Universidade Federal de Ouro Preto

Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica

Um Estudo sobre Vulnerabilidades e Ataques em Sistemas de Automação

Aluno: Paulo Ricardo de Freitas
Orientador: Prof. Dr. Víctor Costa da Silva Campos
Co-orientador: Prof. Theo Silva Lins

Trabalho de Conclusão de Curso

João Monlevade, 26 de fevereiro de 2018

Universidade Federal de Ouro Preto

Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica

Um Estudo sobre Vulnerabilidades e Ataques em Sistemas de Automação

Aluno: Paulo Ricardo de Freitas
Orientador: Prof. Dr. Víctor Costa da Silva Campos
Co-orientador: Prof. Theo Silva Lins

Monografia apresentada ao Curso de Engenharia Elétrica do Instituto de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, como requisito final para conclusão de curso.

João Monlevade, 26 de fevereiro de 2018

F866e Freitas, Paulo Ricardo de.
Um estudo sobre vulnerabilidades e ataques em sistemas de automação [manuscrito] / Paulo Ricardo de Freitas. - 2018.

46f.: il.: color; grafs.

Orientador: Prof. Dr. Víctor Costa da Silva Campos.
Coorientador: Prof. Dr. Theo Silva Lins.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Engenharia Elétrica.

1. Engenharia elétrica. 2. Sistemas de controle. 3. Computadores - Medidas de segurança. I. Campos, Víctor Costa da Silva. II. Lins, Theo Silva. III. Universidade Federal de Ouro Preto. IV. Titulo.

CDU: 621.3:004.056

Catalogação: ficha@sisbin.ufop.br

MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Aplicadas
Colegiado do Curso de Engenharia de Elétrica

ANEXO IV - ATA DE DEFESA

Aos 21 dias do mês de fevereiro de 2018, às 14 horas e 30 minutos, no bloco C deste instituto, foi realizada a defesa de monografia pelo (a) formando (a) Paulo Ricardo de Freitas, sendo a comissão examinadora constituída pelos professores: Marcelo Moreira Tiago, Márcio Feliciano Braga, Theo Silva Lins e Víctor Costa da Silva Campos.

O (a) candidato (a) apresentou a monografia intitulada: Um Estudo sobre Vulnerabilidades e Ataques em Sistemas de Automação. A comissão examinadora deliberou, por unanimidade, pela aprovação do(a) candidato(a), com a nota média 9,0, de acordo com a tabela 1. Na forma regulamentar foi lavrada a presente ata que é assinada pelos membros da comissão examinadora e pelo (a) formando(a).

Tabela 1 – Notas de avaliação da banca examinadora

Banca Examinadora	Nota
Marcelo Moreira Tiago	9,0
Márcio Feliciano Braga	9,0
Theo Silva Lins	9,0
Víctor Costa da Silva Campos	9,0
Média	9,0

João Monlevade, 21 de fevereiro de 2018.

Víctor Costa da Silva Campos
Professor(a) Orientador(a)

Theo Silva Lins
Professor(a) Co-orientador(a)

Marcelo Moreira Tiago
Professor(a) Convidado(a)

Márcio Feliciano Braga
Professor(a) Convidado(a)

Paulo Ricardo de Freitas
Aluno(a)

MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Aplicadas
Colegiado do Curso de Engenharia de Elétrica

ANEXO X - TERMO DE RESPONSABILIDADE

O texto do trabalho de conclusão de curso intitulado "Um estudo sobre vulnerabilidades e ataques em sistemas de automação" é de minha inteira responsabilidade. Declaro que não há utilização indevida de texto, material fotográfico ou qualquer outro material pertencente a terceiros sem a devida citação ou consentimento dos referidos autores.

João Monlevade, 23 de fevereiro de 2018.

Paulo Ricardo de Freitas

Paulo Ricardo de Freitas

Resumo

Este trabalho apresenta um estudo sobre sistemas supervisórios e táticas de controle para introduzir e testar alguns dos métodos de detecção e identificação de ataques a sistemas *ciber-físicos* (CPS, do inglês *Cyber-Physical Systems*). Ao longo da obra, são discutidas algumas técnicas e ferramentas utilizadas em infra-estruturas críticas, desenvolvidas por engenheiros de controle e profissionais da Tecnologia de Informação e Comunicação (TIC), . Há também um estudo sobre a tecnologia Foundation Fieldbus cujo propósito é sustentar as hipóteses propostas e testar os cenários de ataque. Por fim, a função deste trabalho é apresentar algumas referências necessárias para entender o que é um ataque a um CPS e como protegê-los, já que a cada dia mais dispositivos inteligentes se interconectam nas indústrias devido a integração da *Internet of Things* (IoT) na indústria 4.0.

Palavras-Chave: supervisório, sistemas de controle, CPS, TIC, Fieldbus, IoT, indústria 4.0.

Abstract

This work presents a study about supervisory systems and control tactics to introduce and test some of the detection methods and attack identification on cyber-physical systems (CPS). Throughout this work, some techniques and tools used in critical infrastructures, elaborated by control engineers and Information and Communication Technology (ICT) professionals, are discussed. Also, there is a study about the Foundation Fieldbus whose purpose is to support the proposed hypotheses and test the attack scenarios. Lastly, the function of this work is to introduce some references that are useful to understand what is an attack to a CPS and how to protect them, since each day more smart devices are interconnected in industries due to the integration of the Internet of Things (IoT) on the industry 4.0.

Keywords: supervisory, control systems, CPS, ICT, Fieldbus, IoT, industry 4.0.

Lista de Acrônimos

BCIT	British Columbia Institute of Technology
CPS	Cyber-Physical System
CLP	Controlador Lógico Programável
DoS	Denial of Service
DDoS	Distributed Denial of Service
FAS	Fieldbus Access Sublayer
FF	Foundation Fieldbus
FMS	Fieldbus Message Specification
HMI	Human-Machine Interface
IoT	Internet of Things
IP	Internet Protocol
MIMO	Multiple Input Multiple Output
MIS	Management Information System
MTU	Master Terminal Unit
OSI	Open System Interconnection
PLC	Programable Logic Controller
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SO	Sistema Operacional
TCI	Tecnologia de Comunicação e Informação
TCP	Transfer Control Protocol
TI	Tecnologia de Informação

Lista de ilustrações

Figura 1 – Modificação do modelo de comunicação OSI para o modelo FF 10
Figura 2 – Controle de processo com PLC 15
Figura 3 – Evolução dos controladores 17
Figura 4 – Filtro de Kalman 23
Figura 5 – Distribuições normais de erros. 25
Figura 6 – Dinâmica do sistema 28
Figura 7 – Validação do modelo 29
Figura 8 – Simulação do ataque *replay*. 31
Figura 9 – Detecção de *replay*. 32
Figura 10 – Simulação de ataque de integridade 34
Figura 11 – Detecção de ataque integridade 34
Figura 12 – Distribuições qui-quadrado. 40

Sumário

Lista de ilustrações . . . . . **viii**

1 INTRODUÇÃO **2**
1.1 Motivação **3**
1.2 Estrutura do Trabalho **3**

2 CONTEXTUALIZAÇÃO HISTÓRICA **5**
2.1 Incidentes **7**

3 FUNDAMENTAÇÃO TEÓRICA **10**
3.1 Foundation Fieldbus **10**
3.1.1 Camadas do Modelo 11
3.1.1.1 Pilha de Comunicação 11
3.1.1.2 Camada do Usuário 12
3.1.1.3 Camada Física 13
3.2 Controlador Lógico Programável **14**
3.3 Indústria 4.0 . . . . . **16**
3.4 Ataques Comuns **17**
3.4.1 DoS e DDoS 18
3.4.2 Ataque Replay 18
3.4.3 Ataque de Integridade 20
3.4.4 Phishing 20
3.5 Análise Comportamental de Sistemas **20**
3.6 Filtro de Kalman **22**
3.7 Detecção de falhas **24**
3.7.1 Teste de qualidade de ajuste com χ^2 24
3.7.2 Teste de qualidade de ajuste com autocorrelação 26

4 ESTUDO DE CASO **28**
4.1 Modelagem do Sistema **28**
4.2 Ensaios **30**
4.2.1 Identificação de falhas utilizando autocorrelação 30
4.2.2 Identificação de falhas por qui-quadrado 33

5 CONCLUSÕES **36**
5.1 Propostas de melhoria **36**

REFERÊNCIAS **37**

A TABELA DE DISTRIBUIÇÃO DO QUI-QUADRADO **40**

1 Introdução

O controle de um determinado processo é o ato de criar influência sobre o mesmo ocasionando a operação de acordo com alguns requerimentos preestabelecidos. O controle de processo está presente em aviões de carga, geradores de energia elétrica, torres de destilação, sistemas de tratamento de esgoto, dentre outros. É importante admitir que o ambiente no qual o processo está imerso gera influências no sistema que podem ser controláveis ou não. Esse fato explica a necessidade de um sistema supervisório do processo, constantemente realizando adaptações para minimizar a interferência do meio sobre os produtos a fim de obter a resposta mais próxima possível de um valor previamente configurado.

Muitos sistemas de controle têm natureza complexa, com mais de uma entrada e saída além da interação mútua entre elas, são os chamados sistemas de múltiplas entradas e múltipla saídas (MIMO). Alguns sistemas complexos que podem ser citados são aqueles que empregam reatores químicos, colunas de destilação, etc. De acordo com Tatjewski (2007), um exemplo de sistema muito complexo é uma linha de produção com intensa implementação de plantas tecnológicas, a princípio, sua natureza resulta em algo difícil de se controlar. Tatjewski (2007) ainda atribui, como principais dificuldades do controle, as baixas garantias de se obter a segurança apropriada e o envolvimento de um número adequado de pessoas supervisionando o processo.

Ao longo dos anos foi criado o que se chama aproximação hierárquica para controle de processos. Em suma, uma boa aproximação é dividida em multicamadas, cada uma com uma função objetivando resolver uma tarefa menor e menos complexa que a principal. Assim, o problema maior é dividido em partes interligadas entre si de maneira a propiciar a realização da tarefa principal de forma mais simples (TATJEWSKI, 2007).

Dentre as camadas presentes nessa aproximação encontram-se as camadas de controle, supervisão, diagnóstico e adaptação, dentre outras. Tais camadas estão intrinsecamente ligadas à dispositivos eletrônicos como os que se seguem: *Remote Terminal Units* (RTU), *Programable Logic Controllers* (PLCs), *Master Terminal Unit* (MTU) e, por fim, *Human-Machine Interface* (HMI). O sistema supervisório é composto pelas unidades remotas que atuam na coleta de leituras realizadas por sensores na planta e enviam para a MTU (BRANQUINHO et al., 2014) que processa os dados, realiza as decisões necessárias no sistema para controlar e otimizar o processo e, por fim, torna os dados legíveis para seres humanos para serem mostrados na HMI e arquivados para fins de documentação.

O sistema *Supervisory Control And Data Acquisition* (SCADA) que atende a sistema supervisório de controle e aquisição de dados é amplamente difundido em controle de processos industriais sendo o que realmente é implementado em ramos industriais de energia, água, transporte, petroquímico, dentre outros. O que torna esse sistema muito difundido e atrativo é que ele proporciona a redução de gastos operacionais e de manutenção, ao passo que aumenta significantemente a segurança do processo por meio de redução de

seres humanos realizando a coleta de dados e modificações do sistema (BRANQUINHO et al., 2014).

1.1 Motivação

Sistemas SCADA são responsáveis pela automação de praticamente todo tipo de processo industrial. A fácil instalação, manutenção e operação somadas à ótima relação custo/benefício tornam esse sistema cada vez mais presente em basicamente todos os setores: transporte, eletricidade, distribuição de gás natural e água são apenas alguns exemplos mais comuns (DUMONT, 2010; WANG; FANG; DAI, 2010; ZHU; JOSEPH; SASTRY, 2011). Vários órgãos governamentais também implementam SCADA em suas instalações críticas, entretanto, como é de se esperar, a maioria dos sistemas instalados em infra-estrutura de posse governamental são completamente isolados, ou melhor dizendo, não integram a Internet, o que os torna muito seguros (JUNG; SONG; KIM, 2008).

Se por um lado existem sistemas SCADA que ainda estão isolados e são praticamente livres de vulnerabilidades, por outro lado, devido ao aumento da competição no mercado, melhoria tecnológica e maior afinidade com as mesmas, surgiu a necessidade de conectar sistemas de controle à Internet para permitir acesso remoto, controle centralizado e consulta a dados em tempo mínimo (DUMONT, 2010). Para melhorar a eficiência e o nível de automação são utilizados Sistemas Operacionais (SOs) de uso comum e protocolos de comunicação livres (JUNG; SONG; KIM, 2008; WANG; FANG; DAI, 2010). Tais práticas, apesar de funcionarem bem no quesito usabilidade, deixam o sistema SCADA exposto e frágil, com um grande número de vulnerabilidades que podem ser exploradas por usuários maliciosos.

A literatura expõe vários ataques a sistemas de controle ao longo dos anos. Ainda em 2011, o *British Columbia Institute of Technology* (BCIT) possuía um banco de dados com mais de 120 relatório de ataques a instalações de infra-estrutura crítica. Segundo Zhu, Joseph e Sastry (2011), no relatório de 2011 da empresa de segurança virtual McAfee, dentre as 200 instalações avaliadas no programa, a maioria sofria algum tipo de ataque cibernético e em quase metade dos sistemas foram encontrados o *Stuxnet*, programa malicioso presente no ataque à usina nuclear iraniana em Natanz (2010) responsável por modificar a frequência elétrica da corrente que alimentava as centífugas de enriquecimento de urânio-235 causando falhas físicas (MILLER; ROWE, 2012).

1.2 Estrutura do Trabalho

A necessidade de estudar métodos de mitigação de ataques a sistemas de automação motivaram o desenvolvimento deste trabalho. Para entender as técnicas propostas, é necessário ter conhecimentos sobre os protocolos de rede que regem sistemas supervisórios. Esse conhecimento tem finalidade de identificar a possibilidade de manifestação

dos diferentes ataques e como cada um funciona, com suas particularidades. Entender as diferenças entre os ataques permite traçar estratégias de detecção e, por fim, desenvolver um método para identificar falhas.

Este trabalho foi organizado de acordo com o critério apresentado em sequência. No capítulo um, é apresentada a importância dos sistemas SCADA no cenário atual e conclui com um breve discurso sobre a exploração de vulnerabilidades em infraestruturas críticas. No segundo capítulo são apresentados os principais casos de exploração de falhas de segurança em sistemas de automação. No terceiro capítulo é feito um estudo bibliográfico sobre os assuntos que devem ser dominados para realizar análises de detecção e identificação de vulnerabilidades em *Cyber-Physical Systems* (CPS). Dentre tais assuntos destacam-se os protocolos de comunicação, tipos de ataque, Filtro de Kalman (FK) como estimador de estados e emprego de autocorrelação e qui-quadrado (χ^2) para identificação de comprometimento do sistema. No quarto capítulo são apresentados os resultados dos ensaios feitos na planta SMAR PD3-F e as simulações computacionais para identificação de ataques. Por último conclui-se o trabalho, resumindo o que foi observado ao longo dos testes e fazendo propostas para futuros projetos semelhantes a esse.

2

Contextualização Histórica

Dentre os sistemas de supervisão implantados nos vários ramos industriais é notório o destaque dado aos sistemas SCADA. Esse sistema surgiu por volta de 1940 e veio se adaptando às novas necessidades dos negócios, formas de administração e tecnologias (NICHOLSON et al., 2012). Pode-se dividir o sistema SCADA em três tipos devido às características apresentadas ao longo de sua evolução.

O primeiro tipo era um sistema completamente autônomo, seus recursos não se estendiam além do chão de fábrica onde estavam instalados, sem conexão a nenhuma rede de computadores ou outros sistemas. Nesta geração do sistema tanto *softwares* quanto protocolos eram puramente proprietários (CAI; WANG; YU, 2008).

A segunda geração dos sistemas SCADA, também conhecida por SCADA baseado em Intranet, contava com uma rede interna conectando o sistema de controle ao sistema de gestão de informação - *Management Information System* (MIS) (CAI; WANG; YU, 2008).

O grande passo no que diz respeito à comunicação dos sistemas SCADA com o resto do mundo por meio da Internet foi dado na terceira geração (atual). Para que isso fosse possível, muitas mudanças tiveram que ocorrer na estrutura da indústria. Começando pelos computadores, que passaram a ter maior capacidade de processamento, saíram dos 16 bits e foram para 32 ou 64 bits, possibilitando a adoção de sistemas operacionais de propósito geral. Na década de 1990, com a ascensão dos computadores pessoais juntamente com os avanços na tecnologia de comunicação em rede, houve a preferência geral da utilização de arquitetura de sistemas abertos em detrimento das arquiteturas proprietárias (NICHOLSON et al., 2012). Dessa forma, com a utilização de sistemas operacionais e protocolos de uso geral, *softwares* de controle abertos em detrimento de recursos proprietários além da conexão à Internet para controle à distância, ficou registrada, na terceira geração, a grande vantagem dos sistemas SCADA em relação aos demais.

Uma vez permitida a ampla conectividade por meio da Internet entre dispositivos separados fisicamente por uma grande distância, além da maior simplicidade garantida pelo emprego de SOs comuns e protocolos padrões de sistemas de Tecnologia de Comunicação e Informação (TCI), os sistemas SCADA adquiriram vulnerabilidades inexistentes em gerações anteriores.

Como esperado, uma vez que as vulnerabilidades dos sistemas SCADA foram detectadas por usuários maliciosos, infraestruturas críticas passaram a ser alvo de diversos tipos de ataques cibernéticos. É comum encontrar na literatura referências a três tipos de atacantes. Cardenas et al. (2009) explica muito bem a origem dos ataques e coloca as principais fontes como está a seguir:

- **Cibercriminosos:** Este grupo compromete o que encontra pelo caminho, é comum lançarem ataques gerais para usuários de computadores comuns, sem intenção de alarmar os administradores dos sistemas de controle. Alguns exemplos da atuação

desse grupo são:

1. **Oak Harbor — Ohio (2003):** ataque utilizando um *Slammer Worm* que propiciou o desligamentos dos displays de segurança por várias horas da usina nuclear Davis-Besse. O interessante desse ataque é que ele não visava afetar o sistema SCADA, porém como era utilizado um *software* livre na aplicação, este gerou as vulnerabilidades necessárias para a infecção.
2. **Pensylvania (2006):** Infecção de um computador na planta de filtragem de água e posterior utilização do computador para uso próprio em suas operações.

- **(Ex-)Funcionários (*Insiders*):** Atualmente representa a maior fonte de ataques específicos a sistemas de controle. Os motivos que levam a um ataque podem ter diversas naturezas: espionagem, sabotagem, vingança, entre outras (CARDENAS et al., 2009). O que deve ser ressaltado é que por se tratar de um ataque conduzido por alguém que conhece a empresa e suas vulnerabilidades, além de poder possuir acesso autorizado ao sistema de informação do alvo, tais ataques muitas vezes apresentam uma boa organização entre os passos do ataque e maior susceptibilidade ao sucesso e danos (HARPER et al., 2011). O grande diferencial desse tipo de fonte de ataque é que ele seria possível em qualquer geração dos sistemas supervisórios, uma vez que pode ser realizado por acesso físico ou remoto à instalação. Segundo Cardenas et al. (2009), o exemplo mais comum de ataque realizado por um confidente foi orquestrado no sistema de controle de esgoto de Maroochydore (antigo condado de Maroochy Shire), Queensland, Austrália, no ano 2000. Um ex-funcionário que teve acesso privilegiado ao sistema realizou uma séries de ataques que possibilitaram o inundamento de um hotel e um rio nas proximidades com milhões de litros de esgoto e, em seguida, tentou convencer a companhia de tratamento de água a reempregá-lo para resolver a situação.

- **Ativistas, terroristas e grupos de crime organizado:** O comprometimento de sistemas de controle seguido por extorsão se tornou uma realidade em alguns países do mundo. Em 2008, um analista do FBI relatou que havia evidências de intrusão a companhias europeias de distribuição de energia elétrica (GREENBERG, 2008). Cardenas et al. (2009) ainda relaciona essas práticas a uma evolução natural dos ataques físicos já conhecidos efetuados por criminosos, a atuação por meio virtual permite menor risco durante a operação e ainda possibilita execução a distância.

Na Seção 2.1 serão explicitados alguns dos incidentes mais famosos do mundo do controle e sistemas SCADA. Por meio dessa seção, pretende-se alarmar acerca das necessidades de intensificar a segurança digital dos sistemas de controle e supervisórios, redes de comunicação onde estão imersos, além de mostrar a forma como as vulnerabilidades são exploradas.

2.1 Incidentes

O estudo e criação de ferramentas eficazes aos sistemas de controle para evitar (ou ao menos identificar) invasões virtuais devem ser feitos atentando-se à forma com que os ataques são realizados, o *modus operandi* dos perpetrantes e, por fim, explorando ao máximo o sistema e suas vulnerabilidades. Deve-se concordar que um estudo aprofundado exige uma boa base de dados, entretanto isso não é sustentado uma vez que grande parte da documentação não é disponível para literatura. Ao mesmo tempo é comum a omissão de incidentes relacionados a segurança de sistemas SCADA tanto por operadores quanto por donos de instalações (CHRISTIANSSON; LUIIJF, 2007).

Apesar da dificuldade em reunir informações sobre incidentes envolvendo segurança de sistemas de controle, uma exceção que sempre é citada na literatura é o BCIT. Este instituto vem criando um banco de dados com os incidentes revelados desde 1982. No ano de 2004, o acervo contava com a documentação de 94 incidentes (CHRISTIANSSON; LUIIJF, 2007), número este que aumentou para 120 em 2011 (ZHU; JOSEPH; SASTRY, 2011).

A seguir, alguns dos principais incidentes envolvendo falhas de segurança em sistemas SCADA e infra-estruturas críticas serão apresentados a fim de esclarecer a importância de um estudo acerca das vulnerabilidades do meio e como criar defesas aos ataques. Os seguintes incidentes foram organizados observando-se fonte de ataque, método de operação, impacto e setor alvo. Esse material está disponível em Miller e Rowe (2012).

1. Explosão do gasoduto siberiano (1982): Foi o primeiro incidente registrado envolvendo cibersegurança e infra-estrutura crítica. Foi elaborado implantando-se um *Trojan* (Cavalo de Troia) no sistema de controle do gasoduto siberiano. O resultado do ataque de origem desconhecida foi a explosão da tubulação com um impacto equivalente a três mil toneladas de TNT.

2. Chevron (1992): Ataque criado por um ex-funcionário da empresa que invadiu computadores em Nova Iorque e San José, Califórnia, para desabilitar alarmes de segurança. O ataque passou despercebido até que uma emergência ocorreu na refinaria de Richmond, Califórnia, e o sistema não conseguiu alertar os moradores da região sobre a liberação de uma substância nociva. Como resultado da operação, o sistema da indústria ficou desativo por dez horas e milhares de pessoas foram colocadas em risco em 22 estados dos EUA e algumas áreas do Canadá.

3. Aeroporto de Worcester, Massachusetts (1997): Ataque do tipo *Denial of Service (DoS)*, ou negação de serviço, em que uma companhia telefônica foi invadida virtualmente e teve um computador que realizava serviços para o aeroporto de Worcester desligado. Em função disso, houve interrompimento de comunicação com a Administração Federal de Aviação e o voos foram cancelados por seis horas.

4. Gazprom (1999): O ataque à companhia de gás russa teve colaboração de um funcionário, os *hackers* admitiram ter usado um *Trojan* para acessar o painel central de controle do fluxo de gás nos dutos de distribuição.

5. Gasoduto de Bellingham, Washington (1999): Tecnicamente este exemplo não representa uma ataque devido à sua origem, entretanto ainda mostra quão devastador pode ser um incidente em infra-estruturas críticas. Em 2002 foi reportado que uma das causas do ocorrido foi a realização de desenvolvimento do banco de dados enquanto o sistema estava em funcionamento. Entre as consequências do incidentes estão: vazamento de 237 mil galões de gasolina no meio ambiente, incêndio de 1,5 milhas ao longo de um afluente que leva ao *Whatcom Falls Park*, três pessoas mortas e pelo menos oito feridos.

6. Cal-ISO (2001): Os atacantes do *California Independent System Operator (Cal-ISO)* obtiveram acesso a um dos computadores da instalação que tem controle hierárquico sobre algumas redes de computadores pessoais. Entretanto, o ataque não obteve sucesso, apesar de se prolongar por duas semanas.

7. CSX Corp. (2003): A *CSX Corporation* é uma das maiores empresas de transporte norte-americanas. Este ataque utilizou um vírus computacional nomeado *Sobig* que resultou no desligamento do sistema de sinalização de trens na Flórida. O vírus *Sobig* é comparado com um *SQLSlammer worm* por ser capaz de se espalhar em velocidade muito alta. Na época do ataque, o vírus apresentava a maior velocidade de contaminação por anexo em *e-mails*. O ataque resultou no atraso do sistema de transporte da empresa.

8. Stuxnet (2010): O *Stuxnet*, nome dado ao *worm* utilizado para infectar a usina iraniana de enriquecimento de urânio em Natanz, utilizou quatro vulnerabilidades até então desconhecidas, portanto chamadas de vulnerabilidade de dia-zero, para alcançar Sistemas Operacionais Windows e atuar nos dispositivos conversores de frequência fabricados pelas empresas *Fararo Paya* (iraniana) e *Vacon* (finlandesa). A função dos dispositivos atacados era o controle das centrífugas de enriquecimento de isótopos de urânio-235. Uma vez que o *Stuxnet* modificava a frequência elétrica da corrente que alimentava os dispositivos, as centrífugas falhavam numa taxa maior que a normal.

9. Night Dragon (2011): Ataque reportado pela empresa *McAfee*, combinava engenharia social, *Trojans* e exploração de vulnerabilidades do SO Windows. Os alvos eram cinco empresas de energia e óleo que estavam sob ataque por mais de dois anos. Foi concluído que o alvo principal não era o sistema SCADA, pois as ferramentas utilizadas eram comuns de sistemas TI. Portanto, presume-se que os atacantes estavam extraindo dados das empresas ao longo do tempo.

10. DUQU (2011): O *Duqu* é um *Malware* descoberto em 2011 por um grupo de pesquisadores e foi observado que em muito se parecia com o já conhecido *Stuxnet* no que se refere a sua estrutura de código. Entretanto, o *Duqu* não é autorreplicante e apenas conduz ao reconhecido do sistema de controle em que está armazenado.

Durante a leitura desta seção e de Miller e Rowe (2012), nota-se que a maioria dos ataques a sistemas de controle são realizados utilizando ferramentas comuns de sistemas de TI. Fica evidente também que, depois de 2010, vários ataques foram baseados no *Stuxnet*, seja de forma mais sutil, como em *Flame* de 2012, ou mais explícita como em *Duqu*.

O *Stuxnet* também possibilitou a apresentação de um tipo de ataque que é exclusivo para sistemas de controle, são os ataques físicos. Um exemplo é o chamado ataque de ressonância que força o sistema físico a oscilar em frequência de ressonância até colapsar.

Apesar do grande gerador de vulnerabilidades para os sistemas SCADA ser a necessidade de interconexão de sistemas fisicamente separados através da Internet, que foi possibilitado por meio da utilização de protocolos comuns de TCI e SOs amigáveis, um fator preocupante na segurança de sistemas de controle é a atuação de *insiders*. O primeiro motivo para tal preocupação é que ataques seriam possíveis em qualquer geração do sistema SCADA quando causados por funcionários. O segundo motivo, puramente estatístico, é que até o ano 2000, a taxa de incidentes causadas devido a acidentes ou atuação de funcionários insatisfeitos agindo maliciosamente era de quase 70% (IGURE; LAUGHTER; WILLIAMS, 2006). Esses dados enunciam a fragilidade dos sistemas de controle, uma vez que eles podem sofrer ataques de naturezas diversas e muitas vezes, como foi visto na literatura, o ataque tem origem interna.

3 Fundamentação Teórica

3.1 Foundation Fieldbus

Fieldbus é uma rede de comunicação digital bidirecional criada para uso industrial entre dispositivos de campo e supervisório (VERHAPPEN, 2012). O sistema Foundation™ Fieldbus (FF) surgiu em 1995 quando várias empresas norte-americanas desistiram de esperar a normatização mundial do sistema fieldbus e desenvolveram o seu próprio sistema (FELSER; SAUTER, 2002), um que se moldava às suas necessidades.

O sistema Foundation™ Fieldbus foi baseado no modelo de comunicação OSI que contém sete camadas, sendo elas: física, enlace, rede, transporte, sessão, apresentação e aplicação. O modelo de comunicação tem essa divisão pois a arquitetura com multicamadas apresenta a habilidade natural de realizar a multiplexação de protocolos, dessa forma diferentes protocolos coexistem na mesma infraestrutura (FALL; STEVENS, 2011). Em resumo, os graus de abstração proporcionados pelas diferentes camadas torna mais simples a tarefa de comunicar dois nós distintos. Entretanto, por se tratar de um protocolo relativamente simples, não foram empregadas as camadas três a seis do modelo OSI, apenas as camadas física e de enlace seguida por parte da sétima camada, ou seja, a camada de aplicação. A camada de enlace somada com a Especificação de Mensagem Fieldbus (FMS) — do inglês *Fieldbus Message Specification* — mais a Subcamada de Acesso Fieldbus (FAS) — tradução de *Fieldbus Access Sublayer* — (ambas partes da camada de aplicação do modelo OSI) foram nomeadas como pilha de comunicação, sendo assim a organização do modelo de comunicação do Foundation Fieldbus. As diferenças entre os modelos podem ser observadas na Figura 1.

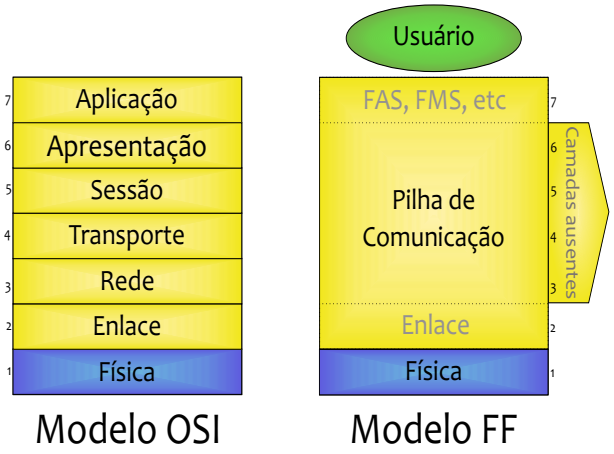


Figura 1 – Comparação entre os modelos de comunicação OSI de sete camadas e o modelo de comunicação utilizado no *Foundation Fieldbus*, as camadas de dois a sete estão embutidas no que foi denominado Pilha de Comunicação. Fonte: *O autor*

Quando é iniciada uma comunicação entre dispositivos, os dados transmitidos passam

por todas as camadas do modelo FF e à medida que a mensagem em transporte é entregue para a camada sucessiva são adicionados cabeçalhos. Cada camada adiciona um novo cabeçalho com a finalidade de possibilitar a interpretação da mensagem no nó de destino (VERHAPPEN, 2012). Esses cabeçalhos, também chamados de *identificadores*, são anexados para determinar qual protocolo ou tipo de informação está sendo utilizado. Um objeto em fluxo dentro da infraestrutura é visto como um dado opaco, diz-se que ele é encapsulado. Além disso, o encapsulamento permite que os dados passem de uma camada para a próxima com a promessa de que a mensagem não será lida enquanto não estiver em seu correto destino, essa é a essência do encapsulamento e a função dos cabeçalhos: permitir o tratamento da mensagem como algo opaco, uma informação não interpretável para intermediários (FALL; STEVENS, 2011). Um exemplo para entender como os dados são transportados entre as camadas é considerar a mensagem como uma carta dentro de um envelope: no envelope estão os dados de cabeçalho, os identificadores, que são adicionados pelas camadas ao longo do caminho, apenas o verdadeiro destinatário poderá ler o que está dentro do envelope.

Na rede Foundation Fieldbus H1, os dados são codificados para transporte utilizando a codificação Manchester. Esse procedimento é realizado visando maior imunidade a ruídos, uma vez que a codificação Manchester exige um sinal de *clock* para verificar o nível lógico da mensagem de tempos em tempos, ou seja, a cada 32 microsegundos ($\pm$ 10%), o sinal transportado é avaliado e codificado, assim ruídos acrescidos entres os intervalos de bordas de subida de clock não serão contabilizados (VERHAPPEN, 2012). Outra vantagem do procedimento é que a codificação Manchester permite uma maior facilidade para recuperar erros de transmissão. O sinal enviado pode ser recuperado mesmo que parte da mensagem se corrompa durante a comunicação, mais informações sobre o método de codificação Manchester e suas aplicações são mostradas em Tanenbaum e Wetherall (2002).

3.1.1 Camadas do Modelo Foundation Fieldbus

Como visto na Figura 1, a topologia Foundation Fieldbus é dividida em camadas baseada no modelo OSI. Serão explicadas as funções de cada uma das camadas presentes no modelo a seguir.

3.1.1.1 Pilha de Comunicação

A principal função da camada de aplicação é criar a sintaxe das mensagens do protocolo. Os principais componentes da camada de aplicação já foram brevemente apresentados no início dessa seção, esses componentes são: FAS e a FMS. Agora esses componentes serão melhor explicados quanto a suas funções na comunicação.

A *Fieldbus Access Sublayer* utiliza-se dos recursos disponibilizados pela camada de enlace para realizar serviços para a *Fieldbus Message Specification*. Por sua vez a FMS permite o envio de mensagens entre aplicações utilizando o barramento de campo. Além disso, essas mensagens devem ter formatos padronizados para diferentes finalidades, se

atentar a isso também é função da FMS. Resumidamente, a FMS se encarrega dos serviços de comunicação, formatos de mensagens, além do comportamento dos protocolos requeridos para construir as mensagens para as aplicações de usuário (VERHAPPEN, 2012).

Os dados transportados através do barramento são descritos por uma estrutura nomeada *descrição de objeto*, sendo que o *Dicionário de Objeto* (DO) armazena todas as descrições de objetos juntas em um encapsulamento. Como foi explicado, o encapsulamento serve para transformar os dados em um pacote opaco além de facilitar a demultiplexação no destinatário para a posterior leitura da mensagem. Portanto, a descrição de cada objeto é obtida em seu respectivo índice do dicionário de objeto. No índice zero do DO, posição nomeada como *header*, são armazenadas informações do próprio dicionário além da posição inicial de alocação das descrições dos objetos gerados na aplicação de usuário. Apenas a partir do índice 255, nas posições determinadas pelo índice zero do DO, estão localizados os dados relativos à descrição dos objetos (VERHAPPEN, 2012).

Cada dispositivo comum empregado no sistema fieldbus contém pelo menos dois *Virtual Field Devices* (VFDs), que são utilizados para verificar, remotamente, os dados descritos no dicionário de objeto. Os dois VFDs obrigatórios nos dispositivos de campo são:

- VFD de Gestão de Sistema e Rede.
- VFD de Aplicação de Usuário.

Por fim, os diferentes VFDs dão acesso à pilha de comunicação, variáveis dinâmicas e estáticas, assim como o *Link Active Scheduler* (LAS) que garante a funcionalidade de *Link Master* a um dispositivo, cuja função é coordenar toda a comunicação no barramento de campo, verificar se os dispositivos estão respondendo, testar dispositivos recém-conectados, remover dispositivos que não estão respondendo e distribuir tokens para acesso ao meio.

3.1.1.2 Camada do Usuário

A camada de usuário define a forma de acessar informações utilizando os dispositivos Fieldbus para posteriormente enviar os dados colhidos para outros dispositivos conectados no barramento (VERHAPPEN, 2012). A arquitetura dos dispositivos fieldbus é baseada em blocos, os chamados blocos de funções, que utilizam-se de algoritmos internos para aferir dados por sensores, realizar cálculos, fechar malhas de realimentação, dentre outras possibilidades. Os blocos de funções contêm algoritmos, armazenamento de dados de entrada e saída e um nome definido pelo usuário (VERHAPPEN, 2012).

Nos dispositivos de campo existem blocos de funções chamados Blocos de Função do Dispositivo Virtual de Campo, que podem ser de três classes: Bloco de Recursos, Bloco de Função e Bloco Transdutor (VERHAPPEN, 2012). Em seguida, serão explicadas as finalidades de cada classe de bloco.

1. Bloco de Recursos: armazena dados sobre o VFD como os seguintes: nome do fabricante, nome do dispositivo, descrição do mesmo, etc. Além disso, essa classe de blocos controla o hardware do dispositivo, os blocos de funções do VFD e status do hardware (VERHAPPEN, 2012).

2. Bloco Transdutor: esse bloco está fortemente atrelado à parte física do dispositivo e dos princípios de mensurações empregados. São capazes de realizar a mensuração de diferentes tipos de energia ou, em outras palavras, propriedade física de um sistema ou parte dele. O valor que será armazenado como dado de leitura apenas é adquirido conhecendo-se o princípio de mensuração do dispositivo. A função do bloco transdutor: possibilitar a criação de dados pela interpretação de diferentes propriedades físicas (VERHAPPEN, 2012).

3. Bloco de Função: nada mais é que um modelo generalizado de mensuração e controle. Dentro dessa classe de blocos existem mais três classes de bloco (VERHAPPEN, 2012):

    a) Bloco Padrão: definido pelo padrão Foundation Fieldbus.

    b) Bloco Melhorado: também definido pelas especificações do Foundation Fieldbus, entretanto essa classe conta com parâmetros e algoritmos adicionais.

    c) Bloco Extendido ou Específico: essa classe é desenvolvida pelos fornecedores e, portanto, não é completamente definida pelos parâmetros Foundation Fieldbus. Entretanto, deve conter parâmetros de Blocos Padrões para assegurar conectividade e comunicação com o barramento.

3.1.1.3 Camada Física

O protocolo Foundation Fieldbus é conhecido por utilizar o par trançado para alimentação e como meio de transporte de dados no barramento de campo. A velocidade padrão de transmissão de dados assegurada no protocolo Foundation Fieldbus é de 31,25 Kbps na rede H1. Entretanto, velocidades superiores podem ser utilizadas desde que sejam empregados *bridges* e *gateways* (VERHAPPEN, 2012).

Com a velocidade padrão de 31.25 Kbps os dispositivos Fieldbus podem se comunicar com até 32 outros dispositivos (máximo) em uma rede, sendo que as seguintes regras devem ser seguidas para a comunicação:

- De 2 a 32 dispositivos desde que a alimentação seja feita utilizando vias diferentes do caminho de transporte de dados e para garantir comunicação com segurança não intrínseca;

- De 2 a 16 dispostivos desde que a alimentação seja feita pelas mesmas vias de transporte de dados e seja garantida segurança intrínseca da comunicação;

- De 1 a 24 dispositivos utilizando vias de alimentação e transporte de dados compartilhadas e não garantindo segurança intrínseca;

Apesar de serem claras, as regras anteriormente apresentadas para conexão de dispositivos de campo não limita o número completamente. Os números de dispositivos permitidos foram calculados considerando a tensão de 24 V_{DC} e corrente aproximada de 9mA nos dispositivos.

3.2 Controlador Lógico Programável

O Controlador Lógico Programável é definido pelo *International Electrotechnical Commission* como (FRANCHI; CAMARGO, 2008):

> Sistema eletrônico operando digitalmente, projetado para uso em um ambiente industrial, que usa uma memória programável para a armazenagem interna de instruções orientadas para o usuário para implementar funções específicas, tais como lógica, sequencial, temporização, contagem e aritmética, para controlar, através de entradas e saídas digitais ou analógicas, vários tipos de máquinas ou processos. O controlador programável e seus periféricos associados são projetados para serem facilmente integráveis em um sistema de controle industrial e facilmente usados em todas suas funções previstas.

O Controlador Lógico Programável (CLP) foi idealizado em 1968 por Richard Morley como uma solução para os problemas intrínsecos dos painéis de relés. Antes do advento dos CLPs, foram utilizados como controladores dispositivos mecânicos que, por sua vez, foram substituídos na década de 1920 por relés e contatores (GROOVER, 2011). Apesar de todos os métodos utilizados para controle serem funcionais, seus pontos negativos são evidentes: baixa vida útil, alto custo e exclusividade do projeto.

Dispositivos mecânicos têm uma pequena vida útil pois se desgastam com o uso, além de serem projetados para uma função única. Por sua vez, os sistemas que empregavam relés exigiam grandes painéis de controle cujo cabeamento era feito manualmente e também deviam ser modificados caso houvesse necessidade de mudança da lógica de controle.

O advento da tecnologia dos CIs possibilitou reduzir o tamanho do circuito de controle de processos, além de serem mais rápidos e terem uma vida útil maior que a dos relés. Por outro lado, a lógica de programação era definida eletricamente através de conexões permanentes, consequentemente a necessidade de mudar o projeto refletia na criação de um novo *design* de máscara de interligações elétricas, o que é um processo demorado e caro.

Alguns computadores comerciais chegaram a ser utilizados em sistemas de controle, mas logo observou-se que era necessário uma alternativa a eles. Os computadores da época tinham grande porte, eram caros e careciam de um cuidado especial das condições do ambiente onde eram instalados. Muitas plantas industriais apresentavam condições hostis que podiam danificar o computador.

Quando os CLPs foram desenvolvidos, o intuito do projeto era suprir todas as deficiências dos outros controladores existentes. Franchi e Camargo (2008) e Groover (2011) destacam as principais qualidades dos CLPs, a saber:

1. Facilidade de programação e reprogramação, o que torna fácil alterar a sequência de operações na linha de montagem;
2. CLPs ocupam menos espaço que painéis de controle de relés;
3. Maior confiabilidade e fácil manutenção;
4. Possibilita expandir sem alterar o sistema, ou seja, executa uma maior variedade de funções que relés;
5. Pode se conectar a sistemas de computadores com facilidade e integrar o banco de dados dos processos a bancos gerenciais da indústria.

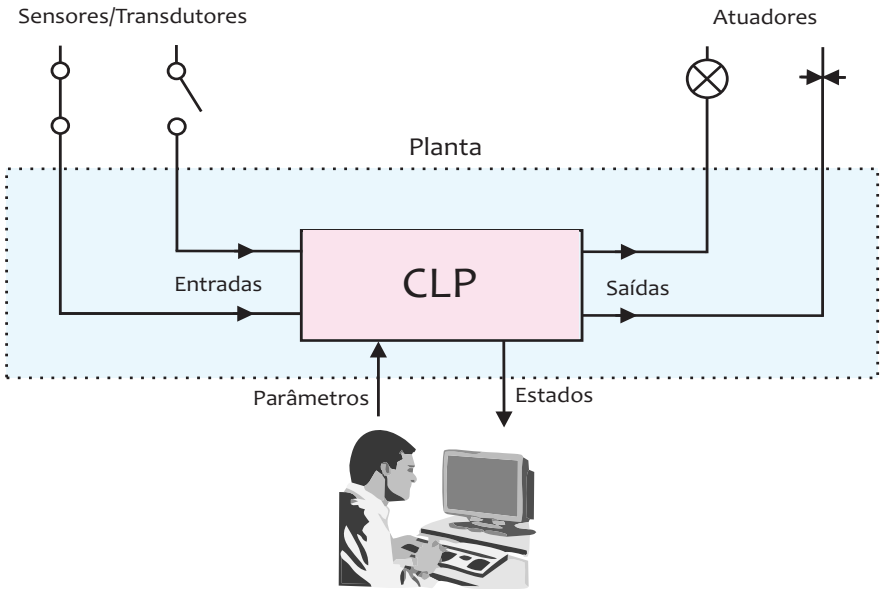


Figura 2 – Controle de processo utilizando um Controlador Lógico Programável e Interface Homem-Máquina: A aquisição de dados através dos transdutores é feita para gerar as entradas do controlador que, por sua vez, determina as saídas através do programa de controle. Fonte: (FRANCHI; CAMARGO, 2008)

Qualquer planta industrial precisa empregar um controlador para garantir o correto funcionamento do processo e viabilizar economicamente o mesmo. Portanto todo sistema de controle pode ser dividido em três partes bem definidas: os transdutores, controladores e atuadores (FRANCHI; CAMARGO, 2008) (vide Figura 2).

- **Transdutores**: são os sensores da planta, dispositivos capazes de converter uma propriedade física em outra. Na planta, ocorre constante monitoramento das entradas, isso é feito utilizando sensores sensíveis a temperatura, pressão, movimento.

Entretanto o CLP apenas reconhece sinais elétricos. O sensor então é capaz de converter uma condição física em um sinal elétrico e assim tornar o dado mensurado legível para o CLP. Um exemplo simples é o botão de pressão momentânea que ao ser pressionado, envia um sinal elétrico ao CLP indicando a condição atual.

- **Atuadores**: são instalados nas saídas do CLP e atuam convertendo o sinal elétrico gerado pelo controlador em uma condição física. É comum a utilização de atuadores que utilizam energia elétrica e pneumática. Os exemplos mais comuns de atuadores são motores, válvulas e pistões.

- **Controladores**: o controlador é a parte fundamental de um sistema de controle pois ele é responsável por gerar saídas de acordo com os estados das entradas utilizando um programa de controle. Os sinais de saída, gerados pelo controlador, são responsáveis por modificar o estado dos atuadores e, assim, ativar motores, acionar bombas, regular válvulas.

3.3 Indústria 4.0

A indústria 4.0 é o fruto da união entre a produção industrial e tecnologias de informação e comunicação. Essa união marcou de tal forma a história que a concedeu o atributo de Quarta Revolução Industrial, a atual revolução e primeira do tipo que foi prevista (HERMANN; PENTEK; OTTO, 2016). Apesar de ser frequentemente discutida, a Indústria 4.0, ainda não é bem definida em texto, apenas definiram-se as principais ideias almejadas pela evolução. Em decorrência disso, em 2016, cinco anos depois do surgimento do nome, ainda não havia uma terminologia para Indústria 4.0, o que dificulta o estudo acadêmico do processo (HERMANN; PENTEK; OTTO, 2016).

Nas recomendações de implementação são adicionados os três componentes principais da Indústria 4.0: Cyber-Physical Systems (CPS), fábricas inteligentes e internet das coisas - tradução para o termo original *Internet of Things* (IoT). De certa, forma todos esses componentes estão intrinsecamente relacionados na Indústria 4.0, fábricas inteligentes são possíveis através de IoT e CPS. Com a IoT é possível interligar dispositivos industriais em uma mesma rede, onde cada dispositivo tem seus próprios endereços MACs. Além disso, é possível realizar armazenamento e análise de dados de forma centralizada ou descentralizada (em cada dispositivo). A convergência dos dispositivos industriais e o mundo virtual resulta nos CPSs que integram máquinas inteligentes, indústrias de produção, sistemas de segurança de seus depósitos, tudo interligado a banco de dados permitindo maior flexibilidade e eficiência da administração e sistema de controle (KAGERMANN et al., 2013).

O emprego de CPS, IoT e demais tecnologias necessárias para formar a indústria inteligente torna a supervisão do processo bem mais simples e reduz o tempo de acesso aos dados. A nova Revolução ainda possibilita o aumento da complexidade de produtos

e sistemas de controle o que é acompanhado, devido a sua natureza, pelo número de vulnerabilidades. Portanto, é necessário haver um melhor controle de acesso ao sistema de controle central, uso de arquiteturas com segurança integrada e empregar profissionais qualificados para detectar e identificar vulnerabilidades ou mitigar/impedir ataques ou uso indevido do sistema de supervisão e controle (KAGERMANN et al., 2013).

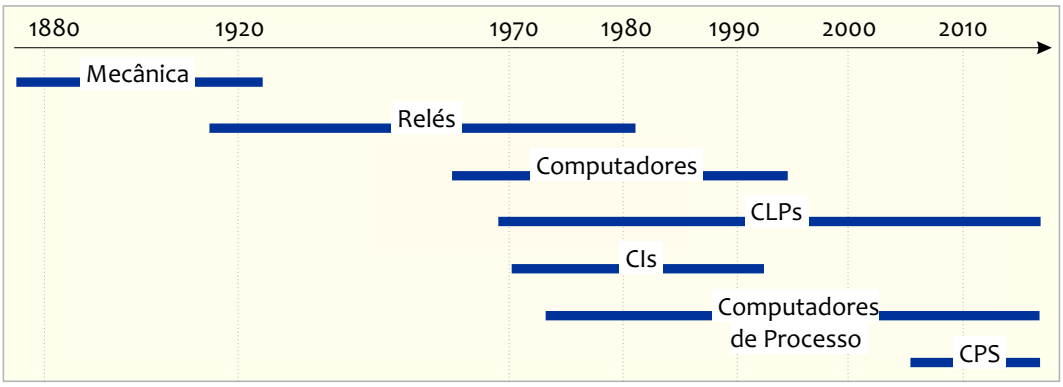


Figura 3 – Evolução dos controladores ao longo da história com representação gráfica do período de utilização de cada tipo. Fonte: O autor

A Figura 3 mostra a evolução da tecnologia empregada em sistemas de controle ao longo da história, sendo que o dispositivo mais utilizado atualmente é o CLP e é de se esperar que no futuro haverá uma participação maior do CPS no ambiente industrial. Como pode ser observado o CPS apenas foi empregado nos últimos anos, é uma tecnologia do século XXI. O CLP já representa uma tecnologia um pouco mais antiga mas que recentemente foi melhor explorado com o uso de tecnologias de comunicação e informação permitindo a conexão com a Internet. Os CPSs foram planejados com o intuito de permitir a interligação virtual entre diferentes dispositivos. Uma vez que a adição de TCI possibilita a adição de vulnerabilidades a um sistema de controle, percebe-se que, uma vez que a tendência é aumentar a interconectividade entre dispositivos e tecnologias, o número de vulnerabilidades está aumentando em sistemas de controle, o que requer medidas de segurança contra ataques.

3.4 Ataques Comuns

A crescente tendência do aumento de interconectividade entre dispositivos de chão de fábrica reforçada pela ascendente Indústria 4.0 modificam completamente o cenário original dos sistemas SCADA o que possibilita a inclusão de vulnerabilidades. É válido lembrar que a tecnologia de comunicação permite a existência de mais de um ponto de acesso em qualquer rede, incluindo as redes SCADA. Além disso a segurança digital não é garantida pelas barreiras físicas pois qualquer nó ou má configuração pode permitir acesso com o exterior da infra-estrutura (IGURE; LAUGHTER; WILLIAMS, 2006). Em Byres e Lowe (2004) é afirmado que a dificuldade de conhecer o protocolo SCADA e suas regras de comunicação tornam o planejamento de um ataque quase impossível devido ao uso dos

protocolos proprietários. Entretanto, a substituição desses protocolos pelos de padrão aberto simplificou os sistemas SCADA e a troca de dados entre dispositivos, em consequência um ataque tornou-se algo relativamente fácil e comum (IGURE; LAUGHTER; WILLIAMS, 2006). Devido a isso, estudar os principais tipos de ataques e o comportamento dos sistemas sob ataque se tornou tarefa de alta importância em infra-estruturas críticas. Nas seguintes subseções serão apresentados alguns dos ataques mais comuns e mais estudados na literatura a fim de tornar o leitor ciente.

3.4.1 Denial of Service (DoS) e Distributed Denial of Service (DDoS)

Os ataques conhecidos como *Denial of Service* (DoS) e *Distributed Denial of Service* (DDoS) são responsáveis por causar a negação de serviço no computador ou dispositivo embarcado alvo, ou seja, a função para qual o dispositivo foi projetado não será mais executada devido ao ataque. Um ataque de negação de serviço é chamado de DDoS quando orquestrado por um número de computadores maior que um, pois nesse caso o ataque tem várias fontes, caracterizando um ataque DoS distribuído (WRIGHT; STEVENS, 1995). Os métodos utilizados por um ataque de negação são complexos e podem variar de acordo com o perfil de tráfego nominal do alvo. Por outro lado, como este trabalho não está interessado na forma como o ataque é executado mas sim no estudo das anomalias causadas durante o mesmo, assim como as técnicas de detecção e mitigação, será fornecida uma definição breve sobre os ataques DoS e DDoS. De acordo com Sun, Ngan e Chao (2009) no ataque DDoS os dispositivos de rede de um computador recebem uma quantidade enorme de dados das fontes maliciosas causando sobrecarga e deixando os dispositivos incapazes de executar a comunicação usual, uma vez que os dados maliciosos estão ocupando todo o *buffer*. Uma forma de mitigar este ataque é mostrada em Sun, Ngan e Chao (2009), onde é desenvolvido um sistema baseado no *Leaky-Bucket* que descarta pacotes que não se parecem com o perfil de tráfego nominal antes de serem processados, reduzindo assim a taxa de dados maliciosos que preenchem o *buffer* e impedem a entrada de dados verdadeiros. O problema do sistema apresentado é que ele não elimina pacotes maliciosos que parecem legítimos, por isso ele apenas mitiga o problema.

3.4.2 Ataque Replay

Em contraste com os ataques DoS, tratados anteriormente, ataques de enganação são mais dificilmente detectados em função de sua natureza. Uma vez que existe fluxo constante de dados, o operador entende que a planta está funcionando perfeitamente, entretanto isso não ocorre devido ao ataque em execução. O *Replay*, nome dado ao procedimento, é em função do processo empregado no sistema. Mo, Chabukswar e Sinopoli (2014) explicam o procedimento empregado utilizando a sequência de passos citados adiante. É necessário supor que o agente malicioso tem acesso à todas leituras dos sensores

da planta e pode modificá-las caso queira. Os valores modificados recebem o nome de y'_k. Também é possível injetar entradas no sistema e elas recebem o nome de u_k^a .

Uma vez definidas as capacidades do agente, será mostrada a estratégia empregada, que pode ser dividida em duas etapas:

1. As saídas y_k do sistema são gravadas durante um tempo desejado sem que nenhuma entrada seja injetada no sistema.

2. O agente reproduz as saídas y_k gravadas no passo anterior enquanto envia uma sequência de comandos para o sistema.

Para ilustrar melhor o uso da técnica, deve-se considerar que a dinâmica do sistema é desconhecida pelo atacante, isso justifica a coleta dos dados. Os dados adquiridos no primeiro passo do ataque correspondem as saídas do sistema em regime permanente, portanto alterações significativas no estado das saídas não são esperadas, logo os dados coletados são utilizáveis. Por fim, durante todo o procedimento, as técnicas de controle ficam comprometidas, o que é de se esperar já que o verdadeiro estado do sistema se torna desconhecido, a reprodução das antigas leituras dos sensores camuflam o que está realmente acontecendo na planta, é neste instante que o atacante inicia o envio de comandos para o sistema.

Nishiya, Hasegawa e Koike (1982) embasam seus métodos de constatação de dados ruins utilizando estimação de estados $\hat{x}_{k|k-1}$ pelo filtro de Kalman e verificando a qualidade da aproximação por χ^2. O grande problema relacionado a esse método reside na obrigatoriedade do sistema ser resiliente ao ataque. Em outras palavras, considerando um sistema descrito em tempo discreto como

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + \omega_k \\ y_k &= Cx_k + \nu_k \end{aligned} \tag{3.1}$$

em que serão adicionados controladores de realimentação e estimação de estados. Em (3.1) as variáveis presentes nas equações representam: vetor de estados (x_k), matriz de estados (A), vetor de entrada (u_k), matriz de entrada (B), ruído de processo ou multiplicativo (ω_k), vetor de saída (y_k), matriz de saída (C), matriz de transmissão direta (D) e ruído de medição ou aditivo (ν_k).

Sendo K e L as matrizes encontradas por meio do Controlador Linear Quadrático Gaussiano LQG (HESPANHA, 2009), a Lei de Estimação de Estados e a Lei de Realimentação por são dadas por

$$\hat{x}_{k+1|k} = (A + BL)(I - KC)\hat{x}_{k|k-1} + (A + BL)Ky'_k$$

$$u_k = -K\hat{x}_k.$$

Por fim, Mo, Chabukswar e Sinopoli (2014) mostram que se $\mathcal{A} \triangleq (A + BL)(I - KC)$ for estável, a taxa de detecção de falhas converge para a taxa de erros de detecção. Em contrapartida, quando $\mathcal{A}$ é instável, o sistema é dito resiliente ao ataque e a taxa de detecção de falhas tende a 1, o que é um resultado excelente.

3.4.3 Ataque de Integridade

Os ataques de integridades são responsáveis por corromper dados de sensores em uma planta de controle, tornando seus dados inúteis para o controle do processo devido a representação errônea de uma ou mais saídas do sistema. Um tipo de ataque de integridade foi tratado com maiores detalhes na Subseção 3.4.2. Um estudo amplo sobre este assunto é feito em (GIANI et al., 2008), em que relata-se o uso de uma bancada de testes projetada para detectar um sensor corrompido por meio de sistemas de detecção que serão abordados adiante.

3.4.4 Phishing - Capturando dados relevantes no tráfego

Segundo Giani et al. (2008) os ataques *Phishing* permitem acesso a informações confidenciais sobre um sistema e comumente estão relacionados a um ou mais ataques que serão executados em seguida (DONDOSSOLA et al., 2008). Uma das formas de realizar o *phishing* é através de um *sniffer* de pacotes, cuja função é interceptar e registrar os pacotes que passam por uma rede ou parte dela. Um *sniffer* pode ser um *software* ou *hardware* que permite colocar a placa de rede de um dispositivo em modo promíscuo, assim todos os pacotes em tráfego serão capturados, decodificados e analisados, nesse modo não importa se o dispositivo é o verdadeiro destinatário do pacote, tudo que for interceptado será registrado (JUNG; SONG; KIM, 2008).

Como qualquer outra ferramenta de TI, o *sniffer* tem várias aplicações, pode ser usado para identificar e retificar problemas de conexão entre servidor e cliente, ou falhas na implementação de protocolos, também pode ser usado para gerar estatísticas de uso de dados ou filtrar conteúdo suspeito. Entretanto o *phishing* pode ser usado para capturar mensagens na rede e permitir a obtenção de informações confidenciais, em seguida o intruso pode ganhar posse dos principais dados e comandos de controle do sistema já que a maioria dos protocolos SCADA não apresentam criptografia de dados (IGURE; LAUGHTER; WILLIAMS, 2006).

3.5 Análise Comportamental de Sistemas de Controle

Nesta Seção serão analisados métodos de detecção e identificação de ataques em um sistema com base nas saídas do mesmo. Como é sabido, ataques de integridade são capazes de alterar os dados referentes às leituras dos sensores tornando-as muitas vezes inutilizáveis para um processo de controle com realimentação. Entretanto, a capacidade

de continuar operando durante um ataque por meio de estimação das saídas do sistema é estudada em (ZHU; MARTINEZ, 2011) e (FAWZI; TABUADA; DIGGAVI, 2011).

Um CPS sob ataque é modelado como Linear e Invariante no Tempo (LIT) em Pasqualetti, Dörfler e Bullo (2013) para posterior estudo de detecção e identificação de ataques. O sistema LIT considerado no trabalho foi

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3.2}$$

esse modelo pode ser utilizado para tempo discreto apenas realizando as modificações de notação necessárias. Entretanto é necessário considerar que as entradas, representadas por $u(t)$, são desconhecidas e cada estado pode ser modificado pelo ataque, dessa forma o estudo mantém a generalização comum em um ataque. As entradas relacionadas aos comandos de ataque são representadas por $u_{\mathbb{K}}(t)$, pois $\mathbb{K}$ representa a presença de ataque no sistema. Após essas considerações seguem as definições realizadas em (PASQUALETTI; DÖRFLER; BULLO, 2012):

1. **Ataque indetectável**: a entrada $u_{\mathbb{K}}$ excita apenas o estado zero do sistema descrito por (3.2). Ou seja, considerando condições iniciais x_1 e $x_2 \in \mathfrak{R}^n$, $\forall\, t \in \mathfrak{R}_+^*$ tem-se $y(x_1, u_{\mathbb{K}}, t) = (x_2, 0, t)$.

2. **Ataque não identificável**: um ataque $\mathbb{R}$ é não identificável caso apresente suas entradas de controle $u_{\mathbb{R}} \neq u_{\mathbb{K}}$ mas as saídas sejam $y(x_{\mathbb{K}}, u_{\mathbb{K}}, t) = y(x_{\mathbb{R}}, u_{\mathbb{R}}, t)$.

Em Pasqualetti, Dörfler e Bullo (2013) são projetados monitores para detectar e identificar ataques. Segundo os autores, um monitor é um algoritmo determinístico com acesso constante às saídas e dinâmica do sistema. As saídas do monitor correspondem a respostas das seguintes perguntas:

- Existe algum ataque em andamento?
- Quais ataques estão sendo efetuados?

O retorno do monitor é visto como $\Psi(\Lambda) = \{\Psi_1(\Lambda), \Psi_2(\Lambda)\}$, sendo que $\Psi_1(\Lambda) \in \{\textit{Sim, Não}\}$ e $\Psi_2(\Lambda) \subseteq \{\mathbb{K}_1, \ldots, \mathbb{K}_n\}$ sendo esse último um conjunto contendo os ataques possíveis. É fácil perceber que um ataque é não identificável caso não esteja presente no conjunto $\Psi(\Lambda)$, além disso ele é não identificável para o caso já mencionado em que as respostas do sistema durante o ataque são exatamente iguais às resposta em um intervalo de funcionamento comum.

Considerando um conjunto de ataques detectáveis, é possível projetar monitores centralizados, descentralizados e distribuídos a partir das técnicas demonstradas em Pasqualetti, Dörfler e Bullo (2012). Para o *design* de um monitor centralizado é utilizado um filtro de detecção baseado no estimador de estados conhecido como observador de

Luenberger. Esse filtro detector é dado em espaço de estados como

$$\begin{aligned} E\dot{w}(t) &= (A + GC)w(t) - Gy(t) \\ r(t) &= Cw(t) - y(t) \end{aligned} \tag{3.3}$$

Sendo que $w(0) = x(0)$, $r(t)$ a matriz do resíduo e a matriz G é tal que $G \in \mathfrak{R}^{n\times p}$ tal que $(E, A + GC)$ é regular e de Hurwitz.

Salienta-se que, na ausência de ataques, a diferença $w(t) - x(t)$, referente ao erro da resposta do filtro, é exponencialmente estável e $r(t) = 0$, $\forall\ t \geqslant 0$. Alguns ataques podem se passar por indetectáveis nesse método. Caso um filtro com sistema semelhante a (3.3) tenha seus autovalores com parte real menor que uma constante real positiva c tal que na ausência de ataques o sistema se estabilize rapidamente em zero, qualquer ataque que utilize entradas cujo tempo de estabilização em zero seja menor que e^{-ct} nunca são detectados pelo filtro escolhido. Para escapar dessa situação é necessário realizar um novo projeto com o valor absoluto de c menor que o tempo de extinção do sinal de controle do ataque (PASQUALETTI; DÖRFLER; BULLO, 2012). Outro problema relacionado a esta técnica se refere a consideração de ruídos de processo e medição no caso estocástico. Para esses casos é recomendado utilizar as técnicas apresentadas à seguir.

3.6 Filtro de Kalman

O Filtro de Kalman (FK) foi desenvolvido em 1960 por Rudolph Emil Kalman com o propósito de resolver problemas lineares relacionados a filtragem de dados discretos. Mais adiante, foram desenvolvidas extensões dessa técnica que permitem sua aplicação para dados contínuos e casos não lineares, o que torna esse filtro largamente utilizado na teoria moderna de controle.

O Filtro de Kalman é um estimador de estados estocástico que apresenta erro quadrático mínimo. O FK pode ser utilizado no processo estático e dinâmico, sendo que para o último é imprescindível o conhecimento da dinâmica do sistema para prever suas características em um instante futuro.

Pode-se entender uma medição obtida por meio de um instrumento como um valor provável traduzido pelo transdutor que representa a condição analisada pelo mesmo. Ou seja, um sensor muitas vezes representa uma condição analisada com um pequeno erro, portanto se for criada uma coleção de dados obtidos pelo mesmo, pode ser observado que essa coleção possui uma média μ e uma variância σ^2 (AGUIRRE, 2004). Para simplificar ainda mais o entendimento do que é proposto, normalmente é atrelada uma curva gaussiana às mensurações realizadas por um sensor, essa curva mostra a função de densidade de probabilidade (FDP) de se representar corretamente a grandeza amostrada.

Kalman propõe a estimação de variáveis relacionadas ao processo com um sistema baseado na Teoria Bayesiana, isso pode ser afirmado porque o Filtro de Kalman tem uma estrutura preditora-corretora.

Considerando sempre o caso discreto, tido que é conhecida a saída do sistema em um instante k_1, o FK é capaz de calcular a saída do sistema em um instante futuro k_2 baseando-se apenas na resposta do sistema no instante inicial e na FDP das leituras dos sensores, como mostrado na Figura 4.

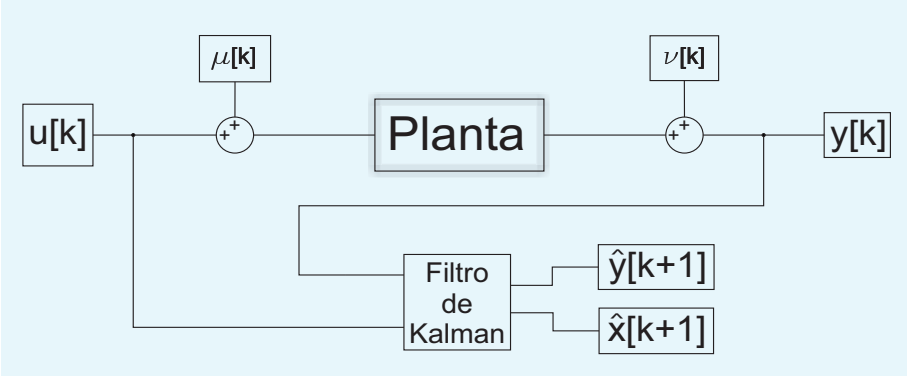


Figura 4 – Configuração de utilização do Filtro de Kalman quanto a suas entradas e saídas. Fonte: O autor

Na primeira etapa do FK é feita a propagação da FDP do k_1 para o k_2 considerando a dinâmica do sistema. Ressalta-se que a variância $\sigma^2_{(k|k)}$ relacionada ao instante k_1 é menor que $\sigma^2_{(k+1|k)}$ porque é levada em consideração a incerteza relacionada à medição. Ou seja, o FK assume diferente de zero o ruídos de medição ou ruído de adição (AGUIRRE, 2004). A etapa de propagação é realizada por

$$\begin{cases} \hat{x}_{(k+1|k)} = A\ \hat{x}_{(k|k)} + B\ u(k) \\ P_{(k+1|k)} = A\ P_{(k|k)}\ A^T + B\ Q\ B^T \\ K = P_{(k+1|k)}\ C^T\ \left[C\ P_{(k+1|k)}\ C^T + R\right]^{-1} \\ \hat{x}_{(k+1|k+1)} = \hat{x}_{(k+1|k)} + K\left(y_{(k+1)} - C\ \hat{x}_{(k+1|k)} - D\ u_{(k)}\right) \\ P_{(k+1|k+1)} = P_{(k+1|k)} - K\ C\ P_{(k+1|k)} \end{cases} \tag{3.4}$$

Em uma segunda etapa do cálculo estabelecido pelo FK é feita a correção da predição feita no passo anterior. Nesse segundo instante, ainda em k_2, considera-se que já é conhecido o valor lido pelo sensor em k_2, portanto é possível identificar as divergências entre as saídas estipulada e lida pelo sensor. Uma vez que a saída encontrada pelo filtro de Kalman no passo anterior recebe a nomenclatura de $\hat{y}_{(k+1|k)}$, a saída obtida pelo sensor é tomada como $y_{(k+1)}$.

Por último, deve se considerar algumas situações para garantir o bom funcionamento da filtragem. Sabendo que as variáveis Q e R são constantes do Filtro de Kalman que se relacionam ao sistema e às medições recebidas dos sensores. Pode-se perceber em (3.4) que Q e R são intrínsecas à matriz de covariância e ao ganho K, respectivamente. No primeiro caso supõe-se que o sistema adicione uma alta parcela de ruídos nas medições, o que nos levar a ter uma maior confiança em nosso modelo de comportamento do que nos dados lidos pelos sensores. Nesse caso, deseja-se que os dados não influam significativamente na próxima saída do filtro, essa interferência, como sabemos, acontece na etapa de correção da

estimação do filtro. Portanto, para reduzir esse impacto, é necessário ter $\sigma^2_{(k+1)}$ e $K_{(k+1)}$ pequeno. O segundo caso é quando a confiança nas medições é maior que no modelo, ou seja, é utilizado quando há pouca confiança no modelo porque ele é incerto. Portanto é necessário que o as medições impactem mais na saída do filtro do que o resultado da propagação, nesse caso é preciso que $\sigma^2_{(k+1)}$ seja grande e $K_{(k+1)} \to 1$ (AGUIRRE, 2004).

Na prática, a escolha dos valores de Q e R que melhor satisfazem às necessidades de cada caso é feita empiricamente. Os valores utilizados devem satisfazer à condição de se obter um resíduo branco gaussiano. Matematicamente $(\xi_{(k+1)} = \hat{y}_{(k+1|k)} - y_{(k+1)})$ deve apresentar média $\mu = 0$ e matriz de correlação dada por $R(a) = \varepsilon\{\tilde{x}_{(k)}\tilde{x}'_{(k+a)}\} = \tilde{X}\delta(a)$, sendo $\delta(a)$ o impulso unitário.

3.7 Detecção de falhas

À medida que se trabalha com dados sensíveis na indústria, deseja-se aumentar a confiança sobre o mesmos. Uma forma comum para atingir uma maior confiabilidade dos dados é utilizando sensores cada vez mais precisos, entretanto essa prática não é imune aos possíveis ataques que uma indústria é vulnerável. Para aumentar a confiança nos dados obtidos por sensores são empregadas técnicas de detecção de falhas no conjunto de dados gerados pelos sensores. Normalmente são realizados testes com os dados agrupando-os de acordo com o sensor de origem.

3.7.1 Teste de qualidade de ajuste com χ^2

Um dos primeiros métodos de detecção de falhas de sensores empregados em sistemas elétricos era bastante simples e apresentava um baixo desempenho. Trata-se do método apresentado em Nishiya, Hasegawa e Koike (1982), que utiliza o Filtro de Kalman para calcular, a cada iteração, o valor normalizado da inovação do processo. Cada iteração representa um novo instante de coleta de amostras do sistema, os cálculos são realizados para procurar mudanças abruptas na dinâmica do sistema, uma forma de contabilizar isso é utilizando o resíduo normalizado. À partir do valor obtido pode-se dizer se o valor utilizado naquela iteração é bom ou ruim, baseando-se em comparações com um limiar definido previamente por meio de ensaios. Caso o valor instantâneo seja caracterizado como ruim, significa que ele sofreu algum erro de transporte ou é fruto de uma medição muito imprecisa, em último caso, se muitas amostras sequenciais forem classificadas como ruins pode significar que o sistema foi bastante modificado e o modelo utilizado no Filtro de Kalman deixou de representar a planta.

Posteriormente, foi introduzido um método de detecção de falhas com a utilização do valor do χ^2. O teste de qualidade de ajuste com o χ^2 possibilita verificar se uma determinada amostra pertence à uma distribuição normal teórica que corresponde ao conjunto de dados coletados precedentemente. Como discutido, a distribuição normal é definida por um valor de média e variância. Portanto se deseja representar um conjunto

de dados coletados por sensores de uma planta como uma distribuição normal devemos ter conhecimento desses parâmetros característicos. Entretanto, como se deseja realizar os cálculos com o menor atraso possível, não é possível realizar um levantamento de um conjunto de dados razoavelmente grande para extrair os parâmetros da distribuição a que seguem.

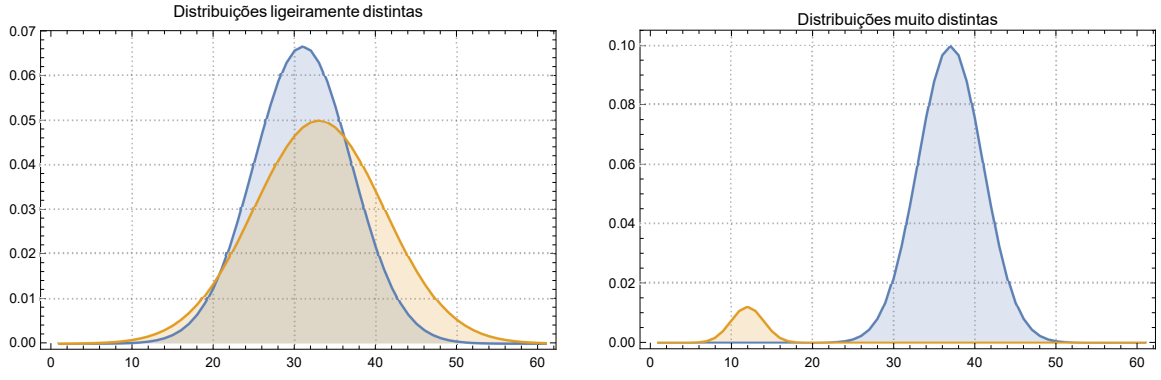
Esse parâmetro relaciona os dados estimados com a média aritmética $\hat{\mu}_k$ e desvio padrão $\hat{\sigma}_k$ das amostras por meio de

$$f'(y_k) = \frac{1}{\sqrt{2\pi\hat{\sigma}_k}} \exp\left[-\frac{(y_k - \hat{\mu}_k)^2}{2\hat{\sigma}_k}\right],$$

em seguida, o valor do χ^2 é dado por

$$\chi^2_k = \sum_{s=1}^{s_{max}} \frac{(f_{k,s} - mF_s)^2}{mF_s}$$

em que s é o número de intervalos utilizados para dividir a distribuição de dados, F_s representa a probabilidade da amostra $y_{k,i}$ existir dentro de um determinado intervalo e $f_{k,s}$ é a frequência com que $y_{k,i}$ pertence a cada intervalo e mF_s mostra a frequência teórica. Finalmente, o dados analisados são classificados como falsos ou ruins caso $\chi^2 \geqslant \chi^2_{max}$, sendo esse último valor determinado pela tabela de distribuição de Qui-Quadrado - χ^2_n que depende da probabilidade do sucesso e o número de graus de liberdade, como mostrado no anexo A.



(a) (b)

Figura 5 – Distribuições normais comparando conjunto de dados do estado estacionário e conjunto com presença de falha: (a) variação repentina de estados do sistema; (b) dados ruins, devem ser descartados pelo algoritmo.

A Figura 5 mostra dois casos de falhas nos sistemas que são estudados pelos teste de qualidade de ajuste por χ^2. A distribuição representada pela curva em azul mostra o funcionamento ordinário de um sistema em estado estacionário, a curva em laranja representa uma distribuição com algum tipo de erro. Como exemplo básico toma-se as Figuras 5a e 5b, o primeiro exemplo mostra um conjunto de dados com um leve deslocamento do valor médio esperado para a distribuição, essa distribuição ocorre quando surge uma

variação brusca dos estados do sistema, normalmente é uma situação circunstancial e de curta duração. O caso mostrado na Figura 5b apresenta um problema diferente, como se observa não existe superposição entre as duas distribuições, logo os elementos de ambas em nada se assemelham. Em um ensaio que se espera obter amostras pertencentes à distribuição em azul, a obtenção de amostras pertencentes à distribuição em laranja configura um comportamento inesperado do sistema, isso significa que o modelo utilizado no Filtro de Kalman não representa fielmente o sistema estudado, o que resulta em altos valores de resíduos.

3.7.2 Teste de qualidade de ajuste com autocorrelação

Partindo do princípio que o controle de um sistema é um processo estocástico torna-se útil verificar a dependência de uma amostra com seus precedentes. A natureza estocástica dos processos de controle fica evidente com o estudo do conjunto de dados gerados pelo sistema supervisório, dados pertencentes a uma distribuição que não é completamente fiel ao sistema de equações diferenciais que regem o modelo. As divergência encontradas entre o resultado esperado e o observado ocorrem devido aos fenômenos não modelados, basicamente devido às aproximações do modelo e adição dos ruídos de processo e medição.

Idealmente as distribuições originadas por processos estocásticos têm seus valores independentes e igualmente espaçados, essa suposição normalmente é violada na prática e isso justifica a utilização do cálculo de autocorrelação para investigar a confiabilidade de uma distribuição. No que se refere à investigação de contaminação dos dados críticos de sistemas de controle por ataque *replay*, a autocorrelação da amostra atual com as amostras anteriores mostra o grau de semelhança do conjunto como um todo. É comum existir uma pequena semelhança entre as amostras, isso é resultado de alguma tendência que o sistema pode ter e que não foi modelada. Na ocorrência do ataque *replay*, o grau de semelhança das amostras cresce exageradamente, ultrapassando o limiar pré-estabelecido e informando que o sistema está sobre ataque. O cálculo da autocorrelação considerando um *lag* k é dado por $r_k = \frac{c_k}{c_0}$, sendo

$$c_k = \frac{1}{N}\sum_{n=1}^{N-k}(y_n - \bar{y})(y_{n+k} - \bar{y})$$

$$c_0 = \frac{1}{N}\sum_{n=1}^{T}(y_n - \bar{y})^2$$

Os valores da autocorrelação para diferentes instantes de tempo são colocadas em um gráfico chamado correlograma, que é montado utilizando as fórmulas elencadas acima (BOX et al., 2015).

A criação de um correlograma exige a manutenção de um conjunto de dados antecedentes para verificação do grau de semelhança, essa é uma diferença importante com relação ao teste apresentado anteriormente, o qui-quadrado. Uma boa forma de criar um sistema de análise de dados que não utilize uma quantidade de memória exagerada e consiga atingir tempo hábil é guardar um número limitado de amostras antigas e realizar

o cálculo de autocorrelação em um sistema de janela móvel. Assim, a cada nova iteração do algoritmo, o dado mais antigo é desprezado e acrescenta-se o valor mais atual recebido. À partir da análise da autocorrelação é verificada a ausência de ataques no sistema e o resultado é enviado para o operador.

4 Estudo de Caso

Os métodos de reconhecimento de ataques a sistemas supervisórios altamente integrados, apresentados no Capítulo 3, foram testados em um ambiente especial e os resultados obtidos são explanados neste capítulo. Inicialmente é necessário salientar que o ambiente de testes criado teve sua implementação baseada em simulação e execução, isso significa que em parte foi utilizado uma planta para os ensaios e em outros momentos foram realizados testes computacionais nos dados coletados pelo supervisório a fim de se encontrar tendências que indiquem instantes sob ataque. As seções seguintes descrevem todos os procedimentos necessários para o estudo feito neste trabalho.

4.1 Modelagem do Sistema

A planta utilizada no estudo é a SMAR PD3-F, dela foi utilizada uma malha responsável pelo contole do nível de água no tanque de aquecimento. O levantamento do sistema foi realizado por meio da modelagem caixa preta ou empírica. Inicialmente colocou-se o sistema em malha aberta com uma entrada fixa $u_v[k] = 0.6$, sendo que esta entrada representa a porcentagem de abertura da válvula pneumática que controla a vazão para o reservatório 1. Após o sistema entrar em estado estacionário, iniciou-se o envio de um sinal $u_v[k]$ pseudoaleatório de média 0.6 para o controle da válvula pneumática e foi feita a coleta dos dados referentes ao nível.

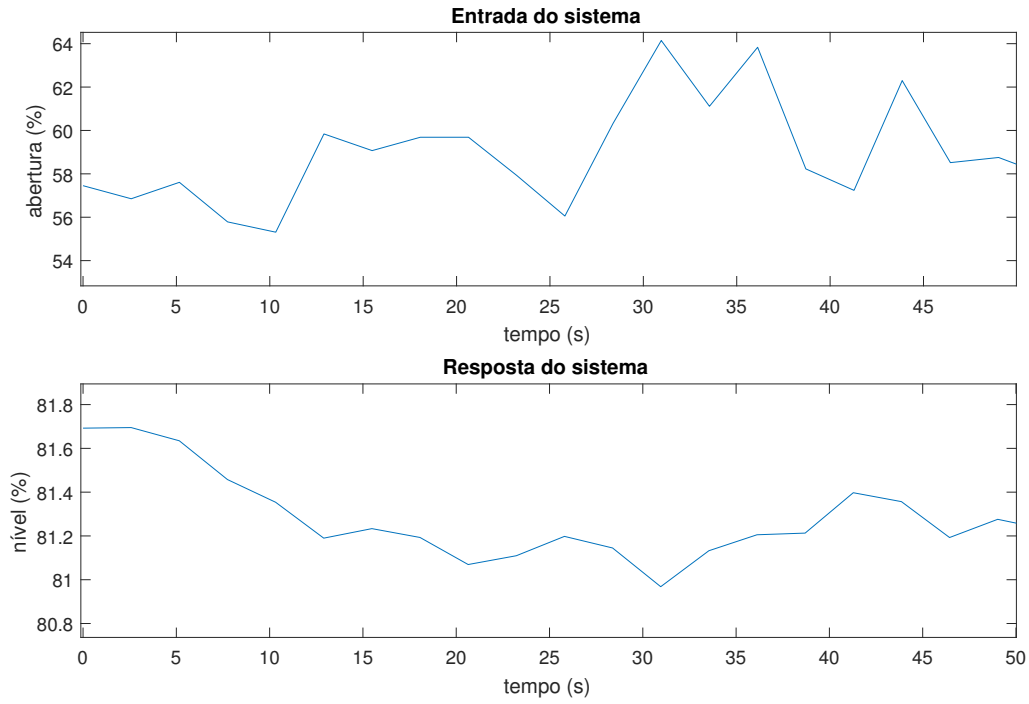


Figura 6 – Representação gráfica da dinâmica do modelo mostrando parte das amostras de entrada e saída do sistema utilizadas na estimação. Fonte: O autor

Na Figura 6 está representada a dinâmica do sistema para um sinal de entrada $u_v[k]$

branco gaussiano durante o intervalo de zero a cinquenta segundos, vale ressaltar que a estimação do modelo utilizou 1980 amostras e o gráfico apenas cumpre a função ilustrativa. É possível observar, mesmo no intervalo selecionado, que a entrada do sistema cumpre o pré-requisito de um sinal aleatório com média 0.6. Como se trata de um sistema de controle de nível de água em um reservatório, trata-se de um sistema relativamente lento, portanto há evidências que o sistema está respondendo adequadamente às entradas, mas isso não é instantâneo.

À partir dos conjuntos de dados gerados por $u_v[k]$ e $y[k]$ foram estimados os parâmetros do sistema para um modelo ARX utilizando o método dos mínimos quadrados descrito em (AGUIRRE, 2004). O modelo estimado está destacado em (4.1).

Modelo ARX estimado :

$$y(k) = -1.07y(k-1) - 0.091y(k-2) + 0.0058u(k-1) - 0.019u(k-2) \quad (4.1)$$

Período de amostragem de 2.58 s

Na Figura 7 são mostradas as comparações das medições com as predições livre e um passo a frente, o que mostra que o modelo é adequado para utilização futura.

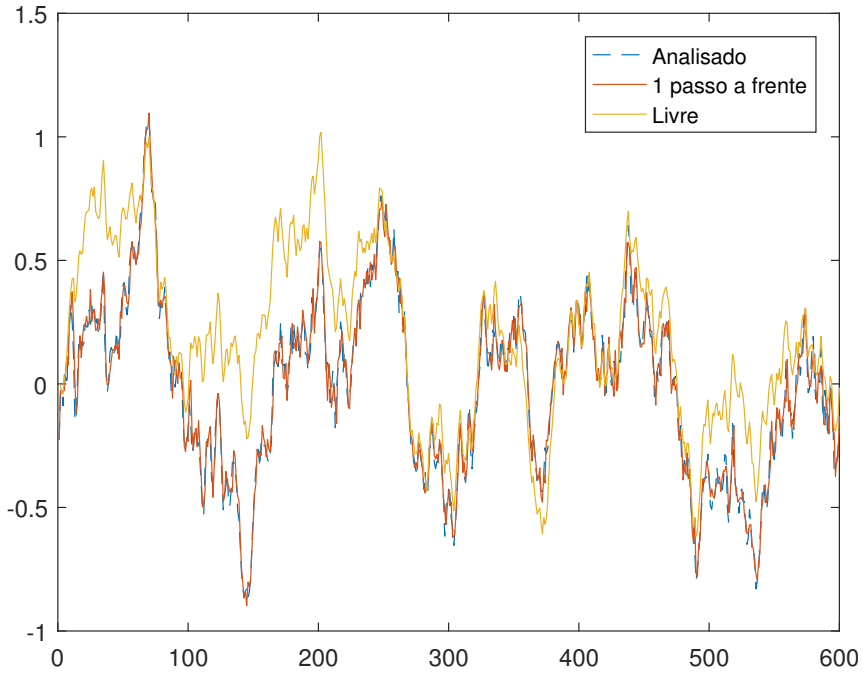


Figura 7 – Validação do modelo estimado com predições um passo a frente e livre. Fonte: O autor

4.2 Ensaios

Após a obtenção do modelo característico do sistema criado na planta SMAR PD3-F, iniciou-se uma sequência de ensaios para coleta e posterior teste dos dados correspondentes à dinâmica do sistema. Os ensaios realizados consistem na inclusão de diferentes controladores PI sintonizados pelo método de Ziegler-Nichols. Foram utilizados diferentes ganhos K_p e K_i nos controladores a fim de aproximar a saída do sistema com o encontrado nas simulações, o que é algo difícil visto que o primeiro ambiente adiciona ruídos de diferentes naturezas e o segundo não.

Foram testados os dois métodos de identificação de falhas abordados nesse trabalho. A seguir os resultados e comentários são apresentados para cada um dos casos.

4.2.1 Identificação de falhas utilizando autocorrelação

A coleta dos dados de cada sistema controlado foi feita por um tempo suficientemente grande para conseguir coletar o nível do tanque de água após atingir o regime permanente, que é quando os dados realmente nos interessam. Os dados coletados são mostrados na Figura 8 com a linha em azul. Esses dados foram utilizados como parâmetros de entrada do Filtro de Kalman para ajustar os valores de Q e R a fim de encontrar resíduos que satisfaçam as condições dos testes de qualidade de ajuste que serão realizados em seguida. Uma vez definidos os parâmetros Q e R do FK foi possível avançar para a próxima etapa dos testes.

Utilizando os dados coletados, foram simulados ataques *replay* (ver subseção 3.4.2) ao sistema a fim de testar os métodos de detecção de falha apresentados na seção 3.7. Como foram realizados diferentes ensaios na planta, foram separados os ensaios que mostraram características particulares que poderiam resultar em testes convenientes para um estudo da eficiência dos algoritmos implementados. Por último, para reforçar a qualidade dos testes, foram feitos algoritmos para simular o ataque *replay* por meio da replicação de uma sequência de amostras de uma região pré-definida do conjunto original. Em seguida, o algoritmo deve realizar um estudo nas amostras extraídas para verificar a melhor forma de inseri-las aos dados que seriam enviados à IHM gerando a transição mais suave possível, esse estudo é feito calculando varição em relação ao valor médio dos conjuntos reais e que serão injetados e também cálculo da derivada da região de início da injeção para assegurar que de ambos os lados a tendência seja a mesma, ou crescente ou decrescente.

O teste realizado para detecção de ataques se firma na teoria apresentada na Seção 3.7.2. Algumas adaptações foram realizadas para mesclar a teoria com a prática por meio do algoritmo implementado. Esse algoritmo considera que o conjunto de dados tem acesso a uma dada quantidade de amostras passadas, a última amostra se refere ao instante atual. Considerando um sistema partindo do instante zero, é necessário existir um intervalo de exclusão, que se refere ao número de amostras em que haverá grande discrepância entre os valores observados e estimados pelo FK, esse intervalo é devido ao tempo gasto para o

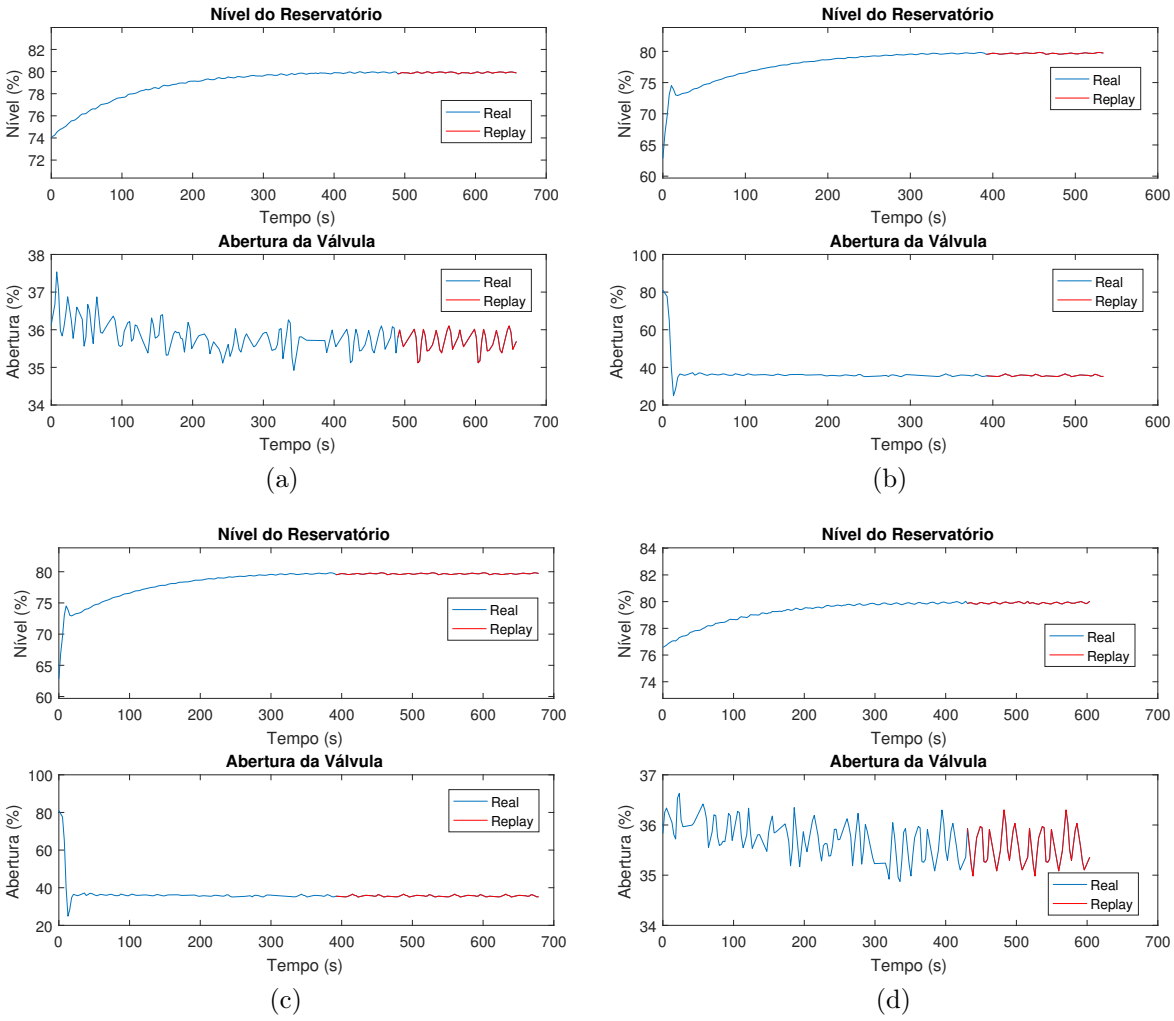


Figura 8 – Simulação do ataque *replay* nos dados coletados de diferentes controladores: (a) *replay* de dados entre os instantes 387 e 482,46 s; (b) *replay* de dados entre os instantes 312,18 e 392,16 s; (c) *replay* de dados prolongado entre os instantes 312,18 e 392,16 s; (d) *replay* de dados entre os instantes 345,72 e 433,44 s. Fonte: o autor.

estimador de estados convergir, todas essas amostras serão desconsideradas dos cálculos futuros. Em seguida, inicia-se a formação do banco de dados, que se refere a um conjunto de amostras boas que serão armazenadas para testar a autocorrelação da amostra atual. Enquanto um número mínimo de amostras pré-determinado não é alcançado, não são realizados testes, apenas quando o número de amostras supera o tamanho estipulado do banco de dados, iniciam-se os testes, sendo que eles consistem em testar a autocorrelação da amostra atual com todas as demais amostras do banco, uma vez terminados os cálculos verifica-se a semelhança entre as amostras. Para limitar o tamanho do banco de dados, a amostra mais antiga é excluída e é adicionada a amostra mais recente.

Considerando que as amostras supracitadas são os resíduos do Filtro de Kalman, um sistema em normal funcionamento apresentaria os resíduos independentes uns dos outros, ou seja, com pouca semelhança definida pela autocorrelação. Entretanto, uma vez contaminado seria alta o valor da autocorrelação, uma vez que seria examinada a

semelhança entre duas amostrais iguais distanciadas por um intervalo Δk.

O que determina se uma amostra está afetada ou não é a extrapolação de um limiar pré-estabelecido do valor da autocorrelação, amostras muito iguais apresentam alto valor de autocorrelação, o que torna fácil sua identificação. Na Figura 9, são mostrados os resultados do sistema de identificação para os conjuntos mostrados na Figura 8.

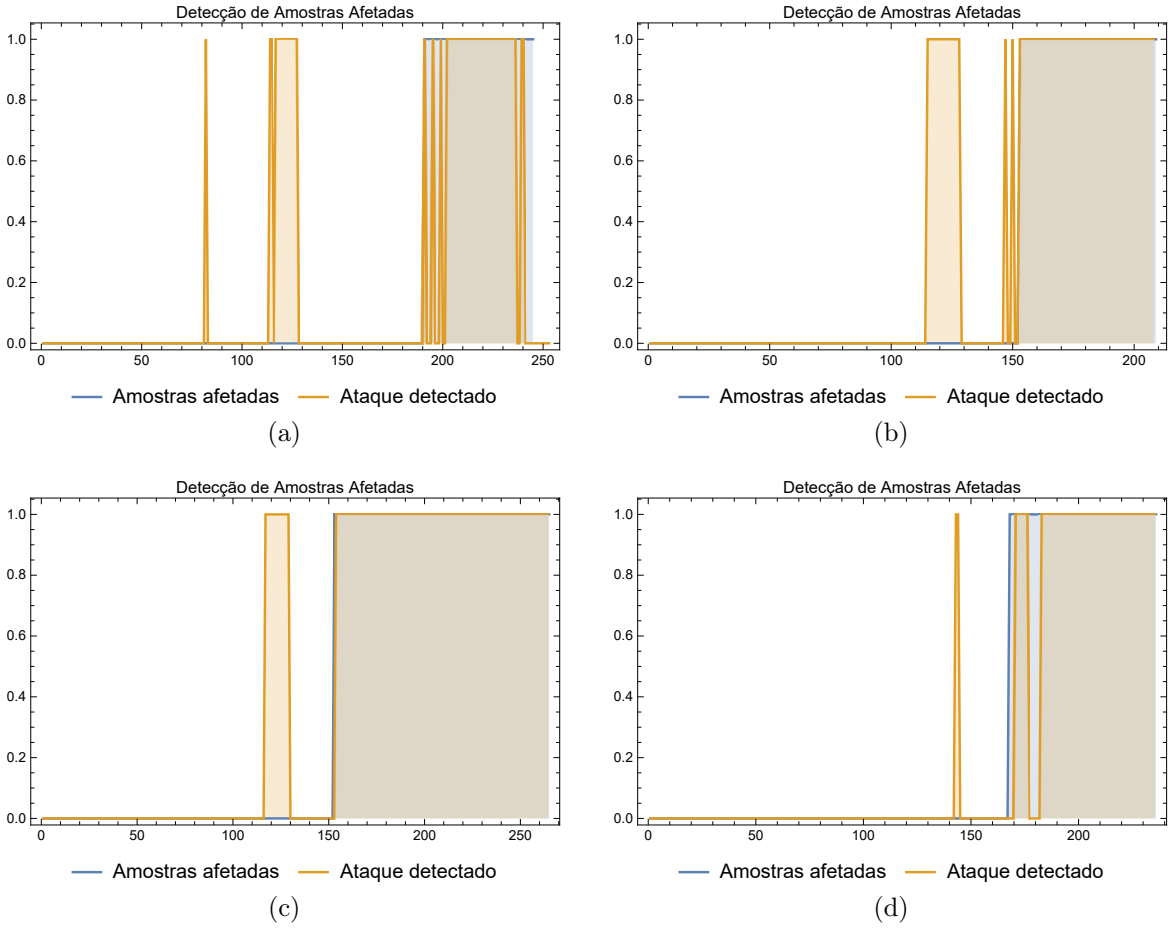


Figura 9 – Resultados dos testes do ataque *replay* nos dados coletados de diferentes controladores. 1 Significa existência ou detecção de ataque, 0 significa ausência de ataque: (a) Comprometimento das amostras 190 em diante; (b) Comprometimento das amostras 152 em diante; (c) Comprometimento das amostras 152 em diante; (d) Comprometimento das amostras 168 em diante. Fonte: o autor.

A partir da análise dos resultados obtidos foi possível tirar algumas conclusões acerca da eficiência do método de detecção de ataque por autocorrelação, a planta e o modelo estimado. Tomando primeiro a eficiência do método, pode-se dizer que é possível utilizá-lo para identificar ataques a um sistema desde que haja melhorias. Essa afirmação é feita baseando-se no fato que, apesar do sistema de detecção conseguir identificar repetição nas amostras, há grande presença de falso-positivos. Por outro lado, quando se compara a duração do nível alto de um falso-positivo com a de um ataque, verifica-se que falso-positivos duram em média tempo muito inferior, dependendo da planta o tempo do falso-positivo sequer resultaria em desastres, caso o tempo de reação fosse lento, como acontece

com o sistema de nível de tanques ou temperatura, por exemplo. Sendo assim, melhorias devem ser feitas a fim de reduzir a frequência dos resultados falso-positivo. Ainda no que se refere aos falso-positivo, acredita-se que o motivo que os tornam frequentes é uma natureza tendenciosa dos sensores. Isso não se mostrou a princípio, durante a calibração do FK, entretanto quando foi introduzida a janela móvel para o cálculo da autocorrelação, observou-se que em determinados instantes apresentavam-se tendências nas medições, o que fazia que a autocorrelação se tornasse alta na ausência de ataque. A explicação para esse fenômeno é a perda natural da calibragem dos sensores e alta sensibilidade à ruídos de processo e medição, que são inconvenientes de todo sistema real.

A última conclusão extraída com esse teste foi observada durante a configuração dos parâmetos Q e R do FK. Uma vez que para alcançar um resíduo do filtro de média zero e independentes no tempo foi necessário utilizar um baixo valor de R e um relativo valor alto de Q, o que configura maior confiança nos sensores do que no modelo. Caso qualquer outro critério fosse considerado para configuração o estimador resultaria em resíduos altamente correlatos e de média crescente. Portanto esses resultados foram uma consequência da baixa repetibilidade do modelo, uma característica frequentemente questionada acerca da planta utilizada.

4.2.2 Identificação de falhas por qui-quadrado

O teste de qualidade de ajuste por χ^2 tem uma particularidade que não foi apresentada na subseção 3.7.1. Diferente do teste de detecção de falhas utilizando a autocorrelação, que tem uma alta aceitação de ruídos de diferentes naturezas, o teste de χ^2 é bastante sensível a ruídos, levando a resultados completamente desprezíveis. A seguir será apresentado um teste de detecção de falhas utilizando o χ^2 e, por último, serão feitas as conclusões e notas necessárias sobre esse método.

Para realização desse teste, vamos considerar um ataque diferente do *replay*. Nessa etapa será considerado um ataque de integridade dos sensores. Portanto ao invés de replicar as entradas e saídas do sistema serão replicadas apenas as saídas e as entradas continuam com os valores corretos. Isso pode representar um ataque aos sensores de uma indústria 4.0 ou, como mostrado em Billings, Chen e Korenberg (1989), ocorrência de medições imprecisas, variação brusca na dinâmica do sistema o que torna o modelo estimado muito distante do real.

A necessidade de criar um conjunto de dados diferente do utilizado nos testes anteriores se dá nas diferenças entre os dois métodos. O método que utiliza a autocorrelação precisa identificar semelhanças nas amostras dos resíduos do Filtro de Kalman, paralelo a isso, a utilização do χ^2 requer variações bruscas do χ^2, o que acontece quando ocorre grande variação entre os parâmetros medidos e estimados da amostra ou variação brusca na variância do conjunto, o que indica que a nova amostra não pertence a ele. A condição para utilizar o teste do χ^2 é a modificação brusca da variância do conjunto com a inclusão de novas amostras, o que se alcança por meio de erros de estimação ou erros de medição.

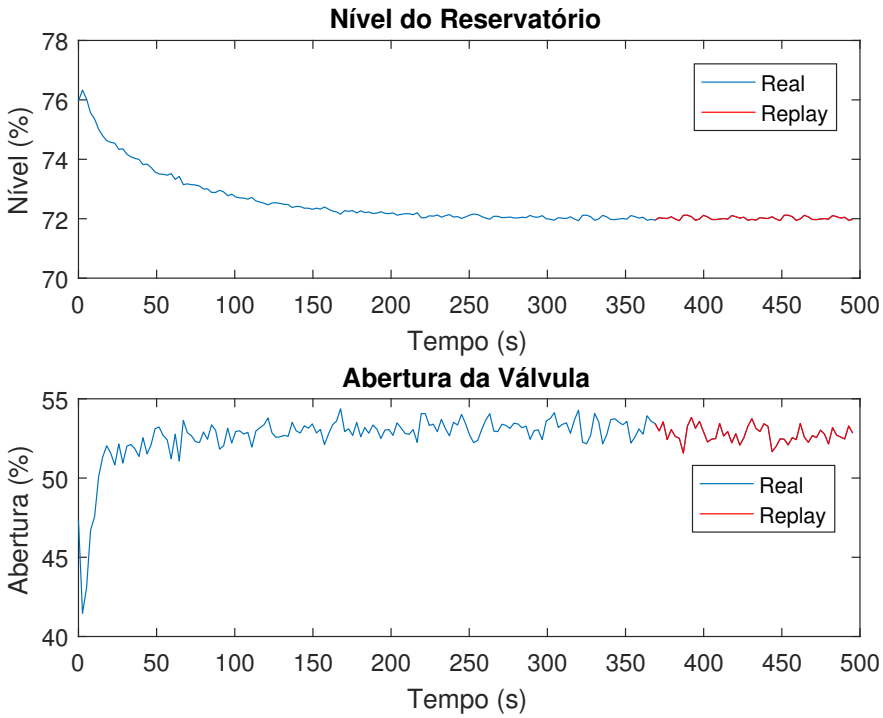


Figura 10 – Simulação de um ataque de integridade aos sensores de nível da planta após o instante t=387 s. Fonte: O autor

A Figura 11 mostra os resultados da simulação puramente computacional de um ataque de integridade aos sensores de nível da planta SMAR PD3-F. Os valores da abertura da válvula pneumática e nível do tanque são mostrados na Figura 10. O conjunto de dados utilizado nessa simulação foi gerado utilizando o modelo estimado da planta para simular a dinâmica do sistema com o *setpoint* de nível em 70% da capacidade do tanque. A partir disso foi simulada a dinâmica incluindo um controlador em cascata do tipo PI. Seus ganhos proporcional e integral foram escolhidos utilizando sintonia de Ziegler-Nichols para obter menor erro em regime permanente. Após a execução da simulação por tempo suficientemente grande, foi escolhida a porção final dos valores correspondentes às

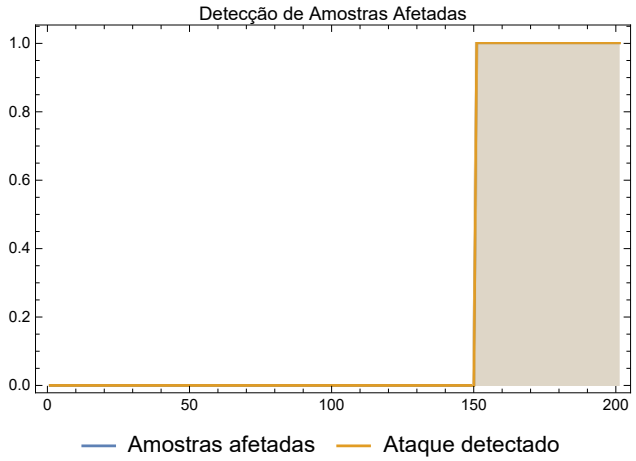


Figura 11 – Detecção de ataque de integridade aos sensores de nível da planta. Da amostra 150 em diante todas amostras estão comprometidas, a detecção é praticamente simultânea. Fonte: O autor

medições de nível para ser substituída por uma quantidade igual de medições passadas, semelhante ao que foi feito no ataque *replay*, com exceção que apenas as medições de nível são manipuladas. Uma alternativa ao que foi realizado é a substituição das medições por valores aleatórios. O que vai permitir que esse ataque seja detectado é variação brusca da dinâmica do sistema, que se reflete no aumento do resíduo do Filtro de Kalman.

Como resultado do teste de detecção de falhas utilizando o cálculo do χ^2, pode-se observar que a detecção é praticamente simultânea. As amostras comprometidas são as de número 150 em diante, a detecção foi feita à partir da amostra 152, o que é um ótimo tempo de resposta. Por outro lado, essa eficiência não pôde ser verificada com os testes utilizando dados obtidos na planta SMAR. A grande quantidade de ruídos somado à estimação pobre do modelo do sistema impossibilitaram a boa execução desse teste levando a resultados inconclusivos. Por esse motivo tais resultado foram omitidos deste trabalho.

5 Conclusões

Sistemas supervisórios desenvolvem função crucial no controle de sistemas industriais. Quando se trata de sistemas críticos, em que a segurança de um grande número de pessoas está envolvido, o cuidado deve ser ainda maior para evitar grandes tragédias.

Ao longo deste trabalho foram revisados alguns incidentes envolvendo sistemas críticos deixando claro porque essa é uma preocupação crescente no cenário industrial. Também foram estudadas algumas formas pertinentes de exploração de vulnerabilidades dos sistemas supervisórios que aparecem constantemente na literatura, como é o caso dos ataques de integridade. Por fim, foram apresentados alguns métodos de detecção e identificação de ataques a sistemas supervisórios.

Foi dada atenção especial para as técnicas de detecção de ataque utilizando autocorrelação móvel de medições e teste de χ^2. Ambos os métodos apresentaram resultados suficientemente bons para conclusão de existência ou não de dados errôneos. Ao longo do desenvolvimento deste trabalho foram feitas ligações entre os métodos e suas particularidades com diferentes técnicas de ataque. Por exemplo, testes de autocorrelação se tornam mais eficientes à medida que se aumenta o banco de dados, por outro lado essa afirmação nem sempre é verdadeira nos testes utilizando o χ^2. Finalmente, os dois testes que tiveram seus resultados discutidos na seção anterior têm validade em casos específicos. Como já foi explicado, o qui-quadrado necessita de variação brusca na dinâmica do sistema para detectar, portanto replay não são detectados, pois replay faz a replicação de dados verdadeiros, logo não há variação do desvio-padrão do conjunto, o que acarreta a não detecção do ataque, o mesmo ocorre quando se utiliza a autocorrelação para identificar outro ataque de natureza distinta do *replay*.

5.1 Propostas de melhoria

Após a discussão dos resultados apresentados no capítulo 4, concluiu-se que o modelo estimado não representava tão fielmente o sistema quanto necessário para ter resultados ótimos. Outro fator que contribuiu fortemente para a alta taxa de falso-positivos foi a sensibilidade a ruídos dos sensores e má calibragem. Portanto, como proposta para trabalhos futuros, recomenda-se recalibrar todos os sensores da planta e, em seguida, estimar um novo modelo do sistema. Por fim, é necessário verificar a repetibilidade do sistema para ter certeza que o modelo é válido durante os testes. Dessa forma, serão apreciados melhores resultados das técnicas implementadas nesse trabalho.

Referências

AGUIRRE, L. A. *Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais*. [S.l.]: Editora UFMG, 2004. 22, 23, 24, 29

BILLINGS, S.; CHEN, S.; KORENBERG, M. Identification of mimo non-linear systems using a forward-regression orthogonal estimator. *International journal of control*, Taylor & Francis, v. 49, n. 6, p. 2157–2189, 1989. 33

BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015. 26

BRANQUINHO, M. et al. *Segurança de Automação Industrial e SCADA*. [S.l.]: Elsevier Brasil, 2014. 2, 3

BYRES, E.; LOWE, J. The myths and facts behind cyber security risks for industrial control systems. In: *Proceedings of the VDE Kongress*. [S.l.: s.n.], 2004. v. 116, p. 213–218. 17

CAI, N.; WANG, J.; YU, X. Scada system security: Complexity, history and new developments. In: *2008 6th IEEE International Conference on Industrial Informatics*. [S.l.: s.n.], 2008. p. 569–574. ISSN 1935-4576. 5

CARDENAS, A. et al. Challenges for securing cyber physical systems. In: *Workshop on future directions in cyber-physical systems security*. [S.l.: s.n.], 2009. p. 5. 5, 6

CHRISTIANSSON, H.; LUIIJF, E. Creating a european scada security testbed. In: SPRINGER. *International Conference on Critical Infrastructure Protection*. [S.l.], 2007. p. 237–247. 7

DONDOSSOLA, G. et al. Effects of intentional threats to power substation control systems. *International Journal of Critical Infrastructures*, Inderscience Publishers, v. 4, n. 1-2, p. 129–143, 2008. 20

DUMONT, D. Cyber security concerns of supervisory control and data acquisition (scada) systems. In: *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*. [S.l.: s.n.], 2010. p. 473–475. 3

FALL, K. R.; STEVENS, W. R. *TCP/IP illustrated, volume 1: The protocols*. [S.l.]: addison-Wesley, 2011. 10, 11

FAWZI, H.; TABUADA, P.; DIGGAVI, S. Secure state-estimation for dynamical systems under active adversaries. In: IEEE. *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*. [S.l.], 2011. p. 337–344. 21

FELSER, M.; SAUTER, T. The fieldbus war: history or short break between battles? In: IEEE. *Factory Communication Systems, 2002. 4th IEEE International Workshop on*. [S.l.], 2002. p. 73–80. 10

FRANCHI, C. M.; CAMARGO, V. L. A. *Controladores lógicos programáveis: sistemas discretos*. [S.l.: s.n.], 2008. 14, 15

GIANI, A. et al. A testbed for secure and robust scada systems. *SIGBED Rev.*, ACM, New York, NY, USA, v. 5, n. 2, p. 4:1–4:4, jul. 2008. ISSN 1551-3688. Disponível em: <http://doi.acm.org/10.1145/1399583.1399587>. 20

GREENBERG, A. Hackers cut cities' power. *Forbes, Jaunuary*, 2008. 6

GROOVER, M. P. *Automação industrial e sistemas de manufatura*. [S.l.]: Pearson Education do Brasil, 2011. 14, 15

HARPER, A. et al. *Gray Hat Hacking The Ethical Hackers Handbook*. 3rd. ed. [S.l.]: McGraw-Hill Osborne Media, 2011. ISBN 0071742557, 9780071742559. 6

HERMANN, M.; PENTEK, T.; OTTO, B. Design principles for industrie 4.0 scenarios. In: IEEE. *System Sciences (HICSS), 2016 49th Hawaii International Conference on*. [S.l.], 2016. p. 3928–3937. 16

HESPANHA, J. P. *Linear systems theory*. [S.l.]: Princeton university press Princeton, 2009. v. 41. 19

IGURE, V. M.; LAUGHTER, S. A.; WILLIAMS, R. D. Security issues in scada networks. *Computers & Security*, Elsevier, v. 25, n. 7, p. 498–506, 2006. 9, 17, 18, 20

JUNG, S.; SONG, J.-g.; KIM, S. Design on scada test-bed and security device. *International Journal of Multimedia and Ubiquitous Engineering*, Citeseer, v. 3, n. 4, p. 75–86, 2008. 3, 20

KAGERMANN, H. et al. *Recommendations for Implementing the strategic initiative INDUSTRIE 4.0: securing the future of German manufacturing industry; final report of the Industrie 4.0 working group*. [S.l.]: Forschungsunion, 2013. 16, 17

MILLER, B.; ROWE, D. A survey scada of and critical infrastructure incidents. In: *Proceedings of the 1st Annual Conference on Research in Information Technology*. New York, NY, USA: ACM, 2012. (RIIT '12), p. 51–56. ISBN 978-1-4503-1643-9. Disponível em: <http://doi.acm.org/10.1145/2380790.2380805>. 3, 7, 9

MO, Y.; CHABUKSWAR, R.; SINOPOLI, B. Detecting integrity attacks on scada systems. *IEEE Transactions on Control Systems Technology*, IEEE, v. 22, n. 4, p. 1396–1407, 2014. 18, 20

NICHOLSON, A. et al. Scada security in the light of cyber-warfare. *Computers & Security*, Elsevier, v. 31, n. 4, p. 418–436, 2012. 5

NISHIYA, K.; HASEGAWA, J.; KOIKE, T. Dynamic state estimation including anomaly detection and identification for power systems. In: IET. *IEE Proceedings C (Generation, Transmission and Distribution)*. [S.l.], 1982. v. 129, n. 5, p. 192–198. 19, 24

PASQUALETTI, F.; DÖRFLER, F.; BULLO, F. Attack detection and identification in cyber-physical systems–part ii: Centralized and distributed monitor design. *arXiv preprint arXiv:1202.6049*, 2012. 21, 22

PASQUALETTI, F.; DÖRFLER, F.; BULLO, F. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, IEEE, v. 58, n. 11, p. 2715–2729, 2013. 21

SUN, H.; NGAN, W.; CHAO, H. J. Rateguard: A robust distributed denial of service (ddos) defense system. In: IEEE. *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. [S.l.], 2009. p. 1–8. 18

TANENBAUM, A. S.; WETHERALL, D. J. *Computer networks. 4th*. [S.l.: s.n.], 2002. v. 222. 748 p. 11

TATJEWSKI, P. *Advanced control of industrial processes: structures and algorithms*. [S.l.]: Springer Science & Business Media, 2007. 2

VERHAPPEN, A. P. I. *Foundation Fieldbus*. 4. ed. [S.l.]: ISA, 2012. ISBN 1937560201,9781937560201. 10, 11, 12, 13

WANG, C.; FANG, L.; DAI, Y. A simulation environment for scada security analysis and assessment. In: *2010 International Conference on Measuring Technology and Mechatronics Automation*. [S.l.: s.n.], 2010. v. 1, p. 342–347. ISSN 2157-1473. 3

WRIGHT, G. R.; STEVENS, W. R. *TCP/IP Illustrated*. [S.l.]: Addison-Wesley Professional, 1995. v. 2. 18

ZHU, B.; JOSEPH, A.; SASTRY, S. A taxonomy of cyber attacks on scada systems. In: *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*. [S.l.: s.n.], 2011. p. 380–388. 3, 7

ZHU, M.; MARTINEZ, S. Stackelberg-game analysis of correlated attacks in cyber-physical systems. In: IEEE. *American Control Conference (ACC), 2011*. [S.l.], 2011. p. 4063–4068. 21

Tabela de Distribuição do qui-quadrado

Figura 12 – Tabela de Distribuição de qui-quadrado.

	$P(\chi^2_n \leq x)$													
n	0,005	0,01	0,025	0,05	0,1	0,25	0,5	0,75	0,9	0,95	0,975	0,99	0,995	
1	3,93E-05	0,000157	0,000982	0,003932	0,016	0,102	0,455	1,323	2,706	3,841	5,024	6,635	7,879	1
2	0,010	0,020	0,051	0,103	0,211	0,575	1,386	2,773	4,605	5,991	7,378	9,210	10,597	2
3	0,072	0,115	0,216	0,352	0,584	1,213	2,366	4,108	6,251	7,815	9,348	11,345	12,838	3
4	0,207	0,297	0,484	0,711	1,064	1,923	3,357	5,385	7,779	9,488	11,143	13,277	14,860	4
5	0,412	0,554	0,831	1,145	1,610	2,675	4,351	6,626	9,236	11,070	12,832	15,086	16,750	5
6	0,676	0,872	1,237	1,635	2,204	3,455	5,348	7,841	10,645	12,592	14,449	16,812	18,548	6
7	0,989	1,239	1,690	2,167	2,833	4,255	6,346	9,037	12,017	14,067	16,013	18,475	20,278	7
8	1,344	1,647	2,180	2,733	3,490	5,071	7,344	10,219	13,362	15,507	17,535	20,090	21,955	8
9	1,735	2,088	2,700	3,325	4,168	5,899	8,343	11,389	14,684	16,919	19,023	21,666	23,589	9
10	2,156	2,558	3,247	3,940	4,865	6,737	9,342	12,549	15,987	18,307	20,483	23,209	25,188	10
11	2,603	3,053	3,816	4,575	5,578	7,584	10,341	13,701	17,275	19,675	21,920	24,725	26,757	11
12	3,074	3,571	4,404	5,226	6,304	8,438	11,340	14,845	18,549	21,026	23,337	26,217	28,300	12
13	3,565	4,107	5,009	5,892	7,041	9,299	12,340	15,984	19,812	22,362	24,736	27,688	29,819	13
14	4,075	4,660	5,629	6,571	7,790	10,165	13,339	17,117	21,064	23,685	26,119	29,141	31,319	14
15	4,601	5,229	6,262	7,261	8,547	11,037	14,339	18,245	22,307	24,996	27,488	30,578	32,801	15
16	5,142	5,812	6,908	7,962	9,312	11,912	15,338	19,369	23,542	26,296	28,845	32,000	34,267	16
17	5,697	6,408	7,564	8,672	10,085	12,792	16,338	20,489	24,769	27,587	30,191	33,409	35,718	17
18	6,265	7,015	8,231	9,390	10,865	13,675	17,338	21,605	25,989	28,869	31,526	34,805	37,156	18
19	6,844	7,633	8,907	10,117	11,651	14,562	18,338	22,718	27,204	30,144	32,852	36,191	38,582	19
20	7,434	8,260	9,591	10,851	12,443	15,452	19,337	23,828	28,412	31,410	34,170	37,566	39,997	20
21	8,034	8,897	10,283	11,591	13,240	16,344	20,337	24,935	29,615	32,671	35,479	38,932	41,401	21
22	8,643	9,542	10,982	12,338	14,041	17,240	21,337	26,039	30,813	33,924	36,781	40,289	42,796	22
23	9,260	10,196	11,689	13,091	14,848	18,137	22,337	27,141	32,007	35,172	38,076	41,638	44,181	23
24	9,886	10,856	12,401	13,848	15,659	19,037	23,337	28,241	33,196	36,415	39,364	42,980	45,558	24
25	10,520	11,524	13,120	14,611	16,473	19,939	24,337	29,339	34,382	37,652	40,646	44,314	46,928	25
26	11,160	12,198	13,844	15,379	17,292	20,843	25,336	30,435	35,563	38,885	41,923	45,642	48,290	26
27	11,808	12,878	14,573	16,151	18,114	21,749	26,336	31,528	36,741	40,113	43,195	46,963	49,645	27
28	12,461	13,565	15,308	16,928	18,939	22,657	27,336	32,620	37,916	41,337	44,461	48,278	50,994	28
29	13,121	14,256	16,047	17,708	19,768	23,567	28,336	33,711	39,087	42,557	45,722	49,588	52,335	29
30	13,787	14,953	16,791	18,493	20,599	24,478	29,336	34,800	40,256	43,773	46,979	50,892	53,672	30
40	20,707	22,164	24,433	26,509	29,051	33,660	39,335	45,616	51,805	55,758	59,342	63,691	66,766	40
50	27,991	29,707	32,357	34,764	37,689	42,942	49,335	56,334	63,167	67,505	71,420	76,154	79,490	50
60	35,534	37,485	40,482	43,188	46,459	52,294	59,335	66,981	74,397	79,082	83,298	88,379	91,952	60
70	43,275	45,442	48,758	51,739	55,329	61,698	69,334	77,577	85,527	90,531	95,023	100,425	104,215	70
80	51,172	53,540	57,153	60,391	64,278	71,145	79,334	88,130	96,578	101,879	106,629	112,329	116,321	80
90	59,196	61,754	65,647	69,126	73,291	80,625	89,334	98,650	107,565	113,145	118,136	124,116	128,299	90
100	67,328	70,065	74,222	77,929	82,358	90,133	99,334	109,141	118,498	124,342	129,561	135,807	140,170	100