Ataques Furtivos em Sistemas de Controle Físicos Cibernéticos

Alan Oliveira de Sá[1,2], **Luiz F. Rust da Costa Carmo**[1,3], **Raphael C. S. Machado**[3]

[1]Programa de Pós-Graduação em Informática - Instituto Tércio Pacitti / IM,
Universidade Federal do Rio de Janeiro, 21.941-901, RJ – Brasil

[2]Centro de Instrução Almirante Wandenkolk – Marinha do Brasil,
Ilha das Enxadas, Baía de Guanabara – Rio de Janeiro – RJ – Brasil

[3]Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro)
Av. Nossa Senhora das Graças, 50, Xerém, Duque de Caxias, 25.250-020, RJ – Brasil

alan.oliveira.sa@gmail.com, {lfrust,rcmachado}@inmetro.gov.br

Abstract. *The advantages of using communication networks to interconnect controllers and physical plants motivate the increasing number of Networked Control Systems, in industrial and critical infrastructure facilities. However, this integration also exposes such control systems to new threats, typical of the cyber domain. In this context, studies have been conduced, aiming to explore vulnerabilities and propose security solutions for cyber-physical systems. In this paper, it is proposed a covert attack for system degradation, which is planned based on the intelligence gathered by another attack, herein proposed, referred as System Identification attack. The simulation results demonstrate that the joint operation of the two attacks is capable to affect, in a covert and accurate way, the physical behavior of a system.*

Resumo. *As vantagens do uso de redes de comunicação para interconectar controladores e plantas físicas tem motivado o crescente número de Sistemas de Controle em Rede, na indústria e em infraestruturas críticas. Entretanto, esta integração expõe tais sistemas a novas ameaças, típicas do domínio cibernético. Neste contexto, estudos têm sido realizados com o objetivo de explorar as vulnerabilidades e propor soluções de segurança para sistemas físico-cibernéticos. Neste artigo é proposto um ataque furtivo de degradação de serviço o qual é planejado com base nos dados colhidos por um outro ataque, ora proposto, denominado de System Identification. Os resultados de simulação demonstram que a operação conjunta dos dois ataques é capaz de afetar de forma furtiva e acurada o comportamento físico de um sistema.*

1. Introdução

A integração de sistemas usados para controlar processos físicos por meio de redes de comunicação visa atribuir a tais sistemas melhores capacidades operacionais e gerenciais, bem como reduzir custos. Em face destas vantagens, existe a tendência de um crescente número de processos industriais e sistemas de infraestruturas críticas controlados por Sistemas de Controle em Rede, ou *Networked Control Systems* (NCS) [Farooqui et al. 2014], também referidos como *Network-Based Control Systems* (NBCS) [Long et al. 2005]. Um NCS, conforme apresentado na Figura 1, consiste de uma planta física, descrita por uma função de transferência $G(z)$, um controlador, o qual executa uma função de controle

$C(z)$, e uma rede de comunicação que interconecta ambos os dispositivos para a transmissão de sinais de controle e de realimentação. Os sinais de controle são transmitidos do controlador para os atuadores da planta. Os sinais de realimentação são transmitidos dos sensores da planta para o controlador.

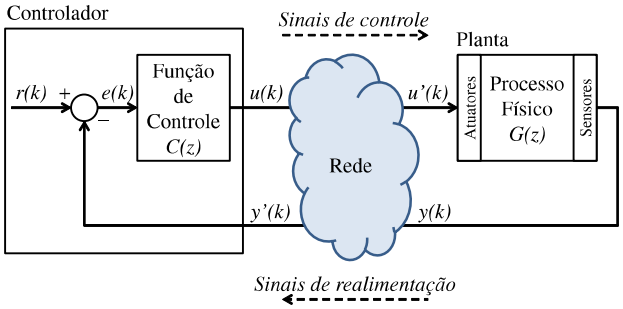


Figura 1. Sistema de Controle em Rede (ou NCS)

Ao mesmo tempo em que traz uma série de vantagens, a integração de controladores e plantas físicas em malha fechada por meio de redes de comunicação também expõe tais sistemas a novas ameaças, típicas do domínio cibernético. Neste contexto, estudos vêm sendo realizados, com o objetivo de caracterizar vulnerabilidades e propor soluções de segurança em NCSs.

Uma possível forma de atacar um NCS se dá pela intervenção em seu *software*, *i.e.* por meio de alterações na configuração ou mesmo no código executado pelo controlador, seguindo estratégia similar àquela utilizada pelo *worm* Stuxnet [Langner 2011]. Outra maneira possível para um atacante afetar um NCS é por meio de interferências no seu processo de comunicação. Basicamente, um atacante pode interferir nos sinais de controle e/ou de realimentação de três diferentes modos: induzindo *jitter* (atrasos variáveis), causando a perda de pacotes de dados, ou mesmo injetando dados falsos na comunicação.

No presente trabalho, desenvolvemos um ataque onde são injetados dados falsos no processo de comunicação de um NCS, demonstrando a possibilidade de degradação do serviço realizado por uma planta por meio de alterações sutis em seu comportamento físico. Esta intervenção tem por objetivo reduzir a eficiência da planta ou mesmo lhe causar danos em médio/longo prazo. Cabe ressaltar que uma intervenção descontrolada no NCS pode levar a uma avaria imediata da planta, ou mesmo causar alterações de grande proporção em seu funcionamento, o que pode resultar na descoberta do ataque e no eventual insucesso da operação. Sendo assim, as alterações impelidas pelo ataque ora proposto são dimensionadas para que a mudança de comportamento da planta seja fisicamente de difícil percepção, motivo pelo qual classificamos o ataque como fisicamente furtivo.

Para garantir que o ataque a um NCS seja fisicamente furtivo, o atacante deve planejar sua ofensiva com base em um conhecimento acurado sobre a dinâmica do sistema, caso contrário, as consequências do ataque podem ser imprevisíveis. Uma forma de adquirir tal conhecimento é por meio de operações de inteligência convencionais, desempenhadas para colher informações sobre o projeto e a dinâmica do NCS. Outra forma de obter informações sobre o sistema a ser atacado é por meio de o que classificamos neste trabalho como ataques de *Cyber-Physical Intelligence*. Neste sentido, também propomos no presente artigo um ataque de identificação de sistemas, ou *System Identification*, que visa obter informações sobre a função de transferência $G(z)$ da planta e da função de controle $C(z)$ do controlador. Este ataque é baseado no Algoritmo de Busca por Retro-

cesso, ou *Backtracking Search Optimization algorithm* (BSA) [Civicioglu 2013]. Note que, tanto o ataque de degradação de serviço por meio da injeção de dados, quanto o ataque *System Identification* requerem acesso aos sinais transmitidos no NCS, o que pode ser dificultado por técnicas de criptografia. Entretanto, não se pode negligenciar a possibilidade de acesso a tais dados por meio de ataques de criptoanálise ou mesmo de força bruta.

O presente trabalho motivou a formalização de uma série de conceitos relacionados a furtividade e inteligência no contexto da segurança físico-cibernética. Sendo assim, uma contribuição complementar do artigo é a proposição de uma nomenclatura que abarque toda uma nova classe de ataques aos sistemas físico-cibernéticos. A taxonomia proposta estabelece uma nova abordagem quanto à furtividade de ataques a sistemas físico-cibernéticos, os quais devem ser analisados sob dois aspectos simultaneamente: o aspecto físico e o aspecto cibernético.

É digno de nota que o objetivo deste trabalho não é facilitar ataques furtivos de degradação de serviço em sistemas de controle físico-cibernéticos. O objetivo deste trabalho é demonstrar o grau de acurácia que pode ser obtido neste tipo de ataque, sobretudo quando apoiado por ataques de *System Identification*, e, portanto, encorajar a pesquisa de contramedidas para tais ataques. O restante do artigo é organizado da seguinte forma: Primeiramente, na Seção 2, são apresentados alguns trabalhos relacionados. Em seguida, na Seção 3, é proposta uma taxonomia referente aos ataques físico-cibernéticos em malhas de controle de NCSs. Na Seção 4, é feita a descrição de um ataque do tipo *System Identification*. Na Seção 5, é definido um ataque furtivo de degradação de serviço. Na Seção 6, são apresentados os resultados obtidos em simulações de ataques furtivos de degradação de serviço, apoiados por ataques *System Identification*. Finalmente, na Seção 7, são apresentadas algumas conclusões e possibilidades de trabalhos futuros.

2. Trabalhos Relacionados

A possibilidade de ataques físico-cibernéticos se tornou uma realidade após o lançamento do *worm* Stuxnet [Langner 2011] e tem motivado pesquisas concernentes à segurança de NCSs. Nesta seção são apresentados alguns trabalhos relacionados ao assunto.

Em [Long et al. 2005] os autores propõem dois modelos de fila para avaliar o impacto do *jitter* e da perda de pacotes em um NCS sob ataque. O ataque não é planejado com base em um conhecimento prévio sobre os modelos do controlador e da planta. Sendo assim, para afetar o comportamento físico dos sistema, o atacante inunda a rede com um tráfego adicional, causando *jitter* e perda de pacotes de forma arbitrária. Nesta tática, o excesso de pacotes na rede pode reduzir a furtividade do ataque, permitindo a adoção de contramedidas tais como a filtragem de pacotes ou o bloqueio do tráfego malicioso na sua origem [Long et al. 2005]. Adicionalmente, a ação arbitrária sobre um modelo desconhecido pode levar o sistema a comportamentos físicos extremos, o que não é desejável se for almejado um ataque furtivo.

Em [Farooqui et al. 2014], os autores apresentam uma plataforma de testes para sistemas SCADA (*Supervisory Control and Data Acquisition*). Os mesmos demonstram um ataque onde são enviados dados falsos para o controlador e para o atuador do NCS. No artigo, os dados falsos injetados durante a comunicação têm valores randômicos e visam fazer com que um motor DC perca a sua estabilidade. Este tipo de ataque não demanda

um conhecimento prévio sobre o NCS. Em contrapartida, o efeito físico desejado e a furtividade não podem ser garantidos em virtude das consequências imprevisíveis que podem surgir da aplicação de sinais aleatórios em um sistema cujo modelo é desconhecido.

Mais recentemente, em [Teixeira et al. 2015], os autores fornecem um quadro geral contendo a análise de uma grande variedade de métodos de ataque em NCSs. Em sua classificação, os mesmos estabelecem que ataques furtivos em NCSs requerem um alto nível de conhecimento sobre o sistema atacado. Exemplos de ataques furtivos são apresentados em [Smith 2011, Smith 2015]. Nestes trabalhos os ataques são desempenhados por um *man-in-the-middle* (MitM), onde o atacante necessita injetar dados tanto no enlace de controle quanto no de realimentação, bem como conhecer o modelo da planta que está sendo controlada. A furtividade destes ataques, que depende da diferença entre o modelo real da planta e o modelo utilizado pelo atacante, é analisada do ponto de vista dos sinais que chegam para o controlador, sem abordar se os efeitos físicos causados na planta são perceptíveis, ou se são furtivos perante um observador humano.

Nos trabalhos [Teixeira et al. 2015, Smith 2011, Smith 2015], onde é requerido um conhecimento sobre o modelo do NCS atacado, não é descrito como este este conhecimento é obtido pelo atacante. Considera-se apenas que o modelo é previamente conhecido para subsidiar o planejamento do ataque. A ação conjunta, ora proposta, de um ataque furtivo de degradação de serviço, apoiado por um ataque *System Identification*, visa preencher este hiato, demonstrando como os dados do NCS podem ser obtidos e como um ataque furtivo pode se beneficiar disto. A Tabela 1 apresenta uma síntese das características dos ataques apresentados nesta seção.

Tabela 1. Síntese dos ataques mencionados

Ataque	Método de ataque	Conhecimento sobre o modelo	Como o modelo é obtido
Long, *et al.* [Long et al. 2005]	Indução de *jitter* e perda de pacotes	Nenhum	N/A
Stuxnet *worm* [Langner 2011]	Modificações no código do PLC	Sim	Experimentos em um sistema real
Farooqui, *et al.* [Farooqui et al. 2014]	Injeção de dados	Nenhum	N/A
Smith [Smith 2011, Smith 2015]	Injeção de dados	Sim	Não descrito
Teixeira [Teixeira et al. 2015]	Perda de pacotes	Nenhum	N/A
	Injeção de dados	Sim	Não descrito

3. Taxonomia

Nesta Seção é apresentada uma taxonomia relativa aos possíveis ataques a sistemas de controle físico-cibernéticos. Na Seção 3.1, os ataques são brevemente descritos e classificados de acordo com a forma como agem no NCS. Na Seção 3.2, é proposta uma nova abordagem para a análise da furtividade de ataques à sistemas físico-cibernéticos.

3.1. Classificação dos ataques

Ataques a sistemas físico-cibernéticos podem atuar tanto nos seus dispositivos – *i.e.* no controlador, atuadores e sensores da planta – quanto em seus sistemas de comunicação, afetando os sinais de controle e de realimentação. Como premissa, devemos considerar que o *serviço* que se pretende atacar/proteger em tal sistema é o trabalho executado pelo processo físico, controlado por um NCS.

Considerando a definição supracitada de serviço em NCSs, os ataques podem ser classificados em três categorias distintas, como apresentado na Figura 2:

- *Denial-of-Service* (DoS) [Hussain et al. 2003]: em um NCS, os ataques DoS compreendem todos os tipos de ataques físico-cibernéticos que neguem a operação do processo físico, interrompendo a execução do serviço que a planta controlada se propõe a fazer. O ataque resulta, por exemplo, em comportamentos que podem desligar a planta ou mesmo destruí-la em um curto prazo.
- *Service Degradation* (SD): os ataques do tipo SD consistem em intervenções maliciosas que são executadas na malha de controle visando reduzir a eficiência do serviço, *i.e.* a eficiência do processo físico, ou mesmo reduzir o tempo médio entre falhas, ou *mean time between failure* (MTBF), da planta em médio/longo prazo.
- *Cyber-physical Intelligence* (CPI): o conceito de *Cyber-physical Intelligence*, aqui proposto, é diferente do conceito onde sistemas físico-cibernéticos são integrados com sistemas inteligentes [Ramos et al. 2011]. Na presente taxonomia, os ataques do tipo CPI compreendem as ações que são desempenhadas na malha de controle do NCS com o objetivo de colher informações sobre a operação do sistema e/ou sobre o seu projeto. Estes ataques têm por objetivo adquirir as informações necessárias para o planejamento de ataques furtivos e controlados, ou mesmo para subsidiar ações de *replay* [Langner 2011].

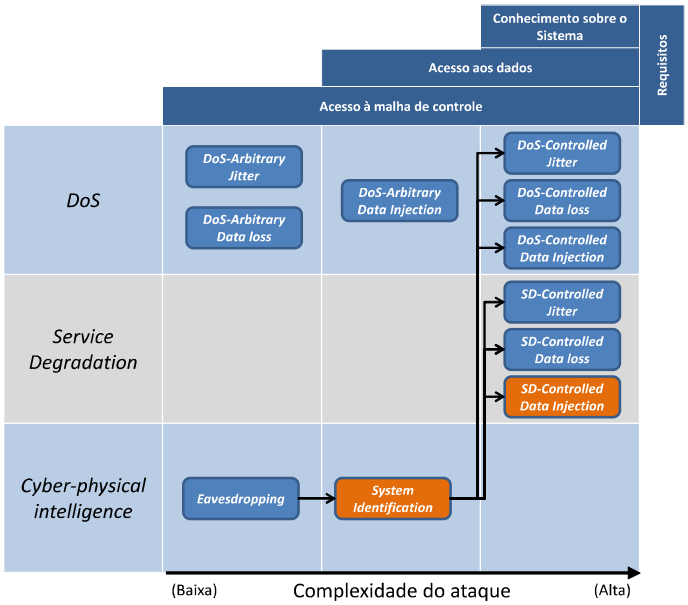


Figura 2. Classificação e requisitos dos ataques físico-cibernéticos atuantes na malha de controle de um NCS.

Na Figura 2, são apresentados seis tipos de ataques DoS, bem como os seus respectivos requisitos. Destes seis tipos de ataque, os menos complexos são os arbitrários:

- *DoS-Arbitrary Jitter*: neste tipo de ataque, o atraso dos sinais de controle e /ou realimentação é alterado arbitrariamente, sem um conhecimento prévio do modelo do NCS, com o objetivo de levar o sistema a uma instabilidade ou a uma condição que cause a interrupção do processo físico. Este ataque requer somente o acesso à malha de controle, uma vez que o mesmo pode se dar pelo simples consumo de recursos do sistema, tal como a banda dos enlaces de comunicação, ou mesmo recursos computacionais dos equipamentos que fazem parte da malha de controle.

- *DoS-Arbitrary Data Loss*: neste tipo de ataque, o atacante impede que os dados cheguem aos atuadores e/ou controladores. O atacante efetua um *jamming* arbitrário nos sinais de comunicação, sem um conhecimento prévio do modelo do NCS, levando o sistema à instabilidade ou a uma condição que cause a interrupção do processo físico. Cabe ressaltar que alguns ataques do tipo *DoS-Arbitrary Jitter* podem evoluir para um ataque *DoS-Arbitrary Data Loss*, caso atrasos de maior proporção venham a causar a perda de pacotes. Assim como no ataque *DoS-Arbitrary Jitter*, este ataque só requer o acesso à malha de controle do NCS.
- *DoS-Arbitrary Data Injection*: nestes ataques, o atacante envia dados falsos e arbitrários ao controlador, como se estes tivessem sido enviados pelos sensores, e/ou para os atuadores, como se tivessem sido enviados pelo controlador. Os dados são injetados na malha de controle do NCS sem o conhecimento prévio de seu modelo. Este ataque é mais complexo que os ataques *DoS-Arbitrary Jitter* e *DoS-Arbitrary Data Loss*, uma vez que requer o acesso aos dados que fluem na malha de controle do NCS.

Os ataques do tipo *DoS-Controlled* – *DoS-Controlled Jitter*, *DoS-Controlled Data Loss* e *DoS-Controlled Data Injection* – apresentados na Figura 2, interferem na malha de controle do NCS da mesma forma que seus respectivos ataques *DoS-Arbitrary*. A diferença entre um ataque *DoS-Controlled* e um ataque *DoS-Arbitrary* é que, no primeiro, a interferência causada pelo atacante é precisamente planejada e executada, visando alcançar com exatidão o comportamento desejado que leva o sistema à interrupção do serviço físico, de uma forma mais eficiente. Assim, para alcançar tal eficiência, um ataque *DoS-Controlled* requer um conhecimento acurado do modelo do NCS, *i.e.* das funções de transferência da planta e do controlador, as quais devem ser analisadas para o planejamento do ataque.

Referente aos ataques SD, devemos considerar três diferentes tipos de ataque – *SD-Controlled Jitter*, *SD-Controlled Data Loss* e *SD-Controlled Data Injection* – conforme apresentado na Figura 2. A diferença entre um ataque *SD-Controlled* e um ataque *DoS-Controlled* é que o primeiro não tem a intenção de interromper o processo físico em um curto prazo. O ataque visa manter o processo funcionando com a eficiência reduzida ou, por vezes, causar a deterioração física e gradual dos dispositivos controlados. Para que isto ocorra, os ataques *SD-Controlled* requerem um conhecimento prévio e acurado sobre o NCS. Caso contrário o ataque pode, por razões não previstas, evoluir para um ataque DoS, causando a interrupção do processo físico.

O conhecimento sobre o sistema, requerido tanto nos ataques *DoS-Controlled* e *SD-Controlled*, pode ser obtido por meio de ataques CPI, conforme apresentado na Figura 2. O primeiro, e mais simples, ataque CPI é o *eavesdropping* [Khatri et al. 2015], que consiste em simplesmente capturar os sinais de controle e de realimentação transmitidos. O segundo ataque CPI, proposto neste artigo, é o *System Identification*, o qual visa obter informações sobre a função de transferência da planta e a função de controle do controlador por meio da análise dos sinais que trafegam na rede. Os ataques CPI por si só não impactam no funcionamento do NCS, mas são uma poderosa ferramenta para planejar ataques *DoS-Controlled* e *SD-Controlled* eficientes.

3.2. Furtividade Cibernética vs. Física

A furtividade de um ataque corresponde à sua capacidade de não ser percebido ou detectado. No caso de ataques físico-cibernéticos em NCSs, a furtividade deve ser analisada

simultaneamente em dois domínios diferentes: o domínio cibernético; e o domínio físico. Neste sentido, é apresentada nesta seção a definição de o que é um ataque *cibernetica-mente furtivo* e o que é um ataque *fisicamente furtivo*:

- Ataques ciberneticamente furtivos: são ataques que têm baixa probabilidade de serem detectados por algoritmos que monitoram os softwares, a comunicação e os dados do sistema, ou por sistemas que monitoram a dinâmica da planta.
- Ataques fisicamente furtivos: são ataques que causam efeitos físicos que não são facilmente percebidos ou identificados por um observador humano. O ataque modifica sutilmente alguns comportamentos do sistema de forma a afetar fisicamente a planta, mas o efeito não é facilmente percebido ou, eventualmente, pode ser entendido como uma consequência cuja causa seja outra, diferente de um ataque.

4. Ataque de Identificação de Sistema

O ataque de Identificação de Sistemas, ou *System identification*, aqui apresentado visa estimar os coeficientes da função de transferência da planta $G(z)$ e da função de controle $C(z)$ do controlador. Ambas as funções são de sistemas Lineares e Invariantes no Tempo (LIT). O ataque usa o Algoritmo de Busca por Retrocesso, ou *Backtracking Search Algorithm* (BSA), proposto em [Civicioglu 2013] e resumidamente descrito em [de Sá et al. 2016], como metaheurística para minimizar a função de aptidão apresentada nesta Seção.

O BSA é um algoritmo evolucionário que utiliza informações obtidas por gerações – ou iterações – passadas para buscar soluções em problemas de otimização. O algoritmo possui dois parâmetros que são empiricamente ajustados: o tamanho da sua população P; e η, descrito em [de Sá et al. 2016], que estabelece a amplitude do deslocamento dos indivíduos de P. O parâmetro η deve ser ajustado visando atribuir ao algoritmo tanto uma boa capacidade exploração, quanto de refinamento da busca.

Se a entrada $i(k)$ e a saída $o(k)$ de um dispositivo real do NCS são conhecidas, seu modelo interno pode ser inferido aplicando a entrada conhecida $i(k)$ em um modelo estimado, que deve ser ajustado até que a sua saída estimada $\hat{o}(k)$ convirja para $o(k)$. Neste sentido, o BSA é usado para ajustar iterativamente o modelo estimado, minimizando uma função de aptidão específica, até que o modelo estimado convirja para o modelo real do dispositivo do NCS, o qual pode ser um controlador ou uma planta.

Para estabelecer a função de aptidão, devemos primeiramente considerar o sistema LIT genérico, cuja função de transferência $Q(z)$ pode ser representada por (1):

$$Q(z) = \frac{O(z)}{I(z)} = \frac{a_n z^n + a_{n-1} z^{n-1} + ... + a_1 z^1 + a_0}{z^m + b_{m-1} z^{m-1} + ... + b_1 z^1 + b_0}, \tag{1}$$

onde $I(z)$ é a entrada do sistema, $O(z)$ é a sua saída, n e m correspondem a ordem do numerador e do denominador, respectivamente, e $[a_n, a_{n-1}, ...a_1, a_0]$ e $[b_{m-1}, b_{m-2}, ...b_1, b_0]$ são os coeficientes do numerador e do denominador, respectivamente, os quais pretende-se estimar com o presente algoritmo de Identificação de Sistemas. Consideremos ainda que $i(k)$ e $o(k)$ representam as amostras da entrada e da saída do sistema, respectivamente, onde $I(z) = \mathcal{Z}[i(k)]$, $O(z) = \mathcal{Z}[o(k)]$, k é o número da amostra e $\mathcal{Z}$ representa a operação da transformada Z.

Neste ataque de identificação de sistemas, $i(k)$ e $o(k)$ são primeiramente capturados por um ataque do tipo *eavesdropping* [Khatri et al. 2015], por exemplo, durante um

período T. Para lidar com eventuais perdas de amostras, que podem não ser recebidas pelo atacante durante T, o algoritmo retém o valor da última amostra recebida, conforme (2), onde $x(k)$ pode ser tanto $i(k)$ quanto $o(k)$.

$$x(k) = \begin{cases} x(k-1) & \text{se a amostra } k \text{ é perdida;} \\ \\ x(k) & \text{senão.} \end{cases} \tag{2}$$

Em seguida, após capturar $i(k)$ e $o(k)$, o sinal $i(k)$ é aplicado à entrada de um modelo estimado, descrito por função de transferência cujos coeficientes $[a_{n,j}, a_{n-1,j}, ...a_{1,j}, a_{0,j}, b_{m-1,j}, b_{m-2,j}, ...b_{1,j}, b_{0,j}]$ são as coordenadas de um indivíduo j do BSA. A aplicação de $i(k)$ ao modelo estimado resulta em um sinal de saída $\hat{o}_j(k)$. Após obter $\hat{o}_j(k)$, a função de aptidão f_j do indivíduo j é calculada comparando a saída $o(k)$, capturada no dispositivo atacado, com a saída do modelo estimado $\hat{o}_j(k)$, de acordo com (3):

$$f_j = \frac{\sum_{k=0}^{N} (o(k) - \hat{o}_j(k))^2}{N}, \tag{3}$$

onde N é o número de amostras que existem durante o período de monitoração T. Note que, se o atacante não perder nenhuma amostra de $i(k)$ e $o(k)$ durante T, então $\min f_j = 0$ quando $[a_{n,j}, a_{n-1,j}, ...a_{1,j}, a_{0,j}, b_{m-1,j}, b_{m-2,j}, ...b_{1,j}, b_{0,j}] = [a_n, a_{n-1}, ...a_1, a_0, b_{m-1}, b_{m-2}, ...b_1, b_0]$, *i.e.* quando o modelo estimado converge para o modelo real.

É possível estabelecer uma analogia entre este ataque de identificação de sistemas e o ataque de criptoanálise do tipo *known plaintext*, onde $i(k)$ e $o(k)$ correspondem aos textos simples e cifrado, respectivamente, o formato da função de transferência genérica $Q(z)$ corresponde ao algoritmo de criptografia e os coeficientes reais de $Q(z)$ correspondem à chave criptográfica.

5. Ataque Furtivo para Degradação do Serviço

Com base na taxonomia apresentada na Seção 3.1, o ataque descrito nesta Seção é classificado como do tipo *SD-Controlled Data Injection*. Seu propósito é reduzir o MTBF da planta e/ou reduzir a eficiência do processo físico que a mesma executa, através da inserção de dados falsos na malha de controle. Ao mesmo tempo, o atacante deseja que o ataque atenda ao requisito de ser fisicamente furtivo, *i.e.* com um efeito físico de difícil percepção por um observador humano, ou entendido como uma consequência cuja causa não seja um ataque – conforme definido na seção Seção 3.2.

Uma das maneiras de degradar um serviço físico é por meio da indução de um *overshoot* durante o regime transitório da planta. *Overshoots*, ou picos no regime transitório, podem causar estresse e, eventualmente, danos à sistemas físicos como por exemplo sistemas mecânicos, químicos e eletromecânicos [El-Sharkawi and Huang 1989, Tran et al. 2007]. Adicionalmente, por ocorrerem em curto espaço de tempo, os *overshoots* são de difícil percepção pelo observador humano. Outra forma de degradar o serviço é causar um erro estacionário constante na planta, ou seja, fazer com que a saída da mesma tenha um erro constante quando $t \to \infty$. Erros estacionários de pequena proporção, além de serem de difícil percepção pelo observador humano, podem reduzir a eficiência do processo físico e, eventualmente, estressar e danificar o sistema em médio/longo prazo.

Neste ataque, para alcançar qualquer um dos dois efeitos citados, *i.e.* um *overshoot* ou um erro estacionário constante, o atacante intervém no processo de comunicação do NCS a fim de injetar, de forma controlada, dados falsos no sistema. Para tal, o atacante atua como um MitM que executa uma função de ataque $M(z)$, conforme apresentado na Figura 3, onde $U'(z) = M(z)U(z)$, $U(z) = \mathcal{Z}[u(k)]$ e $U'(z) = \mathcal{Z}[u'(k)]$. A função $M(z)$ é projetada com base nos dados da planta e do controlador, obtidos no ataque do tipo *System Identification* descrito na Seção 4. A eficácia do ataque, portanto, depende do projeto de $M(z)$, que por sua vez depende da acurácia do ataque de *System Identification*. Cabe ressaltar que, apesar de na Figura 3 o MitM atuar nos sinais de controle, é possível, também, que o mesmo atue nos sinais de realimentação do NCS. O MitM pode ser estabelecido tanto em redes cabeadas quanto, eventualmente, em redes sem fio conforme em [Hwang et al. 2008].

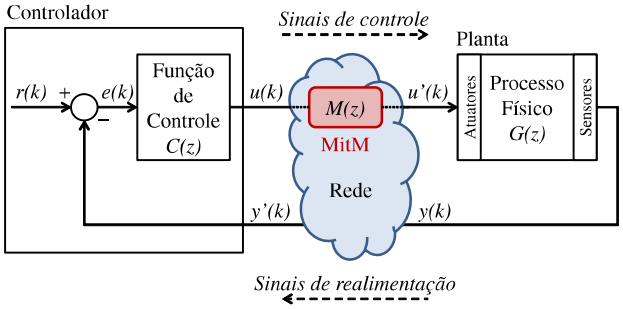


Figura 3. Ataque MitM

6. Resultados

Nesta seção são apresentados os resultados obtidos em simulações que combinam ataques do tipo *System Identification* com ataques *SD-Controlled* fisicamente furtivos. Na Seção 6.1, é apresentado o modelo do sistema atacado. Na Seção 6.2 são apresentados os resultados obtidos pelo ataque do tipo *System Identification*. Na Seção 6.3 são apresentados os resultados obtidos com simulações de ataques do tipo *SD-Controlled Data Injection*, fisicamente furtivos, planejados com base nos dados do ataque de *System Identification*.

6.1. Modelo do Sistema

O NCS atacado tem a mesma arquitetura do NCS apresentado na Figura 1, e consiste em um controlador Proporcional-Integral (PI) que controla a velocidade de rotação de um motor DC. A função de controle $C(z)$ e a função de transferência $G(z)$ do motor DC foram extraídas de [Long et al. 2005]. Tais equações são representadas por (4):

$$C(z) = \frac{c_1 z - c_2}{z - 1} \qquad G(z) = \frac{g_1 z + g_2}{z^2 - g_3 z + g_4} \tag{4}$$

onde $c_1 = 0,1701$, $c_2 = -0,1673$, $g_1 = 0,3379$, $g_2 = 0,2793$, $g_3 = -1,5462$ e $g_4 = 0,5646$. A taxa de amostragem do sistema é 50 amostras/s e o *set point* $r(k)$ é uma função degrau unitário. O atraso na rede não é considerado nestas simulações.

6.2. Resultados da Identificação do Sistema

Nesta Seção, o desempenho do algoritmo de Identificação de Sistemas é avaliado por meio um conjunto de simulações realizadas no MATLAB. A ferramenta SIMULINK foi

utilizada para calcular a saída $\hat{o}_j$ dos modelos estimados, cujos coeficientes são as coordenadas de um indivíduo j do BSA.

A estrutura das equações representadas por (4) são previamente conhecidas pelo atacante, uma vez que, como premissa, este sabe que o alvo é um NCS que controla um motor DC por meio de um controlador PI. Nestas simulações, o objetivo do ataque de *System Identification* é descobrir g_1, g_2, g_3, g_4, c_1 e c_2, levando em consideração cenários em que o atacante eventualmente perde amostras durante a coleta dos sinais de controle e de realimentação.

Toda vez que o motor DC é ligado, os sinais de controle e de realimentação são capturados pelo atacante durante um período $T = 2s$. no momento em que o motor é ligado, todas as condições iniciais são consideradas 0. Os coeficientes de $G(z)$, $[g_1, g_2, g_3, g_4]$, e os coeficientes de $C(z)$, $[c_1, c_2]$, são calculados separadamente considerando que, apesar da malha fechada, $G(z)$ e $C(z)$ são funções independentes. Para estimar $[g_1, g_2, g_3, g_4]$, o atacante considera que o sinal de controle é a entrada e que o sinal de realimentação é a saída da planta. Já para estimar $[c_1, c_2]$, o atacante considera que o sinal e realimentação é a entrada e que o sinal de controle é a saída do controlador.

Para simular a perda de amostras, são consideradas quatro taxas de perda l diferentes: 0, 0,05, 0,1 e 0,2. Assim, uma amostra é perdida pelo atacante toda vez que $l < \mathcal{P}$, onde $\mathcal{P} \sim U(0, 1)$ e U é uma distribuição uniforme. Para cada taxa de perda são executadas 100 simulações diferentes.

No BSA, a população utilizada contém 100 indivíduos e η, empiricamente ajustado, é 1. Para estimar os coeficientes do controlador $[c_1, c_2]$, são executadas 600 iterações do algoritmo. Já para estimar os coeficientes da planta $[g_1, g_2, g_3, g_4]$, o número de iterações é aumentado para 800, devido ao maior número de dimensões do espaço de busca neste caso. Os limites de cada dimensão do espaço de busca são $[-10, 10]$.

A Figura 4 apresenta a média de 100 valores estimados para g_1, g_2, g_3, g_4, c_1 e c_2, com um Intervalo de Confiança (IC) de 95%, considerando diferentes taxas de perda de amostras. Os valores reais dos coeficientes de $C(z)$ e $G(z)$ também são representados na Figura 4. Note que a amplitude das escalas das Figuras 4(a), 4(b), 4(c) e 4(d) é diferente da amplitude das escalas das Figuras 4(e) e 4(f), em virtude dos menores IC de c_1 e c_2. Adicionalmente, algumas estatísticas referentes aos resultados obtidos são apresentadas na Tabela 2.

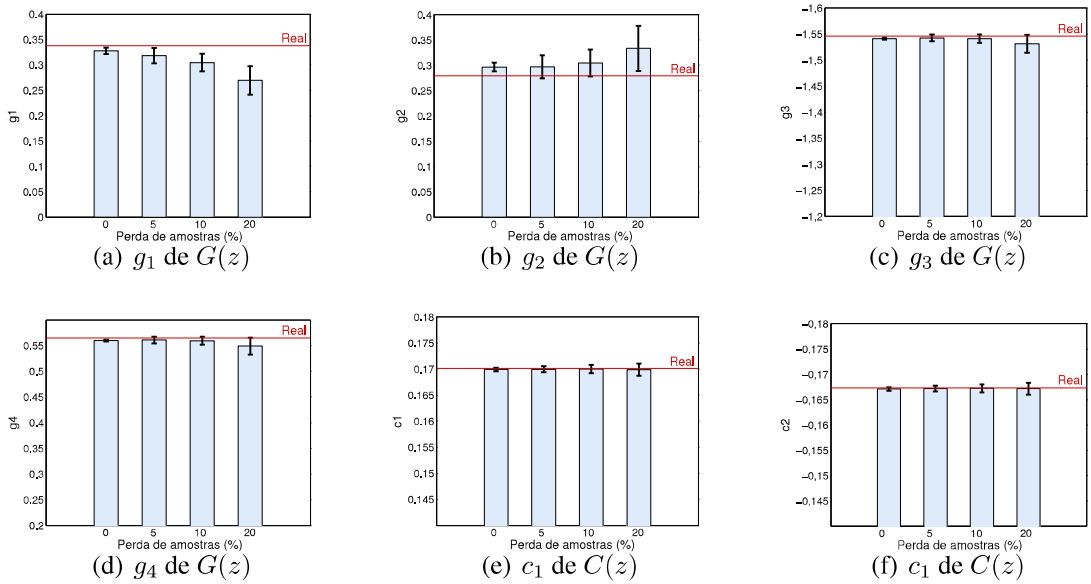
Tabela 2. Estatísticas dos resultados com diferentes perdas de amostras

Perda de amostras	Média						Desvio Padrão					
	g_1	g_2	g_3	g_4	c_1	c_2	g_1	g_2	g_3	g_4	c_1	c_2
0%	0.32793	0.29652	-1.54121	0.55983	0.16991	-0.16712	0.03097	0.04288	0.00986	0.00944	0.00167	0.00178
5%	0.31835	0.29689	-1.54251	0.56085	0.16997	-0.16719	0.07572	0.11523	0.03322	0.03194	0.00287	0.00287
10%	0.30473	0.30461	-1.54110	0.55925	0.16999	-0.16724	0.08781	0.13483	0.04076	0.03922	0.00397	0.00399
20%	0.26963	0.33352	-1.53119	0.54916	0.16989	-0.16716	0.14120	0.22378	0.08596	0.08313	0.00596	0.00598
Perda de amostras	Assimetria(*)						Curtose					
	g_1	g_2	g_3	g_4	c_1	c_2	g_1	g_2	g_3	g_4	c_1	c_2
0%	-1.21214	1.23278	1.75298	-1.73202	-0.64331	0.79458	0.18846	0.19433	0.21259	0.21218	0.15119	0.16472
5%	-2.34607	1.64875	1.35284	-1.41346	-0.42288	0.36037	0.08094	0.10527	0.09412	0.09802	0.02540	0.03118
10%	-2.52938	1.97711	1.18018	-1.26045	-0.23379	0.13377	0.16833	0.17123	0.25041	0.24811	0.24361	0.23429
20%	-3.24122	1.75186	1.68335	-1.71055	-0.40055	0.37927	0.21292	0.21127	0.25054	0.24932	0.23883	0.24441

(*) Calculado de acordo com o 2^o coeficiente de assimetria de Pearson.

De acordo com a Tabela 2 as distribuições de g_1, g_2, g_3 e g_4 possuem forte assimetria, enquanto as distribuições de c_1 e c_2 têm assimetria moderada. Em relação a curtose, as distribuições de todos os coeficientes de $G(z)$ e $C(z)$ são leptocúrticas. Entretanto, analisando a Tabela 2, não é possível identificar uma clara tendência de aumento ou diminuição de assimetria e curtose em face do aumento da perda de amostras.

Na Figura 4, é possível constatar que em todos os casos os ICs tendem a crescer com o aumento da perda de amostras. O mesmo ocorre com os desvios padrão apresentados na Tabela 2. Referente aos coeficientes de $G(z)$, a Figura 4 mostra que a diferença entre a média e o valor real de g_1, g_2, g_3 e g_4 também tende a crescer com o aumento da perda de amostras. Cabe ressaltar que o desempenho do algoritmo no cálculo de g_3 e g_4 é melhor do que no cálculo de g_1 e g_2, tanto no que diz respeito a média quanto a amplitude do IC. Atribuímos este comportamento à maior sensibilidade que a saída de $G(z)$ tem às variações de seus polos do que às variações de seus zeros. Isto significa que, neste problema, f_j cresce mais com os erros de g_3 e g_4 do que com os erros de g_1 e g_2, fazendo com que a população do BSA convirja de forma mais acurada em g_3 e g_4.



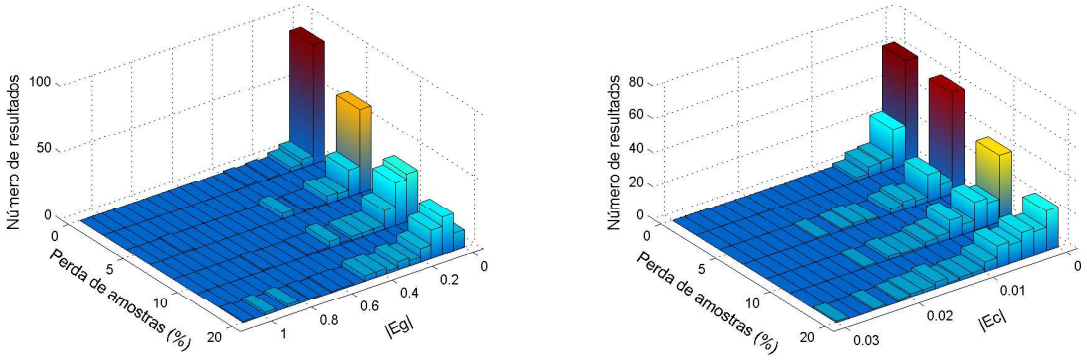
(a) g_1 de $G(z)$ (b) g_2 de $G(z)$ (c) g_3 de $G(z)$

(d) g_4 de $G(z)$ (e) c_1 de $C(z)$ (f) c_1 de $C(z)$

Figura 4. Média, com IC de 95%, dos coeficientes estimados de $G(z)$ e $C(z)$, em face de diferentes taxas de perda de amostras.

Na Figura 4 é possível também verificar que a acurácia obtida no cálculo dos coeficientes de $C(z)$ é melhor do que a acurácia dos coeficientes de $G(z)$, para todas as taxas de perda de amostras. As médias de c_1 e c_2 são mais próximas dos seus valores reais, com um menor IC. De fato, o processo de otimização é mais eficiente no cálculo dos coeficientes de $C(z)$ devido ao menor tamanho do espaço de busca, que possui apenas duas dimensões ao invés das quatro existentes no problema de $G(z)$.

Como uma forma adicional de avaliar o desempenho do algoritmo, foram calculados $|E_g| = |\mathcal{G}_r - \mathcal{G}_e|$ e $|E_c| = |\mathcal{C}_r - \mathcal{C}_e|$ que sintetizam o erro de estimativa dos coeficientes de $G(z)$ e $C(z)$, respectivamente, para cada solução encontrada. $\mathcal{G}_r$ e $\mathcal{G}_e$ são vetores contendo os coeficientes reais e estimados de $G(z)$, respectivamente. Já $\mathcal{C}_r$ e $\mathcal{C}_e$ são vetores contendo os coeficientes reais e estimados de $C(z)$, respectivamente. Os histogramas de $|E_g|$ e $|E_c|$ são apresentados na Figura 5, considerando as diferentes taxas de perda de amostras mencionadas. Os histogramas mostram graficamente que $|E_g|$ e $|E_c|$, que correspondem ao módulo do erro dos coeficientes estimados de $G(z)$ e $C(z)$, respectivamente, tendem a apresentar valores maiores à medida que a perda de amostras aumenta. Isto também pode ser confirmado pelo aumento do desvio padrão dos coeficientes de $G(z)$ e $C(z)$ apresentados na Tabela 2. Entretanto, de acordo com a Figura 5, a moda destes erros permanecem próximas de zero em todos os casos de perda avaliados.

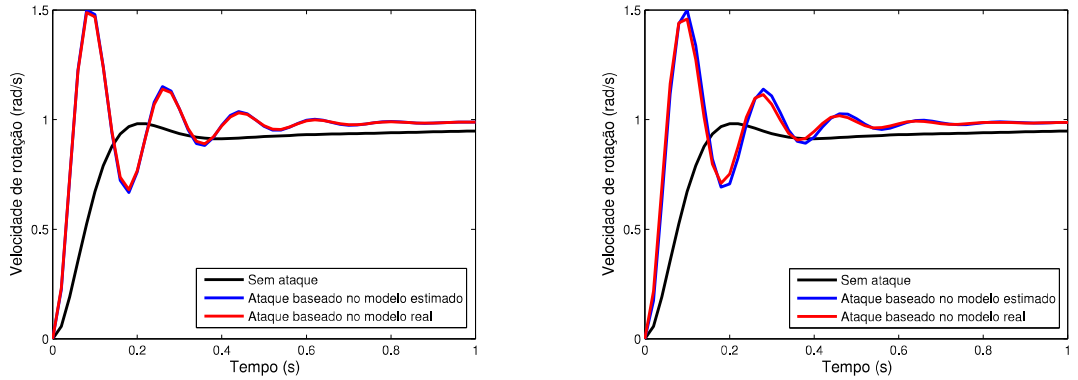


(a) Distribuição dos $|E_g|$ (b) Distribuição dos $|E_c|$

Figura 5. Histogramas de $|E_g|$ e $|E_c|$ em face das diferentes perdas de amostras

6.3. Resultados do Ataque de Degradação do Serviço

Nesta seção são apresentados os resultados obtidos em simulações de ataque do tipo *SD-Controlled Data Injection*, realizados por um MitM atuando no enlace de controle do NCS, conforme na Figura (3). Os ataques foram simulados no MATLAB, com o objetivo de avaliar a acurácia de ataques planejados com base nos resultados da Seção 6.2, obtidos pelo ataque de *System Identification*. Foram realizados dois conjuntos de ataques. O primeiro, visa causar um *overshoot* de 50% na velocidade de rotação do motor. O segundo, visa causar um erro estacionário de -10% na velocidade de rotação do motor em regime permanente.



(a) Ataque baseado nos dados obtidos sem perda de amostras (b) Ataque baseado nos dados obtidos com 20% de perda de amostras

Figura 6. Resposta do sistema a ataques planejados com o propósito de causar um *overshoot* de 50% da velocidade de rotação do motor.

No ataque visando o *overshoot*, a função executada pelo atacante é $M(z) = \mathcal{K}_o$. Por meio da análise do lugar das raízes, traçado com base nos modelos levantados, o atacante ajusta o valor de $\mathcal{K}_o$ para que o sistema se torne subamortecido com um pico de velocidade de rotação 50% maior do que a velocidade em regime permanente. Os valores de $\mathcal{K}_o$ foram ajustados com base nas médias dos coeficientes levantados na Seção 3.2. A Tabela 3 apresenta os valores de $\mathcal{K}_o$, estimados considerando as diferentes situações de perda de amostras no ataque de *System Identification*, bem como os *overshoots* obtidos com os respectivos $\mathcal{K}_o$ no modelo real. Na Figura 6 é possível comparar a resposta do sistema sem ataque com a resposta ao ataque visando o *overshoot* de 50%. É possível verificar, ainda, que o ataque ao modelo real apresenta, no domínio do tempo, uma resposta bem próxima ao ataque projetado com base no modelo obtido pelo ataque de *System*

Identification, tanto no caso em que o sistema foi identificado com 0% de perdas, quanto no pior caso considerado, com 20% de perdas. Cabe ressaltar que todas as respostas apresentadas na Figura 6 convergem para 1 rad/s.

No ataque cujo propósito é causar um erro estacionário de -10% na velocidade de rotação do motor, o atacante executa a função (5):

$$M(z) = \frac{\mathcal{K}_{Ess}(z-1)}{z-0,94}, \tag{5}$$

onde $\mathcal{K}_{Ess}$ é ajustado com base nos dados de identificação do sistema, considerando cada condição de perda de amostras. O pólo de $M(z)$ é adicionado com o objetivo de permitir que ocorra um erro estacionário no sistema. O zero de $M(z)$ visa formatar o lugar das raízes a fim de que haja um $\mathcal{K}_{Ess}$ estável que leve o sistema a um erro estacionário de -10%. A Tabela 3 apresenta os valores de $\mathcal{K}_{Ess}$ adotados considerando as diferentes situações de perda de amostras no ataque de *System Identification*, bem como os respectivos erros estacionários alcançados no modelo real.

Tabela 3. Valores de $\mathcal{K}_o$, $\mathcal{K}_{Ess}$ e resultados obtidos com os ataques

	Perda de amostras no ataque *System Identification*			
	0 %	5 %	10 %	20 %
$\mathcal{K}_o$	4,0451	4,0745	4,0828	3,796
Overshoot no modelo real	48,90 %	49,43 %	49,57 %	45,94 %
$\mathcal{K}_{Ess}$	5,7471	5,7803	5,8140	5,8823
Erro estacionário no modelo real	-10%	-10%	$-9,9\%$	$-9,8\%$

De acordo com os dados na Tabela 3, é possível afirmar que os ataques *SD-Controlled Data Injection*, projetados com base nos dados colhidos pelo ataque *System Identification*, foram capazes de modificar de forma acurada a resposta do sistema físico, considerando todas as condições de perda avaliadas. No pior caso, *i.e.* com 20% de perda de amostras, o *overshoot* foi de $45,94\%$ e o erro estacionário foi de $-9,8\%$, bem próximos dos valores desejados de 50% e -10%, respectivamente. Tal acurácia, permite que a resposta do sistema se mantenha controlada e próxima a um comportamento pré-definido como fisicamente furtivo para o sistema em questão.

7. Conclusões

Este trabalho propõe um ataque fisicamente furtivo de degradação de serviço, cujo desempenho depende do conhecimento sobre a planta atacada e seu controlador. Para adquirir tal conhecimento, é proposto um ataque de *System Identification*, baseado no algoritmo BSA. A eficácia do ataque de *System Identification* é demonstrada e o seu desempenho é estatisticamente analisado em face de diferentes taxas de perda de amostra. Os resultados alcançados nos ataques fisicamente furtivos de degradação de serviço, dimensionados com base nos dados levantados pelo *System Identification*, demonstram o elevado grau de acurácia que pode ser obtido com a combinação dos ataques. No pior caso, *i.e.* com 20% de perda de amostras durante a identificação do sistema, o atacante foi capaz de causar na planta um *overshoot* de $45,94\%$ e um erro estacionário de $-9,8\%$, bem próximos dos valores desejados de 50% e -10%, respectivamente. Em ambas as ações fisicamente furtivas, a acurácia do ataque garante que estas não evoluam para alterações de comportamento fisicamente mais perceptíveis.

Como trabalho futuro, encorajamos a pesquisa de técnicas capazes de evitar, ou dificultar, ataques fisicamente furtivos planejados com dados obtidos por ataques *System Identification*. Neste sentido, planejamos investigar contramedidas que possam dificultar a obtenção de informações sobre os sistemas de controle físico-cibernéticos, as quais são essenciais para o planejamento de ataques furtivos e controlados.

Referências

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, 219(15):8121–8144.

de Sá, A. O., Nedjah, N., and de Macedo Mourelle, L. (2016). Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms. *Neurocomputing*, 172:322–336.

El-Sharkawi, M. and Huang, C. (1989). Variable structure tracking of dc motor for high performance applications. *Energy Conversion, IEEE Transactions on*, 4(4):643–650.

Farooqui, A. A., Zaidi, S. S. H., Memon, A. Y., and Qazi, S. (2014). Cyber security backdrop: A scada testbed. In *Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE*, pages 98–103. IEEE.

Hussain, A., Heidemann, J., and Papadopoulos, C. (2003). A framework for classifying denial of service attacks. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 99–110. ACM.

Hwang, H., Jung, G., Sohn, K., and Park, S. (2008). A study on mitm (man in the middle) vulnerability in wireless network using 802.1 x and eap. In *Information Science and Security, 2008. ICISS. International Conference on*, pages 164–170. IEEE.

Khatri, S., Sharma, P., Chaudhary, P., and Bijalwan, A. (2015). A taxonomy of physical layer attacks in manet. *International Journal of Computer Applications*, 117(22).

Langner, R. (2011). Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE*, 9(3):49–51.

Long, M., Wu, C.-H., and Hung, J. Y. (2005). Denial of service attacks on network-based control systems: impact and mitigation. *Industrial Informatics, IEEE Transactions on*, 1(2):85–96.

Ramos, C., Vale, Z., and Faria, L. (2011). Cyber-physical intelligence in the context of power systems. In *Future Generation Information Technology*, pages 19–29. Springer.

Smith, R. (2011). A decoupled feedback structure for covertly appropriating networked control systems. In *Proceedings of the 18th IFAC World Congress 2011*, volume 18. IFAC-PapersOnLine.

Smith, R. S. (2015). Covert misappropriation of networked control systems: Presenting a feedback structure. *Control Systems, IEEE*, 35(1):82–92.

Teixeira, A., Shames, I., Sandberg, H., and Johansson, K. H. (2015). A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148.

Tran, T., Ha, Q. P., and Nguyen, H. T. (2007). Robust non-overshoot time responses using cascade sliding mode-pid control. *Journal of Advanced Computational Intelligence and Intelligent Informatics*.

