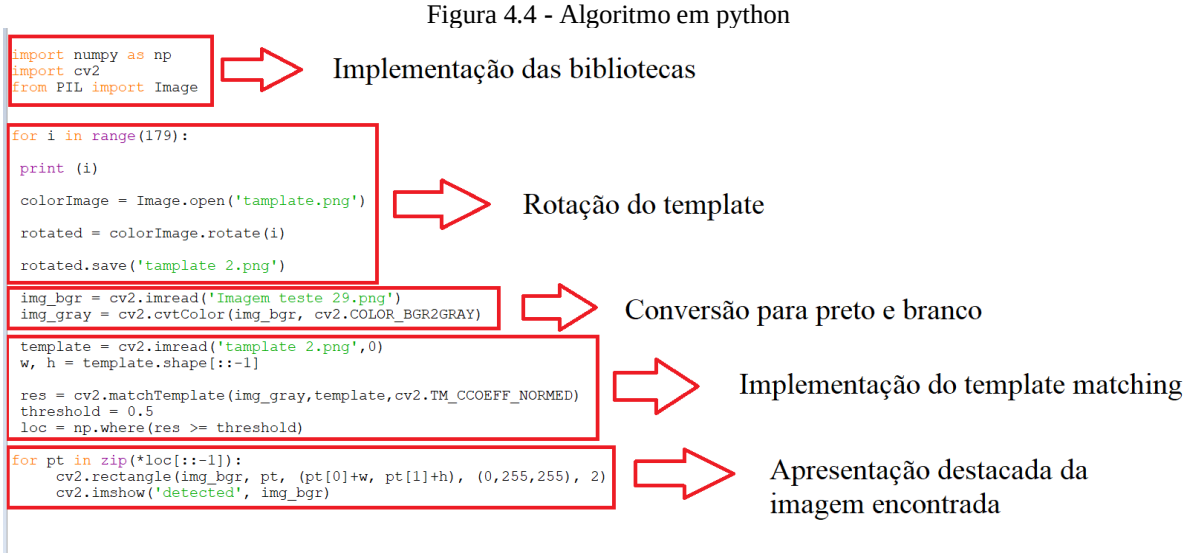
É importante comentar que se considera que um terceiro está emitindo esse sinal, e ele pode estar localizado em qualquer direção da PPI. Assim, é importante considerar que o sinal de ativação pode estar em qualquer direção possível. Para fins de simplificação consideram-se variações de 1 grau, ou seja, o atacante poderia emitir o sinal da direção 000, 001, 002, 003 e assim por diante. Para detectar essas possíveis variações, foi estabelecido que após uma varredura, caso o *template* não seja encontrado, será realizada uma nova busca após girar o *template* em 1 grau. Isso será feito até que o *template* seja encontrado ou até que todas as variações sejam testadas. Percebe-se que utilizaremos apenas 179 graus de rotação, pois

consideram-se, nesse caso, as equivalências em ângulos suplementares, o *template* em 000 é equivalente ao *template* em 180.

Na busca propriamente dita, realizamos a leitura da imagem teste e convertemos para tons de preto de branco. Isso serve para eliminar possíveis variações de cores, realizando apenas a análise da intensidade dos pixels. O *template* já é lido em tons de preto e branco para comparação. A busca é iniciada realizando comparações utilizando a equação 4, a equivalência é considerada se for encontrado um fator de semelhança acima do *threshold*. Por fim, é realizada a apresentação do *template* encontrado com um quadro para destacá-lo na imagem.

Para a confecção do algoritmo, Figura 4.4 e Apêndice B, foram utilizadas as bibliotecas Numpy, cv2 e Pillow. De acordo com o site pypi.org (2020), repositório de bibliotecas em python, Numpy é um módulo para a realização de operações matemáticas, Cv2 é um módulo para operações com imagens e Pillow também realiza operações com imagens.

Figura 4.4 - Algoritmo em python



Fonte: Elaborada pelo autor

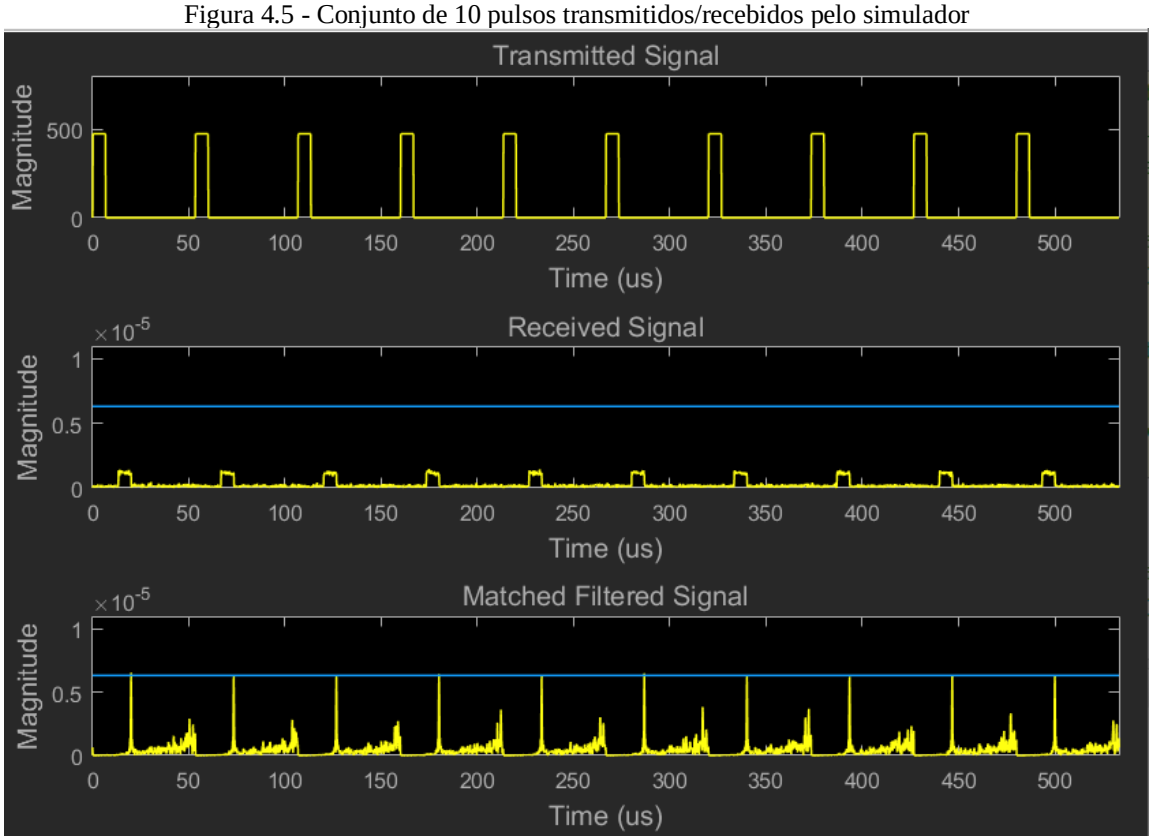
O bloco de apresentação da imagem encontrada, retratado na imagem acima, poderia ser substituído por um bloco de ativação para a função maliciosa do *malware* instalado.

4.4 Ativação baseada no processamento de sinais

O cenário anterior tem como objetivo demonstrar a possibilidade de emprego de um ataque ciber-eletrônico utilizando a localização de uma imagem de ativação na tela do radar alvo, porém, essa imagem pode gerar suspeitas por parte dos operadores. Por isso, foi realizado um experimento que busca gerar uma informação que possa ser localizada pelo *malware* sem

que seja apresentada imagem na PPI. Nesse caso nota-se que a técnica de integração de pulsos pode ser utilizada pelo atacante para obter furtividade. Foi utilizado o sistema computacional MATLAB, empregando o algoritmo *Stream and Accelerate Simulation of Radar System*, Anexo A, que se encontra disponível no banco de dados do *software*.

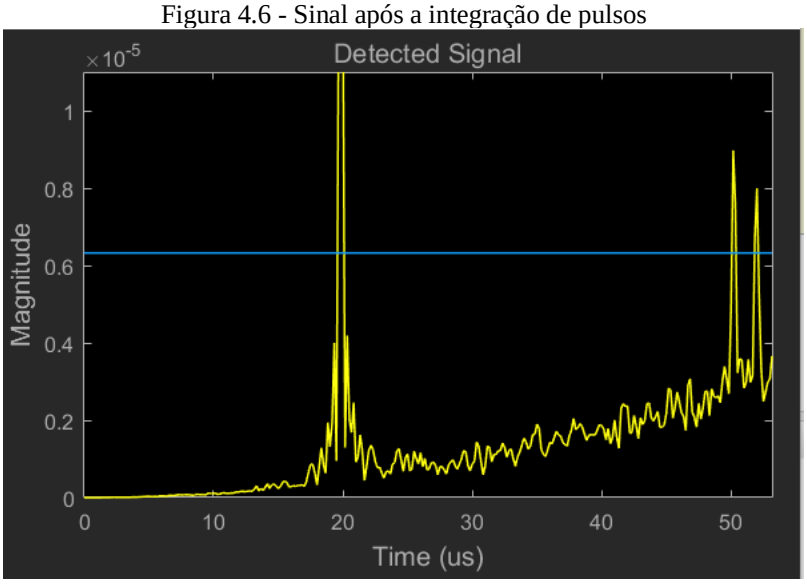
Figura 4.5 - Conjunto de 10 pulsos transmitidos/recebidos pelo simulador



Fonte: MATLAB, 2020

Esse algoritmo objetiva simular um sistema radar que gera e transmite uma forma de onda e simula a reflexão dessa onda em alvos fictícios. A simulação emprega a técnica de integração de pulsos, realizando uma integração a cada 10 pulsos, conforme a Figura 4.5, e considera a presença de três alvos. Também há um nível de *threshold* estabelecido para discriminar alvos de ruídos. Percebe-se, ao visualizar a figura acima, que o sinal transmitido tem a forma de uma onda quadrada e o os sinais recebidos são de uma dimensão bem menor que o sinal transmitido. Após a aplicação do filtro casado temos o sinal que representa a detecção dos alvos em uma determinada marcação. Esses 10 pulsos são então integrados para reduzir a influências do ruído ambiente e aumentar o sinal refletido pelos alvos, de acordo com a Figura 4.6.

Figura 4.6 - Sinal após a integração de pulsos



Fonte: MATLAB, 2020

A linha azul na figura acima representa o limite de *threshold* que representa o que será apresentado na PPI do radar. Nota-se a presença de três alvos, conforme definido no algoritmo inicial. Esse experimento propõe que seria possível emitir um sinal que fosse formatado de forma a ser transmitido apenas no intervalo de um pulso dos dez utilizados para cada integração. Considera-se que o sistema que realizará a integração faz parte do *software* que está rodando no SO *Windows.* Em nosso sistema simulado, a matriz rxsig 320x1 representa o sinal recebido em um intervalo de varredura. Assim, assume-se que o malware é capaz de realizar a leitura constante da primeira matriz rxsig de cada intervalo de recepção. Nossa hipótese considera que após um determinado número de detecções de um sinal de ataque o malware ativaria ações para prejudicar o sistema.

Para verificar a possibilidade de emprego dessa hipótese, foi inserido um sinal de ataque no código original, conforme a figura a seguir e Apêndice D. O sinal rxsig representa os sinais reais, refletidos e ruídos do ambiente. É gerado então um sinal AtSig, que é um sinal gerado por um possível atacante na direção.

Figura 4.7 - Sinal de ataque

```
% Receive target returns at sensor
rxsig = collector(tgtsig,tgtang);

% SINAL ATAQUE
tt = length(rxsig)
AtSig = zeros(tt,1);
if mod(m, num_pulse_int)==1
     AtSig(150:200) = max(real(rxsig));
end
rxsig = rxsig+AtSig;
```

Fonte: Elaborada pelo autor

5 ANÁLISE DOS RESULTADOS

Nesse capítulo serão apresentados os resultados obtidos nas simulações, bem como algumas considerações referentes aos mesmos.

5.1 Ativação baseada do processamento de imagens

Foram realizadas buscas em trinta imagens, conforme o Apêndice A. Os resultados, que podem ser verificados no Apêndice C, foram compilados nas tabelas a seguir:

Tabela 5.1 - Resultados com ativação por processamento de imagens para *threshold* = 0.3

Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo	TOTAL
15	12	1	2	30

Fonte: Elaborada pelo autor

Tabela 5.2 - Resultados com ativação por processamento de imagens para *threshold* = 0.4

Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo	TOTAL
14	2	11	3	30

Fonte: Elaborada pelo autor

Tabela 5.3 - Resultados com ativação por processamento de imagens para *threshold* = 0.5

Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo	TOTAL
14	0	13	3	30

Fonte: Elaborada pelo autor

Tabela 5.4 - Resultados com ativação por processamento de imagens para *threshold* = 0.6

Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo	TOTAL
13	0	13	4	30

Fonte: Elaborada pelo autor

Tabela 5.5 - Resultados com ativação por processamento de imagens para *threshold* = 0.7

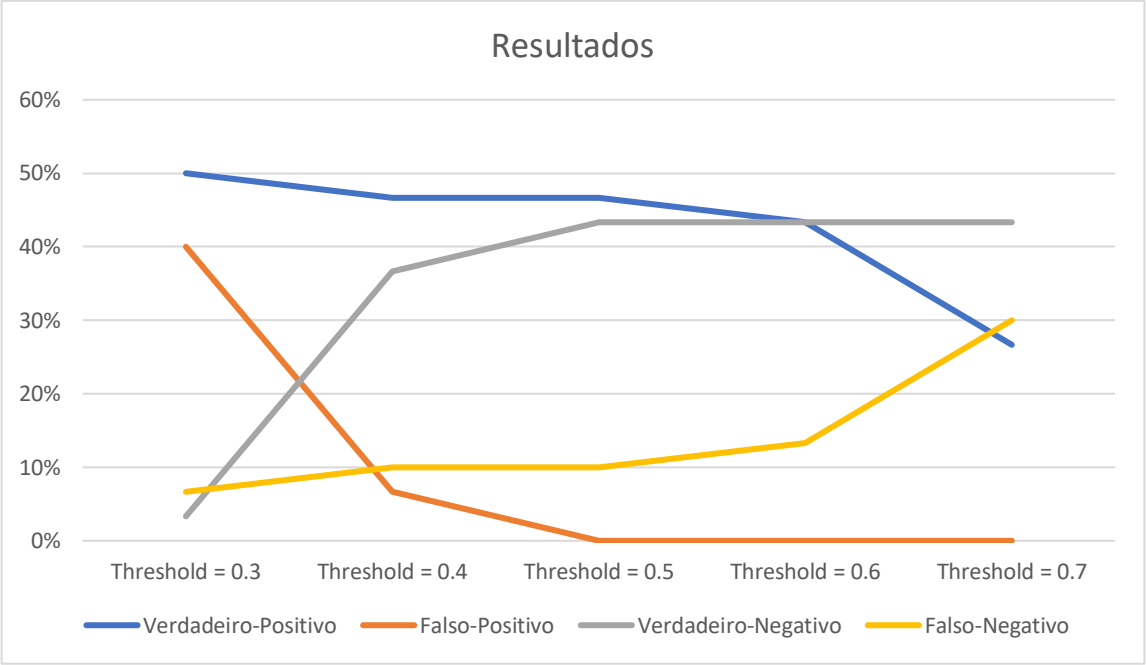
Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo	TOTAL
8	0	13	9	30

Fonte: Elaborada pelo autor

A situação de Verdadeiro-Positivo (VP) se refere ao caso onde o *template* está presente na imagem teste e foi detectado, Falso-Positivo (FP) é o caso em que o *template* não está presente, mas é detectado, Verdadeiro-Negativo (VN) se dá quando o *template* não está na

imagem e não é detectado e Falto-Negativo (FN) quando o *template* está na imagem mas não é detectado.

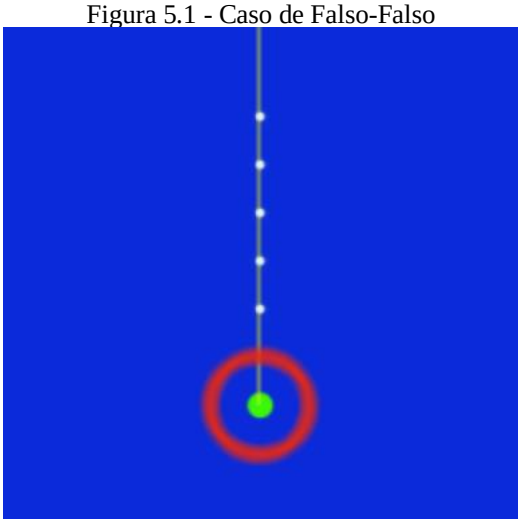
Gráfico 5.1 - Resultados



Fonte: Elaborado pelo autor

O Gráfico 5.1 - Resultados ilustra a influência da variação do *threshold* nos resultados das buscas realizada. Percebe-se que a escolha do valor a ser utilizado depende da necessidade de emprego. A redução do *threshold* aumenta o índice de VP, porém causa um aumento na taxa de FP. Por outro lado, o aumento desse valor diminui as chances de FP porém aumenta a taxa de FN. Dessa forma, é necessário que o usuário avalie o contexto operacional onde o código será empregado para obter uma solução ótima.

Figura 5.1 - Caso de Falso-Falso



Fonte: Elaborada pelo autor

Ao analisar as imagens, nota-se um caso que merece especial destaque pois apresentou FN em todos os valores de *threshold* testados. A Figura 5.1 ilustra um caso onde uma linha, que representa o rumo da embarcação, encontra-se sobre a imagem que se deseja localizar. A linha foi capaz de impossibilitar a detecção da imagem. Esse ponto deve ser levado em consideração para o emprego da técnica e demonstra que ruídos presentes em uma situação real podem alterar os resultados.

Tabela 5.6 - Taxa de acerto para os valores de *threshold* utilizados

Threshold = 0.3	Threshold = 0.4	Threshold = 0.5	Threshold = 0.6	Threshold = 0.7
53.3%	83.3%	90%	86.6%	70%

Fonte: Elaborada pelo autor

O caso apresentado na figura acima apresenta uma baixa taxa de incidência e pode ser facilmente contornado em uma situação tática real. Vale ressaltar que o *malware* está executando uma busca constante, ou seja, logo que essa situação for alterada a detecção volta a ser possível. Por fim, a Tabela 5.6 - Taxa de acerto para os valores de *threshold* utilizados apresenta a taxa de acerto para cada um dos valores de *threshold* utilizados nas simulações. Percebe-se que o melhor desempenho foi obtido utilizando o *threshold* de 0.5, apresentando uma taxa de acerto de 90%.

Figura 5.2 - Código para medição de tempo de execução

```
import numpy as np
import cv2
from PIL import Image

import time

ini = time.time()

for i in range(179):

 print (i)

 colorImage = Image.open('tamplate.png')

 rotated = colorImage.rotate(i)

 rotated.save('tamplate 2.png')

 img_bgr = cv2.imread('Imagem teste 5.png')
 img_gray = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY)

 template = cv2.imread('tamplate 2.png',0)
 w, h = template.shape[::-1]

 res = cv2.matchTemplate(img_gray,template,cv2.TM_CCOEFF_NORMED)
 threshold = 0.7
 loc = np.where(res >= threshold)

for pt in zip(*loc[::-1]):
     cv2.rectangle(img_bgr, pt, (pt[0]+w, pt[1]+h), (0,255,255), 2)
     cv2.imshow('detected', img_bgr)

fim = time.time()

print ("Tempo de execução: ", fim-ini)
```

Início da contagem

Fim da contagem

Fonte: Elaborada pelo autor

O tempo de execução do algoritmo foi medido no caso da Figura A.6, onde não há presença do *template*, para que se noção do tempo máximo necessário para realizar uma varredura. Essa medida foi realizada utilizando os comandos exibidos na Figura 5.2, e apresentou o resultado de aproximadamente 37 segundos. Dessa forma, é possível realizar uma varredura a cada 37 segundos.

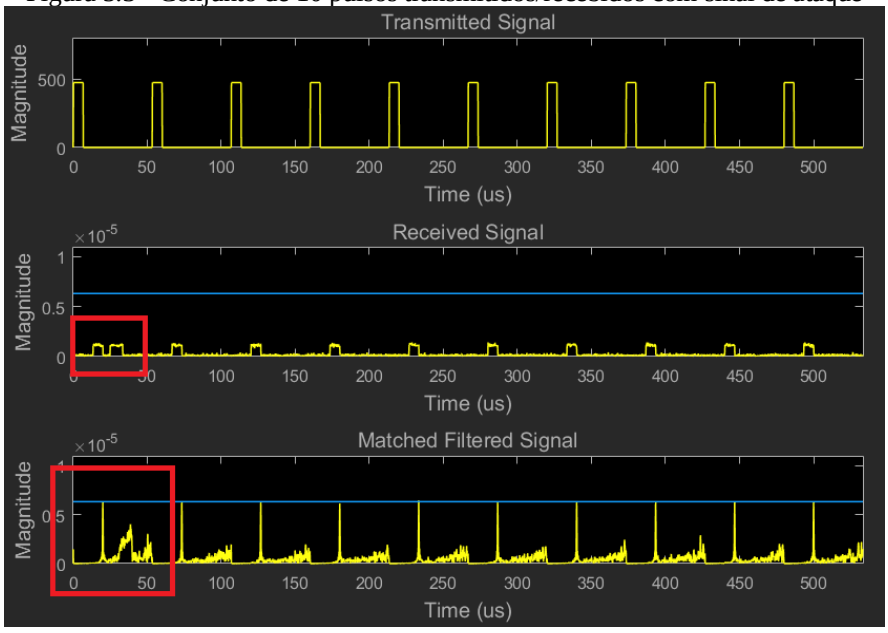
5.1.1 Considerações

Como foi proposto, esse trabalho procura apresentar possibilidades de emprego, por isso não foram abordadas questões relativas à otimização e melhorias no algoritmo aplicado. É importante ressaltar que esse ataque produz uma imagem na PPI do radar alvo e essa imagem não passará desapercebida pelo operador que poderá suspeitar que um ataque está ocorrendo. Esses resultados demonstram que o emprego de processamento de imagem pode servir de maneira eficaz para a detecção de padrões na tela do radar e, como proposto, servir como gatilho para a ativação de um *malware* instalado no sistema alvo com boa chance de sucesso.

5.2 Ativação baseada no processamento de sinais

O sinal de ataque foi somado com os sinais reais e apresentado no processamento conforme a figura abaixo:

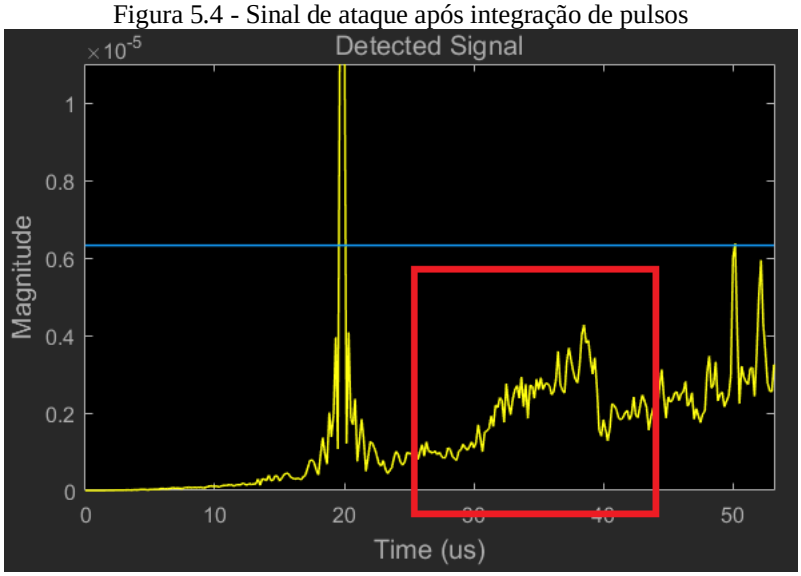
Figura 5.3 - Conjunto de 10 pulsos transmitidos/recebidos com sinal de ataque



Fonte: Elaborada pelo autor

É possível visualizar, com base na figura acima, que o sinal inserido está presente apenas no primeiro intervalo de recepção entre os 10 transmitidos antes de cada integração. Sua característica inconstante faz com o que, após a integração de pulsos, ele não seja capaz de atingir o valor mínimo para a exibição da tela. Apesar disso os dados desse sinal se encontram na variável rxsig referente ao primeiro pulso do conjunto.

Figura 5.4 - Sinal de ataque após integração de pulsos



Fonte: Elaborada pelo autor

Percebe-se, na figura acima, que após a integração de pulsos o sinal de ataque permanece abaixo da linha azul, que representa o *threshold*, não sendo exibido na PPI. Considerando que o *malware* é capaz acessar a matriz e reconhecer esse sinal, seria possível estabelecer um contador que, após atingir um valor que proporcione uma boa confiabilidade, poderá ativar a ação maliciosa oportunamente.

5.2.1 Considerações

Com esse experimento contata-se que é possível realizar um ataque com características furtivas, ou seja, que não gera alteração na PPI radar. Dessa forma seria possível realizar um ataque sem o risco de alarmar o operador do sistema alvo. É importante ressaltar que ainda é necessário desenvolver um algoritmo capaz de identificar o sinal de ataque no estágio pré-integração, que apresenta uma relação sinal ruído menor em relação ao estágio pós-integração.

6 CONCLUSÃO

Considerando o referencial teórico apresentado e os experimentos executados é possível perceber que a guerra ciber-eletrônica possui aplicabilidade no ambiente marítimo. É possível utilizar o processamento de imagens e sinais como gatilho para ativação de um código malicioso previamente instalado em um sistema radar de navegação com bom índice de precisão. Mesmo com todos os artifícios de segurança da informação, todos os sistemas computacionais estão sujeitos ao risco de serem infectados por *malware*, Stallings (2019). Esse código malicioso pode ser utilizado em proveito de uma operação naval, fazendo com que seu efeito nocivo seja ativado no momento mais oportuno para a força atacante. Uma alternativa para mitigar essa ameaça seria a utilização periódica de ferramentas para verificação da integridade dos códigos utilizados em um determinado sistema radar. Além disso, é recomendável obedecer às normas e recomendações para a salvaguarda das informações digitais.

6.1 Considerações Finais

De acordo com as memórias de Karl Doenitz (2012), principal nome na condução da guerra submarina alemã na Segunda Guerra Mundial, os rumos da guerra no mar foram alterados pela pesquisa e desenvolvimento, em especial dos sistemas de guerra eletrônica, como o radar. Dixon (1976), afirma que muitos dos conhecimentos aplicados pelo exército alemão no que se refere a utilização de blindados em ações militares na Segunda Guerra Mundial foram frutos de estudos de militares ingleses. Esses estudos não encontraram aceitação pelos formuladores da doutrina militar inglesa, pois em sua visão a cavalaria ainda seria superior. Em ambos os casos se nota que os rumos da história foram alterados em benefício daqueles que melhor utilizaram as tecnologias disponíveis. Assim, pode-se perceber que é de suma importância para o meio militar que as novas tecnologias sejam estudadas para que se dimensione da melhor maneira possível seu impacto na arte da guerra.

É válido comentar que o presente estudo apresenta peculiaridades que remetem ao conceito do *cisne negro*, que caracteriza determinados eventos. “Cisne Negro se resume ao terceto: raridade, impacto e previsibilidade retrospectiva” (SARRO, 2020). De maneira geral, o *cisne negro* se refere a um evento com baixa probabilidade de ocorrência, mas de grande impacto, que não foi devidamente explorado antes de ocorrer. Nota-se que a presença de códigos maliciosos em equipamentos eletrônicos não pode ser negligenciada, especial por

instituições de defesa. Dessa forma, promover estudos voltados a esse assunto é importante para evitar o advento de um *cisne negro*.

6.2 Sugestões para Futuros Trabalhos

Sugere-se que nas próximas edições do C-Ap-A sejam incluídos na lista de temas de interesse tópicos relacionados aos conceitos apresentados. Esses conceitos seriam a utilização de técnicas de processamento de imagem para identificação de padrões em imagens radar e a utilização de técnicas de processamento de sinais para implantação de informações maliciosas em sinais eletromagnéticas. Além desses tópicos, seria interessante pensar o desenvolvimento das demais ações maliciosas que podem estar relacionadas ao *malware* como técnicas de infecção, técnicas para execução furtiva e ações preventivas contra essas mesmas ameaças.

REFERÊNCIAS

ADEE, S. The Hunt for the Kill Switch. **IEEE Spectrum**, v. 45, n. 5, p.34-39, 2008.

BHATTI, J.; HUMPHREYS, T. E. Hostile control of ships via false GPS signals: Demonstration and detection. **NAVIGATION: Journal of the Institute of Navigation**, v. 64, n. 1, p. 51-66, 2017.

BRASIL. Centro de Guerra Eletrônica da Marinha. GE- 204 – **Processamento de Sinais Radar e Sistemas Radar**. Niterói, 2016.

BRASIL. Comando de Operações Navais. ComOpNav- 521 - **Manual de Guerra Eletrônica**. Rio de Janeiro, 2003.

BRASIL. Diretoria Geral do Material da Marinha. DGMM-540 – **Normas de Tecnologia da Informação da Marinha**. Rio de Janeiro, 2019.

BRASIL. Estado Maior da Armada. EMA-416 – **Doutrina de Tecnologia da Informação da Marinha**. Brasília, 2007.

BRASIL. Ministério da Defesa. **Estratégia Nacional de Defesa**, 2ª ed. Brasília, 2012.

COVELLI, M. **Cinematic Radar Simulator**. Versão 2.0. Disponível em: https://www.amazon.com/CINEMATIC-RADAR-SIMULATOR-2-0-Download/dp/B075R1PLP9. Acesso em: 08 de jan. de 2020.

DIXON, N. **On the Psychology of Military Incompetence**. London: PIMLICO, 1994

DOENITZ, K. **Memoirs**: Ten Years and Twenty Days. Maryland: Naval Institute Press, 2012.

FALLEIRO, F. D. **Conversor Analógico-Digital com Capacitores Mínimos Integrado na Tecnologia CMOS**. Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2015.

FALLIERE, N.; MURCHU, L.; CHIEN, E. **W32.Stuxnet Dossier**. Symantec Corporation, 2011. Disponível em: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf. Acesso em: 08 de jan. de 2020.

GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 4ª ed. São Paulo: Atlas, 2002.

KUMAR, A.; PANDA, S. P. A Survey: How Python Pitches in IT-World. **International Conference on Machine Learning, Big Data, Cloud and Parallel Computing**. Índia, 2019.

LEITÃO, M. J. M. **Sistemas de Radar**. 29 Slides. Disponível em: https://web.fe.up.pt/~mleitao/SRCO/Teoricas/SRCO_RAD.pdf. Acesso em: 08 de jan. de 2020.

LOCKHEED MARTIN. **Cyber Electronic Warfare.** Disponível em: https://www.lockheedmartin.com/en-us/capabilities/electronic-warfare/cyber-electronic-warfare.html. Acesso em: 19 de jan. de 2020.

MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**. 5ª ed. Rio de Janeiro: LTC, 2013a.

MACHADO, F. B.; MAIA, L. P. **Material Suplementar para Acompanhar Arquitetura de Sistemas Operacionais**. 5ª ed. Rio de Janeiro: LTC, 2013b.

MATHWORKS. **MATLAB**. Versão R2018a. Disponível em: https://www.mathworks.com/products/new_products/release2018a.html. Acesso em: 08 de jan. de 2020.

MCLAUGHLIN, S. *et al*. The Cybersecurity Landscape in Industrial Control Systems. **Proceedings of the IEEE**, v. 104, n. 5, p.1039-1057, 2016.

MENDONÇA, C. S. **Guerra Cibernética**: Desafios de uma Nova Fronteira. Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2014.

MIGUENS, A. P. **Navegação:** a Ciência e a Arte. Niterói: Diretoria de Hidrografia e Navegação da Marinha, 1996.

PARCHARIDIS, M. D. **Simulation of Cyber Attacks Against Scada Systems**. International Hellenic University. Thessaloniki, 2018.

PYPI.ORG. **The Python Package Index**. Disponível em: https://pypi.org/. Acesso em: 08 de jan. de 2020.

PYTHON.ORG. **Python**. Versão 3.8. 14 out. 2019. Disponível em: https://www.python.org/. Acesso em: 08 de jan. de 2020.

REGAN, J. **O que é Malware? Como Malwares Funcionam e como se Livrar Deles**. AVG, 10 de jul. de 2019. Disponível em: https://www.avg.com/pt/signal/what-is-malware. Acesso em: 08 de jan. de 2020.

SÁ, A. O.; MACHADO, R. C. S.; ALMEIDA, N. N. O Encontro da Guerra Cibernética com as Guerras Eletrônica e Cinética no Âmbito do Poder Marítimo. **Revista da Escola de Guerra Naval**, Rio de Janeiro, v. 25, n. 1, p. 89-128. Janeiro/abril. 2019.

SANGER, D. E. **The Perfect Weapon**: War, Sabotage and Fear in the Cyber Age. Nova Iorque: Broadway Books, 2018.

SARRO, T. J. **Pesquisando o Cisne Negro nas Águas do Cisne Branco**:
um breve ensaio sobre a lógica do Cisne Negro. Disponível em:
https://www.marinha.mil.br/spp/node/60. Acesso em: 08 de jan. de 2020.

SCHLEHER, D. C. **Introduction to Electronic Warfare**. Norwood: Artech House, 1994.

STALLINGS, W. **Criptografia e Segurança de Redes: princípios e práticas.** 6ª ed. São Paulo: Pearson Education do Brasil, 2015.

STATCOUNTER. **Desktop Operating System Market Share Wordwide**. Disponível em:
https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-201812-201912.
Acesso em: 08 de jan. de 2020.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson Prentice Hall, 2009.

TAVARES, Y. M. **Sistema Integrado de Hardware/Software para Rastreamento de Alvos**. Universidade do Estado do Rio de Janeiro. Rio de Janeiro, 2016.

ZETTER, K. **Countdown to Zero Day:** Stuxnet and Launch of the World's First Digital Weapon. Nova Iorque: Broadway Books, 2015.

APÊNDICE A – Imagens Utilizadas em Testes

Figura A.1 - Imagem teste 1

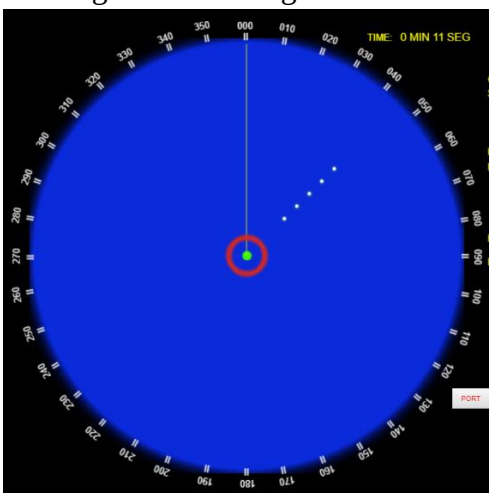


Figura A.2 - Imagem teste 2

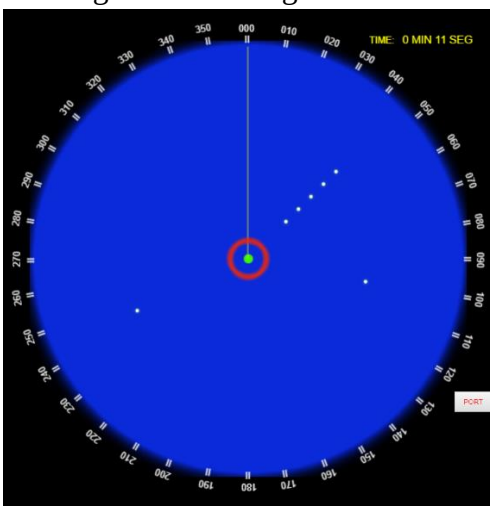


Figura A.3 - Imagem teste 3

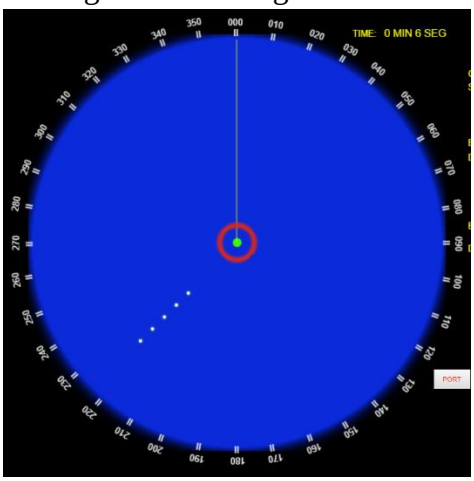


Figura A.4 - Imagem teste 4

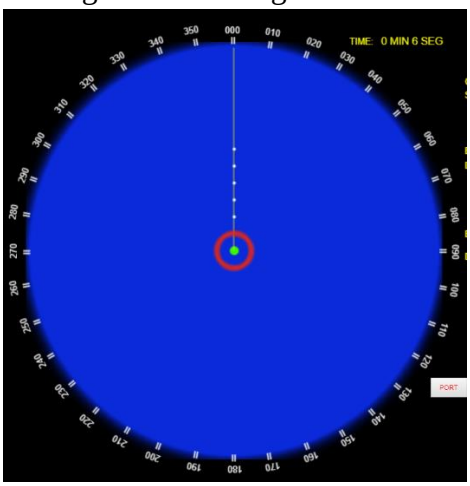


Figura A.5 - Imagem teste 5

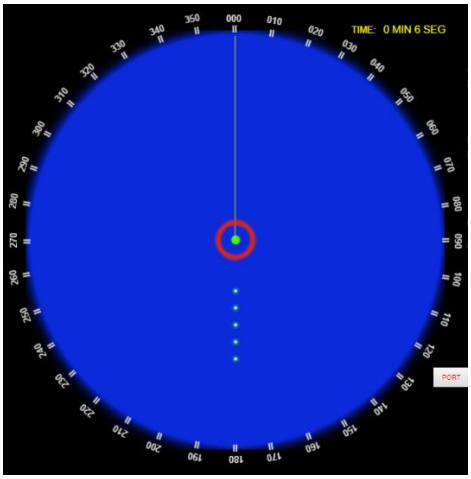


Figura A.6 - Imagem teste 6

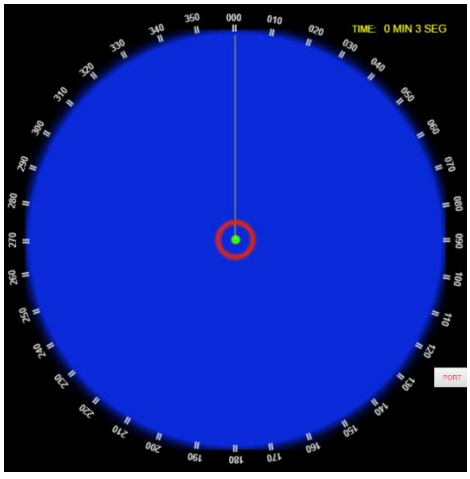


Figura A.7 - Imagem teste 7

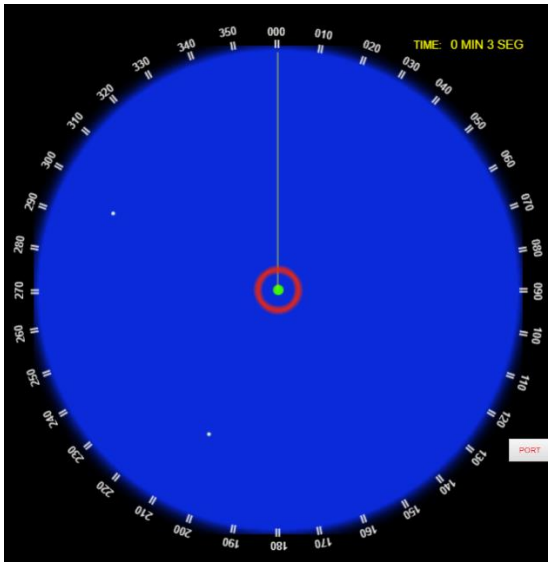


Figura A.8 - Imagem teste 8

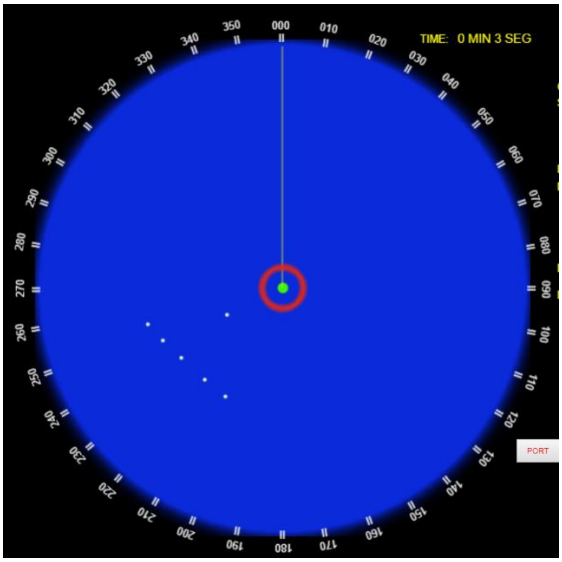


Figura A.9 - Imagem teste 9

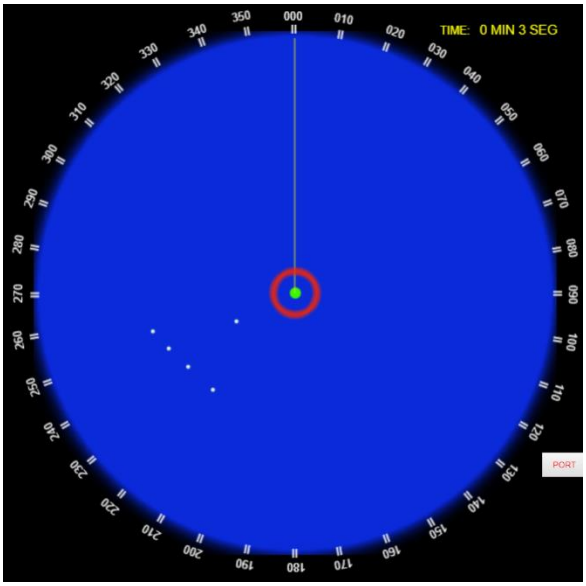


Figura A.10 - Imagem teste 10

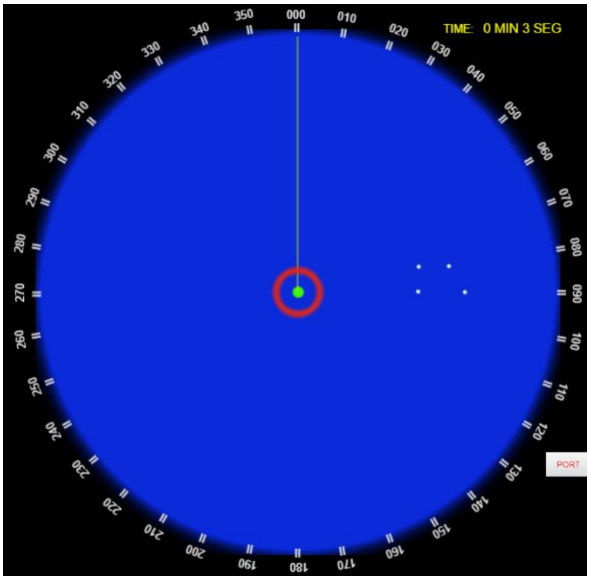


Figura A.11 - Imagem teste 11

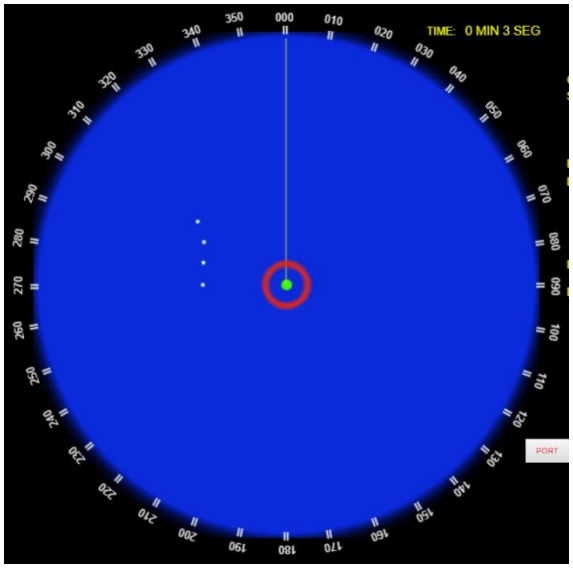


Figura A.12 - Imagem teste 12

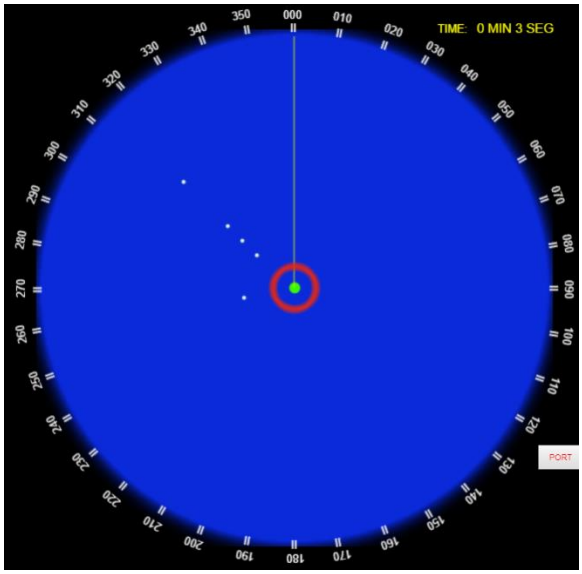


Figura A.13 - Imagem teste 13

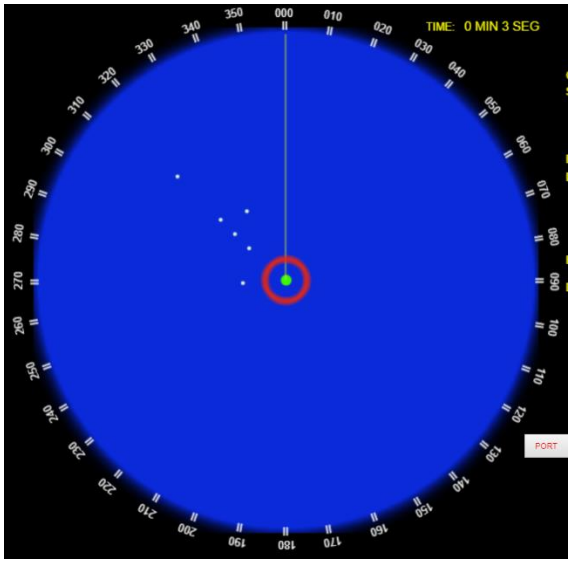


Figura A.14 - Imagem teste 14

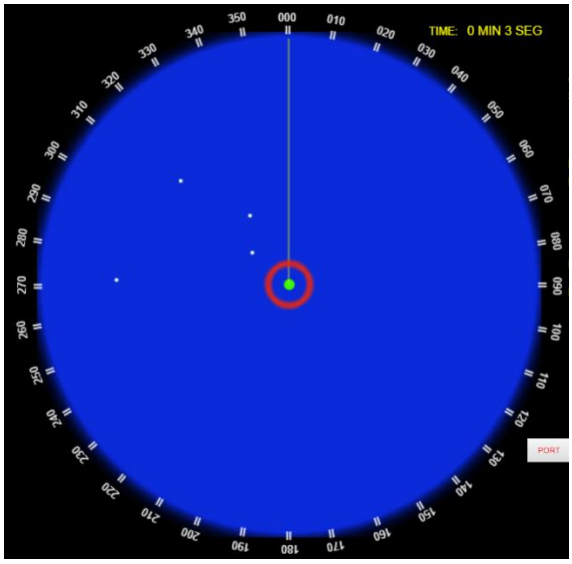


Figura A. 15 - Imagem teste 15

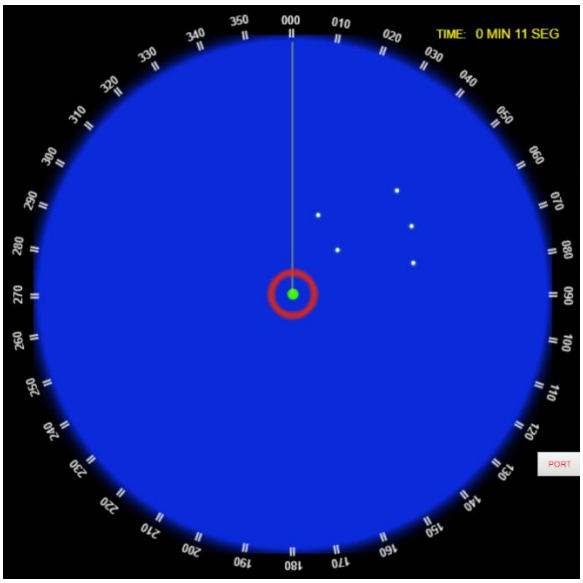


Figura A. 16 - Imagem teste 16

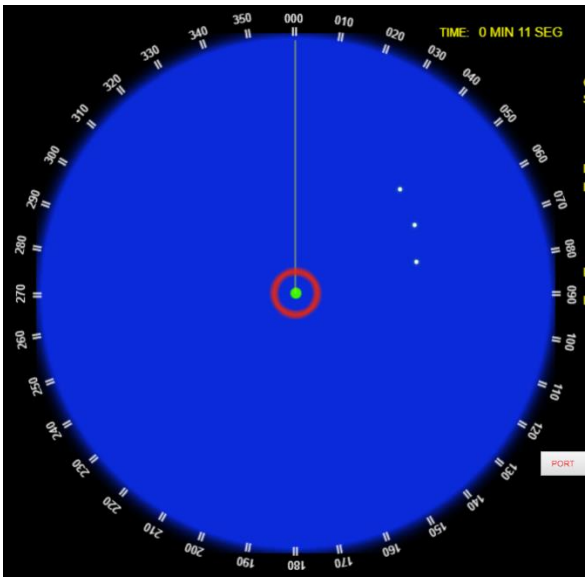


Figura A. 17 - Imagem teste 17

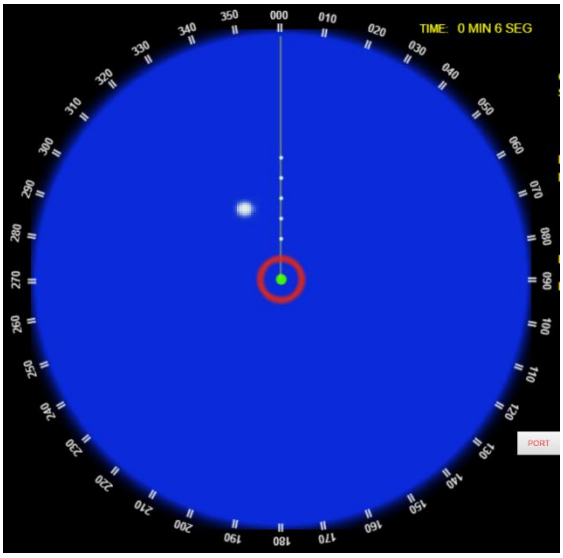


Figura A. 18 - Imagem teste 18

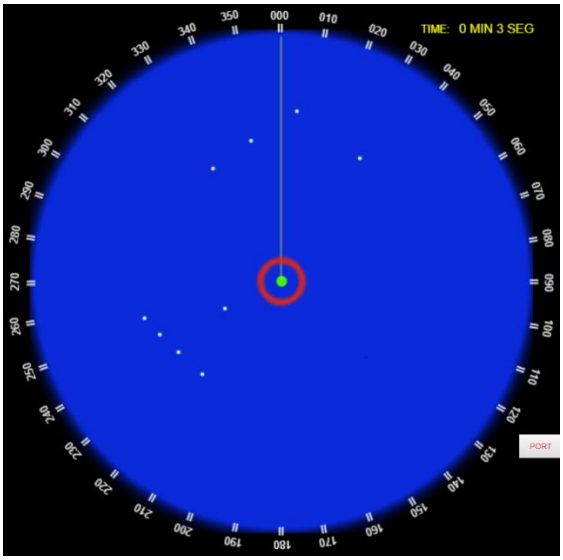


Figura A. 19 - Imagem teste 19

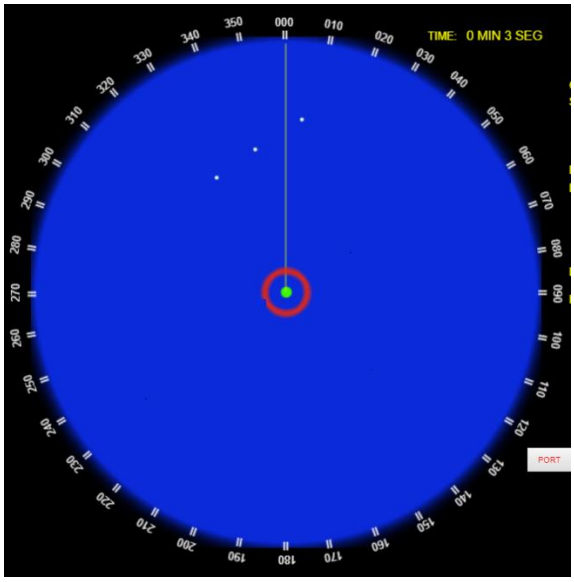


Figura A. 20 - Imagem teste 20

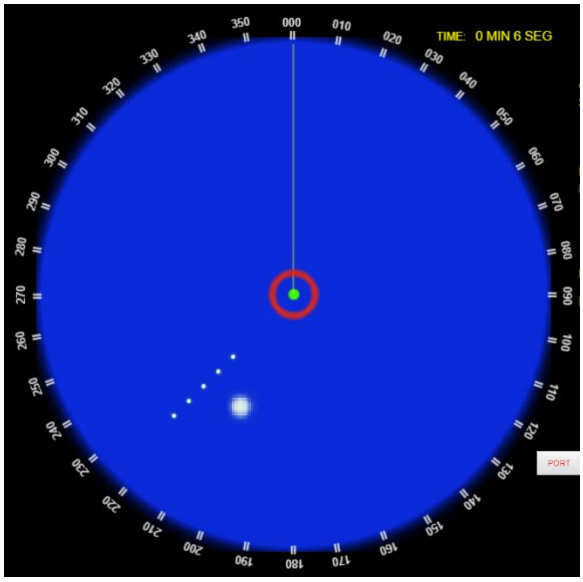


Figura A. 21 - Imagem teste 21

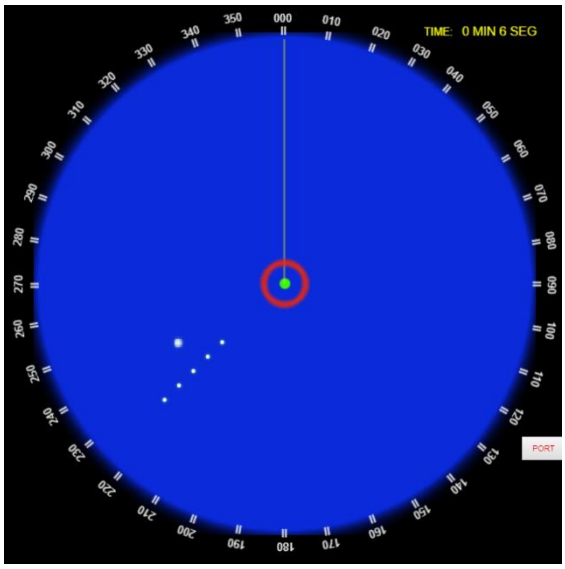


Figura A. 22 - Imagem teste 22

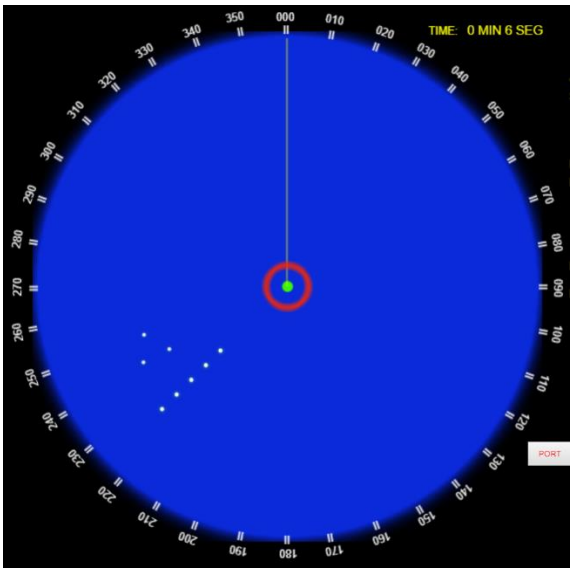


Figura A. 23 - Imagem teste 23

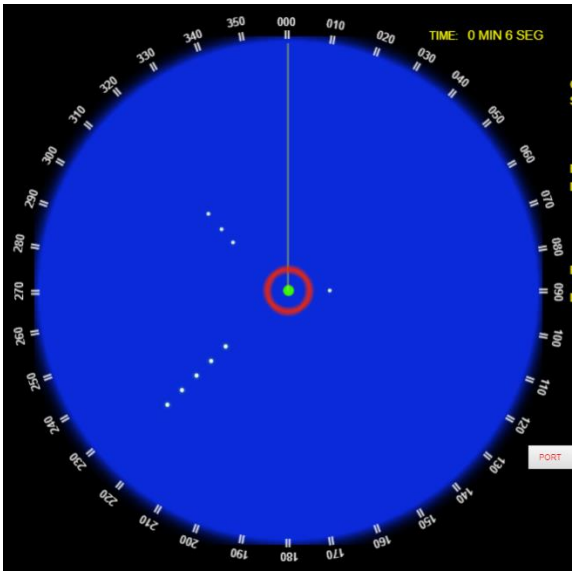


Figura A. 24 - Imagem teste 24

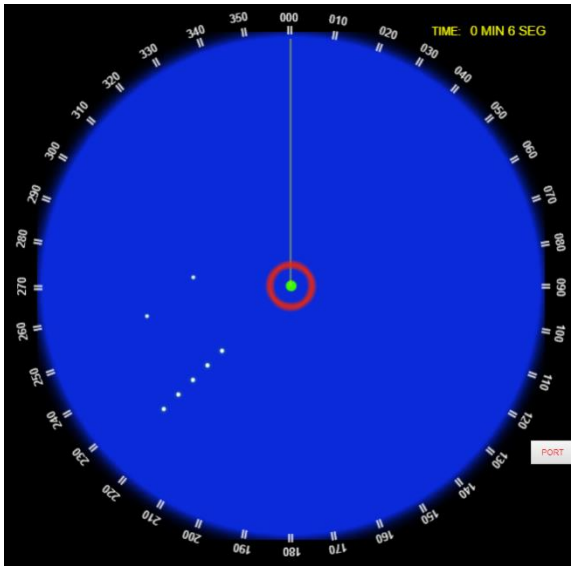


Figura A. 25 - Imagem teste 25

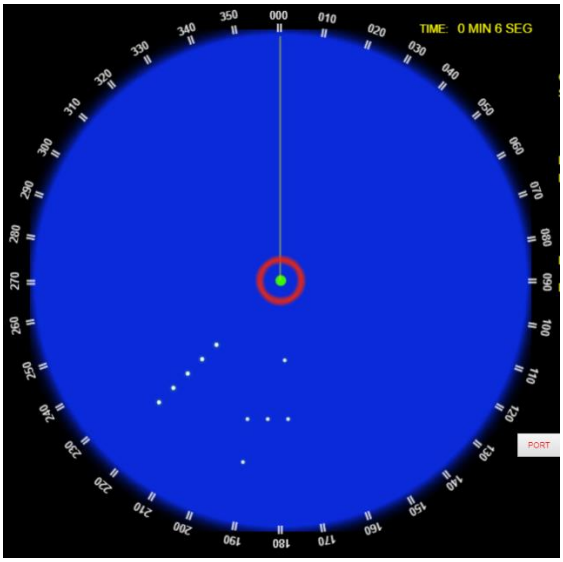


Figura A. 26 - Imagem teste 26

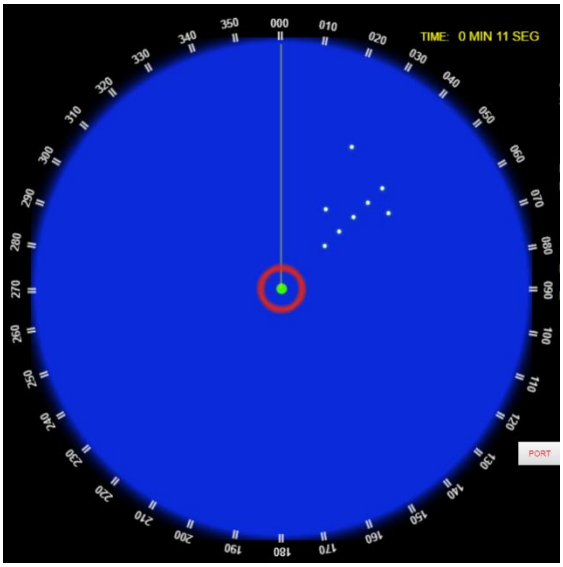


Figura A. 27 - Imagem teste 27

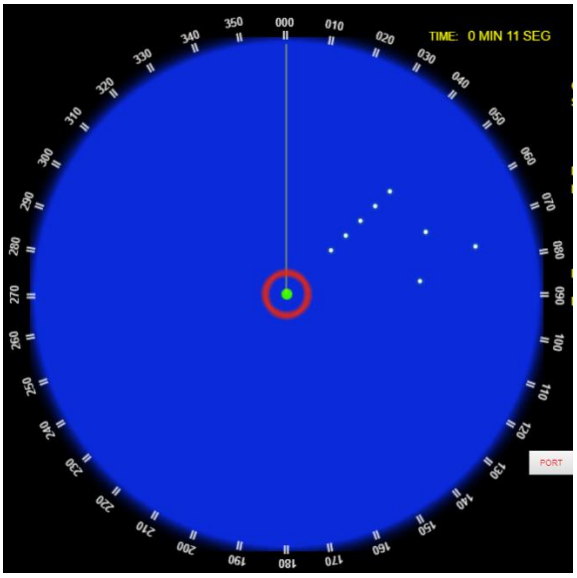


Figura A. 28 - Imagem teste 28

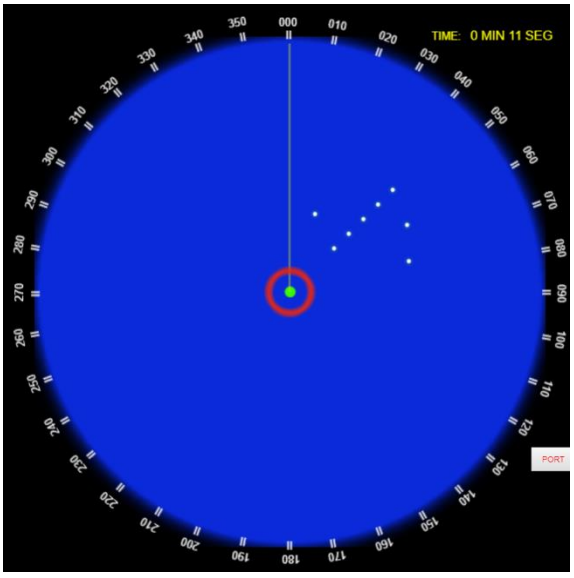


Figura A. 29 - Imagem teste 29

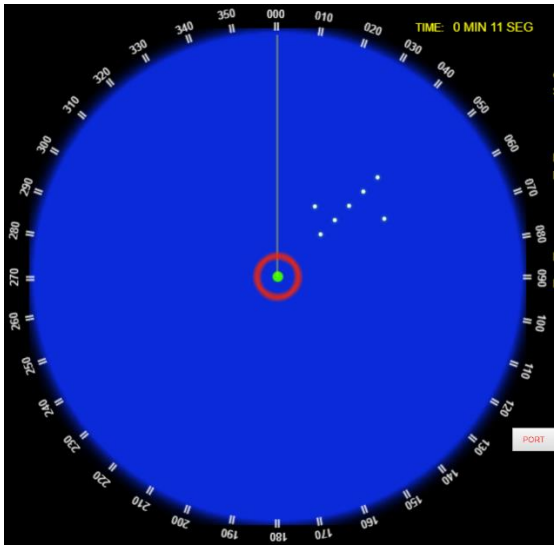
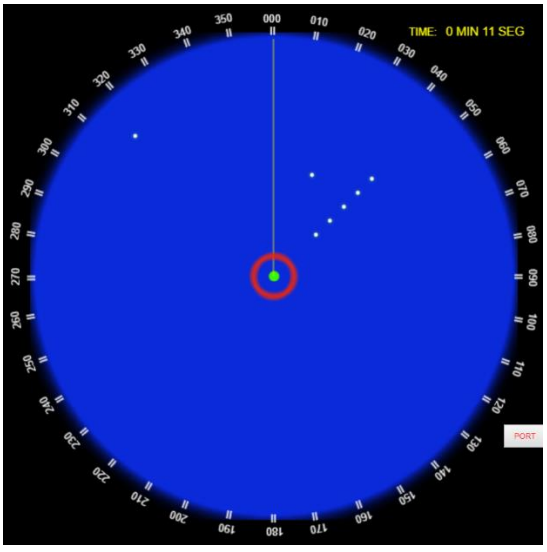


Figura A. 30 - Imagem teste 30



APÊNDICE B – Algoritmo utilizado no experimento em Python

```
import numpy as np
import cv2
from PIL import Image


for i in range(179):

 print (i)

 colorImage = Image.open('template.png')

 rotated = colorImage.rotate(i)

 rotated.save('template 2.png')

 img_bgr = cv2.imread('Imagem teste 29.png')
 img_gray = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY)

 template = cv2.imread('template 2.png',0)
 w, h = template.shape[::-1]

 res = cv2.matchTemplate(img_gray,template,cv2.TM_CCOEFF_NORMED)
 threshold = 0.5
 loc = np.where(res >= threshold)

for pt in zip(*loc[::-1]):
    cv2.rectangle(img_bgr, pt, (pt[0]+w, pt[1]+h), (0,255,255), 2)
    cv2.imshow('detected', img_bgr)
```

APÊNDICE C – Resultados obtidos no experimento em Python

Threshold = 0.3				
Teste	Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo
1	X			
2	X			
3	X			
4				X
5	X			
6			X	
7		X		
8		X		
9		X		
10		X		
11		X		
12		X		
13		X		
14		X		
15		X		
16		X		
17				X
18		X		
19		X		
20	X			
21	X			
22	X			
23	X			
24	X			
25	X			
26	X			
27	X			
28	X			
29	X			
30	X			
Total	15	12	1	2

Threshold = 0.4				
Teste	Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo
1	X			
2	X			
3	X			
4				X
5				X
6			X	
7			X	
8			X	
9			X	
10			X	
11			X	
12			X	
13			X	
14			X	
15		X		
16		X		
17				X
18			X	
19			X	
20	X			
21	X			
22	X			
23	X			
24	X			
25	X			
26	X			
27	X			
28	X			
29	X			
30	X			
Total	14	2	11	3

Threshold = 0.5				
Teste	Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo
1	X			
2	X			
3	X			
4				X
5				X
6			X	
7			X	
8			X	
9			X	
10			X	
11			X	
12			X	
13			X	
14			X	
15			X	
16			X	
17				X
18			X	
19			X	
20	X			
21	X			
22	X			
23	X			
24	X			
25	X			
26	X			
27	X			
28	X			
29	X			
30	X			
Total	14	0	13	3

Threshold = 0.6				
Teste	Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo
1	X			
2	X			
3	X			
4				X
5				X
6			X	
7			X	
8			X	
9			X	
10			X	
11			X	
12			X	
13			X	
14			X	
15			X	
16			X	
17				X
18			X	
19			X	
20				X
21	X			
22	X			
23	X			
24	X			
25	X			
26	X			
27	X			
28	X			
29	X			
30	X			
Total	13	0	13	4

Threshold = 0.7				
Teste	Verdadeiro-Positivo	Falso-Positivo	Verdadeiro-Negativo	Falso-Negativo
1	X			
2	X			
3	X			
4				X
5				X
6			X	
7			X	
8			X	
9			X	
10			X	
11			X	
12			X	
13			X	
14			X	
15			X	
16			X	
17				X
18			X	
19			X	
20				X
21				X
22				X
23	X			
24	X			
25	X			
26				X
27	X			
28				X
29				X
30	X			
Total	8	0	13	9

APÊNDICE D – Algoritmo do *software* MATLAB adaptado

```
%% Stream and Accelerate Simulation of Radar System
% Phased Array System Toolbox can be used to model an end-to-end radar
% system - generate a transmitted waveform, simulate the target return, and
% then process the received signal to detect the target. This is shown in
% the examples: <docid:phased_examples.example-ex97528254> and
% <docid:phased_examples.example-ex12077916>. This example shows how to
% simulate such a system in streaming mode so you can run the simulation
% for a long time and observe the system dynamics.

%   Copyright 2013-2017 The MathWorks, Inc.

%% Simulation Setup
% First, set up the radar system with some basic parameters. The entire
% radar system is similar to the one shown in the
% <docid:phased_examples.example-ex12077916> example.

clc;
clear all;
close all;

fs = 6e6;
bw = 3e6;
c = 3e8;
fc = 10e9;
prf = 18750;
num_pulse_int = 10;

[waveform,transmitter,radiator,collector,receiver,sensormotion,...
    target,tgtmotion,channel,matchedfilter,tvg,threshold] = ...
    helperRadarStreamExampleSystemSetup(fs,bw,prf,fc,c);
```

```
%% System Simulation
% Next, run the simulation for 100 pulses. During this simulation, four
% time-scopes are used to observe the signals at various stages. The first
% three scopes display the transmitted signal, received signal, and the
% post-matched-filter and gain-adjusted signal for 10-pulses. Although the
% transmitted signal is a high-power pulse train, scope 2 shows a much
% weaker received signal due to propagation loss. This signal cannot be
% detected using the preset detection threshold. Even after
% matched-filtering and gain compensation, it is still challenging to
% detect all three targets.

% pre-allocation
fast_time_grid = 0:1/fs:1/prf-1/fs;
num_pulse_samples = numel(fast_time_grid);
rx_pulses = complex(zeros(num_pulse_samples,num_pulse_int));
mf_pulses = complex(zeros(num_pulse_samples,num_pulse_int));
detect_pulse = zeros(num_pulse_samples,1);

% simulation loop
for m = 1:10*num_pulse_int
    % Update sensor and target positions
    [sensorpos,sensorvel] = sensormotion(1/prf);
    [tgtpos,tgtvel] = tgtmotion(1/prf);

    % Calculate the target angles as seen by the sensor
    [tgtrng,tgtang] = rangeangle(tgtpos,sensorpos);

    % Simulate propagation of pulse in direction of targets
    pulse = waveform();
    [pulse,txstatus] = transmitter(pulse);
    txsig = radiator(pulse,tgtang);
    txsig = channel(txsig,sensorpos,tgtpos,sensorvel,tgtvel);

    % Reflect pulse off of targets
```

```
    tgtsig = target(txsig);

    % Receive target returns at sensor
    rxsig = collector(tgtsig,tgtang);

    % SINAL ATAQUE
    tt = length(rxsig)
    AtSig = zeros(tt,1);
    if mod(m, num_pulse_int)==1
        AtSig(150:200) = max(real(rxsig));
    end
    rxsig = rxsig+AtSig;

    nn = mod(m-1,num_pulse_int)+1;
    rx_pulses(:,nn) = receiver(rxsig,~(txstatus>0));

    % Detection processing
    mf_pulses(:,nn) = matchedfilter(rx_pulses(:,nn));
    mf_pulses(:,nn) = tvg(mf_pulses(:,nn));

    % Perform pulse integration every 'num_pulse_int' pulses
    if nn == num_pulse_int
        detect_pulse = pulsint(mf_pulses,'noncoherent');
    end

    helperRadarStreamDisplay(pulse,abs(rx_pulses(:,nn)),...
        abs(mf_pulses(:,nn)),detect_pulse,...
        sqrt(threshold)*ones(num_pulse_samples,1));
end
```

ANEXO A – Algoritmo disponível no *software* MATLAB

```
%% Stream and Accelerate Simulation of Radar System
% Phased Array System Toolbox can be used to model an end-to-end radar
% system - generate a transmitted waveform, simulate the target return, and
% then process the received signal to detect the target. This is shown in
% the examples: <docid:phased_examples.example-ex97528254> and
% <docid:phased_examples.example-ex12077916>. This example shows how to
% simulate such a system in streaming mode so you can run the simulation
% for a long time and observe the system dynamics.

%   Copyright 2013-2017 The MathWorks, Inc.

%% Simulation Setup
% First, set up the radar system with some basic parameters. The entire
% radar system is similar to the one shown in the
% <docid:phased_examples.example-ex12077916> example.

fs = 6e6;
bw = 3e6;
c = 3e8;
fc = 10e9;
prf = 18750;
num_pulse_int = 10;

[waveform,transmitter,radiator,collector,receiver,sensormotion,...
    target,tgtmotion,channel,matchedfilter,tvg,threshold] = ...
    helperRadarStreamExampleSystemSetup(fs,bw,prf,fc,c);

%% System Simulation
% Next, run the simulation for 100 pulses. During this simulation, four
% time-scopes are used to observe the signals at various stages. The first
% three scopes display the transmitted signal, received signal, and the
% post-matched-filter and gain-adjusted signal for 10-pulses. Although the
```

```
% transmitted signal is a high-power pulse train, scope 2 shows a much
% weaker received signal due to propagation loss. This signal cannot be
% detected using the preset detection threshold. Even after
% matched-filtering and gain compensation, it is still challenging to
% detect all three targets.

% pre-allocation
fast_time_grid = 0:1/fs:1/prf-1/fs;
num_pulse_samples = numel(fast_time_grid);
rx_pulses = complex(zeros(num_pulse_samples,num_pulse_int));
mf_pulses = complex(zeros(num_pulse_samples,num_pulse_int));
detect_pulse = zeros(num_pulse_samples,1);

% simulation loop
for m = 1:10*num_pulse_int
    % Update sensor and target positions
    [sensorpos,sensorvel] = sensormotion(1/prf);
    [tgtpos,tgtvel] = tgtmotion(1/prf);

    % Calculate the target angles as seen by the sensor
    [tgtrng,tgtang] = rangeangle(tgtpos,sensorpos);

    % Simulate propagation of pulse in direction of targets
    pulse = waveform();
    [pulse,txstatus] = transmitter(pulse);
    txsig = radiator(pulse,tgtang);
    txsig = channel(txsig,sensorpos,tgtpos,sensorvel,tgtvel);

    % Reflect pulse off of targets
    tgtsig = target(txsig);

    t3 = 0:1/1e3:60;
    d3 = [sin(2*pi*10e9*(0:2:60))]';
    x3 = @rectpuls;
```

```
    y3 = pulstran(t3,d3,x3);
    m2 = imresize(y3, [320,1]);

    % Receive target returns at sensor
    rxsig = collector(tgtsig,tgtang)+m2;
    nn = mod(m-1,num_pulse_int)+1;
    rx_pulses(:,nn) = receiver(rxsig,~(txstatus>0));

    % Detection processing
    mf_pulses(:,nn) = matchedfilter(rx_pulses(:,nn));
    mf_pulses(:,nn) = tvg(mf_pulses(:,nn));

    % Perform pulse integration every 'num_pulse_int' pulses
    if nn == num_pulse_int
        detect_pulse = pulsint(mf_pulses,'noncoherent');
    end

    helperRadarStreamDisplay(pulse,abs(rx_pulses(:,nn)),...
        abs(mf_pulses(:,nn)),detect_pulse,...
        sqrt(threshold)*ones(num_pulse_samples,1));
end

%% Improve Simulation Speed Using Code Generation
% Because radar systems require intensive processing, simulation speed is a
% major concern. After you have run 100 pulse to check out your code, you
% may want to run 1000 pulses. When you run the simulation in interpreted
% MATLAB mode, you can measure the elapsed time using:

tic;
helperRadarStreamRun;
time_interpreted = toc

%%
% If the simulation is too slow, you can speed it up using MATLAB
```

```
% Coder. MATLAB Coder can generate compiled MATLAB code resulting in
% significant improvement in processing speed. In this example, MATLAB
% Coder generates a helperRadarStreamRun_mex function from the
% helperRadarStreamRun function. The command used is shown below:
%
%  codegen helperRadarStreamRun.m
%
% When the mex version is invoked, the simulation speed is improved.

tic;
helperRadarStreamRun_mex;
time_compiled = toc

%%
% Speedup improvement depends on several factors such as machine CPU speed
% and available memory but is typically increased 3-4 times. Note that the
% visualization of data using scopes is not sped up by MATLAB Coder and
% is still handled by the MATLAB interpreter. If visualizations are not
% critical to your simulation, then you can remove them for further speed
% improvement.
%
% Below are several trade-offs to consider when adopting this approach:
%
% # The visualization capability in generated code is very limited compared
% to what is available in MATLAB. If you need to keep the visualization in
% the simulation, use the |coder.extrinsic| trick; but this slows down the
% simulation.
% # The generated code does not allow dynamic changes of variable type and
% size compared to the original MATLAB code. The generated code is often
% optimized for a particular variable type and size; therefore, any change
% in a variable type and size, which can be caused, for example, by a
% change in PRF, requires a recompile.
% # The simulation speed benefit becomes more important when the MATLAB
% simulation time is long. If MATLAB simulation finishes in a few seconds,
```

```
% you do not gain much by generating the code from the original MATLAB
% simulation. As mentioned in the previous bullet, it is often necessary to
% recompile the code when parameters change. Therefore, it might be better
% to first use MATLAB simulation to identify appropriate parameter values
% and then use generated code to run long simulations.

%% Summary
% This example shows how to perform the radar system simulation in
% streaming mode. It also shows how the code generation can be used to
% speed up the simulation. The tradeoff between using the generated code
% and MATLAB code is discussed at the end.
```