Learning-based stealthy attacks and adaptive defenses in cyber-physical systems

by

Bipin Paudel

B.S., Tribhuvan University, 2017

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Abstract

Cyber-physical systems increasingly rely on networked sensing, computation, and supervisory control, creating attack surfaces through which adversaries can manipulate physical processes while remaining undetected. Stealthy attacks are particularly challenging because they can exploit partial observability and system dynamics to alter internal states without immediately producing anomalous behavior in observed measurements. Classical zero-dynamics attacks generally assume substantial knowledge of the underlying system model. This dissertation investigates how learning-based adversaries can relax this assumption, how detection mechanisms can adapt to increasingly capable attackers, and how transient operating conditions and privacy-preserving mechanisms influence this interaction.

The dissertation first studies model-free generation of zero-dynamics-like attacks using deep reinforcement learning. Using the quadruple-tank process, a Soft Actor-Critic-based adversary is trained to manipulate control inputs and drive unobserved states away from desired operating conditions without access to the plant's state-space matrices. An iterative attacker-detector framework is then developed in which progressively stronger attackers are trained against existing detectors, while new learning-based detectors are trained using trajectories generated by the latest attacker. The results show that reinforcement learning can reproduce important characteristics of theoretically constructed zero-dynamics attacks and can provide challenging attack data for detector development.

The framework is next extended to inverter-based power systems. Deep reinforcement learning attackers manipulate control-layer signals of a grid-forming microgrid while supervisory detection relies primarily on point-of-common-coupling measurements. Iterative adversarial training is used to construct increasingly robust detectors and to examine attack impact and detection timing.

The dissertation then considers attacks during transient rather than steady-state operation. Load-induced transients naturally produce deviations in voltage and frequency that can make malicious behavior more difficult to distinguish from legitimate dynamics. An adaptive adversary is trained to induce mislocalization, in which a disturbance originating from one compromised inverter is attributed to another. The study also compares detector aggregation strategies, showing that strict aggregation can progressively suppress successful attacks, whereas consensus-based aggregation can preserve exploitable ambiguity and allow persistent mislocalization under some conditions.

Finally, the dissertation examines a complementary defense that changes the measurements available for system identification. Real and surrogate measurements are combined for feedback control and attacker identification, while a separate trusted detector retains real measurements. The mixture weight is selected from disturbance recovery, and the surrogate parameters are optimized to reduce damage before alarm, defined as a physical-limit violation before detection. The optimized design reduces the observed successful attack rate relative to real-only feedback and an unperturbed surrogate, while meeting the aggregate tracking limit and tested recovery requirements. This benefit is obtained at the cost of increased normal-operation tracking error.

Together, these studies present an adaptive view of cyber-physical security in which attackers and defenders co-evolve, and in which resilience must be evaluated across model uncertainty, transient operation, detector design, and control requirements.

Learning-based stealthy attacks and adaptive defenses in cyber-physical systems

by

Bipin Paudel

B.S., Tribhuvan University, 2017

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Approved by:

Major Professor
George Amariucai, PhD

Copyright

© Bipin Paudel 2026.

Abstract

Cyber-physical systems increasingly rely on networked sensing, computation, and supervisory control, creating attack surfaces through which adversaries can manipulate physical processes while remaining undetected. Stealthy attacks are particularly challenging because they can exploit partial observability and system dynamics to alter internal states without immediately producing anomalous behavior in observed measurements. Classical zero-dynamics attacks generally assume substantial knowledge of the underlying system model. This dissertation investigates how learning-based adversaries can relax this assumption, how detection mechanisms can adapt to increasingly capable attackers, and how transient operating conditions and privacy-preserving mechanisms influence this interaction.

The dissertation first studies model-free generation of zero-dynamics-like attacks using deep reinforcement learning. Using the quadruple-tank process, a Soft Actor-Critic-based adversary is trained to manipulate control inputs and drive unobserved states away from desired operating conditions without access to the plant's state-space matrices. An iterative attacker-detector framework is then developed in which progressively stronger attackers are trained against existing detectors, while new learning-based detectors are trained using trajectories generated by the latest attacker. The results show that reinforcement learning can reproduce important characteristics of theoretically constructed zero-dynamics attacks and can provide challenging attack data for detector development.

The framework is next extended to inverter-based power systems. Deep reinforcement learning attackers manipulate control-layer signals of a grid-forming microgrid while supervisory detection relies primarily on point-of-common-coupling measurements. Iterative adversarial training is used to construct increasingly robust detectors and to examine attack impact and detection timing.

The dissertation then considers attacks during transient rather than steady-state operation. Load-induced transients naturally produce deviations in voltage and frequency that can make malicious behavior more difficult to distinguish from legitimate dynamics. An adaptive adversary is trained to induce mislocalization, in which a disturbance originating from one compromised inverter is attributed to another. The study also compares detector aggregation strategies, showing that strict aggregation can progressively suppress successful attacks, whereas consensus-based aggregation can preserve exploitable ambiguity and allow persistent mislocalization under some conditions.

Finally, the dissertation examines a complementary defense that changes the measurements available for system identification. Real and surrogate measurements are combined for feedback control and attacker identification, while a separate trusted detector retains real measurements. The mixture weight is selected from disturbance recovery, and the surrogate parameters are optimized to reduce damage before alarm, defined as a physical-limit violation before detection. The optimized design reduces the observed successful attack rate relative to real-only feedback and an unperturbed surrogate, while meeting the aggregate tracking limit and tested recovery requirements. This benefit is obtained at the cost of increased normal-operation tracking error.

Together, these studies present an adaptive view of cyber-physical security in which attackers and defenders co-evolve, and in which resilience must be evaluated across model uncertainty, transient operation, detector design, and control requirements.

Table of Contents

List of Figures xv

List of Tables xix

Acknowledgements xxii

Dedication xxiv

1 Introduction 1

1.1 Background and Motivation 1

1.2 Research Problem and Challenges 3

1.2.1 Limited Measurements and Hidden Physical Changes 3

1.2.2 Attack Generation without Explicit State-Space Models 4

1.2.3 Defenders Evaluated against Fixed Attack Patterns 5

1.2.4 Detection Timing and Correct Localization 5

1.3 Research Aim and Objectives 6

1.4 Research Approach and Evaluation 7

1.4.1 Learning-Based Attack Generation 7

1.4.2 Iterative Detector Development 8

1.4.3 Measurement Design under Control Constraints 8

1.4.4 Physical Outcomes and Attack Success 9

1.5 Overview of Subsequent Chapters 9

1.5.1 Conclusion and Future Work 10

2 Reinforcement Learning Approach to Generate Zero-Dynamics Attacks on Control Systems Without State Space Models . . . 11
2.1 Introduction . . . 11
2.2 Related Works . . . 14
2.3 Preliminaries . . . 16
2.3.1 Quadruple Tank Process . . . 16
2.3.2 State space-based Theoretical Attack . . . 17
2.3.3 Reinforcement learning (RL) . . . 18
2.3.4 Deep reinforcement learning . . . 19
2.4 Attacking the quadruple tank process (QTP) . . . 21
2.4.1 Description of the QTP model . . . 22
2.4.2 Deep Reinforcement Learning based attacker . . . 23
2.4.3 Chi-square based anomaly detector (χ^2) . . . 25
2.4.4 LSTM based Detector . . . 25
2.4.5 Two proportion Z-test . . . 27
2.5 Results . . . 27
2.5.1 RL based attackers . . . 27
2.5.2 LSTM based detectors . . . 28
2.5.3 Test of Convergence . . . 29
2.5.4 RL vs. Theoretical attack . . . 31
2.6 Conclusion . . . 31
2.7 Discussion and Implications . . . 32
2.7.1 What the Model-Free Result Establishes . . . 32
2.7.2 Interpreting the Connection to Zero Dynamics . . . 33
2.7.3 Reading the Iterative Results . . . 33
2.7.4 Connection to the Microgrid Studies . . . 34

3 Robust Detection in Power Systems: Iterative Reinforcement Learning Based Adversarial Training . . . 36
3.1 Introduction . . . 36
3.2 Literature Review . . . 41
3.3 Compromising the microgrid . . . 43
3.3.1 Impact of FDIA on microgrids . . . 44
3.3.2 Attacker's goal . . . 44
3.3.3 Detectors . . . 44
3.3.4 Deep Reinforcement Learning (DRL) based attacker . . . 45
3.4 Results . . . 47
3.4.1 DRL based attackers . . . 48
3.4.2 LSTM based detectors . . . 49
3.4.3 LSTM based detectors . . . 49
3.4.4 Impact of attackers and detectors in the system . . . 50
3.5 Conclusion . . . 52
3.6 Discussion and Practical Implications . . . 53
3.6.1 The Gap between Local Behavior and Supervisory Evidence . . . 53
3.6.2 Why an Eventual Alarm Is Not Sufficient . . . 53
3.6.3 What Changes across Adversarial-Training Rounds . . . 54
3.6.4 Detection, Response, and the Next Research Question . . . 55

4 Mislocalization Attacks on Grid-Forming Microgrids Under Transient Conditions: An Iterative Reinforcement Learning Framework . . . 56
4.1 Introduction . . . 56
4.2 Related Work and Positioning . . . 60
4.2.1 Learning-Based Attacks and Defenses in Power Systems . . . 60
4.2.2 Iterative Attacker–Detector Learning . . . 62
4.2.3 Detector Paradigms: Supervised, Unsupervised, and RL . . . 62

4.2.4 Why Transient Operation Matters . . . 63
4.2.5 Localization as an Adversarial Objective . . . 64
4.2.6 Partial Observability and Stealth . . . 64
4.3 Experiment Setup . . . 65
4.3.1 Microgrid Model . . . 65
4.3.2 Supervisory Layer and Partial Observability . . . 67
4.3.3 Detector Design . . . 67
4.3.4 Multi-Detector Output Aggregation . . . 69
4.3.5 Threat Model and Offline-Training/Online-Inference Separation . . . 70
4.3.6 RL-Based Attacker Using Soft Actor Critic . . . 72
4.3.7 Iterative Adversarial Training Procedure . . . 77
4.3.8 Implementation Details . . . 78
4.4 Results . . . 80
4.4.1 Iterative Attacker–Detector Training Dynamics . . . 80
4.4.2 Progressive Suppression Under Strict Aggregation . . . 82
4.4.3 Persistent Mislocalization Under Consensus Aggregation . . . 82
4.4.4 Representative Attack Trajectories . . . 83
4.4.5 Implications for Aggregation Design . . . 85
4.5 Conclusion . . . 86
4.6 Discussion . . . 87
4.6.1 Detection, Localization, and Mitigation Are Different Tasks . . . 87
4.6.2 The Meaning of Strict Aggregation . . . 88
4.6.3 How Averaging Can Suppress Correct Evidence . . . 88
4.6.4 Physical Responses Do Not Uniquely Identify Their Source . . . 89
4.6.5 What the Aggregation Comparison Establishes . . . 90
4.6.6 Implications across the Dissertation . . . 90

5 Surrogate Dynamics and Damage Before Alarm . . . 92

5.1 Introduction 92
5.2 Literature Review 95
5.2.1 Zero-Dynamics Attacks and Attacker Knowledge 95
5.2.2 Protecting System Information 96
5.2.3 Detection, Measurement Access, and Physical Impact 96
5.3 Preliminaries 97
5.3.1 Quadruple-tank process 97
5.3.2 Surrogate model and measurement routing 99
5.3.3 Controller and trusted detector 100
5.3.4 Learned zero-dynamics attack 102
5.3.5 Damage before alarm 103
5.4 Experimental Setup 104
5.4.1 Plant, controller and detector settings 104
5.4.2 Disturbance recovery and measurement-weight selection 105
5.4.3 Supporting disturbance and perturbation studies 106
5.4.4 Surrogate optimization and final configurations 106
5.4.5 Final identification and attack evaluation 107
5.4.6 Normal-operation utility 108
5.4.7 Statistical analysis 108
5.5 Results 110
5.5.1 Disturbance recovery and the selected measurement weight 110
5.5.2 Physical constraints and arbitrary perturbations 111
5.5.3 Optimized surrogate parameters 113
5.5.4 Pooled damage before alarm 114
5.5.5 SI-level vulnerability 116
5.5.6 Normal-operation tracking and recovery 117
5.5.7 Variation across validation banks 119

5.6 Discussion . . . 119
5.6.1 Effect of optimizing the surrogate . . . 119
5.6.2 Identification accuracy with an unperturbed surrogate . . . 120
5.6.3 Recovery-based selection and tracking cost . . . 122
5.6.4 Statistical and experimental scope . . . 123
5.7 Conclusion . . . 123

6 Conclusion . . . 125
6.1 Overall Findings . . . 125
6.2 Contributions . . . 126
6.2.1 Learning-Based Zero-Dynamics-Like Attacks . . . 126
6.2.2 Iterative Adversarial Training for Microgrid Detection . . . 127
6.2.3 Transient Mislocalization and Detector Aggregation . . . 127
6.2.4 Utility-Constrained Mixed-Measurement Defense . . . 128
6.2.5 Scope of the Contributions . . . 128
6.3 Future Work . . . 129
6.3.1 Imperfect System Knowledge and Model Mismatch . . . 129
6.3.2 Measurement Selection for Detection and Localization . . . 130
6.3.3 Detector Diversity and Robust Aggregation . . . 131

Bibliography . . . 132

A Chapter 1 . . . 142
A.1 Proof for Attack Generation . . . 142
A.2 Architectures and Hyperparameters . . . 143
A.2.1 Design Rationale and Limitations . . . 145
A.3 Attacker Rewards and Performance . . . 148
A.4 Individual-Detector Classification Diagnostics . . . 148

B Chapter 4 150
B.1 Supporting Disturbance Studies 150
B.2 Measured and unmeasured disturbances 150
B.3 Fixed-direction perturbation screen 150
B.4 Smaller-outlet sensitivity 155

List of Figures

2.1 QTP consisting of four connected tanks [1] 14

2.2 Architecture of the process during an attack 24

2.3 Iterative attacker–detector evaluation. The label $DET_{j:0}$ denotes the accumulated set $\{DET_j, \ldots, DET_0\}$. Diagonal arrows report attack success probabilities before adding the next detector, and horizontal arrows report them after its addition. Both evaluations use the entire accumulated detector set shown. 30

3.1 Power system 40

3.2 Iterative attacker–detector evaluation against accumulated detector sets, where $DET_{j:0} = \{DET_j, \ldots, DET_0\}$. Diagonal arrows report attack success probabilities before adding the next detector, and horizontal arrows report them after its addition. 48

3.3 Normal operation of microgrid with load change at time $t = 4s$ 50

3.4 Successful attack by $Attacker_3$; inverter 1 is still connected to the system. Attack at: $4.05s$, Success at: $4.117s$ 50

3.5 Successful attack by $Attacker_1$; inverter 1 is disconnected post detection. Attack at: $4.0759s$, Success at: $4.278s$, Detected at: $4.340s$ 50

3.6 Successful attack by $Attacker_3$; inverter 1 is disconnected post detection. Attack at: $4.05s$, Success at: $4.117s$, Detected at: $4.138s$ 51

4.1 Single-line diagram of the simulated grid-forming microgrid operating under droop control. 66

4.2 Separation between offline adversarial training/security stress testing and online frozen-policy attack execution/evaluation. Iterative attacker/detector training occurs only offline; the physical transient is evaluated using frozen models. 73

4.3 Recorded signals from three simulations of attacks targeting inverter 1. Columns identify the measured inverter, while line colors and styles identify the attack on a particular setting. Each setting uses the same simulation across all columns. The vertical dashed line marks attack onset, shading indicates the pre-attack interval, and dots mark recording endpoints. Each signal row uses the same vertical scale across inverters. 85

5.1 Mixed-measurement architecture. The operating Kalman filter (KF) and linear quadratic regulator (LQR) receive $y^c = 0.75y^r + 0.25y^f$, while the trusted KF and cumulative sum (CUSUM) detector receive real measurements and the known command. Both plants receive u, but the attack a reaches only the real plant. The attacker learns from exposed input–output records without access to the trusted channel. 101

5.2 Worst p95 recovery times across separate positive and negative disturbances in each measured tank. Dashed lines are 25% above the matched real-only values, while stars identify the selected weight, $\alpha = 0.75$. 111

5.3 Supporting fixed-direction screen with fake-only feedback and $\rho = 0.20$. Real Tank 3 crosses below 0 cm at 7 s, before the Tank-1 disturbance. The screen uses a different feedback arrangement from the final mixed configuration. . . 112

5.4 Signed relative changes in the optimized surrogate parameters. The components vary within the common $\pm 50\%$ envelope, whose limits are shown by dashed lines. 113

5.5 Pooled DBA rates with 95% Wald intervals. Labels show DBA events divided by eligible attacks, and both mixed configurations use $\alpha = 0.75$. 115

5.6 Overall normal-operation RMSE with 95% Student-t confidence intervals for the mean, calculated from 250 no-attack episodes per configuration. Open circles show the five bank means, each based on 50 episodes, and the dashed line marks $J_{\max}$. The optimized mean is below the limit, but its upper confidence bound and the Bank-0 mean are slightly above it. 118

A.1 Reward plots of five consecutive attackers 147

A.2 Cosine similarity as described in 2.5.4. 147

A.3 Individual-detector confusion matrices for D2 (initial attack-trained checkpoint), D4 (later strict checkpoint), and D6 (later consensus checkpoint), evaluated on their respective saved datasets. Rows denote true classes and columns denote predictions. Values are percentages within each true-class row. Outlined cells indicate inverter-labelled windows classified as Normal. These matrices describe individual classifiers, not the aggregated ensemble decisions. 149

B.1 Illustrative Tank-1 disturbance trajectories across real-only, fake-only and mixed feedback. The weight grid and two-jump schedule differ from the formal recovery-selection experiment. 151

B.2 Illustrative trajectories with disturbances in unmeasured Tank 3. Changes in the lower tanks provide an indirect, real-time feedback path. 152

B.3 Fixed-direction screen at $\rho = 0$, with identical real and surrogate parameters and fake-only feedback. 152

B.4 Fixed-direction screen at $\rho = 0.10$. The real trajectory remains within the physical range for this run. 153

B.5 Fixed-direction screen at $\rho = 0.30$. The real trajectory crosses the physical boundary during startup. Values outside the operating range indicate failure of the linear simulation's physical validity check. 153

B.6 Additional Tank-3 comparison across perturbation bounds. These illustrative trajectories are not additional samples in the primary DBA analysis. 154
B.7 Supporting comparison of original and halved outlet areas. Subtracting the matched no-jump trajectory isolates the height changes caused by the disturbances. 155

List of Tables

2.1 Performance of different attackers. Each attacker's (ATT_i) attack rate indicates their capability to beat collective previous detectors d_{i-1} to d_0. 28

2.2 Reported test-set performance of the individual LSTM detectors. FPR and TPR are proportions. TP, FP, FN, and TN denote true positives, false positives, false negatives, and true negatives, respectively. 29

2.3 Reported two-proportion tests for successive attackers and detectors. Each evaluation identifies the attacker and the accumulated detector set it faces; p_1 and p_2 are successful attack probabilities. 31

3.1 Signal description and their mnemonics 40

3.2 Decision thresholds, false positive rates (FPR), and true positive rates (TPR) of the individual LSTM detectors. 49

4.1 Comparison with representative learning-based cyberattack and defense frameworks. "Transient-targeted" means that the attacker is explicitly designed to exploit the transient interval, not merely that load changes are included in validation. 61

4.2 Microgrid parameters 66

4.3 SAC hyperparameters. 79

4.4 Per-inverter attack success rate (ASR, %) across iterative detector expansion, under both aggregation settings. For each step, we report the ASR against the current detector ensemble ("Before") and after adding the new detector ("After"). 80

4.5 Two-proportion z-tests of ASR changes across iterative steps under strict and consensus aggregation. Attacker-side tests hold the attacker fixed and compare ASR before and after detector addition. Detector-side tests hold the detector ensemble fixed and compare successive attackers. "Sig." denotes $|z| > 1.96$; dashes indicate unavailable next-iteration comparisons. 81

5.1 Settings for measurement-weight selection and final validation. The surrogate-design bound and attacker prior serve different purposes. 104

5.2 Configurations compared in final validation. The trusted detector receives real measurements in every configuration. 107

5.3 Measurement-weight selection with the unperturbed surrogate. Times are the worse p95 values across positive and negative disturbances. The limits are 193.125 s for Tank 1 and 212.5 s for Tank 2. 110

5.4 Supporting fake-only configuration for one fixed perturbation direction. Crossing times and minima concern the real plant; negative heights denote physical-limit violations in the linear simulation. 112

5.5 Nominal and optimized surrogate parameters. Values are rounded for display; the full-precision vector is retained in the accompanying data. Gravity remains fixed at 981 cm/s^2. 114

5.6 Final pooled DBA counts, rates and 95% Wald intervals. 115

5.7 Pooled two-proportion comparisons. A difference is the first named configuration minus the second, in percentage points (pp). All three cross the unadjusted $|Z| > 1.96$ threshold. The first two comparisons are primary. . . 116

5.8 SI-level vulnerability in final validation. A model is counted as vulnerable when at least one of its five attack repetitions produces an eligible DBA event. Brackets give 95% Wilson confidence intervals for the vulnerable-model rate, based on 500 SI models per configuration. 116

5.9 SI-level two-proportion comparisons. Differences are the first configuration minus the second; pp denotes percentage points. All three cross the unadjusted two-sided threshold $|Z| > 1.96$. 117

5.10 Overall normal-operation tracking and worst p95 recovery times from the dedicated disturbance experiments. Brackets below each mean RMSE give its 95% Student-t confidence interval, based on 250 no-attack episodes per configuration. The RMSE limit is 0.177009 cm; Tank-1 and Tank-2 recovery limits are 193.125 s and 212.5 s. 118

5.11 Final bank-level DBA and normal-operation tracking results. Banks are numbered 0–4. Optimized Bank 0 is slightly above the RMSE limit; the optimized aggregate is below it. 120

5.12 Mean held-out prediction accuracy against the real plant, averaged over 500 identified models per configuration. Brackets below each mean give its 95% Student-t confidence interval across models. Prediction RMSE is reported in sensor-output units. 121

A.1 Architecture of the neural network models associated with the attackers. Attackers (ATT_1, ATT_2) and (ATT_3, ATT_4 and ATT_5) share the same architecture 144

A.2 Hyper Parameter Setting for the attackers based on SAC algorithm 144

A.3 Architecture of the LSTM-based detectors 145

Acknowledgments

I am sincerely grateful to my advisor, George Amariucai, for his guidance, encouragement, and support throughout my Ph.D. journey. When I began my doctoral studies, I had very little research experience, and his mentorship played an important role in helping me grow into an independent researcher. I feel truly fortunate to have been his student and to have learned from him throughout this journey.

I would also like to thank my Ph.D. committee members, Eugene Vasserman, Doina Caragea, and Behrooz Mirafzal, for their thoughtful feedback, valuable suggestions, and continued support, all of which helped shape and strengthen my research. I am grateful to have had them as members of my committee. Additionally, I would like to thank Shuangqing Wei from Louisiana State University for his guidance, insightful discussions, and valuable contributions to my research, particularly our work on privacy in large language models.

I extend my sincere gratitude to the Department of Computer Science at Kansas State University for supporting me throughout my Ph.D. journey. Their support went beyond providing a stipend and included funding for research tools, conference participation, and other resources that contributed greatly to my academic and professional development.

I owe a deep debt of gratitude to my family. I am especially thankful to my father and mother, whose love, sacrifices, and support since my childhood gave me the foundation and opportunities that eventually made pursuing a Ph.D. possible. I am also grateful to my brother for his encouragement and support throughout this journey.

I would also like to thank Bishwash Mandal for the many helpful discussions on privacy and machine learning, and for helping me identify an important direction in one of my research projects.

Finally, I want to express my deepest gratitude to my wife, Nikita Gautam, for her constant love, patience, encouragement, and support in every aspect of my life. This journey would not have been possible without you. Thank you for always believing in me, for standing

beside me through the difficult moments, and for being a wonderful wife and an amazing mother to our sweet little daughter.

Dedication

To my wife, for always supporting me, believing in me, and standing by my side.

To my daughter, for filling my life with happiness and giving it a deeper sense of purpose.

Chapter 1

Introduction

1.1 Background and Motivation

Cyber-physical systems combine physical processes with digital sensing, communication, and control. Sensors provide measurements of the process, controllers use these measurements to determine suitable actions, and actuators apply those actions to the physical system. This interaction allows a system to respond to disturbances and maintain its desired operation. Water-level regulation and the coordination of micro-grid based power systems are two examples considered in this dissertation. In both cases, the quality of a digital decision can directly affect the behavior of physical equipment [2].

The same interaction creates security concerns. An attacker who can modify a control command may change the physical process even without changing the measurements delivered to the supervisor. The measurements can be genuine, but they may not reveal every important internal change. Protecting communication and preventing unauthorized access remain important. However, it is also necessary to understand what happens after a component or control channel has been compromised. This dissertation studies that stage of the problem: how malicious control actions affect a system, what the defender can observe, and whether the defender can respond appropriately.

The focus is on attacks that exploit a difference between physical behavior and super-

visory awareness. Some attacks produce large and immediate changes that are easy to recognize. Others are designed to keep the monitored behavior close to what a detector considers acceptable while changing less visible parts of the system. These attacks are referred to as stealthy attacks. Their importance comes not only from their ability to avoid an alarm, but also from the time they provide an attacker to influence the physical process [3].

Zero-dynamics attacks provide a mathematical example of this problem. Under suitable system conditions, an attack can excite internal dynamics while producing no change, or only a small change under practical initialization and modeling conditions, in the monitored outputs. Classical constructions use a mathematical description of the system to identify these attack directions [3]. This raises a question about the information an attacker actually needs. The absence of an explicit state-space model may prevent one method of constructing an attack, but it does not necessarily prevent an attacker from learning a harmful sequence of actions through interaction.

Deep reinforcement learning offers a way to examine this question. Rather than receiving a complete attack sequence in advance, a learning agent tries actions, observes their effects, and improves a policy according to a reward. For the attacks studied here, the reward encourages a harmful physical outcome or an incorrect supervisory decision while discouraging behavior that would defeat the attack. The resulting policy can be used to evaluate a defender under an adversary that has learned specifically against it.

Learning also provides a way to improve the defender. Measurements generated by a successful attacker can reveal weaknesses in the current detector and become training data for a new detector. A new attacker can then be trained against the expanded defense. The dissertation uses this repeated interaction to move beyond evaluating a detector against only one fixed collection of attacks.

The studies progress from a quadruple-tank process to grid-forming microgrids. They first examine hidden-state disruption, then timely detection in an interconnected electrical system, and finally incorrect localization during transient operation. The central argument is that cyber-physical defense should be evaluated against adaptive behavior and the operational decision it must support. Detecting an abnormal pattern is useful, but protection

also depends on detecting it in time and, when required, identifying the correct source.

The final study returns to the quadruple-tank process and examines a complementary question: can the defender reduce attack effectiveness by changing the information used to construct the attack? Instead of only developing a stronger detector, it modifies the measurements exposed during system identification. However, those same measurements must still allow the controller to respond to real disturbances. This introduces a security–control trade-off: limiting the value of the exposed response for attack construction while retaining the feedback required for acceptable operation.

1.2 Research Problem and Challenges

1.2.1 Limited Measurements and Hidden Physical Changes

The physical state of a system and the measurements available to its supervisor are not the same thing. A state describes an internal condition, such as a tank height or an inverter phase angle. A measurement is the information provided by the sensor and monitoring architecture. A supervisor may therefore need to assess the entire system from measurements of only part of it.

In the quadruple-tank process, four water levels characterize the physical state, whereas the sensor arrangement studied directly measures the two lower tanks [4]. In microgrid studies, supervisory detectors rely on voltage and frequency measurements at the point of common coupling. These observations provide information about the connected electrical system, but they do not directly expose every internal controller signal or inverter state.

Having fewer measurements than internal states does not, by itself, prove that a system is unobservable or vulnerable to a zero-dynamics attack. Measurements collected over time can contain information about states that are not directly measured. The security question is whether the available measurement history distinguishes the harmful behavior from behavior that the defender accepts. This distinction avoids treating limited sensing as automatic proof of an undetectable attack [2].

For this reason, the dissertation examines both the detector's internal physical behavior and the measurements it produces. Looking only at the physical states would give the defender information it does not possess. Looking only at the detector output would hide the physical consequences of a missed or delayed decision. Evaluating both allows us to describe the gap an attacker is trying to exploit.

1.2.2 Attack Generation without Explicit State-Space Models

Model-based and learning-based attack construction requires different types of information. A model-based method may use state-space matrices to calculate an attack direction. A reinforcement learning method instead uses interactions with an environment to learn which actions lead to the desired outcome. Therefore, removing the explicit matrices from the learning algorithm changes the attack-generation procedure, but it does not remove every assumption about the attacker's knowledge.

In the first study, the attacker has access to a useful replica or a digital twin of the target system for training and state approximation. The policy does not need the plant's state-space matrices as direct inputs to its learning procedure. However, the digital twin provides information through its responses. The contribution is therefore an attack learned without explicit state-space access, rather than an attack learned without any knowledge of the process [5].

This distinction matters when interpreting the later microgrid studies as well. Simulation allows a capable attacker to explore actions without repeatedly disturbing real equipment. It also gives the offline attacker access to physical variables needed to evaluate success. Results obtained under this informed-attacker setting provide evidence about the studied defense, but they should not be interpreted as showing that every attacker can obtain the same information or achieve the same performance.

1.2.3 Defenders Evaluated against Fixed Attack Patterns

A detector can perform well on recorded attacks and still fail against a new strategy. The recorded data describe the attacks that were generated, not every attack that could occur. An adaptive attacker can search for behavior that differs from those examples while still satisfying its objective.

The approach taken here is to make the attacker part of the evaluation process. After a detector is developed, the next attacker is trained against that detector and the earlier defenses retained in the framework. The new attack trajectories then provide additional evidence for detector development. This process tests whether a gain in detection survives another round of adversarial adaptation [5, 6].

This procedure does not guarantee that every later attacker is stronger or that every measured success rate decreases. Learning is performed with finite training resources, and each new policy faces a changed defense. The purpose is to repeatedly search for remaining weaknesses.

1.2.4 Detection Timing and Correct Localization

The meaning of a correct decision depends on the task. In a physical-disruption experiment, a detector that raises an alarm after the harmful condition has already been reached may be too late. In a localization experiment, a detector may recognize an abnormal event but attribute it to the wrong component. These are different ways in which a supervisory defense can fail.

The first two research chapters emphasize attacks that reach a specified physical condition before detection. The latter microgrid chapter considers a different objective: sustained attribution of a disturbance to an uncompromised inverter. In that setting, predicting the normal class is not the same as selecting the wrong inverter, and the two outcomes should not be combined into one success category.

Transient operation adds context to this problem. A load change produces a legitimate response that the detector should not automatically classify as malicious. An attacker can

time a manipulation to occur during that response. The resulting measurements reflect both the normal disturbance and the malicious action. The question is whether the supervisory system can still make the required decision from those measurements. The transient study addresses this question within its tested conditions; it does not assume that every transient is more vulnerable than every steady-state condition.

1.3 Research Aim and Objectives

The aim of this dissertation is to investigate how learning-based attackers expose weaknesses in partially observed cyber-physical systems and how iterative detector development changes the resulting security outcomes. The work considers the attacker, the physical process, and the supervisory defense together rather than evaluating any one of them in isolation.

The first objective is to determine whether a reinforcement learning policy can produce attacks with characteristics similar to model-based zero-dynamics attacks without receiving the system's state-space matrices. This includes examining the physical direction of the resulting state changes, not only whether the attacker receives a high reward.

The second objective is to investigate whether attack-generated measurement data can improve a defender against later adaptive attacks. The goal is to evaluate a sequence of attacker–detector interactions and determine what remains vulnerable after the defense is expanded.

The third objective is to study the physical meaning of attack and detection outcomes in a grid-forming microgrid. This includes the relationship between internal inverter behavior, supervisory voltage and frequency measurements, and the timing of detection relative to the attack objective.

The fourth objective is to examine sustained mislocalization during load-induced transients and determine how detector aggregation changes the outcome. This objective shifts attention from the performance of an individual classifier to the decision produced by the complete ensemble.

The fifth objective is to determine whether mixed real and surrogate measurements can

reduce a successful physical violation before alarm from attacks constructed using identified plant models. This objective considers disturbance recovery, normal-operation tracking, and physical limits alongside the security outcome.

Together, these objectives examine cyber-physical security from attack generation through detection and localization, with emphasis on how an adaptive attacker can exploit the difference between physical behavior and the measurements available to the defender.

1.4 Research Approach and Evaluation

1.4.1 Learning-Based Attack Generation

The research uses Soft Actor–Critic (SAC) to train continuous-action attack policies. SAC is an off-policy reinforcement learning algorithm that combines reward maximization with an entropy term that encourages exploration [7]. Continuous actions are appropriate for the studied manipulations because the attacker changes numerical control signals rather than selecting only among a small number of discrete commands.

The specific observations, actions, and rewards differ across chapters. In the tank process, the attacker seeks to move unmeasured tank states outside the accepted operating range while avoiding detection. In the first microgrid study, the attacker manipulates selected control-layer signals to disturb inverter phase relationships. In the transient study, the objective is to influence the supervisory localization decision. Using the same learning algorithm does not make these tasks equivalent; each requires its own physical model, information assumptions, and success definition.

Training and execution are also distinct. Training requires repeated simulations and policy updates. A trained policy can then generate actions through inference without repeating that training procedure during the event being evaluated. In particular, the study evaluates frozen attacker policies and frozen detector ensembles within each test episode.

1.4.2 Iterative Detector Development

The detector-development process begins with an initial defense and an attacker trained against it. Measurements from the attack are used to train an additional detector. The next attacker faces the resulting defense, and the procedure repeats. Earlier detectors are retained so that adding a new detector does not simply discard the earlier models.

The tank and microgrid studies use Long Short-Term Memory (LSTM) networks to analyze measurement sequences. LSTM networks are recurrent models designed to learn dependencies across sequential observations [8]. Their use here allows the detector to consider the development of a measured response rather than only its instantaneous magnitude. This is a modeling choice to be evaluated, not an assumption that a neural detector is always better than a statistical or threshold-based detector.

The final research chapter also examines how multiple detector outputs are combined. Under strict aggregation, the true compromised-inverter class remains flagged when any detector predicts it. Under consensus aggregation, confidence vectors are averaged and one final class is selected. These rules use the available detector outputs differently, so ensemble evaluation must include the aggregation decision rather than only the accuracy of the individual members.

1.4.3 Measurement Design under Control Constraints

Chapter 5 uses a different approach from the reinforcement-learning studies. The attacker fits a plant model through grey-box system identification and constructs a zero-dynamics attack from that model. The defense selects the real-measurement weight from disturbance recovery and then uses the covariance matrix adaptation evolution strategy (CMA-ES) to optimize the surrogate parameters. The objective is to reduce physical damage before alarm while satisfying the declared control requirements. The controller design and trusted detector remain fixed.

1.4.4 Physical Outcomes and Attack Success

For the physical-disruption studies, a useful common interpretation is damage-before-alarm (DBA): the specified harmful condition is reached before the detector raises an alarm. Here, "damage" refers to the simulation's defined physical limit or loss-of-operation condition. It should not be read as a measurement of irreversible hardware damage.

To express the timing relationship, let t_{phys} be the first time at which the specified physical condition is reached, let t_{alarm} be the first alarm time, and let T be the evaluation horizon. An event that does not occur within the horizon is assigned time $+\infty$. The strict before-alarm event can then be written as

$$\mathcal{E}_{\text{DBA}} = \{t_{\text{phys}} \leq T, \quad t_{\text{phys}} < t_{\text{alarm}}\}. \tag{1.1}$$

This notation explains the temporal objective; it does not replace the chapter-specific thresholds or terminal rules. Events recorded at the same sample require an explicit tie convention in an implementation and are not strictly ordered by this definition.

The mislocalization study uses a separate success criterion. It asks whether the selected aggregation rule sustains an incorrect inverter attribution. Its attack success rate is therefore not a DBA rate. Keeping these outcomes separate is necessary because correct localization, timely detection, and avoidance of physical disruption are related but distinct requirements.

1.5 Overview of Subsequent Chapters

Chapter 2 studies whether a reinforcement-learning attacker can generate zero-dynamics-like behavior on the quadruple-tank process without directly using the plant's state-space matrices. The chapter also introduces the iterative development of attackers and measurement-based detectors.

Chapter 3 extends this attacker–detector framework to a three-inverter grid-forming microgrid. It examines how attacks on control-layer signals affect the connected physical system and how successive detectors respond to adaptive attacks.

Chapter 4 considers a different supervisory objective: identifying the source of an attack during load-induced transients. It studies sustained mislocalization and compares alternative rules for combining the outputs of multiple detectors.

Chapter 5 investigates mixed real and surrogate measurements as a defense against attacks constructed from identified plant models. It explains why real feedback is needed for disturbance recovery, selects the measurement weight using a control requirement, and evaluates surrogate optimization against real-only feedback and an unperturbed surrogate. The results are assessed through damage before alarm and control performance.

Chapter 6 summarizes the main findings and contributions, discusses their scope, and presents directions for future work.

1.5.1 Conclusion and Future Work

Chapter 6 brings together the main findings of the dissertation and explains the contributions of the three research studies. It discusses the scope of the results and outlines future directions involving imperfect system knowledge, measurement selection, detector aggregation, information protection, and hardware-supported evaluation.

Chapter 2

Reinforcement Learning Approach to Generate Zero-Dynamics Attacks on Control Systems Without State Space Models

2.1 Introduction

Control Systems regulate the behavior of most of the cyber-physical systems, ranging from smart homes to critical infrastructures. Since control systems typically consist of several interconnected components, they exhibit multiple open surfaces for attacks. Multiple such attacks have been recorded to date, including one that targeted one of Iran's nuclear sites, where the attacker gained access to the industrial control systems and damaged around 1000 centrifuges by increasing their speed beyond normal [9], or the 2015 attack on the power grid in Ukraine [10], which left approximately 225,000 people without electricity, causing damage to almost half of the system's substations, which took more than two months to fix. Protecting critical infrastructures against such attacks is therefore crucial to the security of any household, company, or state.

Stealthy attacks are a type of false data injection attacks (FDIAs) on control systems, characterized by the fact that they typically go undetected. While there are many works focused on stealthy attacks, this paper focuses on a specific kind of stealthy attack, also known as a **zero-dynamics** attack. In a zero-dynamics attack, the attack signals are injected into the control signals, eventually disrupting the states of the system in such a way that the immediate effect of the state disruption is not seen in the system's measurements. Zero-dynamics-based attacks in control systems are described in [3], and [1] demonstrates the application of such an attack in power grids.

Several conditions must be fulfilled for a successful zero-dynamics-based attack to occur, as explained in [3]. One of them is that the attacker has access to the state space equations of the system to generate stealthy attack signals [3] – note that [11] demonstrates that the attack can be successful at a reasonable rate even with imperfect system knowledge, so an approximate model should suffice. Furthermore, the systems have to be of non-minimum phase, and only a subset of the states should be measured directly. The latter two conditions are primarily fulfilled in large power systems and smart grids where the supervisory control and data acquisition (SCADA) system does most of the control work based on the subset of the measured states. Nevertheless, due to the high complexity of such systems, it is usually not feasible to derive even approximately accurate state space models of the entire system. In such a case, the standard method for generating the zero-dynamics attack becomes infeasible. To address this problem, we explore the use of a reinforcement learning technique to perform an attack similar to a zero-dynamics one, in situations in which the state space model of the system is unavailable.

The recent advancements in artificial intelligence, particularly deep learning models, have led to a rise in using these models for stealthy attacks in cyber-physical systems. This paper examines the use of a reinforcement learning (RL) framework for training an agent to generate attack signals on a cyber-physical system (CPS) based on the system's current state. Given that such systems operate in a state/action space where the attack signals, state, and measurement values are continuous, we chose the deep reinforcement learning-based framework, which is known to be effective in approximating continuous functions.

To the best of our knowledge, this is the first attempt to investigate a zero-dynamics-based stealthy attack on a cyber-physical system using a deep reinforcement learning framework. We test our approach on the Quadratic Tank Process (QTP) described in [4]. The QTP is a classic example of a multivariable control system consisting of four interconnected water tanks arranged in a specific configuration as shown in figure 2.1. Each of the bottom two tanks has its own water level sensor. The water levels in the four tanks can be controlled by two pumps that regulate (through two pre-set splitting valves) the flow of water into the tanks. The goal of the control system is to maintain the water levels in the tanks at desired setpoints despite disturbances or changes in the system. For the attack to be successful, the agent (or attacker) must be capable of causing at least one of the system state components to diverge while keeping the output measurement constant (the system state is a 4-dimensional vector representing the water levels in the four tanks).

To accomplish our attack-generation goal, we initially constructed an attacker that can successfully move the system's states away from the desired value without being detected by a simple chi-square test statistics based detector. Subsequently, we developed a more robust LSTM-based detector to detect such attacks. We then created new attackers to beat the new detectors, and *added* new detectors to catch the new attacker, and the process continued iteratively until both the attacker and the *composite* detector converged, and no further improvements were observed. At this point, we claimed that the attacker and detector at the convergence stage represented the most effective attackers and detectors for this particular setup.

The main contributions of this paper are as follows:

- We developed a framework for training RL-based attackers that learn to approximate theoretical zero dynamics-based attacks. The framework involves training cumulative LSTM-based detectors until they converge to the point where they show no further improvement.

- We empirically showed that our optimal RL-based attacker approximates zero dynamics attackers rather well, in the sense that it diverges the system's states in the same

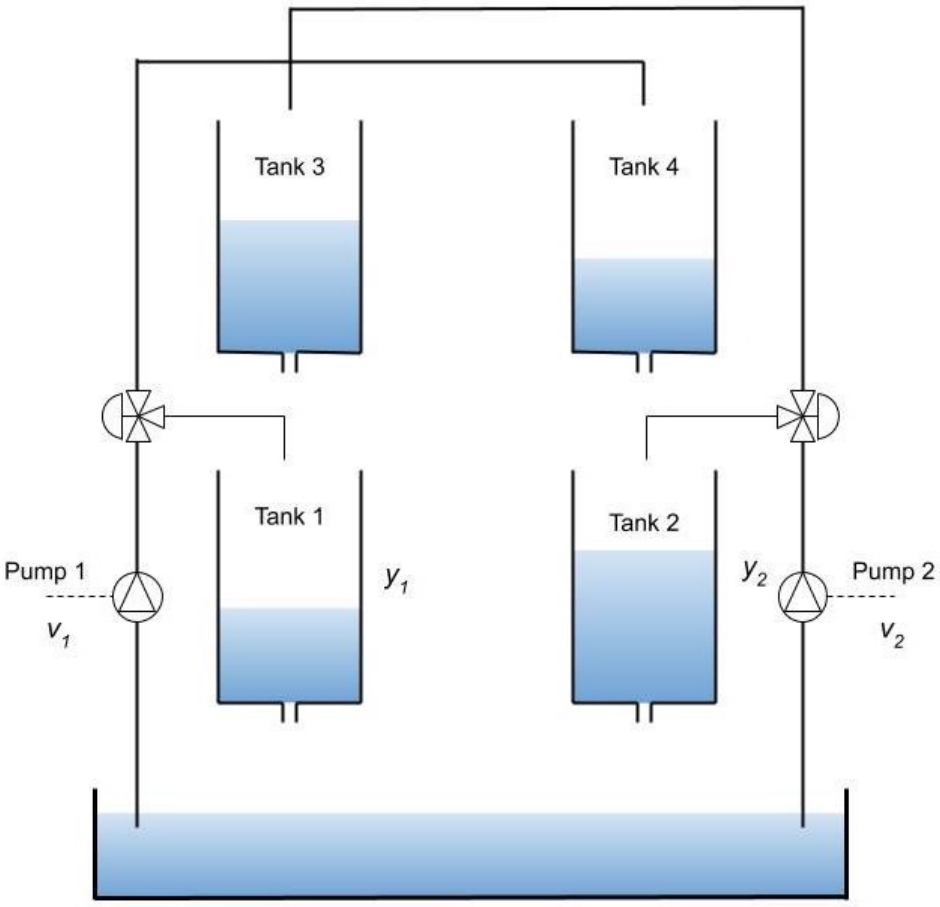


Figure 2.1: QTP consisting of four connected tanks [1]

manner (i.e., in the same direction of the state space) as the theoretical attacks of [1, 3].

2.2 Related Works

The majority of research on the security of cyber-physical systems focuses on false data injection attacks, also known as FDIAs. For example, works such as [12–14] delve into the overall process of creating FDIAs, their impact, and potential mitigation strategies. The effects of FDIAs on electricity market operations are studied in [13], while stealthy attacks on robotic vehicles by disrupting their state estimation techniques are demonstrated in [14].

While much of the existing research on FDIAs has focused on attacks that aim to modify

measurements, a zero dynamics attack [3] involves injecting attack signals into the control signals. The authors of [1] demonstrate how zero-dynamics-based stealthy attacks can drive the states of cyber-physical systems based on the quadruple tank process (QTP) beyond the desired value. The same QTP is also used in [3, 15, 16].

The authors of [15] provide a comprehensive analysis of several attacks, such as denial-of-service(DoS), replay, false data injection, and zero dynamic attacks, classifying such attacks into three dimensions, and providing insights into how attackers can exploit various vulnerabilities of complex systems. Similarly, [17] studies the trade-off between the increase in the quadratic cost and the stealthiness of the attack based on a linear quadratic Gaussian (LQG) controller. The vulnerability of industrial control systems, such as power systems, to stealthy attacks is well-established. However, protecting power systems can be incredibly challenging due to the outdated and difficult-to-replace equipment. Due to this reason, the need arises for a proper framework to enable risk analysis on each measurement and to identify the buses in a power system that are most vulnerable to attack [16].

Due to the tremendous success of deep learning models in various fields, such as recommendation engines, natural language processing, and computer vision, applying these models into cyber-physical systems is inevitable. By utilizing the power of deep learning models, we can train them to attack and protect these systems. However, the intriguing properties of neural network models make them prone to attack. For example, by perturbing an input to a deep learning detector optimally, an adversary can cause this detector to misclassify an input, eventually decreasing the detector's accuracy [18].

GANs (Generative Adversarial Networks) are a semi-supervised approach to generative modeling that automatically discover and learn patterns or regularities in input data. GANs can successfully conduct stealthy attacks on industrial control systems. By intercepting sensor measurements and control signals, GAN-based models can deviate measurements by a certain threshold without being detected by an existing black box detector [19]. In addition, GANs can be leveraged to attack power systems such that an attack's effects increase while the number of nodes to attack reduces [20]. GAN-based models can also be used for system protection by developing anomaly detectors in industrial control systems [21–23].

Deep reinforcement learning models have made their way into the optimal control setting and can often successfully replace the conventional controller. A data-driven, model-free, and closed-loop control agent based on a deep reinforcement learning (DRL) model that controls the voltage in the 200-bus test system was demonstrated in [24]. In addition, [25] also demonstrates the use of reinforcement learning in cyber-physical systems along with several case studies.

Similarly, reinforcement learning techniques can be used to perform FDIAs, as shown in [26], where the authors introduce a novel FDIA to distort the normal operations of a power system regulated by automatic voltage controls (AVCs). The attack is based on a Partially Observable MDP reinforcement learning-based framework where an agent learns to apply actions to collapse the voltage in the system with knowledge of the entire power system.

2.3 Preliminaries

2.3.1 Quadruple Tank Process

We consider a discrete time state space representation of the quadratic tank process. In a general form, such a state space representation is defined by the following equations:

$$x_{k+1} = Ax_k + Bu_k + v_k \quad \text{and} \quad y_k = Cx_k + w_k, \tag{2.1}$$

where x_k, y_k, and u_k are the states, observations, and control signal at time k, respectively.

In control systems, a controller is used to minimize the difference between the actual and desired states of the system. This paper proposes using a linear quadratic Gaussian (LQG) controller comprising a Kalman filter and a linear quadratic regulator (LQR). The LQR helps to drive the system's states to their desired values, while the Kalman filter is used for state estimation since the controller does not have direct access to state values.

The conventional LQR primarily aims to drive the system states to 0. However, to drive the states toward the desired values, we can change the traditional LQR by utilizing the backward recursion technique and solving the matrix Ricatti difference equation to obtain

optimal control signal as demonstrated in [11].

2.3.2 State space-based Theoretical Attack

Zero-dynamics-based stealthy attacks based on the state space model are described in [3]. In this version of a zero-dynamics-based attack, the attacker injects the attack signal a_k at time k into the control signal u_k, and the state value at time $k+1$ becomes

$$x_{k+1} = Ax_k + B(u_k + a_k) + v_k, \tag{2.2}$$

where $a_k = Fz_k$ is chosen in such a way that the output measurement y_k does not change during the time of an attack.

F is chosen such that $(A+BF)V^* \subseteq V^*$, where V^* is the maximal output-nulling invariant subspace of the system. The algorithm for calculating V^* is described in [27], and one way to obtain F is provided in appendix A.1. Finally, z_k is defined by a recursive equation, $z_k = (A + BF)z_{k-1}$. The initial value of z, denoted as z_0, is the Perron eigenvector of $(A + BF)$. As the attack progresses, the contribution of an attack to the diverging states is of the form $(A+BF)^k z_0$. For the attack to work at all, the system should be a non-minimum phase one. If this is the case, then at least one of the system's zeros must be outside the unit circle, leading to one of the eigenvalues of $(A + BF)$ outside the unit circle.

As illustrated above, creating attack signals necessitates an attacker's complete knowledge of the A, B, and C matrices. While it is easier to determine the state space equations for a less intricate system, such as a QTP model, accurately establishing the state space equation for larger and more complicated systems, such as power systems and smart grids, is quite challenging. As a result, a different attack method must be explored when executing zero-dynamics based attacks in these intricate systems.

2.3.3 Reinforcement learning (RL)

There are two high-level entities in RL problems: the agent and the environment. The agent observes the current state of the environment, takes action, gets some reward, and gets to a new state. The agent's primary goal is to get the maximum possible reward based on its interaction with the environment.

Markov decision process (MDP): The RL problem can be formalized as MDP containing the following elements:

- S, a set of states
- A, a set of actions
- $r : S \times A$, a reward function
- $S \times A \times S' \in [0,1]$, transition probabilities between the states

The Markov condition states that the future state only depends on the current state and current actions [28].

The main elements of the RL agent are the policy and value functions. A policy is a mapping from a state s to an action a, denoted as $\pi(a|s)$, where π is the policy function. Similarly, a value function $V_\pi(S)$ can be defined for a given state s as follows:

$$V^\pi(s) = \sum_a \pi(s|a) \sum_{s'} \sum_r p(s', r|s, a)(r + \gamma . v_\pi(s')). \tag{2.3}$$

The following equation is a Bellman equation that returns the expected return for the agent starting from state s and following a policy π. The reward value, denoted by r, is obtained by taking action a while in state s and transitioning to state s'. $\gamma \in [0, 1]$ is a discount factor that diminishes the value of future rewards. RL algorithms use various methods to learn the optimal policy, such as Q-learning, policy gradient methods [29], and actor-critic methods [28].

2.3.4 Deep reinforcement learning

Deep reinforcement learning integrates reinforcement learning with deep Learning. Here, deep learning models are used as function approximations for RL functions, such as value functions and policy functions. In addition, deep learning models can also be used to learn the model dynamics. We will now explore the main functions of deep reinforcement learning.

For a real-world scenario that consists of large-dimensional and continuous state/action pairs, we opt for the value function approximation based on the deep learning model. The value function, in this case, is represented as $V(s; w)$ and is a function approximation given by a neural network model and parameterized by the weight vector w.

Deep RL based soft actor-critic:

Generally, RL problems are concerned with maximizing long-term rewards. In addition, soft actor-critic (SAC) also seeks to maximize the entropy of the policy. SAC is an off-policy actor-critic deep reinforcement learning algorithm based on the maximum entropy RL framework.

Entropy refers to the degree of randomness or uncertainty. In SAC, entropy measures the diversity of policy actions for a given state. High entropy corresponds to the situation in which a policy function's output consists of a range of actions for a particular state. By encouraging high entropy, SAC encourages the policy to explore more diverse actions and avoid getting stuck in suboptimal policies. In other words, maximum entropy helps SAC address the brittleness issue by promoting policy exploration rather than giving a higher probability to any one action.

SAC uses a parametric probability distribution $\pi_\theta(a|s) = Pr(a|s; \theta)$. Consequently, an actor produces a probability distribution over the actions the agent can take in a given state, rather than a deterministic action. The objective function with the expected entropy of the policy over $p_\pi(s_t)$ is represented as

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim p_\pi}[r(s_t, a_t) + \alpha\mathbb{H}(\pi(.|s_t))], \tag{2.4}$$

where α is the temperature parameter that determines the relative importance of the entropy term against the reward, thus determining the optimal policy's stochasticity. Several advantages of the given objective function are described in [7]. First, this policy is incentivized to explore more; second, this policy can capture multiple modes of near-optimal behavior.

In SAC, neural network-based function approximators are used for both the Q-function and the policy. Furthermore, both these networks are optimized with stochastic gradient descent alternately. The three networks associated with SAC are a *state value function* $V_\psi(s_t)$ parameterized by ψ, a *soft Q-function* $Q_\theta(s_t, a_t)$ parameterized by θ and a *policy function* $\pi_\phi(a_t|s_t)$.

The soft value function is trained to minimize the squared residual error

$$J_v(\psi) = \mathbb{E}_{s,t\sim\mathbb{D}}\big[\frac{1}{2}(V_\psi(s_t) - E_{a_t\sim\pi_\phi}[Q_\theta(s_t, a_t) - \\ - \log \pi_\phi(a_t|s_t)])^2\big], \tag{2.5}$$

where $\mathbb{D}$ is the distribution from the experience replay.

The soft Q-function is trained to minimize the soft Bellman residual

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim\mathbb{D}}\big[\frac{1}{2}(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t))^2\big], \tag{2.6}$$

with $\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim p}[V_{\bar{\psi}(s_{t+1})}]$.

Finally, the policy parameters can be learned by directly minimizing the expected KL-divergence as given in [7], which finally yields the objective function as:

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim\mathbb{D},\epsilon_t\sim\mathbb{N}}\big[log\pi_\phi(f_\phi(\epsilon_t; s_t)|s_t) - Q_\theta(s_t, f_\phi(\epsilon_t; s_t))\big]. \tag{2.7}$$

The complete algorithm is described in [7], and it alternates between gathering data by interacting with the environment while applying the current policy and updating the function approximators using stochastic gradients from samples taken from a replay buffer.

2.4 Attacking the quadruple tank process (QTP)

Cyber-physical systems exhibit intricate structures, incorporating components from various companies or vendors across different countries. When one of these components turns malicious, it gains unhindered access to measurements, control signals, and system configurations linked to the entire system. We assume that these compromised components can replicate the target system, a digital twin of the QTP system, which is the subject of the attack [1], but not to the state-space representation of this system. This emulates a real-world attack on a more complex system, where the state-space representation is unavailable. Using the digital twin, the attacker develops a deep reinforcement learning model to orchestrate a zero-dynamics attack.

During the attack process, the attacker only possesses the measurements of the QTP being attacked. Along with these measurements, the attacker estimates state values from the digital twin as explained in 2.4.2. Subsequently, an attack signal is generated by the attacker and injected into the control signals. This manipulation causes the measurements to deviate, all while avoiding detection by the detector system. The attack cannot be initiated at the start of the system (i.e., $t = 0s$) due to the transient phase of the system [1]. Therefore, the attack is initiated after the system has reached its steady state.

Subsequently, we develop an LSTM-based detector that analyzes the directly observed measurements to detect any presence of an attack in the system. We then create new attackers to beat the new detectors, and the process continues iteratively until no further improvements are observed.

The following steps explain how we build attackers and detectors iteratively:

1. Start with a chi-square detector (DET_0);
2. Train an attacker ATT_i to beat detectors $DET_{i-1}...DET_0$;
3. Build DET_i to detect ATT_i;
4. Go back to 2.

2.4.1 Description of the QTP model

In section 2.3.1, we described the system model of the quadruple tank. As illustrated in figure 2.1, the system can be modeled by the following differential equations:

$$\begin{aligned}
\frac{dx_1}{dt} &= -\frac{a_1}{S_1}\sqrt{2gx_1} + \frac{a_3}{S_1}\sqrt{2gx_3} + \frac{\gamma_1 k_1}{S_1}v_1, \\
\frac{dx_2}{dt} &= -\frac{a_1}{S_2}\sqrt{2gx_2} + \frac{a_4}{S_2}\sqrt{2gx_4} + \frac{\gamma_2 k_2}{S_2}v_2, \\
\frac{dx_3}{dt} &= -\frac{a_3}{S_3}\sqrt{2gx_3} + \frac{1-\gamma_2 k_2}{S_3}v_2, \\
\frac{dx_4}{dt} &= -\frac{a_4}{S_4}\sqrt{2gx_4} + \frac{1-\gamma_1 k_1}{S_4}v_1,
\end{aligned}$$

where S_i is the cross-section of tank i, x_i is the water level of tank i, v_i is the voltage applied to pump i, causing a flow $k_i v_i$, a_i is the cross-section of the outlet hole and γ_1, $\gamma_2 \in (0,1)$ are the flow ratios set by the fixed-position valves.

Based on this model, our next goal is to instantiate the parameters and linearize the model to obtain the state matrices.

$$A = \begin{bmatrix} 0.975 & 0 & 0.042 & 0 \\ 0 & 0.977 & 0 & 0.044 \\ 0 & 0 & 0.958 & 0 \\ 0 & 0 & 0 & 0.956 \end{bmatrix}, B = \begin{bmatrix} 0.0515 & 0 \\ 0 & 0.0447 \\ 0 & 0.0737 \\ 0.085 & 0 \end{bmatrix} \text{ and } C = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \end{bmatrix}$$

.

The maximum value of the height of the tank is 20 cm. The initial height of the tank is given by $x_0 = \hat{x_0} = [0\ 0\ 0\ 0]^T$ and the desired values of the states at steady state are given as $x_d = [10\ 10\ 10\ 10]^T$.

The control system is exposed to various external disturbances in a real-world scenario. This experiment simulates these real-world disturbances through process and measurement noise, modeled as white Gaussian noises with a variance of 10^{-3}.

The system is controlled by an LQG controller, and a Kalman Filter state estimation technique is used to estimate the system states, as described in 2.3.1.

2.4.2 Deep Reinforcement Learning based attacker

In our experiment, the control signals correspond to the system's voltage, so the attack signals must be continuous. Therefore, we train the attacker to generate continuous attack signals. In this study, we employ the soft actor-critic (SAC) algorithm as the basis for our attacker, as explained in section 2.3.4.

The input to the RL-based attacker comprises current observations of the system. In our experiment, the observations (states) for the RL agent are based on a sliding window of size 10 of past observations. We train the attacker to produce attack signals based on a combination of states (x) and measurements (y). However, in a real-world situation, an attacker does not have access to the system's states (x) but only to the measurements (y), and control signals (u). To address this limitation, the attacker runs a parallel copy of the system to estimate the state values (x) while using measurements (y) from the actual system where the attack is being performed. Figure 2.2 illustrates the attack process. Despite using the state values (x) from another clone system, the process and measurement noises of the two systems are different, as the noises are sampled randomly from a Gaussian distribution. This variation in noise makes it more challenging for an attacker to generate the desired attack signals without detection. Nonetheless, we have observed that the attacker generalizes well in this scenario and can successfully attack the system with a high probability.

The attacker's RL based framework involves multiple Neural Network models to approximate the function of the actor and critic, in addition to another value network. The overall architecture of these networks for all the attackers is shown in appendix A.2 table A.1.

Reward shaping.

The rewards received by an attacker in a cyber-physical system are proportional to their performance. The states in the system correspond to the height of individual tanks, and if

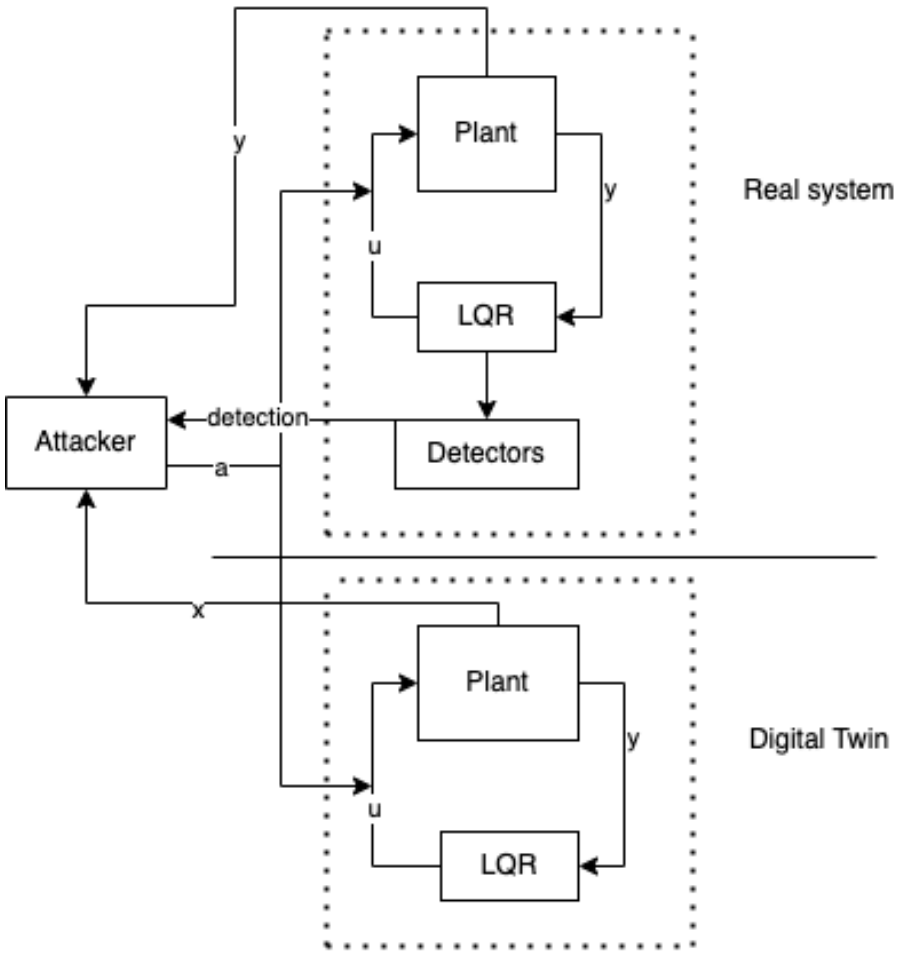


Figure 2.2: Architecture of the process during an attack

an unobserved state falls outside the range of 0 to 20 cm, the system is considered to have diverged beyond the model's bounds, indicating a successful attack. The agent is rewarded based on the degree of divergence in unobserved states and penalized based on the degree of divergence in observed states. There are three different kinds of rewards the attacker receives at each timestep.

1. *Success*: A scenario where the attacker successfully attacks the system
   - $reward = -15 + 10 \cdot (2 + d_2)^3$.
2. *Failure*: A scenario where the detector detects an attack before an attack is successful.
   - $reward = -(25 + 10 \cdot (2 + d_1)^3)$ if DET_0 detects an attack

- $reward \mathrel{-}= (20 + (5 \cdot (i-1)) + (2 + d_1)^3)$ for all n LSTM-based detectors DET_i, $i \in (1, n)$ if DET_i detected an attack.

3. - An attacker receives a small reward at each timestep based on how close it is to a goal. $reward = -15 - a \cdot (1 + d_1)^2 + b \cdot (1 + d_2)^2$.

where d_1 represents the Euclidean distance between the desired and current values of $[x_1, x_2]$, while d_2 represents the Euclidean distance between the desired and current values of $[x_3, x_4]$. Here, x_1 and x_2 are directly measured states, while x_3 and x_4 are non-measured states. Similarly, the values of a and b are different for different attackers, ATT_i, $\forall i \in (1, n)$ as shown in appendix A.2 table A.2. The reward function incorporates numerical values derived from experimental evaluations, lacking a formal scientific derivation method.

Furthermore, it is worth noting that reward functions in deep reinforcement learning are often non-linear, which enables more intricate connections between the agent's actions and the rewards it obtains. The rationale behind this is that deep reinforcement learning agents operate in high-dimensional spaces where a linear reward function may not be capable of comprehending the complexity of the task or environment.

2.4.3 Chi-square based anomaly detector (χ^2)

Our initial detector is a windowed χ^2 test statistics-based detector that detects changes in the variance of the observations. Here, we calculate the sample variance from the $y_residuals$, which is the difference between the actual and estimated measurements. Our detector is identical to the one employed in [11].

2.4.4 LSTM based Detector

Once we successfully create an attacker that can beat the Chi-square-based anomaly detector, we train an LSTM-based detector to identify if the system was being attacked [8]. The time series measurements that correspond to the directly observed states are inputted into the LSTM detector to analyze whether the system was under attack. The architecture of the

LSTM-based detector, along with its dataset, is explained below.

Architecture of LSTM

Each measurement at time t is represented by $y^{(t)} = \{y_1^{(t)}, y_2^{(t)}\}$, with $y_1^{(t)}$ and $y_2^{(t)}$ representing the measurements for states x_1 and x_2 respectively. We configured the LSTM detector to have an input size of 50 time series measurements, such that at each timestep t, the input to the LSTM network was a sequence of the most recent 50 measurements, i.e., $Y = \{y^{(t)}, y^{(t-1)}, y^{(t-2)}, ..., y^{(t-49)}\}$. On the other hand, the output consists of a single node representing whether or not the input system is under attack.

The main advantage of the LSTM-based detector over a simple Chi-square-based anomaly detector is that it can learn the trajectory of the directly measured states $y^{(t)} = \{y_1^{(t)}, y_2^{(t)}\}$ of the QTP to identify any anomalies in the system. All the detectors in this paper share the same architecture. The architecture of these LSTM detectors is shown in appendix A.2 table A.3.

Dataset Collection

The dataset contains two distinct categories of measurements: those taken during an attack and those taken during a non-attack scenario. To obtain measurements for the non-attack scenario, we conducted multiple simulations of the system without any attack and recorded the corresponding measurements. For the attack measurements, we ran the simulation until the system reached a steady state and then launched an attack at time-step 100. We generated measurements for the attack scenario until the system diverged completely or the attack was detected. Additionally, we ensured that our dataset included measurements corresponding to the transient state of the system. While generating these measurements, we added random process and measurement noise to these simulations to make the simulation more realistic and to replicate real-world phenomena.

2.4.5 Two proportion Z-test

We utilize a statistical hypothesis test called the two-proportion z-test to assess the performance of two consecutive attackers or detectors to determine if two proportions are identical where these proportions represent the binomial distribution. To conduct the test, we determine the rate of successful attacks (p) across n trials for the attacker and the rate of successful detection for the detector before any successful attacks across n trials. We then take two independent samples from these proportions and compute the z-statistic, assuming the samples were randomly selected.

$H_0 : p_1 = p_2$, $H_a : p_1 \neq p_2$ and z value is calculated using

$$z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_1} + \frac{1}{n_2})}}, \text{where } \hat{p} = \frac{n_1\hat{p}_1 + n_2\hat{p}_2}{n_1 + n_2}$$

and $\hat{p}_1$ and $\hat{p}_2$ are the rates of success of the two proportions used during this test, and n_1 and n_2 are the number of trials performed. After calculating the test statistic z, we calculate the corresponding critical region to compare the test statistic. If the absolute value of z-statistic is greater than or equal to the critical value, we can reject the null hypothesis or conclude that the two distributions are not the same.

2.5 Results

As explained in section 2.4, we iteratively develop attackers that can overcome all previous detectors and detectors that can recognize the new attacker until further developing new attackers would not improve attack success rates, and developing new detectors would not reduce the probability of an RL-based attacker's success.

2.5.1 RL based attackers

The success rate of attackers is presented in table 2.1. From this, we can infer that the attacker can still carry out successful attacks even when the states (x_s) are estimated using

the parallel-running digital twin system. Furthermore, we observe that the likelihood of a successful attack decreases as the number of detectors increases, making it more challenging for an attacker to succeed. However, we cannot assert that the system is secure from zero-dynamics attacks unless the successful attack rate is 0%.

Likewise, table 2.1 also illustrates the duration it takes for the system to deviate after an attack has been initiated. It is worth noting that our experiment's sampling time is set to one second, meaning that each timestep corresponds to a one-second interval. An increment in the number of detectors results in more time for an attacker to devise an optimal attack signal to accomplish a successful attack on the system. This trend can be seen in the mentioned results table.

The success probabilities of our last two attackers were 1.9% and 0.9%, respectively, which are lower than those of the earlier attackers. Despite this, all the attackers could attack the QTP system successfully. Figure A.1 in appendix A.3 depicts the reward diagram for the five attackers during the training process. Specifically, figures A.1(d) and A.1(e) illustrate that the agents were unable to reach the global optima and obtain less reward as compared to the previous attackers.

Table 2.1: Performance of different attackers. Each attacker's (ATT_i) attack rate indicates their capability to beat collective previous detectors d_{i-1} to d_0.

Attacker	**Successful Attack Rate**	**Timesteps for successful attack**
ATT_1	70%	27
ATT_2	59%	42
ATT_3	50%	63
ATT_4	1.9%	49
ATT_5	0.9%	71

2.5.2 LSTM based detectors

Table 2.2 presents the results of our experiment on the performance of LSTM-based detectors. Each detector in the table is associated with a specific threshold value. If the detector's output is below the threshold, it indicates the presence of an attack; otherwise, it is consid-

ered normal. The threshold value was selected to minimize the false positive or false alarm rates. Using this concept, we chose the threshold value that ensures the false positive rate remains below 0.0001.

In addition, selecting the threshold value involves balancing the false positive rate and the true positive rate (TPR). Although the TPR in table 2.2 does not seem to be perfect, it should be noted that the system does not fail immediately, and it requires multiple time steps of attack signals to cause system divergence. Additionally, all four detectors in our experiment can detect an attack generated by their corresponding attacker before the attack succeeds at a reasonable rate, as evidenced by the results in figure 2.3 where an arrow pointing from DET_i to ATT_i indicates the successful attack rate of ATT_i against DET_i which is pretty low. Meanwhile, the arrow pointing from ATT_i to DET_i indicates the success rate of an attacker's attack towards its corresponding detector and the detectors preceding it. Consequently, these results illustrate the feasibility of constructing functional LSTM-based detectors tailored to RL-based attackers.

Table 2.2: Reported test-set performance of the individual LSTM detectors. FPR and TPR are proportions. TP, FP, FN, and TN denote true positives, false positives, false negatives, and true negatives, respectively.

Name	DET_1	DET_2	DET_3	DET_4
Threshold	0.001	0.001	0.002	0.001
Test set TP	20810	68223	88345	67175
Test set FP	7	4	2	0
Test set FN	8037	21850	1733	13795
Test set TN	89930	89635	89632	90038
Test set FPR	0.00007	0.00004	0.00002	0.0
Test set TPR	0.72	0.76	0.98	0.83

2.5.3 Test of Convergence

Figure 2.3 shows the success rate of attackers against the corresponding detectors as explained in section 2.4. We conducted a hypothesis test to evaluate the performance of the two consecutive attackers and detectors, as explained in section 2.4.5. The test was conducted at a 95% confidence level, and we compared the absolute value of the test statistic

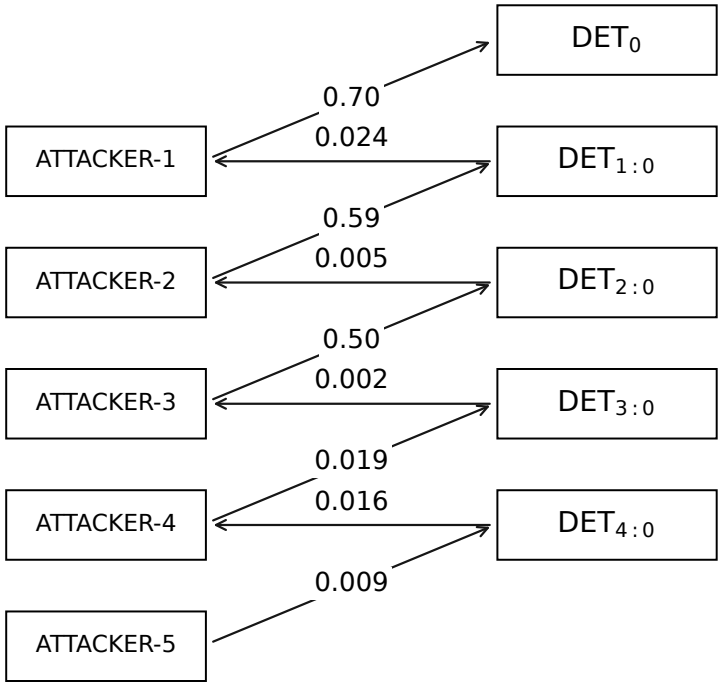


Figure 2.3: Iterative attacker–detector evaluation. The label $DET_{j:0}$ denotes the accumulated set $\{DET_j, \ldots, DET_0\}$. Diagonal arrows report attack success probabilities before adding the next detector, and horizontal arrows report them after its addition. Both evaluations use the entire accumulated detector set shown.

(Z) with the critical region value of $z_{\alpha/2} = 1.96$. If the absolute value of Z is less than 1.96, we accepted the null hypothesis that the two proportions are equal, which implies that the two subsequent attackers or detectors follow identical binomial distributions.

Table 2.3 displays the results of our hypothesis testing. The results show that the Z-value of 1.4 indicates no significant difference in the success rate of ATT_4 against DET_4 and the success rate of ATT_5 against DET_4 and the previous detectors. On the other hand, the Z-value of 0.51 indicates that the DET_3 (along with the earlier detectors) and DET_4 detection rates against ATT_4 are similar. Through this empirical evidence, we concluded that neither the attacker nor the detector would improve further. Therefore, based on our experiment, we affirmed that the best attacker and the best detector in this scenario are the ATT_5 and the DET_4, respectively.

Table 2.3: Reported two-proportion tests for successive attackers and detectors. Each evaluation identifies the attacker and the accumulated detector set it faces; p_1 and p_2 are successful attack probabilities.

$DET_{j:0} = \{DET_0, \ldots, DET_j\}$: all detectors accumulated through iteration j.				
Evaluation 1	Evaluation 2	$p_1,\ p_2$	Z	Decision
$ATT_1 \to DET_{1:0}$	$ATT_2 \to DET_{1:0}$	0.024, 0.590	-27.4388	Reject
$ATT_2 \to DET_{2:0}$	$ATT_3 \to DET_{2:0}$	0.005, 0.500	-25.477	Reject
$ATT_3 \to DET_{3:0}$	$ATT_4 \to DET_{3:0}$	0.002, 0.019	-4.31	Reject
$ATT_4 \to DET_{4:0}$	$ATT_5 \to DET_{4:0}$	0.016, 0.009	1.4	Not rejected
$ATT_1 \to DET_0$	$ATT_1 \to DET_{1:0}$	0.700, 0.024	31.45	Reject
$ATT_2 \to DET_{1:0}$	$ATT_2 \to DET_{2:0}$	0.590, 0.005	28.6	Reject
$ATT_3 \to DET_{2:0}$	$ATT_3 \to DET_{3:0}$	0.500, 0.002	25.68	Reject
$ATT_4 \to DET_{3:0}$	$ATT_4 \to DET_{4:0}$	0.019, 0.016	0.51	Not rejected

2.5.4 RL vs. Theoretical attack

In figure A.2 of appendix A.3 it is observed that the states (x_3, x_4) that diverge follow the direction of the highest eigenvector associated with the largest eigenvalue of $(A + BF)$, which is the underlying principle of the theoretical attack explained in 2.3.2. The cosine similarity between the attack contributions to the diverging states and the highest eigenvector associated with the largest eigenvalue of $(A + BF)$ is shown in the figure. The results demonstrate that for all five attackers, the cosine similarity approaches 1 at the point of system divergence. This outcome indicates that the RL-based attacker emulates the theoretical zero-dynamics attack.

2.6 Conclusion

We introduced a novel concept of zero-dynamics attack on the QTP process using deep reinforcement learning, which differs from previous works focused on theoretical zero-dynamics based attacks. The theoretical attacks required an attacker to have perfect knowledge of the system state space to generate attack signals. In contrast, our work showed that an RL-based attacker does not require such knowledge and is robust to different system noises. Furthermore, our study highlights the superiority of LSTM-based detectors over chi-square

statistics based detectors. We conclude that not all detectors are equally effective in detecting system anomalies. We also used a cosine similarity technique to show that the RL-based attacker imitates a theoretical attacker.

2.7 Discussion and Implications

2.7.1 What the Model-Free Result Establishes

The main result of this chapter concerns the construction of the attack. In the theoretical approach, the attacker uses the system matrices to calculate a sequence that exploits the plant dynamics. In the learning-based approach, the attacker discovers a policy through repeated interaction with a digital twin. The successful learned attacks therefore show that explicit matrix access is not necessary for this attack-generation procedure. They do not show that information about the plant is unnecessary.

The digital twin is important in two ways. It provides an environment in which the policy can explore attack actions, and it supplies approximate state information during the evaluated attack process. This distinguishes the experiment from a setting in which the attacker observes only a short record of unknown input–output behavior and has no useful simulator. The result should be interpreted within the information access described in Section 2.4.2.

The use of different noise realizations in the digital twin and target process also gives the result a specific meaning. Successful attacks are not limited to replaying one identical noisy trajectory. However, variation in process and measurement noise is not the same as uncertainty in the plant parameters, controller structure, or sensor configuration. The experiment supports the reported performance under its noise model; broader transfer across different plants would require a separate evaluation.

2.7.2 Interpreting the Connection to Zero Dynamics

The comparison in Section 2.5.4 adds information that an attack success rate alone cannot provide. A policy might push a tank beyond a limit through a large, easily recognized disturbance. Such an outcome would show that the process can be disrupted, but would not explain whether the policy learned behavior related to a zero-dynamics attack.

Here, the learned trajectories are also compared with the state-space direction associated with the theoretical attack. Their reported directional agreement suggests that the policies exploit a related physical mechanism rather than merely producing arbitrary disruption. This is useful because the policy is trained through rewards and does not receive an instruction to follow the theoretical eigenvector.

Directional agreement nevertheless has a limited interpretation. Cosine similarity measures alignment, not the magnitude of the state change or the time required to reach a physical limit. Two trajectories can point in nearly the same direction while producing different output deviations and different detection outcomes. The similarity result should therefore be read together with the state trajectories and the attack success measurements, rather than as a replacement for them.

There is also a distinction between an exact output-nulling construction and an attack that remains below a particular detector's sensitivity. The learned attacks are evaluated in the presence of noise, finite observation windows, and decision thresholds. An attack can resemble the theoretical mechanism while leaving small measurement patterns that another detector can recognize. This helps explain why comparing the learned behavior with zero dynamics and improving detection through LSTM models are not contradictory results.

2.7.3 Reading the Iterative Results

The progression from the initial attacker to the later attackers shows the value of repeatedly challenging the defense. Each new detector adds patterns learned from a previously successful attack. A later attacker must find behavior that avoids the expanded collection, rather than only the initial statistical test. Retaining earlier detectors also preserves their decision

functions instead of replacing them with a model trained only on the latest data.

The successful attack rates in Table 2.1 should be interpreted as outcomes of this changing interaction. The first and last attackers do not face the same defense. Their rates therefore describe the progression of the framework, not a direct ranking of policies against one fixed opponent. A later policy can be more specialized while succeeding less often because the defense it faces is stronger.

The statistical comparisons provide an empirical stopping criterion. A non-significant difference means that the test did not establish a difference at the selected significance level. Accordingly, the final attacker and detector are best understood as the final evaluated models under this procedure, rather than globally optimal solutions.

2.7.4 Connection to the Microgrid Studies

The quadruple-tank process makes the relationship between measured outputs and unmeasured physical states easy to examine. Its role is not to reproduce the full complexity of a power network. Instead, it provides a controlled setting in which attack generation, detector adaptation, and physical consequences can be studied together.

The next chapter carries this experimental idea to an interconnected microgrid. The physical variables and time scales change, but the evaluation question remains similar: can a learned manipulation reach its physical objective before the supervisor recognizes the event? This transition allows the dissertation to examine whether the iterative attacker-detector approach remains useful when the consequences of a local manipulation spread across several connected devices.

Acknowledgements

This publication was made possible by NPRP grant #12C-33905-SP-165 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors. We appreciate the anonymous reviewers' valuable

suggestions and comments.

Chapter 3

Robust Detection in Power Systems: Iterative Reinforcement Learning Based Adversarial Training

3.1 Introduction

The ever-increasing integration of renewable energy sources into the power grid has accelerated the adoption of microgrids. These microgrids are characterized by their ability to operate autonomously and independently from the main grid while utilizing local generation to meet energy demands. These microgrids operate in a decentralized fashion, which enhances the reliability and resiliency of the microgrid. Furthermore, the autonomous and seamless operation of the microgrid is highly dependent on the continuous communication between the inverters, control layers, and measurement systems. However, this extensive reliance on communication increases their vulnerability towards cyber intrusions, especially stealthy attacks, which remain undetected by the intrusion detection system while disrupting the network [30].

Stealthy attacks represent a sophisticated category of cyberattacks designed to avoid detection by traditional security measures, employing advanced techniques to manipulate

system behavior. Research studies have explored stealthy attacks or false data injection attacks (FDIAs) in control systems, including injection of attack signals into measurement signals to evade detection by security detectors [31–34], and direct injection of attack signals into control signals, leading to undetected deviations in system states [5, 15, 35, 36]. These attacks can severely impact system integrity while maintaining the appearance of normal operation.

Recent advancements in artificial intelligence have enabled the execution of sophisticated stealthy attacks by leveraging AI's capabilities [5, 24, 26, 37, 38]. The sophistication of such attacks arises from the fact that attackers do not necessarily require knowledge of the system parameters [5], unlike mathematical-based attack models [36]. The authors of the paper [5] conduct zero-dynamics attacks using a digital twin model of the system, training the attack solely based on the system's input and output.

Recent studies on detecting stealthy attacks, particularly those employing deep learning, have demonstrated high effectiveness in identifying system vulnerabilities. These vulnerabilities often go undetected by conventional detectors, such as statistic-based detectors [5] or Kalman Filter-based state estimation detectors [38]. Consequently, there has been a significant surge in developing robust deep learning-based detectors [39–41]. However, state-of-the-art time series classification based detectors [41] often underperform because they are not trained on datasets containing diverse attack scenarios. This limitation opens the possibility of training sophisticated attackers capable of evading these detectors. Furthermore, the resulting dataset is highly imbalanced due to the scarcity of attack instances, as reflected in their F1 scores.

In this paper, we address the issues outlined above by initially developing soft actor-critic (SAC) DRL based attackers that interrupt the normal operation of a microgrid [7]. The goal of these attackers is to manipulate signals in the physical layer, causing them to deviate from their normal set points while leaving signals in the supervisory layer unaffected. As a result, irreversible damage may occur by the time any detector in the supervisory layer detects an anomaly. Subsequently, we develop a robust detector to identify attacks from these attackers, with the aim of minimizing their impact.

A similar approach to train detectors is studied in [5] by conducting stealthy attacks on a quadratic tank process (QTP). In this study, the authors employed an iterative training approach to optimize both attackers and detectors to their highest capabilities. Initially, an attacker was trained to execute a stealthy attack on the QTP, outperforming a conventional Chi-square-based detector. Subsequently, the authors developed a robust neural network-based detector to identify the previously trained attacker. Following this, a new attacker was trained to surpass both the initial and neural network-based detectors. This iterative process continued until a point of convergence was reached, where no further improvements were observed in the attacker or detector's performance. At this stage, the authors concluded that the attacker and detector had reached their optimal capabilities within this setup. This iterative training process of both attackers and detectors proves particularly beneficial because, in each iteration, the attacker evolves to outperform all previous detectors, while the detector is refined to detect the most recent attacker. Leveraging multiple intelligent attackers enables the generation of diverse datasets, facilitating the training of a more robust detector with each iteration. The authors of the paper [38] discuss a similar approach to training a robust detector. In this approach, a single attacker undergoes training to attack the system, generating a dataset utilized for online training of a neural network based detector. Throughout this process, the attacker and detector evolve, leading to final models that differ from the initial ones. However, in the iterative training process explained in [5], the effectiveness of the initial models in the final iterations of attackers and detectors is maintained.

The authors of [5] conduct a series of statistical tests to ascertain convergence. Their system, a quadratic tank process (QTP), is notably simpler in complexity compared to the microgrid we are examining. Moreover, they do not assess the effectiveness of various attackers and detectors and their implications on the system. In contrast, our main objective is to construct robust detection systems to prevent any successful attacks empirically. We have achieved this goal without relying on statistical tests for validation. In our training process, as we iteratively develop attackers and detectors, we may reach a point where a new attacker cannot successfully disrupt the microgrid, achieving a zero success rate. In such

cases, we conclude our experiment by determining that the detector has achieved optimal performance within this specific setup. However, it is important to note that the final detector is not universally the best detector. With the advancement of AI-based models, like large language models (LLMs), it may be possible that superior LLMs could be trained to execute the same stealthy attack and surpass our latest detector. However, it's important to acknowledge that deploying such LLMs in this context is farfetched and faces practical challenges due to their size, training/deployment costs, and execution speed, especially given the high sample rate of power systems.

Our microgrid simulator in Figure 3.1 includes three grid forming Inverters (GFMIs) that are connected to the point of common coupling (PCC) through an LCL filter to smooth out the output voltage by eliminating the unwanted harmonics, and to enhance the power quality of the microgrid. Each GFMI is controlled via a virtual synchronous generator (VSG) that ensures voltage and frequency support at the PCC of each inverter. The model predictive control generates the switching sequences according to the generated internal reference voltage and measurement units. Table 3.1 provides the signals and their corresponding descriptions linked to this system.

The main contributions of this paper are as follows:

- We design an iterative framework to develop attackers and robust detectors for the microgrid in Figure 3.1. In each iteration, we train a new detector that detects an attack from the recent attacker. In this way, the subsequent attacker and detector become more robust than their predecessors, enhancing their ability to attack the system and detect an attack successfully. In the end, the new attacker can no longer perform a successful attack on the system, and the last detector is the best possible detector in this framework.

- We investigate the potential impact of our DRL-based stealthy attackers on the microgrid and assess the effectiveness of detectors across multiple iterations. Our findings demonstrate that detectors developed through this mechanism are more effective at quickly identifying anomalies, thereby mitigating system impacts.

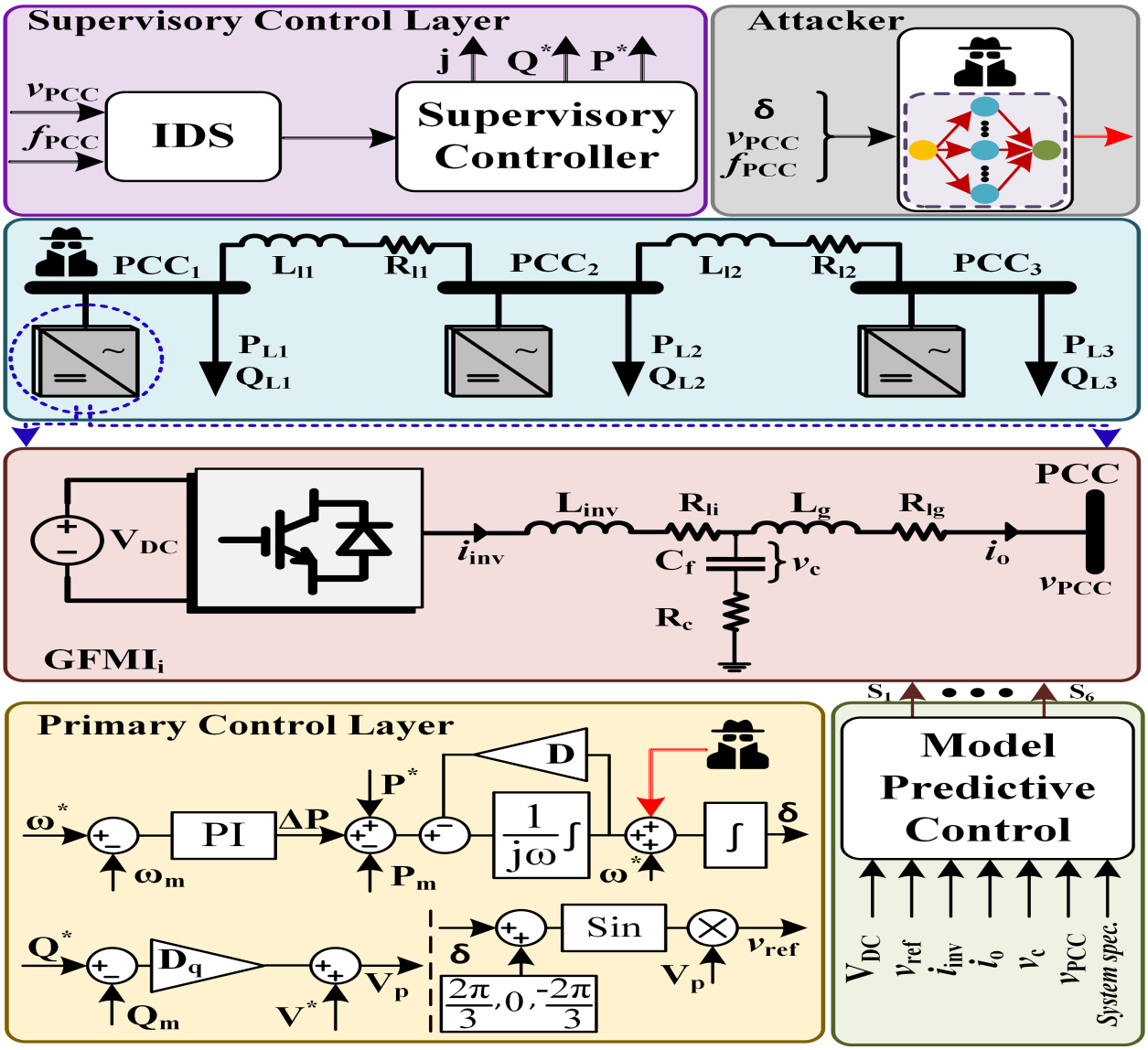


Figure 3.1: Power system

Symbol	Description	Units
V_{PCC}	PCC voltage	V
f_{PCC}	PCC frequency	Hz
δ	Phase angle	Radians
J	Moment of inertia	Kgm^2
f^*	Nominal frequency	Hz

Table 3.1: Signal description and their mnemonics

3.2 Literature Review

Cyber-physical systems (CPS) face many cyber threats that could disrupt their normal operations. The authors of [15] analyze various attacks such as denial-of-service (DoS), replay, false data injection, and zero dynamics attacks. These attacks are classified into three dimensions based on their nature: system knowledge, disclosure, and disruption resources. CPS attacks span across different sectors, including power systems, electricity markets, autonomous vehicles, and electric vehicle charging infrastructures [14, 26, 31, 35, 36, 42–44].

Modern power systems are increasingly complex, driven by growing energy demands and the integration of various energy sources. As a result, there's a growing emphasis on distributed control strategies, particularly in microgrids, to enhance efficiency and scalability. However, the lack of a centralized controller for monitoring global information leaves these systems vulnerable to cyberattacks and FDIAs [31–33].

While conventional cyber attacks are relatively easy to detect for distributed observer-based techniques, stealthy attacks present a challenge. A well-executed stealthy attack on measurements can evade traditional detection methods by achieving control objectives without causing physical errors [26, 31–34, 37]. Similarly, a separate class of stealthy attacks known as zero-dynamics attacks (ZDAs) is explored in [3, 5, 15, 35, 36, 42, 45].

The authors in [31] explore a stealthy attack on a distributed observer-based microgrids strategy, introducing false data into voltage measurements that go undetected. Similarly, the authors in [32] offer robust mitigation strategies against such attacks, leveraging discordant elements and consensus theory. Additionally, the importance of machine learning algorithms for detecting FDIAs is explained in [33, 34]. The main insights from these papers [31–33] is the need for data-based techniques to reduce dependency on mathematical models for attack detection, especially when these models are unavailable, which also minimizes computational overhead.

Similarly, in ZDAs, attack signals are injected into control signals, disrupting the system's states in a manner that doesn't immediately reflect in the system's measurements. For this to occur, the system should be non-minimum phase, and only a subset of states should be

directly measured. The analytical framework for this attack is shown in [3, 15], while the authors of [35, 36, 42] demonstrate its application in a quadratic tank process (QTP) [4] and power systems. Additionally, [5] uses deep reinforcement learning on QTP to show that attackers can execute sophisticated ZDAs without knowing system parameters.

While Deep Reinforcement Learning (DRL) techniques have recently gained popularity in optimal control settings and are utilized in cyberphysical systems [24, 25], they can also be exploited for various types of attacks on CPSs, as demonstrated in [5, 26, 37, 38]. The authors of the paper [26] present a novel FDIA technique that disrupts the normal operations of a power system regulated by automatic voltage controls (AVCs), causing a voltage collapse. Similarly, a DRL-based attack targeting a CPS with modular multilevel converters (M2Cs) is discussed in [37]. Likewise, the authors in [38] demonstrate an attack on sensor measurements that detect and report power consumption by appliances in a network. In this attack, the signal is injected into the capacitor voltage measurements of M2Cs to evade the FDIA detector based on a Kalman filter at the control layer. Additionally, [5] shows that ZDAs can be executed without knowing the system parameters using DRL-based attackers. The papers [26, 37] demonstrate the effectiveness of DRL-based attackers in bypassing existing detectors. However, it is worth noting that these detectors are trained online, continuously updating their parameters, thereby rendering previous attack strategies ineffective. On the other hand, [5] introduces an iterative framework for training attackers and detectors. This iterative process allows both attackers and detectors to evolve and improve through adversarial training over successive iterations.

Given the vulnerability of modern power systems to a wide range of attacks, the development of robust detection mechanisms is of utmost importance. The necessity for employing data-based techniques in building these detectors is highlighted in [5, 31–33, 38, 41, 46, 47]. These detectors vary in their implementations and architectures, showcasing diverse approaches in the literature. For instance, [47] proposes an implementation based on Support Vector Machines, while [5] delves into the application of LSTM-based detectors. Additionally, [46] discusses utilizing an optimized extreme learning machine for detecting FDIAs using autoencoders. However, many of these techniques encounter an issue with imbalanced

datasets due to the inherent difficulty in generating attack-related data without a comprehensive understanding of the system's vulnerabilities. To address this challenge, [5] explores iterative training methods for attackers to uncover system vulnerabilities, subsequently facilitating the development of robust detectors using this acquired information. Our study employs this framework to establish robust detection mechanisms within the microgrids.

3.3 Compromising the microgrid

In our experiment, the attacker is based on a Deep Reinforcement Learning (DRL) framework. This attacker observes the current state of the system, generates attack signals, and injects them into control signals transmitted from the supervisory layer, which ultimately affects the system without being detected by a detector, as shown in Figure 3.1.

Digital-twin scope The digital twin represented here is a three-inverter microgrid. The attacker is assumed to have a digital twin of this system to train attack policies, while the defender has its own digital twin to develop attackers whose trajectories are subsequently used to train detectors. Each twin represents the electrical network, loads, inverter dynamics, local control loops, and supervisory measurement and command interfaces. Conceptually, the two parties maintain independent twins and train their own sets of attackers and detectors through the iterative procedure as described in this study. The attacker uses the resulting policies to manipulate the selected inertia and nominal-frequency control signals, whereas the defender uses the generated normal and attack trajectories to develop detectors to be deployed at the supervisory layer. These detectors monitor PCC voltage and frequency. Attacker training uses the specified inverter observations and detector feedback available within its training environment; access to the supervisory controller's internal implementation is not required, although the relevant closed-loop responses must be represented. Because both roles use the same system model and training procedure under the assumptions of this study, the reported experiments represent them using a single simulator. This shared simulation framework does not imply that the two parties share their trained models.

3.3.1 Impact of FDIA on microgrids

FDIA significantly affects microgrid operation by tampering with data exchange among energy sources, control layers, and measurement units. This manipulation can disrupt normal microgrid function, leading to inaccurate decision-making by the supervisory control layer and potential supply/demand imbalances. Additionally, FDIA can delay the detection of system failures, potentially causing cascading failures, inverter damage, and blackouts.

3.3.2 Attacker's goal

A successful attack occurs when the phase angle (δ) in the physical layer diverges without causing any alterations in the output f_{PCC} and V_{PCC} that goes to the supervisory layer for a sustained time duration. The attack fails if detected by a detector in the supervisory layer before a successful attack.

3.3.3 Detectors

The supervisory control layer monitors the voltage and frequency of the grid at the PCC to evaluate the operating stage of the microgrid. If the V_{PCC} and f_{PCC} deviate from their nominal values within a predefined threshold according to IEEE-1547 [48], the supervisory layer is notified by its intrusion detection system.

Threshold based detector

The threshold-based detector monitors the V_{PCC} and f_{PCC} signals over a specific period-7 line cycle in our case. When these signals in the supervisory layer deviate beyond the thresholds for approximately 7 line cycles, this indicates a system under attack as detected by this initial detector. In our case, at any given time, if more than 50% of the signals in the past 7 line cycles are beyond the threshold, the detector detects it as an anomaly. This threshold is defined as either one of the boundaries of the normal operation intervals $V_{PCC} \in [125, 150]$ and $f_{PCC} \in [59, 61]$. This detector represents the first line of defense integrated into the power system simulator.

Long Short-Term Memory (LSTM) based detector

Once we trained an attacker that can successfully beat the initial detector described in section 3.3.3, our next step is to develop a robust LSTM-based detector for identifying this attack [8]. The objective here is to identify anomalies in the signals that go to the supervisory layer, as explained in the operation of the threshold-based detector.

The training dataset for this LSTM-based detector comprises two distinct classes: anomalies and non-anomalies. Anomalous data is generated by executing a new RL attacker within the system while collecting signals such as V_{PCC} and V_{PCC}. For the non-anomalous dataset, the system is operated under normal conditions, and the same signals are collected. These datasets include the system's behavior during load changes, reflecting real-world scenarios. Once these datasets are prepared, we train an LSTM classifier for attack detection.

3.3.4 Deep Reinforcement Learning (DRL) based attacker

As mentioned earlier, the attacker is based on a Deep reinforcement learning framework. This paper uses soft actor-critic based deep reinforcement learning to train an attacker [7]. In the soft actor-critic framework, the actor π_ϕ, with parameters ϕ, selects actions using its policy and interacts with the environment, while the critic $\mathcal{Q}^\theta$, with parameters θ, evaluates these actions by estimating their expected values. The actor aims to maximize expected rewards and minimize the negative log-likelihood of the selected actions, which is called entropy regularization. This regularization promotes policies that are both effective and diverse, leading to better exploration and, ultimately, improved learning performance. The critic minimizes the mean squared error (MSE) between its value predictions and the actual returns. More details about this algorithm can be found in [7].

In our experimental setup, the actor and critic networks are implemented using LSTM networks. The inverter system operates at a high sampling rate of 20×10^{-6} seconds, producing numerous samples within a short period. In contrast, we train our attacker with a sampling rate 5 times lower than that of the inverter, reducing the computational load. Our experiment results indicate that the LSTM network is more effective in accurately capturing

the system's state by considering the historical values of the signal data, even at this lower sampling rate.

State

A state S represents the current observations of the environment at a given time. The agent or actor π_ϕ observes the current state to decide the action to take. The input to the DRL-based attacker comprises current observations of the system. In our experiment, a state S is based on a sequence of 300 past observations comprising normalized δ, V_{PCC}, and f_{PCC} for all three inverters. The length of this sequence is a hyperparameter, chosen to help the attacker achieve its goal while the state effectively captures the trend of the signals.

Action

Action $A \in \mathcal{R}^2$ represents the two-dimensional attack signals generated by an attacker based on the current system observations. These continuous attack signals from the policy network π_ϕ are injected into the control signals specified by the supervisory layer, specifically moment of inertia (J) and nominal frequency (f^*).

Reward

A reward R is a scalar feedback signal the environment provides to the agent after each action. The agent's goal is to maximize the cumulative reward over time. This reward is based on the attacker's performance to get to the desired outcome, which in our case is to diverge the δ without being detected by the detector.

The attacker aims to diverge at least one of the signals δ from the others. We measure this divergence using the exponential distance between δ values of the three inverters at time t. The nonlinearity of the exponential function assists the network in stabilizing and reaching the goal. Thus, the greater the desynchronization of δ, the higher the reward accumulated by the agent.

Similarly, a crucial objective is to remain undetected by the supervisory layer's detector.

This requires that the signals V_{PCC} and f_{PCC} not deviate significantly from their desired values within a certain time margin. This deviation is also quantified using the exponential distance between the corresponding values of the three inverters for V_{PCC} and f_{PCC}. A larger distance results in a lower reward. All of these formulations can be combined together to form a reward $\mathcal{R}$ at time t

$$\mathcal{R} = \alpha e^{d_\delta} - \beta e^{d_{fPCC}} - \gamma e^{d_{VPCC}} - \sum_{i=1}^{n} e^{det_i}, \tag{3.1}$$

where, d_δ and d_{VPCC} denote the distance among δs and V_{PCC}s of all three inverters, respectively. Likewise, d_{fPCC} indicates the distance among f_{PCC} among all three inverters and the measure of how far they are from their average values. This ensures they remain within predefined boundaries to evade detection by the supervisory layer's detector. Additionally, det_i represents the softmax output of the LSTM-based detector $Detector_i$, indicating its confidence that the current signal at a given time is malicious. α, β, and γ are the hyperparameters that define the importance we want to give to each distance.

Terminal States

A positive terminal state corresponds to an attacker successfully attacking the system without being detected. To be more precise, at any given time, if the average of the distance among all the signals in (δ) in the past 7 line cycles is beyond some threshold (5 in our case), it is a successful attack. In this situation, the episode ends with a high positive reward. Similarly, the negative terminal state occurs when an attacker is detected during an attack before reaching a positive terminal state, as detailed above. In this case, the episode ends with a high negative reward.

3.4 Results

We stick to our attack framework by iteratively developing attackers and detectors, where attackers can overcome all detectors and detectors can detect attacks from all attackers.

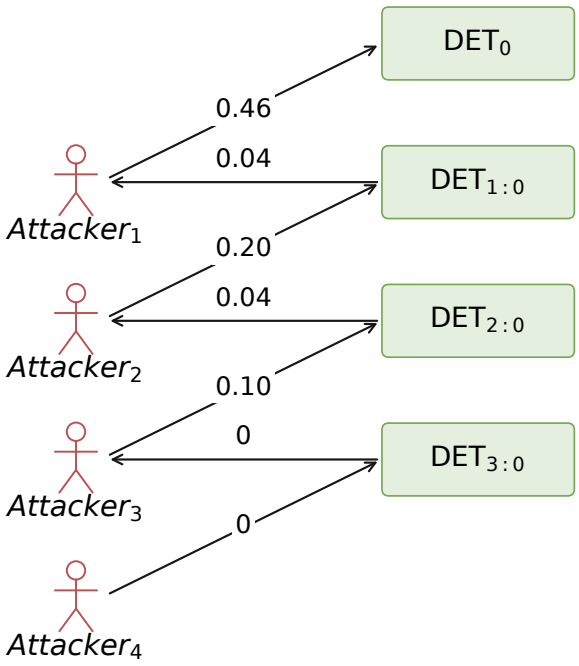


Figure 3.2: Iterative attacker–detector evaluation against accumulated detector sets, where $DET_{j:0} = \{DET_j, \ldots, DET_0\}$. Diagonal arrows report attack success probabilities before adding the next detector, and horizontal arrows report them after its addition.

3.4.1 DRL based attackers

We conducted experiments using the framework in [5], as illustrated in Figure 3.2. The figure depicts the success rate of an attacker in successfully attacking the system. The results show that the RL-based attacker can identify patterns to execute successful attacks on the system. In the figure, arrows from $Attacker_i$ to $Detector_{i-1}$ denote the successful attack rate of $Attacker_i$ against detectors $Detector_{i-1..0}$. Similarly, arrows from $Detector_i$ to $Attacker_i$ indicate the successful attack rate of $Attacker_i$ against a new detector $Detector_{i..0}$ where $Detector_i$ is constructed using data from $Attacker_i$. This low success rate suggests that the new detector $Detector_i$ effectively prevents successful attacks by the latest attacker. Subsequently, successive attackers can beat the ensemble of detectors, although the success rate decreases as the number of detectors increases. The final detector $Detector_3$ successfully prevents all attacks from attackers $Attacker_3$ and $Attacker_4$.

3.4.2 LSTM based detectors

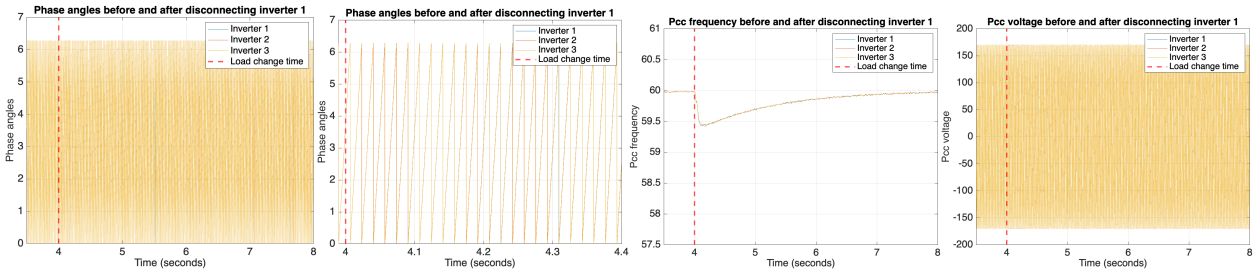
We have developed more robust LSTM-based detectors as depicted in Figure 3.2. It is important to note that the power system operates at a very high sampling rate, and excessive false alarms from the detector could be detrimental. To address this, we adjust the threshold of the network's output accordingly to minimize the false alarm rate. Table 3.2 presents the selected decision threshold for each detector and the corresponding FPR and TPR. The relatively low TPR indicates a tradeoff with a decrease in FPR. It's important to mention that it requires some time for the attack to impact the system. Hence, a certain compromise in TPR to lower the FPR is reasonable.

Table 3.2: Decision thresholds, false positive rates (FPR), and true positive rates (TPR) of the individual LSTM detectors.

Detector	Decision threshold	FPR	TPR
DET_1	0.85	0.0248	0.7129
DET_2	0.85	0.0748	0.7479
DET_3	0.95	0.0165	0.7071

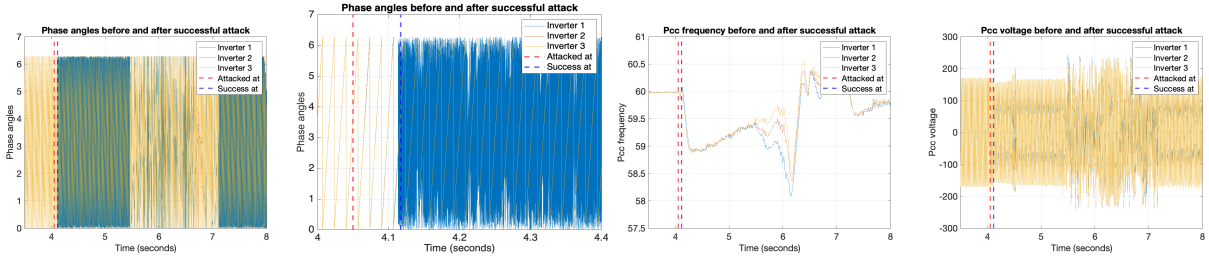
3.4.3 LSTM based detectors

We have developed more robust LSTM-based detectors as depicted in Figure 3.2. It is important to note that the power system operates at a very high sampling rate, and excessive false alarms from the detector could be detrimental. To address this, we adjust the threshold of the network's output accordingly to minimize the false alarm rate. Considering this, the false positive rates (FPR) for detectors $Detectors_{1..3}$ are 0.0248, 0.0748, and 0.0165, with the corresponding thresholds set at 0.85, 0.85 and 0.95, respectively. Similarly, the true positive rates (TPR) are 0.7129, 0.7479, and 0.7071, respectively. The relatively low TPR indicates a tradeoff with a decrease in FPR. It's important to mention that it requires some time for the attack to impact the system. Hence, a certain compromise in TPR to lower the FPR is reasonable.



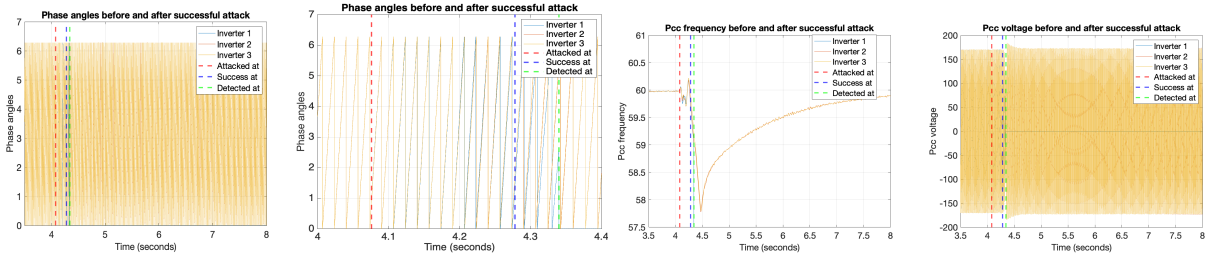
((a)) Transition of phase angles ((b)) Zoomed phase angles ((c)) Transition of PCC frequency ((d)) Transition of PCC voltage

Figure 3.3: Normal operation of microgrid with load change at time $t = 4s$



((a)) Transition of phase angles ((b)) Zoomed phase angles ((c)) Transition of PCC frequency ((d)) Transition of PCC voltage

Figure 3.4: Successful attack by $Attacker_3$; inverter 1 is still connected to the system. Attack at: $4.05s$, Success at: $4.117s$

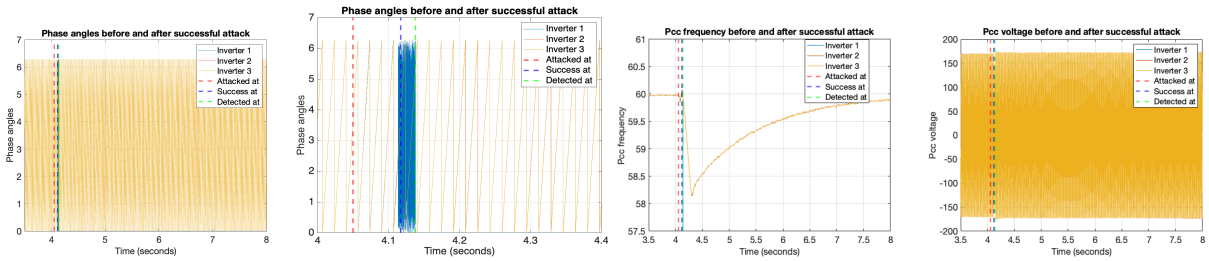


((a)) Transition of phase angles ((b)) Zoomed phase angles ((c)) Transition of PCC frequency ((d)) Transition of PCC voltage

Figure 3.5: Successful attack by $Attacker_1$; inverter 1 is disconnected post detection. Attack at: $4.0759s$, Success at: $4.278s$, Detected at: $4.340s$

3.4.4 Impact of attackers and detectors in the system

Figure 3.3 depicts the microgrid simulation without an attack but with a load change at 4 seconds. Similarly, Figure 3.4 illustrates a successful attack and its impact on the system. Additionally, Figures in 3.5 and 3.6 show successful attacks by $Attacker_1$ and $Attacker_3$, respectively, resulting in the disconnection of inverter 1 after detection and their impact on



((a)) Transition of phase angles ((b)) Zoomed phase angles ((c)) Transition of PCC frequency ((d)) Transition of PCC voltage

Figure 3.6: Successful attack by $Attacker_3$; inverter 1 is disconnected post detection. Attack at: 4.05s, Success at: 4.117s, Detected at: 4.138s

the overall operation of the system. We will now examine each of these cases individually.

Case 1 Without an attack, the microgrid behaves as depicted in 3.3. The load change at time 4 generates transients in f_{PCC} but eventually stabilizes to a steady state.

Case 2 Figure 3.4 illustrates a successful attack by $Attacker_3$ and its impact on the system's performance. Following the successful attack at $t = 4.117s$, both the (f_{PCC}) and (V_{PCC}) of all the inverters deviate from their reference points, losing synchronization. This disturbance disrupts the normal operation of the entire system if left unaddressed.

Case 3 Here, we discuss the implications of $Detector_2$ and $Detector_0$ as shown in figures 3.5 and 3.6, where attacks are executed by $Attacker_1$ and $Attacker_3$, respectively. In both cases, inverter 1 is compromised initially, followed by disconnection of this inverter post-detection. Subsequently, a transient is observed in the frequency (f_{PCC}) of the two inverters in both figures. However, it's notable that with $Attacker_3$, the frequency Nadir is improved by 0.36 Hz at its lowest point, contributing to enhanced system resilience. This shows the superiority of $Detector_2$ over $Detector_0$. Furthermore, the detection time for $Attacker_1$ (3.5) is longer compared to $Attacker_3$ (3.6), indicating varying detector effectiveness. Through iterative refinement, the detectors gradually become efficient at identifying diverse attacks within shorter timeframes. This is evident with the final detector, $Detector_3$, which effectively prevents successful attacks as shown in 3.2.

Case 4 Here, we compare the implications of $Attacker_3$ and $Attacker_1$ on the system with varying detectors. In the case of figure 3.6, $Attacker_3$ swiftly diverges the phase angle (δ) of inverter 1, although with a low success rate as shown in figure 3.2; however, they are promptly caught by the detector with minimal impact on the other two inverters compared to $Attacker_1$. Since the goal was to diverge the δ of inverter 1 as quickly as possible, $Attacker_3$ has learned to achieve this goal by beating all the existing detectors.

Furthermore, the divergence rate of δ of inverter 1 by $Attacker_3$ in figure 3.6(b) has the potential to introduce significant noise to the entire system, which might affect the long-term reliability of the system. In contrast, with $Attacker_1$, the divergence of δ of inverter 1 is slow and prolonged 3.5(b). This type of divergence may have a lesser impact on the long-term reliability of the system. Investigating the precise extent of inverter damage across the system remains a subject for future research in this experiment. Additionally, the sophistication of $Attacker_3$ compared to $Attacker_1$ is evident in the behavior of f_{PCC} from the initiation of the attack until it reaches the success state. In Figure 3.5(c), during an attack by $Attacker_1$, there is a notable fluctuation in f_{PCC} during this period. However, in the case of $Attacker_3$ in Figure 3.6(c), this fluctuation is minimal, and the attacker has learned to mimic the attack process to resemble a system load change to prevent the detection.

Our findings reveal that iteratively constructing detectors within this framework increases the detection rate without significantly affecting the system's normal operation. Nonetheless, it's evident from our $Attacker_3$ that attackers can evolve to a level that might impact the system's long-term reliability, albeit with a relatively low occurrence rate.

3.5 Conclusion

In this paper, we developed DRL-based attackers and robust detectors using an iterative training process outlined in Section 3.1. We demonstrated that with each iteration, detectors improved their detection capability. Additionally, we investigated the influence of these DRL-based attackers and detectors on the microgrid. Through our evaluation, we assessed the impact of these attackers on the system and highlighted how the detectors trained using this

process offer superior protection compared to initial detectors, thereby enhancing system security and minimizing significant disruptions in power systems.

3.6 Discussion and Practical Implications

3.6.1 The Gap between Local Behavior and Supervisory Evidence

The microgrid study extends the earlier tank experiment to a system in which one device's behavior influences several connected devices. The attacker modifies selected control-layer signals associated with an inverter, while the supervisory detectors assess voltage and frequency measurements. Internal phase-angle behavior is therefore part of the physical objective without being a direct input to the supervisory detector.

This distinction is central to the interpretation of stealth. The attack does not need to erase every consequence of its actions from the electrical system. It needs to reach the specified physical objective before the monitored response causes the detector to intervene. A phase-angle disturbance can become important before the corresponding voltage and frequency measurements satisfy the defense's alarm rule.

The observation available to the attacker is also richer than the observation available to the detector in this setup. That asymmetry makes the experiment a test of the selected supervisory architecture against a capable adversary. It does not establish the same attack performance for an adversary that lacks the specified inverter observations or control access. Stating both information sets is necessary to understand which part of the problem the learned policy solves.

3.6.2 Why an Eventual Alarm Is Not Sufficient

The trajectory examples make the difference between detecting an attack and preventing its objective concrete. In Figure 3.5, the reported success time occurs before the reported detection time. The same ordering occurs in Figure 3.6. These examples, therefore, remain successful attacks under the before-alarm criterion, even though the inverter is subsequently

disconnected.

Post-detection disconnection can still be useful. It can limit how long the compromised inverter continues influencing the other inverters and can change the later system response. However, that benefit answers a different question from whether the attack reached its defined objective before detection. A clearer description is that mitigation may reduce subsequent disruption even when it does not prevent the initial successful attack.

The physical thresholds in this chapter describe loss of acceptable modeled operation. They do not directly quantify inverter lifetime or irreversible damage. A phase-angle disturbance, a frequency excursion, and a hardware failure are related concerns, but they are not interchangeable measurements. The simulations demonstrate the reported electrical behavior without establishing a component-damage model.

3.6.3 What Changes across Adversarial-Training Rounds

Adding a detector changes the problem faced by the next attacker. A policy that previously succeeded through a gradual disturbance may no longer avoid the expanded defense. The next learned policy may use a different timing pattern or a different balance between phase disturbance and supervisory visibility. For this reason, adaptation should be examined through both attack success and representative trajectories.

Attack-generated data give the defender examples that would not necessarily arise from normal operation or from a manually selected disturbance. Each new detector can learn features associated with a newly discovered weakness. Retaining previous detectors preserves earlier decision rules as well. The resulting defense is therefore more than a single classifier repeatedly fitted to a changing dataset.

Nevertheless, detector diversity is useful only through the decisions made from the collection. The first microgrid study evaluates the accumulated defense against physical disruption. The following chapter takes the next step by examining the aggregation rule explicitly, asking whether correct evidence from one detector survives when the system produces a final localization decision.

3.6.4 Detection, Response, and the Next Research Question

The experiment links supervisory detection with a physical response through inverter disconnection. In a deployed system, the usefulness of an alarm would also depend on the delay associated with processing measurements, issuing a response, and completing the protective action. The simulation's decision timing does not, by itself, establish that an implementation meets the same timing on a particular controller or communication network.

There is also a separate question about which inverter should be disconnected. Recognizing that the microgrid is behaving abnormally does not necessarily identify the device responsible. Because electrical coupling spreads the response, an uncompromised inverter may display a substantial disturbance caused elsewhere. A response based on the wrong location can remove a healthy device while leaving the compromised one active.

This observation motivates Chapter 4. The next study does not simply repeat the same disruption experiment with another detector. It changes the adversarial objective to wrong-inverter attribution and examines how transient behavior and detector aggregation influence that decision. The progression moves from timely recognition of a harmful event to the correctness of the supervisory action that the event is expected to trigger.

Chapter 4

Mislocalization Attacks on Grid-Forming Microgrids Under Transient Conditions: An Iterative Reinforcement Learning Framework

4.1 Introduction

Modern cyber-physical power systems increasingly couple inverter controls, communication networks, supervisory measurements, and software-defined operating logic. This integration improves flexibility, but it also creates attack surfaces in which malicious commands, compromised firmware, manipulated measurements, or unsafe supervisory inputs can influence physical behavior [49, 50]. Grid-forming (GFM) microgrids are particularly relevant because converter controls act on fast electrical dynamics and because supervisory decisions are often made from a restricted set of measurements rather than from complete access to every inverter state.

Existing cyberattack studies consider a range of operating conditions, including load changes and other dynamic scenarios. However, comparatively less attention has been given

to attacks explicitly synchronized with a benign transient for the purpose of manipulating source localization. In practice, however, microgrids repeatedly undergo non-steady-state operation caused by load steps, switching actions, faults, protection events, islanding/reconnection, and externally driven disturbances. These events naturally produce short-lived deviations in voltage, frequency, current, and power. Transient-aware detection has therefore been studied explicitly in microgrids and distribution systems [51, 52]. From an attacker's perspective, the same operating interval is attractive because a malicious deviation can be superimposed on a physical deviation that the operator already expects to observe. The resulting problem is not simply whether an anomaly is present, but whether the supervisory layer can correctly attribute the observed transient to its physical source.

This distinction motivates the threat studied in this work. We consider an informed attacker that compromises one inverter and times its learned control manipulation to coincide with a load-induced transient. The objective is not necessarily to produce the largest immediate physical deviation. Instead, the attacker attempts to make the supervisory detector attribute the disturbance to a different inverter. Such *mislocalization* is operationally important because localization directly informs mitigation and isolation decisions; prior work on cyber-physical anomaly localization similarly emphasizes that identifying the correct source is necessary for effective response [52]. If the wrong inverter is isolated while the compromised inverter remains active, a correct detection alarm can still lead to an incorrect mitigation action.

Reinforcement learning (RL) and adversarial learning have already been used to generate sophisticated cyberattacks and to develop adaptive defenses in power systems. Examples include RL-based false-data injection against voltage control [26], multi-agent RL vulnerability discovery and remediation in DC microgrids [33] , adversarial deep-RL attack/defense design for inverter-based microgrids [53], RL-based attack generation for load-frequency control [54], and undetectable RL attacks against automatic generation control [55]. These studies establish that RL/DRL-assisted attacks and DRL-assisted defenses are not new by themselves.

The iterative attacker–detector learning paradigm is also not introduced for the first

time in this paper. The earlier work [5] developed successive RL attackers and detectors iteratively until neither side exhibited further practical improvement, and the subsequent power-system study [6] applied iterative RL-based adversarial training to a GFM microgrid. More recently, Wan *et al.* [56] proposed an iterative MARL-based attack-surface assessment and data-driven detector refinement framework for DC microgrids. Accordingly, the novelty of the present paper is not the existence of an iterative attacker–detector loop. Rather, this work uses that previously established paradigm to investigate a different security question: *whether adaptive attacks launched during transient operation can cause persistent inverter mislocalization, and how detector aggregation changes the outcome.*

A second important modeling point is the separation between *training* and *attack execution*. The expensive attacker–detector adaptation loop is performed offline in the simulation/digital-twin environment. During this stage, the attacker is deliberately granted strong information access so that the framework acts as a worst-case security stress test. Once an attacker policy is trained, its parameters are frozen for evaluation. The online stage consists only of policy inference and attack-signal application at the compromised controller; the detector ensemble is also frozen during each evaluation run. Thus, the framework does not assume that operators retrain detectors during a sub-second physical transient, nor that a remote attacker repeatedly trains a new DRL model while the transient is occurring. This offline-training/online-execution distinction is central to the practical interpretation of the proposed framework and is detailed in Section 4.3.5.

The use of a high-fidelity simulation or digital twin should likewise be understood as an *informed-attacker* assumption rather than as a claim that every real adversary possesses an exact replica of the target microgrid. Modern inverter ecosystems contain software, firmware, and external interfaces that introduce cybersecurity concerns and potential high-privilege attack surfaces [49, 50]. We therefore use an intentionally strong attacker model to expose vulnerabilities that would remain hidden under a weaker black-box attacker. At deployment, the digital twin itself is not part of the attacked physical system; it is the environment used to learn the policy beforehand.

The supervisory layer in our study is partially observable and uses PCC voltage mag-

nitude and frequency for classification. The attacker is more informed than the supervisor during offline learning. This asymmetry is intentional: the goal is to evaluate whether a limited-observability supervisory architecture remains robust against a high-capability adversary. Prior work has shown that partial observability is directly relevant to stealthy and zero-dynamics attacks [1, 5, 16]. In the present paper, however, we use the more conservative term *persistent stealthy mislocalization* unless the required invariant-zero/state-divergence conditions are explicitly verified; a non-zero attack success rate alone is not treated as proof of a formal zero-dynamics attack.

Finally, we investigate two ways of combining successive detectors. Under *strict aggregation*, a correct localization by any detector is sufficient to defeat the attack. Under *consensus aggregation*, confidence vectors are averaged and a single aggregate class is selected. This comparison isolates an architectural question not addressed by the earlier iterative-training papers [5, 6]: whether a detector ensemble that is individually improved over time can remain exploitable because of the rule used to combine its members.

The main contributions of this paper are as follows:

- Building on the iterative attacker–detector framework introduced in prior work [5, 6], we formulate a transient-aware adversarial evaluation for a GFM microgrid in which DRL attacks are synchronized with load-induced non-steady-state behavior.

- We formulate inverter *mislocalization* as the adversarial objective, emphasizing the operational consequence of inducing the supervisory layer to identify and potentially isolate the wrong inverter.

- We compare strict and consensus detector aggregation and show that aggregation logic, rather than detector count alone, strongly affects the ability of an adaptive attacker to maintain mislocalization.

- Through extensive simulation studies, we demonstrate that detectors trained iteratively on attack-generated datasets achieve improved robustness and significantly reduce the success rate of increasingly sophisticated

The three-inverter test system is used as a controlled proof-of-concept for isolating these mechanisms. The mathematical workflow is parameterized by the number of detectors and classes and is not intrinsically restricted to three inverters; however, we do not claim that the present experiments constitute a scalability validation for arbitrary-size microgrids. Larger-system and hardware validation are therefore treated as future work rather than as evidence already established by this study.

4.2 Related Work and Positioning

Prior literature relevant to this study can be grouped into four overlapping directions: (i) learning-based attack generation, (ii) supervised/unsupervised/RL-based detection, (iii) adversarial attack–defense design, and (iv) iterative attacker–detector refinement. Table 4.1 summarizes representative work and highlights the specific gap addressed here.

4.2.1 Learning-Based Attacks and Defenses in Power Systems

RL has been used both offensively and defensively across power-system cybersecurity. Chen *et al.* [26] formulated an RL-based FDI attack against automatic voltage control, demonstrating that attack strategies can be learned under incomplete information. Abianeh *et al.* [33] used multi-agent DRL to discover vulnerable points in a DC-microgrid detector and generate coordinated stealthy FDI attacks; a DQN-based detector was then incorporated to improve identification. Wang and Pal [53] formulated both a destabilizing inverter attack and a robust defense using adversarial DRL, explicitly targeting droop-control gains and validating on both a four-IBR microgrid and an IEEE 123-bus configuration. Mohamed and Kundur [54] used DRL to synthesize attacks against load-frequency control and then used generated data to study supervised and unsupervised defenses. Shereen *et al.* [55] formulated undetectable AGC attacks as a POMDP and showed that both supervised and unsupervised ML detectors can outperform traditional detectors against the learned attacks.

These studies closely align with our study's premise, validating the use of RL to craft

Table 4.1: Comparison with representative learning-based cyberattack and defense frameworks. "Transient-targeted" means that the attacker is explicitly designed to exploit the transient interval, not merely that load changes are included in validation.

Work	**System**	**Learning role**	**Primary security objective**	**Iterative attacker–detector**	**Transient-targeted**	**Localization / aggregation focus**
Chen *et al.* [26]	Power-grid AVC	Q-learning attacker	Learn stealthy FDI strategy with limited information	No	No	No
Abianeh *et al.* [33]	Islanded DC microgrid	Multi-agent DDPG attacker + DQN detector	Discover detector vulnerabilities and remediate FDI attacks	No explicit co-evolution loop	No	Attack-point identification; no ensemble aggregation
Wang and Pal [53]	Inverter-based microgrid; IEEE 123-bus case	TD3 attacker + adversarial DRL defense	Least-effort destabilizing attack on droop-control gains and robust defense	No successive detector expansion	No	No
Paudel and Amariucai [5]	Quadratic-tank CPS	DRL attacker + successive learned detectors	Model-free zero-dynamics attack generation and iterative defense improvement	**Yes**	No	No
Paudel *et al.* [6]	GFM microgrid	DRL attacker + successive learned detectors	Iterative adversarial training for robust power-system detection	**Yes**	No	No
Mohamed and Kundur [54]	Load-frequency control	DRL attack generation + supervised/unsupervised detector study	Vulnerability discovery and data-driven defense design	No	No	No
Shereen *et al.* [55]	Three-area AGC	PPO attacker; supervised and unsupervised detector evaluation	Learn high-impact attacks that evade ACE/UIO-style detection	No	No	No
Choi *et al.* [57]	Islanded AC GFM microgrid	GP regression + one-class SVM	Unsupervised FDI detection and point-of-attack isolation	No	No	Localization/isolation; no adversarial co-training
Wan *et al.* [56]	DC microgrid	MARL attacker + iteratively retrained ANN detector	Iterative attack-surface discovery and detector coverage expansion	**Yes**	No; load steps used for robustness validation	Node attack detection; no mislocalization/aggregation study
This work	**GFM microgrid**	**SAC attacker + successive detectors**	**Transient-synchronized stealthy mislocalization**	**Yes; building on [5, 6]**	**Yes**	**Inverter mislocalization; strict vs. consensus aggregation**

stealthy attacks in power systems, where accurate models may be hard to obtain, but RL can still find effective attack strategies

4.2.2 Iterative Attacker–Detector Learning

The authors of the paper [5] introduced a successive attacker–detector process in which a DRL attacker is trained against the current detectors, attack data are collected, and a new detector is trained to recognize the resulting behavior; the process is repeated until practical improvement stabilizes. We subsequently applied iterative RL-based adversarial training to a GFM power-system setting in [6]. Hence, the attacker–detector iteration used in the present manuscript is a continuation of the earlier framework rather than a new mechanism introduced here.

Wan *et al.* [56] more recently present a closely related iterative adversarial-learning workflow for DC microgrids: MARL attackers generate attacks against the current detection scheme, operational data augment the detector training set, and the updated detector is fed back to the attacker in the next offline iteration. Their objective is proactive attack-surface assessment and expansion of detectable attack coverage. Their study is an important recent comparison because it confirms the relevance of iterative adversarial learning in converter-dominated systems. The distinction of the present work lies in the attack objective and operating regime: we study transient-synchronized *mislocalization* in a GFM microgrid and evaluate how strict versus consensus detector aggregation changes the attacker's ability to remain successful.

4.2.3 Detector Paradigms: Supervised, Unsupervised, and RL

The microgrid literature contains several detector paradigms, and their capabilities differ. Supervised detectors can achieve high classification accuracy when representative labeled attack data are available; for example, data-driven supervised estimation and residual analysis have been used for DC-microgrid FDI detection and mitigation [58]. Their main limitation is dependence on training coverage. Unsupervised methods avoid the need for labeled attack traces; Choi *et al.* [57] combine Gaussian-process prediction with a one-class SVM and validate the method in both simulation and hardware-in-the-loop experiments. RL-based detectors can instead learn detection or remediation policies through interaction, as illustrated

by the DQN-based complementary detector in [33]. These paradigms are complementary rather than directly interchangeable.

The detectors used in our experiments are supervised classifiers because the central question is not which detector family is universally superior; it is whether an adaptive attacker can exploit a progressively expanded detector ensemble and, in particular, its aggregation rule. Instead, we discuss these alternatives and make the detector-architecture scope explicit, rather than implying that our selected detector represents all modern microgrid defenses.

4.2.4 Why Transient Operation Matters

Transient behavior is not simply noise around a steady-state operating point. During load switching, protection actions, faults, and recovery, physically legitimate signals can change rapidly and can resemble the deviations that a cyberattack is designed to produce. Prior work has therefore explicitly developed transient-model-based FDI detection for microgrids [51]. Likewise, FALCON [52] uses transient short-circuit voltage/current information to jointly distinguish and localize cyber and physical anomalies in distribution systems. These studies demonstrate that the transient regime is a distinct inference problem in which the detector must separate malicious behavior from legitimate physical disturbances.

Our threat model intentionally exploits this ambiguity. The attacker does not create the benign transient; a load event does. The attacker synchronizes its action with that event so that the malicious contribution is superimposed on a disturbance the supervisory layer already expects. In a severe-weather, switching, fault-recovery, or other disturbance scenario, an operator may therefore observe a plausible transient while the true source of the abnormal behavior is partly cyber-induced. The relevant security question is then whether the detector correctly distinguishes and localizes the malicious contribution rather than simply whether voltage or frequency departed from nominal values.

It is important to state the scope of the claim carefully. The present experiments establish attacker behavior during load- induced transients. They do not include a matched steady-state baseline, and therefore, the contribution is instead to study a practically important

transient attack surface that prior steady-state evaluations do not capture.

4.2.5 Localization as an Adversarial Objective

Attack detection alone is insufficient when mitigation depends on knowing which component is compromised. Ganjkhani *et al.* [52] motivate cyber–physical anomaly localization precisely because effective response requires both anomaly type and location. In our setting, the attacker exploits this decision layer: it seeks to manipulate the compromised inverter so that the supervisory classifier assigns the disturbance to another inverter. This objective differs from the more common goal of maximizing frequency/voltage deviation or directly destabilizing the system [53]. Mislocalization can be harmful even when the system remains temporarily stable because an operator may isolate the wrong asset while leaving the compromised device active.

4.2.6 Partial Observability and Stealth

Stealthy attack theory shows that limited observability can permit malicious internal behavior to remain weakly reflected in monitored outputs [2, 16]. Imperfect model knowledge does not necessarily eliminate the ability to construct approximately stealthy attacks [1], and previous work showed that DRL can learn stealthy/zero-dynamics behavior without an explicit state-space model [5]. The present work uses the same security intuition but avoids equating persistent classifier error with a formal invariant-zero attack. We therefore describe the observed outcome as persistent stealthy mislocalization unless additional state-space evidence is supplied.

Relative strengths and limitations Relative to the cited frameworks, the present study contributes the combination of (i) transient-synchronized attack execution, (ii) inverter *mislocalization* rather than only attack detection or destabilization, and (iii) explicit adversarial comparison of strict and consensus aggregation. At the same time, the current evaluation is intentionally narrower in several respects: the detector family is supervised, the

attacker is given strong model/inverter-side information during offline training, the test system contains three inverters, and no matched steady-state baseline or cross-family supervised/unsupervised/RL detector benchmark is reported. Table 4.1 should therefore be read as a positioning comparison, not as a claim that the present framework dominates all prior detector architectures.

4.3 Experiment Setup

4.3.1 Microgrid Model

Our experiments are conducted on a simulated grid-forming microgrid comprising three grid-forming inverters (GFMIs). The microgrid operates in islanding mode (no connection to the main grid) and is subject to load changes that induce transient behavior in voltage and frequency. Each inverter is responsible for regulating the local voltage and frequency at the PCC through its control loops. Fig. 4.1 illustrates the single-line diagram of the simulated microgrid. The system is three-phase; however, for simplicity of illustration, it is represented using a single-line diagram. The direct controller employs two independent pathways to produce phase-angle and voltage amplitude references, which are then fed into the PWM generator. θ^{ref} determines the phase angle and subsequently the system frequency by applying the active power–frequency droop coefficient m_p. Within the other path, a PI controller determines the inverter's reference voltage v^{ref}, ensuring that the output voltage, v^{pcc}, tracks the nominal voltage V_n by applying the reactive power–voltage droop coefficient m_q. These two controller structures provide both active and reactive power sharing among the inverters operating in GFM, known as droop control [59]. The network parameters are clearly presented in Table 4.2.

We model the electrical network and inverter dynamics, including line impedances and inverter output limits. Transient events are introduced by step changes in load power consumption at random time intervals. Such load disturbances represent routine non-steady-state operation (e.g., load connection/disconnection or scheduled load transfer) and naturally

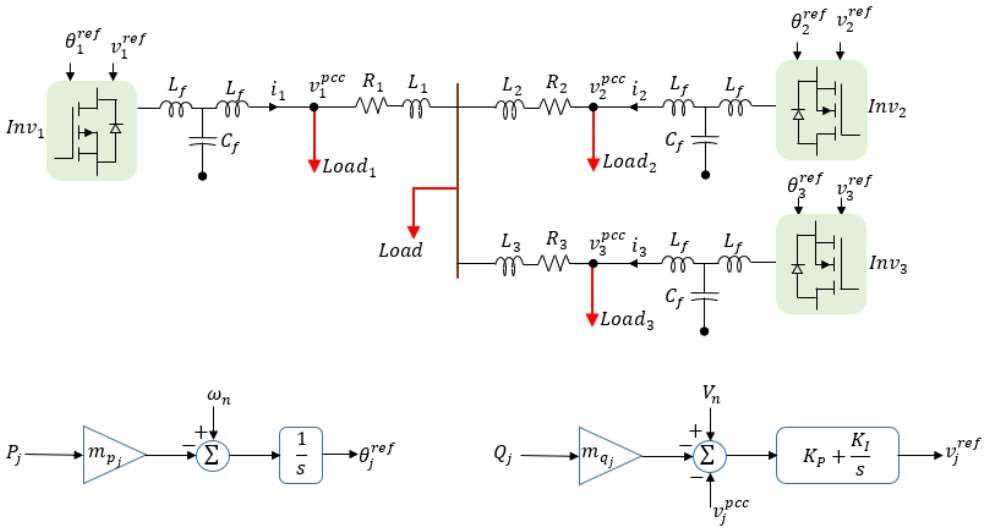


Figure 4.1: Single-line diagram of the simulated grid-forming microgrid operating under droop control.

Table 4.2: Microgrid parameters

Parameter	Value	Parameter	Value
V_n	$208\sqrt{2}$ V	L_2	10 mH
f_n	60 Hz	R_3	2 Ω
L_f	1 mH	L_3	15 mH
C_f (Δ)	5 μF	m_p	1×10^{-3}
R_1	0.1 Ω	m_q	1×10^{-3}
L_1	5 mH	K_P	0.5
R_2	1.5 Ω	K_I	10

cause momentary frequency and voltage excursions. In a real microgrid, analogous transients can also be initiated by switching, faults, protection actions, restoration, or externally driven disturbances. The important security feature is that these intervals already contain legitimate physical deviations. An attacker that synchronizes a manipulation with such an event can therefore superimpose a malicious contribution on a response that the supervisory layer expects to be non-stationary. This makes transient periods easier for attackers to remain unnoticed at the supervisory level.

4.3.2 Supervisory Layer and Partial Observability

The supervisory layer monitors the microgrid using a limited subset of measurements available at the PCC, specifically PCC voltage magnitude and PCC frequency. These measurements are consistent with IEEE 1547 guidelines and reflect realistic monitoring capabilities in practical systems [60]. Detectors deployed at this layer do not have access to internal inverter states or control signals, resulting in a partially observable environment, and based on this information alone, must infer whether the system is operating normally or if an event (fault or attack) has occurred. It also follows the assumptions of prior studies that highlighted how limited measurements can be exploited by stealthy attacks [5, 6, 16]. In summary, the standard assumption here is that supervisory detectors operate with the same kind of information that an actual microgrid controller would have: essentially, bus voltage and frequency, updated in real-time.

Why localization from PCC measurements is meaningful The framework does not assume that inverter identity is analytically observable from two instantaneous scalars. Instead, the classifier receives a *time window* of PCC voltage and frequency and learns transient signatures created by the different electrical paths, line impedances, and inverter/control interactions in the modeled microgrid. The baseline localization task is therefore an empirical pattern-recognition problem under partial observability. If two inverter-origin events produce indistinguishable PCC trajectories, no classifier using only these measurements can be expected to localize them reliably; that ambiguity is a limitation of the sensing architecture rather than information available to the detector. The adversarial objective studied here is precisely to exploit the remaining ambiguity and move an attack trajectory toward the signature of the wrong inverter.

4.3.3 Detector Design

We deploy multiple neural network-based detectors at the supervisory layer. Each detector is a classifier that processes the stream of PCC measurements (voltage and frequency) and

is trained to classify system behavior into four classes: normal operation and faults or disturbances attributed to each of the three inverters. In other words, the detectors are trained to both detect anomalies and attribute (localize) them to a specific inverter, under the assumption that at most one inverter is compromised or faulty at any given time. The detector takes as input a window of recent measurements to account for the transient shape of events.

Training data for detectors Initially, the detectors are trained on a dataset that includes examples of normal operation (including benign load-induced transients with no attacks) and examples of fault events (e.g., line faults or inverter trips) that are labeled according to which inverter's region they originated in. Thus, a benign transient is not automatically labeled as a cyberattack: the initial detector is explicitly trained to recognize normal transient behavior, while fault classes provide localized abnormal-event signatures. We intentionally do not include any attack data in the first round of detector training; this mimics a realistic scenario where detectors are developed based on known fault signatures and normal conditions, but have not yet seen intelligent adversarial behavior. Each detector is trained (via supervised learning) to minimize errors on this training set. The result after initial training is a detector that performs well in differentiating normal vs. fault scenarios in transient conditions. However, as prior work suggests, such detectors may not generalize to novel attack patterns [5].

As our framework evolves iteratively, we update the training data for new detectors using data generated by the current attacker. After each attacker's training, we simulate that attacker acting on the microgrid to produce a time series of measurements along with the ground truth of which inverter was attacked. These traces, combined with previously collected data, form the training set for the next detector. In this way, each successive detector is exposed to more "sophisticated" attack patterns, enabling it to learn subtle signatures that earlier detectors missed. This approach mirrors an incremental learning process and has parallels in ensemble learning and adversarial training in ML, where a model's weaknesses are addressed by training on adversarial examples [5, 6].

4.3.4 Multi-Detector Output Aggregation

Assume a set of M detectors $\mathcal{D} = \{D_1, \ldots, D_M\}$. Each detector D_m produces a confidence (probability) vector over K classes (in our case, $K = 4$ corresponding to the normal and three compromised inverters):

$$\mathbf{p}^{(m)}(x) = \left[p_1^{(m)}(x), \ldots, p_K^{(m)}(x)\right], \quad \sum_{k=1}^{K} p_k^{(m)}(x) = 1, \tag{4.1}$$

where x denotes the observed measurement vector (e.g., PCC-level signals).

The class predicted by detector D_m is

$$\hat{k}^{(m)}(x) = \arg \max_{j \in \{1,\ldots,K\}} p_j^{(m)}(x).$$

Equivalently, its one-hot prediction can be written as

$$\hat{y}^{(m)}(x, k) = \begin{cases} 1, & \text{if } \hat{k}^{(m)}(x) = k, \\ 0, & \text{otherwise.} \end{cases}$$

We consider two aggregation schemes. We refer to logical OR-based aggregation as *strict aggregation*, since detection is triggered if any detector identifies the true class. In contrast, we refer to weighted confidence fusion as *consensus aggregation*, since the final decision depends on agreement across detectors.

Strict Aggregation (Logical OR) In the strict aggregation setting, class k is flagged if at least one detector predicts class k. Thus, the aggregated multi-label output is

$$\hat{y}_{\text{strict}}(x, k) = \bigvee_{m=1}^{M} \hat{y}^{(m)}(x, k), \qquad k \in \{1, \ldots, K\}. \tag{4.2}$$

The corresponding detected class set is

$$\mathcal{S}_{\text{strict}}(x) = \{k \in \{1, \ldots, K\} : \hat{y}_{\text{strict}}(x, k) = 1\}.$$

Hence, strict aggregation may flag multiple attacked inverters simultaneously if different detectors predict different compromised classes.

Consensus Aggregation (Soft Voting) In the consensus aggregation setting, detector confidence vectors are combined by weighted averaging:

$$p_k^{\text{agg}}(x) = \sum_{m=1}^{M} w_m \, p_k^{(m)}(x),$$

where $w_m \geq 0$ and $\sum_{m=1}^{M} w_m = 1$. In our experiments, we use equal weights $w_m = \frac{1}{M}$.

The final decision is then

$$\hat{k}_{\text{consensus}}(x) = \arg \max_{k \in \{1,\ldots,K\}} p_k^{\text{agg}}(x).$$

This corresponds to a soft-voting ensemble classifier and always produces a single final class prediction.

Strict aggregation is more conservative, since it flags a class whenever any detector predicts it, which can reduce false negatives but may increase false positives. In contrast, consensus aggregation promotes consensus through score averaging, which may reduce false alarms but can miss attacks that are only detected by a minority of detectors.

4.3.5 Threat Model and Offline-Training/Online-Inference Separation

Compromise assumption and attack surface We consider a high-capability adversary that obtains control authority over one inverter while the other inverters remain uncompromised. The abstract attack surface is a compromise of the inverter-side control path rather than a compromise of the supervisory detector itself. In practice, such authority could arise from compromised embedded software or firmware, a malicious or vulnerable controller component, an exposed vendor/maintenance interface, manipulated supervisory commands, or

another supply-chain or communication compromise that permits modification of the inverter command path [49, 60]. The study does not assert that one particular exploit has occurred in the test system; instead, it asks what a supervisory localization architecture can withstand *after* one inverter has suffered a high-privilege compromise. This assumption is especially relevant to larger heterogeneous grids, where equipment, firmware, and service interfaces may originate from multiple vendors and therefore cannot all be treated as a single trusted implementation boundary.

Digital-twin scope The physical system considered here is a three-inverter microgrid subject to load-induced transients. The attacker is assumed to have a digital twin for learning policies that manipulate one inverter and cause the supervisory detectors to attribute the disturbance to another inverter. The defender maintains its own twin to develop attackers, generate training trajectories, and improve its localization detectors. Each twin represents the electrical network, loads, inverter dynamics, local control loops, transient events, and relevant measurement and command interfaces. Conceptually, both parties maintain independent twins and train their own attackers and detectors using the iterative procedure described in this chapter. During offline attacker training, the specified inverter observations and detector classification outputs are available within the corresponding training environment for policy learning, reward calculation, and episode termination. These capabilities constitute the informed-attacker assumption. Access to the supervisory controller's internal implementation is not required, although its relevant control responses must be represented. The defender's trained detectors operate at the supervisory layer using PCC voltage and frequency histories to detect disturbances and identify their source. Because both roles use the same system model and training procedure under the assumptions of this study, the reported experiments represent them using a single simulator. This shared simulation framework does not imply that the two parties share their trained models.

Offline training phase The computationally expensive attacker–detector loop is performed offline. Starting from the initial detector, an SAC policy is trained against the

current detector ensemble in the simulated microgrid. The trained policy is then exercised under randomized transient conditions to generate attack trajectories, and these trajectories are used to train the next detector. Repeating this process produces a sequence of attackers and a progressively expanded detector ensemble. This phase may require many simulated episodes and gradient updates.

Online inference/evaluation phase After a policy has been trained, its parameters are frozen. A test episode then represents deployment of that fixed policy at the compromised inverter while the target microgrid undergoes a transient. Online operation consists only of evaluating the learned policy and applying its scalar command modification; no SAC training occurs during the transient. Likewise, the detector ensemble used for a given evaluation is frozen. A new detector is trained only *between* adversarial-training rounds, not dynamically while the real system is experiencing the sub-second event. Thus, the high agent update rate reported in Section 4.3.8 is an inference/control sampling rate, not a model-retraining rate. Fig. 4.2 summarizes this separation.

4.3.6 RL-Based Attacker Using Soft Actor Critic

The attacker's goal is to manipulate the compromised inverter so that the supervisory layer attributes the resulting disturbance to one of the other two inverters. We implement the attacker as an RL agent trained using Soft Actor–Critic (SAC). During offline training, the attacker interacts with the simulated microgrid and learns a continuous control policy under randomized transient conditions. During evaluation, the learned policy is frozen and executed at the compromised inverter. The reward function combines the mislocalization objective with penalties that discourage conspicuous PCC behavior, and episode termination is based on sustained success or failure conditions.

At discrete time step t, the attacker observes a feature vector $o_t \in \mathbb{R}^d$ constructed from recent time-series measurements of voltages, frequencies, and auxiliary inverter signals. Based

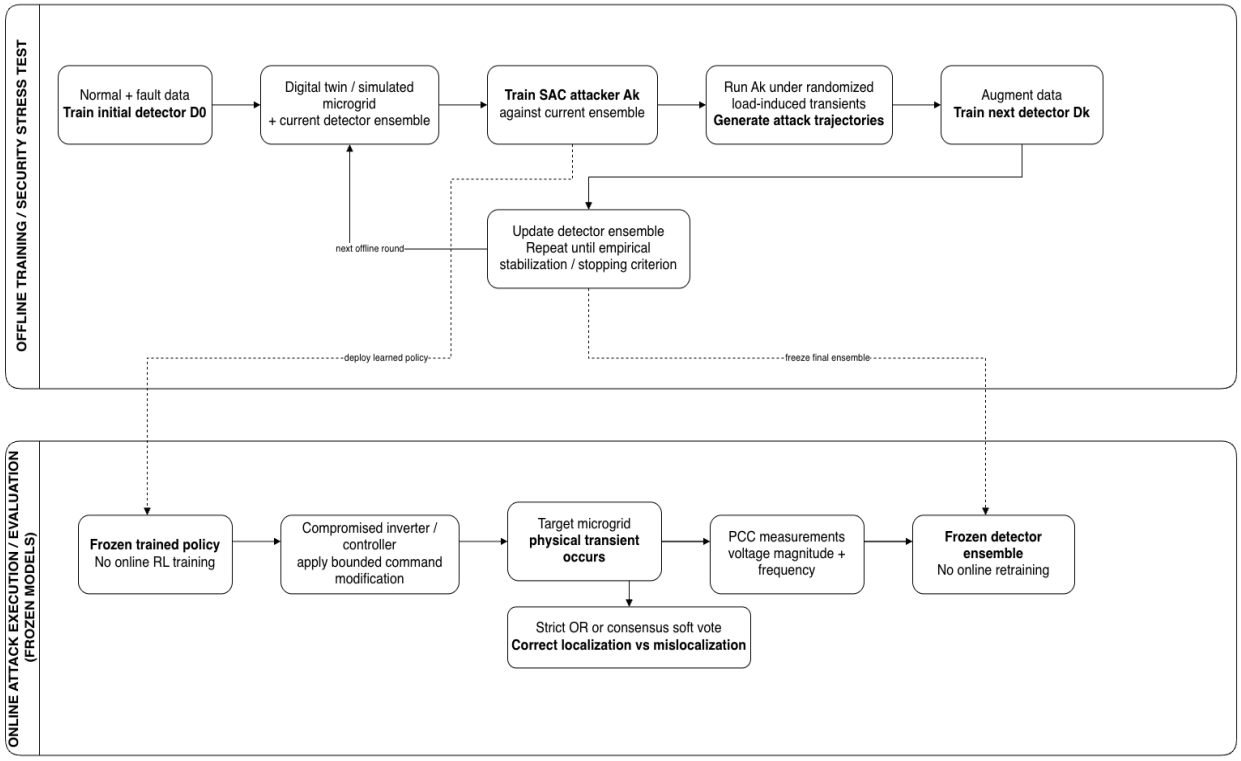


Figure 4.2: Separation between offline adversarial training/security stress testing and online frozen-policy attack execution/evaluation. Iterative attacker/detector training occurs only offline; the physical transient is evaluated using frozen models.

on this observation, the attacker selects a continuous action

$$a_t \sim \pi_\theta(\cdot \mid o_t), \tag{4.3}$$

where π_θ denotes the parameterized stochastic policy learned by SAC.

Let $u^0_{k^\star,t}$ denote the nominal command of the compromised inverter and let $\Delta u(a_t)$ denote the bounded command modification induced by the policy. We write the attacked command generically as

$$u^{\mathrm{atk}}_{k^\star,t} = u^0_{k^\star,t} + \Delta u(a_t). \tag{4.4}$$

The microgrid then evolves according to the underlying simulation model, producing the next observation o_{t+1} and reward r_t.

The attacker policy is trained to maximize the expected discounted return:

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right], \tag{4.5}$$

where $\gamma \in (0, 1)$ is the discount factor and T is the episode length determined by the termination logic.

We employ SAC due to its stability in continuous action spaces and its entropy-regulated objective, which encourages sufficient exploration while maintaining training robustness. The entropy terms prevent premature convergence to deterministic strategies, which is particularly important in adversarial settings where the detector behavior may vary.

Observation Vector The attacker observation is a fixed-length feature vector constructed from a windowed history of inverter-side voltage, current, active-power, reactive-power, and frequency signals, denoted $(v_i, i_i, P_i, Q_i, f_i)$ for $i \in \{1, 2, 3\}$. This intentionally strong information set is available in the offline informed-attacker model described above and provides temporal context for learning transient-synchronized behavior.

Detector-in-the-Loop Signals During attacker training, the environment also evaluates the deployed supervisory detectors on normalized inputs derived from the PCC-level signals. Specifically, the detector input at time t is constructed by stacking and time-ordering the voltage and frequency at the point of common coupling (PCC), $\big[v_{\text{pcc}}(t-N+1), \ldots, v_{\text{pcc}}(t),\ f_{\text{pcc}}(t-N+1), \ldots, f_{\text{pcc}}(t)\big]$, over the most recent N samples.

Reward Design The attacker reward combines (i) a mislocalization shaping term that encourages behaviors that lead to incorrect localization, and (ii) penalties that enforce *stealth* by discouraging excessive deviations/variability in PCC voltage/frequency beyond what is typical for benign load transients.

Let the true attacked inverter class be $k^\star$, and let $\mathcal{K}_{\text{mis}}$ denote the set of incorrect target classes the attacker wants detectors to select (e.g., the other inverter indices). We reuse the

per-detector probability vector $\mathbf{p}_t^{(m)} = \left[p_{1,t}^{(m)}, \ldots, p_{K,t}^{(m)}\right]$ introduced in (1), now indexed by time t, where $p_{k,t}^{(m)}$ is the predicted probability for class k from detector D_m at time t.

For detector m, we define the mislocalization score as

$$\phi\Big(\mathbf{p}_t^{(m)}\Big) \;=\; \sum_{k \in \mathcal{K}_{\text{mis}}} p_{k,t}^{(m)} \;-\; p_{k^\star,t}^{(m)}. \tag{4.6}$$

We aggregate multiple detector scores into a single shaping signal

$$\Phi_t \;=\; \sum_m \alpha_m \, \phi\Big(\mathbf{p}_t^{(m)}\Big)\,, \tag{4.7}$$

with fixed weights α_m (equal weighting in our current implementation).

Rather than enforcing strict stealth (i.e., remaining undetected), we constrain the attacker to induce mislocalization using physically plausible perturbations. In practice, large deviations in PCC voltage and frequency would make the attack easily detectable. Therefore, we penalize excessive deviations and variability, encouraging the attacker to operate within the range of normal transient dynamics while still achieving incorrect localization.

Let D_t denote a normalized deviation measure and V_t denote a normalized variability (e.g., variance) measure computed over a recent window of PCC voltage and frequency signals. To avoid penalizing natural transient behavior, we apply hinge thresholds:

$$\text{pen}_D(t) = \max(0, D_t - \tau_D), \qquad \text{pen}_V(t) = \max(0, V_t - \tau_V), \tag{4.8}$$

where τ_D and τ_V are predefined tolerance levels.

Reward Formulation The per-step reward balances mislocalization objectives and stealth constraints:

$$r_t = w_{\text{det}} \, \tanh\big(\beta \, \Phi_t\big) \;-\; w_D \, \text{pen}_D(t) \;-\; w_V \, \text{pen}_V(t), \tag{4.9}$$

where Φ_t is the mislocalization shaping term defined earlier, and w_{det}, w_D, w_V control the trade-off between attack effectiveness and stealth.

To further encourage confident mislocalization, we include an auxiliary confidence-based shaping term:

$$b_t \;=\; \sum_m \mathbf{1}\Big(\max_k p_{k,t}^{(m)} \geq \theta\Big) \cdot \begin{cases} +1, & \arg\max_k p_{k,t}^{(m)} \in \mathcal{K}_{\text{mis}}, \\ -1, & \arg\max_k p_{k,t}^{(m)} = k^\star, \\ \;\;0, & \text{otherwise}, \end{cases} \tag{4.10}$$

where θ is a confidence threshold.

The final reward is given by

$$r_t \;\leftarrow\; \text{clip}(r_t + \lambda\, b_t,\; r_{\min},\; r_{\max})\,, \tag{4.11}$$

where $\text{clip}(x, a, b) \triangleq \max(a,\, \min(b, x))$ bounds its argument to the interval $[r_{\min}, r_{\max}]$, and λ controls the influence of the auxiliary term.

Episode Termination and Aggregation-Specific Success Criteria An episode ends when the attacker achieves sustained mislocalization (*success*) or the aggregation rule sustainably identifies the true compromised inverter (*failure*). Consecutive-step counters are used so that an isolated classifier fluctuation does not terminate the episode.

Let $c_t^{(m)} = \arg\max_k p_{k,t}^{(m)}$ be detector m's class prediction and let $k^\star$ be the true compromised-inverter class. Under *strict aggregation*, defense success at time t is

$$\text{Fail}_t^{\text{strict}} = \bigvee_{m=1}^{M} \mathbf{1}[c_t^{(m)} = k^\star], \tag{4.12}$$

because one correct detector is sufficient to flag the true source. Attacker success requires every detector to select an incorrect inverter class,

$$\text{Succ}_t^{\text{strict}} = \bigwedge_{m=1}^{M} \mathbf{1}[c_t^{(m)} \in \mathcal{K}_{\text{mis}}]. \tag{4.13}$$

Predictions of the Normal class, therefore, do not count as successful mislocalization.

Under *consensus aggregation*, the final class is the soft-voting decision $\hat{k}_{\text{consensus},t}$ defined above. The corresponding criteria are

$$\text{Succ}_t^{\text{cons}} = \mathbf{1}[\hat{k}_{\text{consensus},t} \in \mathcal{K}_{\text{mis}}], \tag{4.14}$$

$$\text{Fail}_t^{\text{cons}} = \mathbf{1}[\hat{k}_{\text{consensus},t} = k^\star]. \tag{4.15}$$

The same sustained-counter logic is then applied to the selected aggregation rule. This definition makes the reported consensus ASR directly interpretable: a successful episode is one in which the *aggregate* decision repeatedly localizes the event to the wrong inverter, rather than merely one in which an individual detector is wrong.

4.3.7 Iterative Adversarial Training Procedure

We build on the iterative attacker–detector procedure introduced in our prior work [5, 6]. At each offline stage, a new attacker is trained against the current detector ensemble and a new detector is subsequently trained using trajectories generated by that attacker. The procedure is a security stress-testing/adversarial-training loop; it is not an online detector-update mechanism.

Initialization The first detector is trained only on normal-operation and fault data, without any adversarial examples. This baseline detector reflects a realistic deployment scenario in which detection models are designed using known operational disturbances but have not yet encountered adaptive attacks.

Iterative Loop Let $\mathcal{D}^{(k)}$ denote the ensemble of detectors available at iteration k. Given $\mathcal{D}^{(k)}$, we train an attacker policy $\pi^{(k+1)}$ using SAC, where the reward incorporates mislocal-

ization and detection outcomes produced by $\mathcal{D}^{(k)}$. Formally, the attacker seeks to maximize

$$J^{(k+1)}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T-1} \gamma^t r_t(\mathcal{D}^{(k)})\right], \tag{4.16}$$

where the reward explicitly depends on the decisions of the current detectors.

Based on the recent attack, we generate a dataset of attack trajectories by deploying the trained policy under diverse transient scenarios. These trajectories are labeled according to the true compromised inverter for further training of a new detector.

A new detector $D^{(k+1)}$ is then trained using supervised learning on this new dataset, which now includes attack traces produced by $\pi^{(k+1)}$. The updated detector ensemble becomes

$$\mathcal{D}^{(k+1)} = \mathcal{D}^{(k)} \cup \{D^{(k+1)}\}. \tag{4.17}$$

Unlike the initial detector, all subsequent detectors are exposed to attack-generated data and therefore learn to recognize increasingly sophisticated adversarial patterns.

Empirical stopping/stabilization criterion The iterative process is continued until attack success is suppressed or successive measured ASRs cease to exhibit statistically meaningful change under the chosen test. We use the term *empirical stabilization* rather than claiming formal game-theoretic convergence: the attacker and detector objectives are nonconvex, and the experiments do not establish a Nash equilibrium, monotonicity theorem, or contraction property. This distinction is important because individual ASR values can exhibit small reversals even when the overall trend improves.

This loop progressively exposes the detector family to newly discovered attack trajectories while repeatedly challenging the expanded ensemble with a newly trained attacker.

4.3.8 Implementation Details

Simulation and Sampling The microgrid is simulated in MATLAB/Simulink with a fixed-step solver of $T_s = 40, \mu$s (25 kHz), while the SAC policy is evaluated every 1×10^{-4},s

(10 kHz). Each episode has a maximum simulation horizon of 1.5 s, corresponding to at most 15,000 agent steps. The attack onset is randomized within the interval $[1.0, 1.15]$ s, after which the attack may remain active until the maximum episode horizon or until an early-termination criterion is satisfied. The timing of the load-induced transient is varied relative to the attack onset so that the learned policy is exposed to different transient phases.

Observation and Action The attacker's observation stacks the most recent $N_w = 200$ samples of the inverter-side quantities $(v_i, i_i, P_i, Q_i, f_i)$ for $i = 1, 2, 3$, yielding a windowed observation. The action is a single continuous scalar $a_t \in [-1, 1]$ that produces the bounded modification $\Delta u(a_t)$ in the compromised inverter command defined above.

Networks Both the actor and the twin critics use ReLU activations, He weight initialization, and a layer-normalization layer at the observation input. The actor is a Gaussian policy, and there are two critics, initialized independently, as is standard for SAC.

SAC Hyperparameters Table 4.3 lists the SAC hyperparameters used in all experiments.

Table 4.3: SAC hyperparameters.

Parameter	Value
Discount factor γ	0.99
Target smoothing factor	0.995
Replay buffer size	2×10^5
Mini-batch size	128
Warm-start steps	500
Actor / critic optimizer	Adam
Actor / critic learning rate	5×10^{-4}
L_2 regularization	5×10^{-4}
Gradient clipping	ℓ_2-norm
Entropy weight (initial)	2.0
Entropy-weight learning rate	3×10^{-4}
Target entropy	-1
Max episodes	500
Max steps per episode	15000
Score-averaging window	100
Agent sample time	1×10^{-4} s

Table 4.4: Per-inverter attack success rate (ASR, %) across iterative detector expansion, under both aggregation settings. For each step, we report the ASR against the current detector ensemble ("Before") and after adding the new detector ("After").

Step	Target Inv.	Strict Aggregation		Consensus Aggregation	
		Before (%)	After (%)	Before (%)	After (%)
Step 1 ($\|\mathcal{D}\| = 1$)	Inv1	100.0	7.0	100.0	100.0
	Inv2	100.0	90.0	100.0	100.0
	Inv3	36.0	31.0	36.0	39.0
Step 2 ($\|\mathcal{D}\| = 2$)	Inv1	75.0	0.0	100.0	49.0
	Inv2	100.0	12.0	100.0	0.0
	Inv3	34.0	34.5	45.0	42.0
Step 3 ($\|\mathcal{D}\| = 3$)	Inv1	0.0	0.0	58.0	12.5
	Inv2	85.5	0.0	92.0	100
	Inv3	37.0	0.0	44.0	31.5
Step 4 ($\|\mathcal{D}\| = 4$)	Inv1	0.0	0.0	73.0	8.0
	Inv2	15.0	0.0	100	100
	Inv3	0.0	0.0	69.5	31.5
Step 5 ($\|\mathcal{D}\| = 5$)	Inv1	0.0	0.0	0	0
	Inv2	0.0	0.0	100	100
	Inv3	0.0	0.0	26	33.5

4.4 Results

4.4.1 Iterative Attacker–Detector Training Dynamics

All experiments reported in this section are conducted under transient operating conditions induced by load changes. As described in Section 4.3.1, the reported attack success rates (ASRs) therefore quantify attacker performance in the studied non-steady-state regime; they are not a direct transient-versus-steady-state comparison.

Table 4.4 and Table 4.5 summarize the evolution of ASR as the detector ensemble is expanded under strict and consensus aggregation. Each offline iteration trains a new attacker against the current detector ensemble and then trains a new detector using trajectories generated by that attacker. The two-proportion z-test measures whether the observed ASR changes between the compared conditions are statistically significant.

Table 4.5: Two-proportion z-tests of ASR changes across iterative steps under strict and consensus aggregation. Attacker-side tests hold the attacker fixed and compare ASR before and after detector addition. Detector-side tests hold the detector ensemble fixed and compare successive attackers. "Sig." denotes $|z| > 1.96$; dashes indicate unavailable next-iteration comparisons.

Step	Inv.	Strict Att.		Strict Det.		Cons. Att.		Cons. Det.	
		z	Sig.	z	Sig.	z	Sig.	z	Sig.
1	Inv1	18.65	Yes	-13.83	Yes	0.00	No	0.00	No
	Inv2	4.59	Yes	-4.59	Yes	0.00	No	0.00	No
	Inv3	1.06	No	-0.64	No	-0.62	No	-1.22	No
2	Inv1	15.49	Yes	0.00	No	11.70	Yes	-1.80	No
	Inv2	17.73	Yes	-14.70	Yes	20.00	Yes	-18.46	Yes
	Inv3	-0.11	No	-0.52	No	0.61	No	-0.40	No
3	Inv1	0.00	No	0.00	No	9.52	Yes	-12.23	Yes
	Inv2	17.28	Yes	-5.69	Yes	-4.08	Yes	0.00	No
	Inv3	9.53	Yes	0.00	No	2.58	Yes	-7.60	Yes
4	Inv1	0.00	No	0.00	No	13.24	Yes	4.08	Yes
	Inv2	5.69	Yes	0.00	No	0.00	No	0.00	No
	Inv3	0.00	No	0.00	No	7.60	Yes	1.22	No
5	Inv1	0.00	No	–	–	0.00	No	–	–
	Inv2	0.00	No	–	–	0.00	No	–	–
	Inv3	0.00	No	–	–	-1.64	No	–	–

Interpretation of statistical comparisons Tables 4.4 and 4.5 report two complementary comparisons at each iteration. The attacker-side test compares the "Before" and "After" ASRs within the same step, holding the attacker fixed while adding a new detector to the ensemble. In contrast, the detector-side test compares the "After" ASR at step k with the "Before" ASR at step $k+1$. This holds the detector ensemble fixed while evaluating the newly trained attacker. Positive z-values indicate that the ASR is higher in the first condition, whereas negative values indicate the opposite. At the final iteration, detector-side comparisons are unavailable because no subsequent attacker is evaluated. Under strict aggregation, the zero ASRs at Step 5 indicate that the newly trained attacker did not recover successful mislocalization in the evaluated trials, providing further empirical evidence of attack suppression.

4.4.2 Progressive Suppression Under Strict Aggregation

Strict aggregation produces an overall suppression of attack success as detectors are added. ASR is initially high, but successive detector additions reduce the attacker's ability to mislocalize the compromised inverter, and by the final reported iteration, the strict ensemble reaches 0% ASR for all three inverters in the studied configuration.

However, the trend should not be described as mathematically monotonic. Table 4.4 contains a small local reversal for Inv3 at Step 2 (34.0% $\rightarrow$ 34.5%). Consequently, we do not assert the relation $\mathrm{ASR}_{k+1} \leq \mathrm{ASR}_k$ as a theorem or an empirical invariant. The results show that strict aggregation progressively suppresses ASR overall. Although the newly trained attacker at Step 4 temporarily recovers a 15% ASR against inverter 2, adding the new detector again reduces it to 0%. In the final iteration, even the newly trained attacker fails to recover successful mislocalization across all three inverters in the evaluated trials. The latter statistically insignificant changes are interpreted as empirical evidence of stabilization in the measured performance.

This behavior is directly explained by the strict decision rule. To achieve a successful mislocalization, the attacker must simultaneously evade every detector in the ensemble; a correct localization by any member of the ensemble is sufficient to defeat the attack. Therefore, adding detectors increases the number of decision boundaries the attacker must defeat at once. In the present experiments, this increasingly restrictive condition eventually eliminates successful mislocalization.

4.4.3 Persistent Mislocalization Under Consensus Aggregation

Consensus aggregation exhibits a qualitatively different pattern. Because the final class is selected from the averaged confidence vector, a correct minority prediction can be outweighed by other detector outputs. ASR therefore fluctuates rather than decreasing consistently. For example, Inv2 reaches 0% ASR at an early iteration but later returns to high attack success, while Inv3 retains non-zero success across multiple stages. These results show that adding individually useful detectors does not by itself guarantee a more robust *aggregate* decision.

Later iterations contain smaller or statistically insignificant ASR differences under the reported test. We refer to this only as *empirical stabilization*: the measured attacker–detector outcomes stop changing materially under the finite experiment. We do not claim a formal equilibrium, monotonicity property, or convergence guarantee for the non-convex learning dynamics.

Why consensus remains exploitable The difference follows from the aggregation objective. Under strict aggregation, one correct detector defeats the attack. Under consensus aggregation, the attacker can instead benefit from detector disagreement: it is sufficient for the weighted aggregate confidence to favor an incorrect inverter. Thus, a detector that correctly identifies the compromised inverter can be suppressed by the aggregate vote. The aggregation rule itself therefore becomes part of the adversarial decision surface.

Relation to stealthy and zero-dynamics attacks Persistent non-zero ASR under partial observability is compatible with the broader intuition behind stealthy and zero-dynamics attacks [5, 16]: monitored outputs may provide insufficient information to uniquely expose malicious internal behavior. However, non-zero ASR or persistent misclassification alone does *not* establish a formal zero-dynamics attack. Demonstrating that claim would require the corresponding invariant-zero/state-space conditions and evidence that relevant internal state behavior remains hidden from the monitored outputs. We therefore characterize the observed result conservatively as *persistent stealthy mislocalization under consensus aggregation.*

4.4.4 Representative Attack Trajectories

Comparison setup Figure 4.3 presents three selected simulations of attacks targeting inverter 1: an initial attacker and later attackers trained under strict and consensus aggregation. Columns identify the measured inverter, not different attack targets; each setting therefore uses the same simulation across all three columns. Attack onset is aligned at $t = 0$, and the available pre-attack history is retained. The initial, strict, and consensus records

end 16.67, 112.45, and 21.22 ms after onset, respectively. These are recording endpoints, not measurements of detection delay.

Physical responses and localization significance The initial case shows an early current rise and power fluctuations. The selected strict case exhibits smaller early current changes, followed by a sharp increase later in the recording. At the targeted inverter, current peaks occur at approximately 11.2 ms in the initial case and 107.4 ms in the strict case. The consensus case exhibits current and power changes earlier than the late increase in the strict recording. These examples illustrate different temporal responses; because their pre-attack histories and recording durations differ, they do not establish a systematic timing advantage for either aggregation setting.

The current peaks are largest at the targeted inverter in these selected runs, but current and power also change at uncompromised inverters 2 and 3. These spatially distributed responses are consistent with electrical coupling and illustrate why observing a disturbance at an inverter does not, by itself, identify that inverter as its source. The supervisory detectors receive PCC voltage and frequency, rather than the additional current and power signals shown in the figure. Their task is therefore to distinguish the compromised source from the responses of the connected system using limited measurements. Mistaking a response at an uncompromised inverter for the disturbance's origin could direct mitigation toward the wrong asset.

Connection to ensemble attack success The trajectories provide physical context for the trial-level ASRs in Table 4.4, rather than directly establishing the localization outcome of each saved recording. All plotted attacks target inverter 1; the persistent final consensus mislocalization reported for inverters 2 and 3 is supported by the ASR experiments, not by these illustrative traces. Importantly, ASR measures sustained wrong-inverter attribution, so zero measured ASR does not imply the absence of physical disturbances. Supplementary individual-detector confusion matrices are provided in Appendix A.4, where their window-level classification errors are distinguished from ensemble attack success.

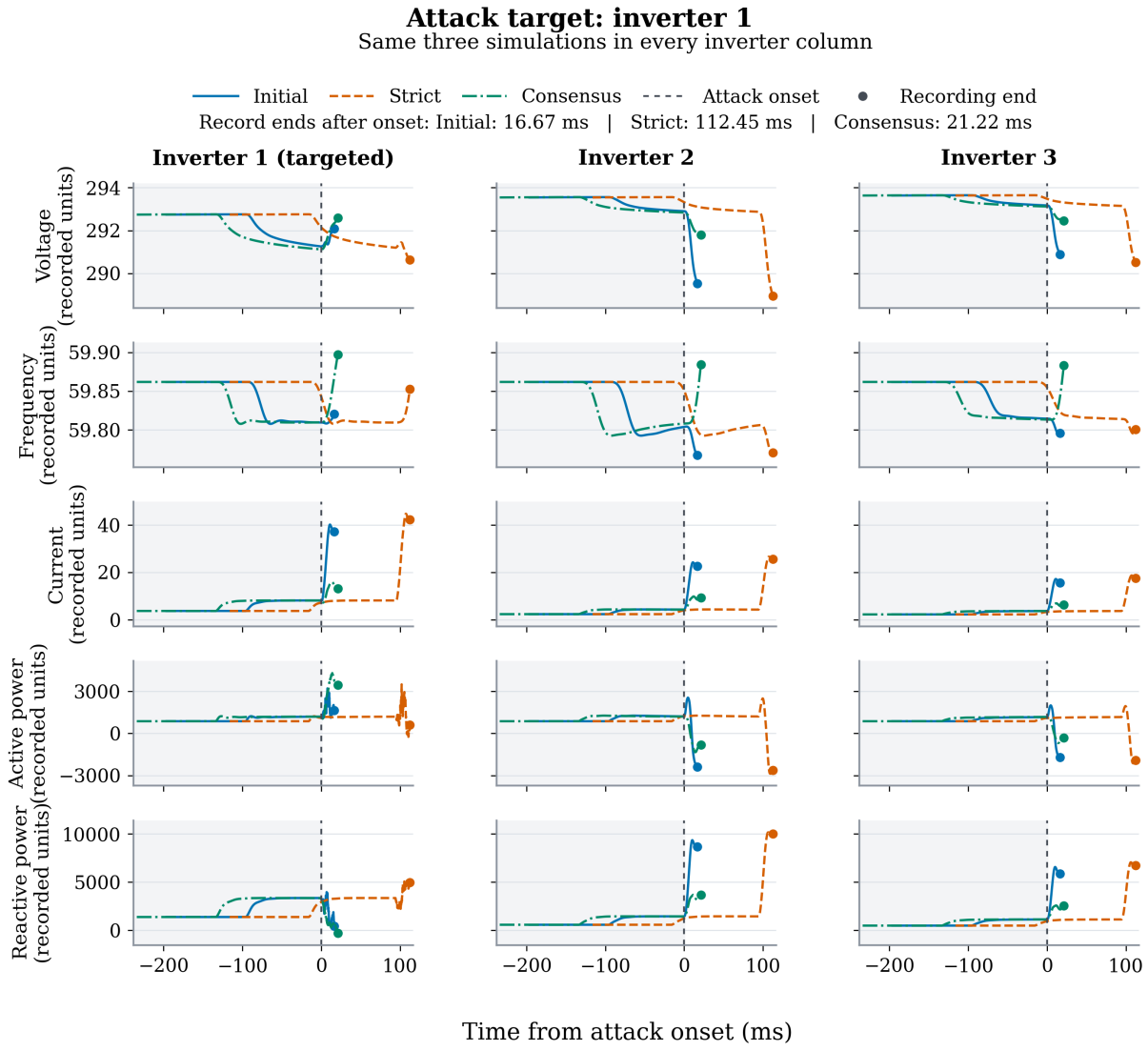


Figure 4.3: Recorded signals from three simulations of attacks targeting inverter 1. Columns identify the measured inverter, while line colors and styles identify the attack on a particular setting. Each setting uses the same simulation across all columns. The vertical dashed line marks attack onset, shading indicates the pre-attack interval, and dots mark recording endpoints. Each signal row uses the same vertical scale across inverters.

4.4.5 Implications for Aggregation Design

The results expose a security–sensitivity trade-off between the two aggregation rules. Strict aggregation is adversarially conservative because any correct detector can identify the true source, but this same OR-like rule can also increase the chance that a single erroneous detec-

tor produces an extra alarm. Consensus aggregation can suppress isolated detector outputs, but it creates a different vulnerability: correct minority evidence can be overwhelmed by the averaged scores. The principal conclusion is therefore not that one should simply add more detectors, but that the aggregation rule must itself be evaluated adversarially.

The present results establish this effect only for the studied detector family and three-inverter test system. Claims about false-alarm advantages, detection latency, other detector paradigms, or scalability to larger microgrids require the corresponding measurements and are not inferred from ASR alone.

4.5 Conclusion

This work investigated transient-synchronized mislocalization attacks in a grid-forming microgrid by building on the iterative attacker–detector framework introduced in our earlier studies. The expensive attacker–detector adaptation loop is performed offline; online evaluation uses a frozen SAC policy and a frozen detector ensemble. This separation removes any requirement for detector retraining during a sub-second physical transient and positions the digital twin as an informed-attacker security stress-testing environment rather than as an online component of the attack.

The experiments show that detector aggregation materially changes adversarial robustness. Strict aggregation progressively suppresses attack success and reaches 0% ASR for all three inverters in the final reported configuration, whereas consensus aggregation permits persistent non-zero mislocalization for some inverters because correct minority predictions can be overridden by the aggregate confidence. These findings motivate treating the aggregation mechanism itself as part of the cyberattack surface. We intentionally describe the result as persistent stealthy mislocalization rather than as a formal zero-dynamics attack, because ASR alone does not establish invariant-zero behavior or hidden-state divergence.

The evaluation is a controlled proof-of-concept on a three-inverter simulated microgrid and does not establish scalability to arbitrary network sizes. It also does not include a matched steady-state baseline, so the results demonstrate the importance of studying the

transient attack surface rather than proving that every transient regime is more vulnerable than steady-state operation. Future work will extend the evaluation to larger and hardware-supported systems, quantify false-alarm and detection-latency trade-offs, study attackers with imperfect model knowledge, compare additional supervised/unsupervised/RL detector families, and develop aggregation rules that retain the benefits of ensemble diversity without allowing correct minority evidence to be suppressed.

4.6 Discussion

4.6.1 Detection, Localization, and Mitigation Are Different Tasks

This chapter changes the meaning of attacker success from the physical-disruption objective studied earlier. The attacker seeks sustained attribution to the wrong inverter. A supervisory system can therefore recognize that an event is abnormal and still fail the task being evaluated. Detecting an anomaly is only the first step when the response depends on identifying its source.

The distinction also separates mislocalization from concealment. Under the success rules in this chapter, a normal-class prediction does not count as a successful wrong-inverter attribution. Such a prediction may raise another security concern, but it is not the objective measured by the reported mislocalization success rate. Likewise, a wrong prediction at one instant is insufficient when success requires the error to persist across the specified decision interval.

The operational concern is the possibility of an incorrect response. A supervisor that attributes an event to a healthy inverter may direct isolation toward that inverter. However, the reported mislocalization rates do not themselves measure the physical consequences of carrying out every possible isolation decision. They establish a decision vulnerability that can affect mitigation, rather than a complete end-to-end assessment of all resulting protection actions.

4.6.2 The Meaning of Strict Aggregation

Strict aggregation preserves a class prediction whenever at least one detector produces it. Its output can therefore contain several candidate classes. In the experimental success rule, the defense defeats the attacker when the true compromised-inverter class is identified with the required persistence. This evaluates whether correct source information remains present somewhere in the aggregate output.

The rule does not imply that the deployed supervisor knows which detector is correct. Ground-truth inverter identity is available to the evaluator for scoring the experiment, not as an extra observation supplied to the supervisor. When detectors disagree, strict aggregation can retain the true source alongside an incorrect source. Additional decision logic would be needed to decide whether to gather more evidence, flag several candidates, or take a protective action.

The zero-ASR result under strict aggregation should therefore be interpreted according to that set-valued decision rule. It does not necessarily mean that the ensemble always supplies one unambiguous and correct inverter label. Nor does it mean that no unnecessary alarm or physical disturbance occurs. This distinction is important when comparing strict aggregation with consensus, which is required to produce a single final class.

4.6.3 How Averaging Can Suppress Correct Evidence

The vulnerability of confidence averaging can be illustrated without changing the trained detectors. Consider three detector confidence vectors, with entries ordered as normal operation, inverter 1, inverter 2, and inverter 3. Suppose inverter 1 is the true source and the vectors are

$$\begin{aligned} \mathbf{p}^{(1)} &= (0.02,\ 0.70,\ 0.23,\ 0.05), \\ \mathbf{p}^{(2)} &= (0.02,\ 0.10,\ 0.83,\ 0.05), \\ \mathbf{p}^{(3)} &= (0.02,\ 0.10,\ 0.83,\ 0.05). \end{aligned} \tag{4.18}$$

These values are an illustrative example, not additional experimental measurements. The first detector correctly selects inverter 1, whereas the other two select inverter 2. Equal-weight averaging produces

$$\mathbf{p}^{\text{agg}} = (0.02,\ 0.30,\ 0.63,\ 0.05), \tag{4.19}$$

so the consensus output is inverter 2 despite the correct prediction from the first detector.

Strict aggregation would preserve both inverter candidates in this example. Consensus instead converts the same evidence into one incorrect decision. The example demonstrates the difference at a single decision instant. An actual successful episode still requires the chapter's sustained mislocalization condition; one averaged error alone is not sufficient.

This mechanism explains why detector count is not an adequate measure of ensemble robustness. Adding a useful detector changes the average, but it does not guarantee that its correct predictions determine the final class. The outcome depends on disagreement, confidence values, and the rule that combines them. The appropriate evaluation target is the complete decision procedure.

4.6.4 Physical Responses Do Not Uniquely Identify Their Source

The representative trajectories show that a manipulation at one inverter can produce responses at the other inverters. Consequently, the location with a visible current or power change is not automatically the origin of the event. The coupled network carries the effect beyond the compromised component.

The additional inverter signals shown in the trajectory figure help explain that physical response, but the supervisory detector does not receive all of them. Its classification task remains based on the specified PCC voltage and frequency history. Reading the figure from the researcher's viewpoint must not accidentally grant the detector access to measurements excluded by the threat model.

The illustrative trajectories and the ASR experiments also provide different evidence.

A trajectory shows how a selected run develops; the trial-level result establishes whether sustained wrong-inverter attribution occurs under the evaluation rules. Neither the end of a plotted record nor the location of its largest peak should be treated as a detection timestamp without the corresponding decision log. This keeps the interpretation of the physical plots consistent with the measured classification outcome.

4.6.5 What the Aggregation Comparison Establishes

The two aggregation settings are evaluated within an adaptive framework. Attackers are trained against the selected rule, and later detectors are developed from the resulting attack data. The reported outcomes therefore describe defenses together with adversaries that adapt to them. They are not simply a comparison of two formulas applied once to identical frozen detector outputs.

This makes the comparison relevant to an adaptive threat, but it also defines its scope. The results show different outcomes for the complete strict and consensus configurations in the studied system. Isolating only the immediate effect of fusion would require evaluating both rules on the same detector predictions and trajectories. The illustrative example above explains a possible fusion mechanism; it does not replace such a matched experimental comparison.

Similarly, the transient experiments show that sustained mislocalization can occur during the modeled load-induced events. Without a matched steady-state baseline, they do not establish how much more or less vulnerable transient operation is. The contribution is the explicit study of this operating regime and decision objective, rather than a general ranking of all operating conditions.

4.6.6 Implications across the Dissertation

The results suggest two related directions for defense design. One concerns the information available to the supervisor: measurements must contain enough source-specific evidence to support localization. The other concerns how that evidence is used: an aggregation rule

should not be assumed robust merely because some of its members are accurate. More sensing and different fusion rules are possible responses, but their benefit would need to be measured together with false alarms, decision ambiguity, and response requirements.

Across the dissertation, the common lesson is that security depends on the relationship between physical behavior and the decision made from limited evidence. The tank study examines hidden-state disruption. The first microgrid study examines detection relative to a physical deadline. This chapter examines whether the correct source survives the final decision process. Iterative adversarial evaluation connects these tasks by actively searching for the behavior that the current defense does not handle well.

Chapter 5

Utility-Constrained Mixed Measurements for Reducing Zero-Dynamics attack after System Identification

5.1 Introduction

Measurements are essential for controlling a physical process and detecting abnormal behavior. However, they may not reveal every harmful change in the system's internal state. Zero-dynamics attacks exploit this limitation by manipulating control inputs that change internal states while canceling their effect on the measured outputs [16]. The term zero dynamics refers to internal behavior that can evolve while the output deviation remains zero. When these dynamics are unstable, an attack can drive internal states beyond their safe limits even though the measured outputs appear normal. Under ideal model and initialization conditions, such an attack produces no change in the output-based detector's residual, which represents the difference between measured and predicted outputs.

Constructing this cancellation depends on the attacker's knowledge of the plant. Al-

though the exact parameters may be unavailable, an attacker can observe input–output data and use grey-box system identification to estimate them. The authors in [61] showed that learned state-space models can support zero-dynamics attacks and that their success depends on the detector. Because a learned model generally differs from the real plant, an attack designed to remain hidden in that model may produce observable deviations when applied to the real system. Consequently, attacks constructed using the zero dynamics of a learned model are not necessarily perfectly stealthy against the actual plant.

This distinction makes the timing of detection important. An attack may eventually trigger an alarm, but still drive the plant beyond a physical limit before that alarm occurs. We therefore use damage before alarm (DBA) as the primary security measure. In this study, a DBA event occurs when the real plant violates a specified physical limit before the trusted detector raises an alarm. Here, damage is represented by this violation of the physical limit. This measure captures the attacker's ability to cause harm during the period before the detector responds, even if the attack becomes detectable later.

These observations motivate changing the information available for system identification so that the attacker learns a model whose attacks are less effective against the real plant. The aim is to reduce DBA by making those attacks less able to cause a physical-limit violation before detection. However, the measurements exposed to the attacker are also used for feedback control, so changing them can affect the controller's response. A useful defense must therefore limit the value of the exposed information to attackers while preserving enough information for reliable control.

A possible approach, proposed here, is to introduce a simulated surrogate with different plant parameters. Both plants receive the same nominal command, while the surrogate supplies the measurements used by the controller and exposed to the attacker. Therefore, the controller regulates the surrogate while also driving the real plant. Because the attacker learns from the surrogate's response, the resulting attack may be less effective against the real plant.

Although surrogate-only feedback can change what the attacker learns, it also hides disturbances that affect only the real plant. The controller may therefore continue to observe

normal behavior while the real state moves away from its reference. Experiments with the quadruple-tank process (QTP) illustrate why this limitation can be difficult to recognize from a trajectory alone. Its stable drainage dynamics can cause a temporary height disturbance to decay under the continuing pump command, even when the controller receives no information about that disturbance. Such passive recovery does not establish active disturbance rejection, and it cannot be assumed for other plants or operating conditions.

To retain a feedback path from the real plant, we combine real and surrogate measurements as $y_{\text{mix}} = \alpha y_{\text{real}} + (1 - \alpha) y_{\text{fake}}$, where $\alpha \in [0, 1]$ is the real-measurement weight. The controller uses this mixture, and the attacker learns its model from records formed with the same measurement rule. Thus, a real disturbance can affect the controller's feedback, while the surrogate contribution changes the response exposed to the attacker. A separate trusted detector continues to monitor real measurements.

This mixture introduces a further design choice: how much real information should it retain? A smaller real-measurement weight preserves a larger surrogate contribution, but it may also slow the response to a real disturbance. We therefore select the weight using recovery time, requiring the response to remain within a prescribed margin of the real-feedback baseline. This criterion directly measures the disturbance response, whereas a tracking error averaged over a long observation period may place little weight on a temporary but slow recovery. Normal-operation tracking and physical limits provide additional requirements for the design.

Once the measurement weight is fixed, the surrogate parameters must be selected with similar care. Preliminary experiments show that some arbitrary perturbations violate the physical limits, so perturbation size alone cannot determine whether a surrogate is useful. Moreover, the parameters affect the system in different ways, and their combined effect must be evaluated. We address this problem using the covariance matrix adaptation evolution strategy (CMA-ES) [62], which searches for parameter combinations that reduce damage before alarm (DBA). Here, DBA means that a physical-limit violation occurs before the trusted detector raises an alarm. The search allows different signs and magnitudes of parameter change while imposing tracking, safety and disturbance-recovery requirements.

We evaluate real-only feedback and mixed feedback with either an unperturbed or optimized surrogate, separating the effect of introducing a surrogate from that of changing its dynamics. Under the evaluated threat model and stated statistical assumptions, the optimized configuration produces a significant reduction in DBA relative to both baselines, while meeting the overall tracking limit and the declared recovery requirements. These findings support a surrogate design that accounts for both the information exposed to the attacker and the feedback needed by the controller.

5.2 Literature Review

The literature most relevant to this chapter concerns zero-dynamics attacks, protection against system identification, and the relationship between detection and physical consequences. These studies provide a basis for changing the information available to an attacker while preserving the feedback needed for control.

5.2.1 Zero-Dynamics Attacks and Attacker Knowledge

The authors of [16] analyzed attacks under different assumptions about system knowledge, information access, and signal manipulation. The authors in [2] established conditions for attack detection and identification, showing that some limitations arise from system dynamics and available measurements rather than detector design alone. These studies explain how harmful internal behavior can remain hidden from an output-based monitor.

Later work considers attackers without exact plant models. The authors of [63] developed robust zero-dynamics attacks using nominal model information and disclosed input–output signals. Additionally, the authors of [5] used reinforcement learning to generate attacks without directly using state-space matrices, whereas [61] examined zero-dynamics attacks constructed from learned state-space models. The latter setting is particularly relevant here: an attacker first identifies a model and then constructs an attack from it. Because cancellation in the learned model may not hold in the real plant, attack outcomes must be

evaluated against the real dynamics and detector.

5.2.2 Protecting System Information

Adding input and output noise to increase identification error for finite impulse response systems while constraining performance degradation is another direction of research to prevent the attack [64]. Their work directly connects model protection with legitimate system performance. The authors of [65] studied differentially private filtering, where released signals protect individual data streams while limiting distortion. However, protecting signal privacy, increasing model estimation error, and preventing a successful physical attack are distinct objectives. The present chapter evaluates attack outcomes rather than claiming a differential privacy guarantee.

Another approach changes the conditions that enable an attack to remain stealthy. The authors of the paper [3] studied changes to system structure that reveal zero-dynamics attacks. The authors of [66] developed moving-target defenses with stochastic, time-varying parameters that limit the attacker's system knowledge. In contrast, the present design leaves the real plant unchanged and holds the selected surrogate parameters fixed. Its defense operates through the response exposed by mixed real and surrogate measurements, rather than continually changing the physical plant.

5.2.3 Detection, Measurement Access, and Physical Impact

Physical watermarking introduces an excitation known to the defender, allowing manipulated sensor reports to be checked against the expected response. The authors of the paper [67] studied watermarked control inputs for replay detection, while another work [68] developed dynamic watermarking against malicious sensor reporting. Similarly, the authors of [69] combined watermarking with a cumulative sum test and examined detection delay under a constraint on additional control cost. These studies connect active detection with control performance, although their sensor-attack settings differ from the learned-model actuator attacks considered here.

Detection also depends on what the monitor can observe. The authors of the paper [70] showed that monitoring location affects attack visibility and developed a field-level monitor that detects attacks hidden from supervisory monitoring. Meanwhile, the authors of [71] evaluated how detector configuration and false-alarm constraints affect the physical impact of stealthy attacks. These findings motivate stating the detector's information access and measuring physical consequences, rather than reporting detection alone.

The present chapter retains a separate trusted detector that uses real measurements, while control and attacker identification use mixed measurements. It does not introduce a watermark or redesign the detector. Instead, it selects the mixture weight from disturbance recovery and optimizes the surrogate for damage before the first eligible alarm: a physical-limit violation. Tracking, physical limitations, and recovery constraints accompany this security objective. Therefore, the comparison concerns whether modifying the exposed information reduces DBA while maintaining the declared control requirements.

5.3 Preliminaries

The proposed design changes the measurements used for control and identification while keeping the real plant fixed. To describe its effect, we first define the plant and surrogate models, then specify the feedback and detection channels. These definitions also distinguish the security outcome, measured by DBA, from the control requirements used to select a surrogate.

5.3.1 Quadruple-tank process

The QTP contains four connected tanks and two pumps. Tanks 1 and 2 are the lower tanks and provide the measured outputs, whereas Tanks 3 and 4 drain into Tanks 1 and 2, respectively. This arrangement provides a useful setting for studying multivariable dynamics and transmission zeros [72]. The experiments use a linearized model in discrete time, so the results concern simulated operation around a nominal operating point.

At sample k, let $x_k^r \in \mathbb{R}^4$ be the real tank-height deviations from the nominal heights, $u_k \in \mathbb{R}^2$ the nominal pump command, and $a_k \in \mathbb{R}^2$ an additive actuator attack. With superscript r denoting the real plant, the state and measurement equations are

$$x_{k+1}^r = A_T x_k^r + B_T(u_k + a_k) + w_k^r, \tag{5.1}$$

$$y_k^r = C_T x_k^r + v_k^r. \tag{5.2}$$

Here, A_T, B_T, and C_T are the true state, input, and output matrices, and $y_k^r \in \mathbb{R}^2$ contains the measured outputs. The zero-mean Gaussian noises w_k^r and v_k^r have covariance matrices Q_w and R_v, respectively. Setting $a_k = 0$ therefore gives the model used for no-attack evaluation.

Because physical safety depends on absolute water levels, the deviation state is converted to real heights as $h_k^r = h_0 + x_k^r$, where

$$h_0 = [12.6, 13.0, 4.8, 4.9]^\top \text{ cm}. \tag{5.3}$$

The symbol $\top$ denotes transpose. Reaching or crossing either 0 cm or 20 cm in any real tank is treated as a violation of the physical limit. The same limits consequently define both the damage event during an attack and the height check during a no-attack operation.

The surrogate is constructed by changing physical parameters, so we also express the model matrices in terms of those parameters. For Tank i, let $\mathcal{A}_i$ be its cross-sectional area, a_i^{out} its outlet area, and h_{i0} its nominal height. Its drainage time constant is

$$\tau_i = \frac{\mathcal{A}_i}{a_i^{\text{out}}} \sqrt{\frac{2h_{i0}}{g}}, \tag{5.4}$$

where g is gravitational acceleration. In addition, let γ_1, γ_2 be the lower-tank flow fractions,

κ_1, κ_2 the pump gains, and k_c the sensor gain. The continuous-time model matrices are then

$$A_c = \begin{bmatrix} -1/\tau_1 & 0 & \mathcal{A}_3/(\mathcal{A}_1\tau_3) & 0 \\ 0 & -1/\tau_2 & 0 & \mathcal{A}_4/(\mathcal{A}_2\tau_4) \\ 0 & 0 & -1/\tau_3 & 0 \\ 0 & 0 & 0 & -1/\tau_4 \end{bmatrix}, \tag{5.5}$$

$$B_c = \begin{bmatrix} \gamma_1\kappa_1/\mathcal{A}_1 & 0 \\ 0 & \gamma_2\kappa_2/\mathcal{A}_2 \\ 0 & (1-\gamma_2)\kappa_2/\mathcal{A}_3 \\ (1-\gamma_1)\kappa_1/\mathcal{A}_4 & 0 \end{bmatrix}, \qquad C = k_c \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5.6}$$

For sampling interval T_s, zero-order-hold discretization gives $A = \exp(A_c T_s)$ and $B = \int_0^{T_s} \exp(A_c s) B_c \, ds$. Evaluating these expressions at the true parameters yields A_T, B_T, C_T; evaluating them at the surrogate parameters yields the second model used below.

5.3.2 Surrogate model and measurement routing

The surrogate evolves alongside the real plant, with its own state x_k^f and output y_k^f. Using superscript f for the fake plant, its equations are

$$x_{k+1}^f = A_F x_k^f + B_F u_k + w_k^f, \tag{5.7}$$

$$y_k^f = C_F x_k^f + v_k^f. \tag{5.8}$$

Both plants receive the nominal command u_k, but only the real plant receives the attack a_k. Although the noise covariances are the same in both branches, their realizations are independent within an episode. The surrogate is also allowed to evolve independently rather than being reset to the real state.

To define its parameters, let $\theta_T \in \mathbb{R}^{17}$ contain the true outlet areas, tank areas, linearization heights, flow fractions, pump gains, and sensor gain. The corresponding surrogate

vector is

$$\theta_F = \theta_T \odot (\mathbf{1} + \rho z), \qquad z \in [-1, 1]^{17}, \tag{5.9}$$

where $\odot$ denotes elementwise multiplication and $\mathbf{1}$ is a vector of ones. The bound ρ sets the maximum relative change, while each component of z determines its sign and magnitude. Thus, the final bound $\rho = 0.50$ permits a 25% increase when $z_i = 0.5$, without requiring the other parameters to change by the same amount. Gravity remains fixed, and the selected surrogate parameters are held constant during identification and attack evaluation.

When $z = 0$, the surrogate matrices equal the true matrices. Nevertheless, the two states and outputs can differ because their noise realizations are independent. When $z \neq 0$, the surrogate can also have different dynamics. In particular, changing its linearization heights changes its coefficients while leaving the real nominal heights, references, and safety limits unchanged.

The controller receives a mixture of the two outputs:

$$y_k^c = \alpha y_k^r + (1 - \alpha) y_k^f, \qquad 0 \leq \alpha \leq 1, \tag{5.10}$$

where α is the real-measurement weight. Accordingly, $\alpha = 0$ gives fake-only feedback, whereas the real-only baseline contains no surrogate and uses $y_k^c = y_k^r$. Both mixed configurations in final validation use $\alpha = 0.75$, giving 75% real and 25% surrogate measurements. Because the attacker observes records formed with the same mixing rule, the choice of α affects both control and identification. Section 5.4.2 explains how disturbance recovery determines this weight.

5.3.3 Controller and trusted detector

As Figure 5.1 shows, control and detection use different measurement routes. The operating controller combines Kalman state estimation with reference-tracking LQR. Let $r_k \in \mathbb{R}^4$ be the reference in height deviations, $\widehat{x}_k^{\text{op}}$ the operating observer estimate, K the state-feedback

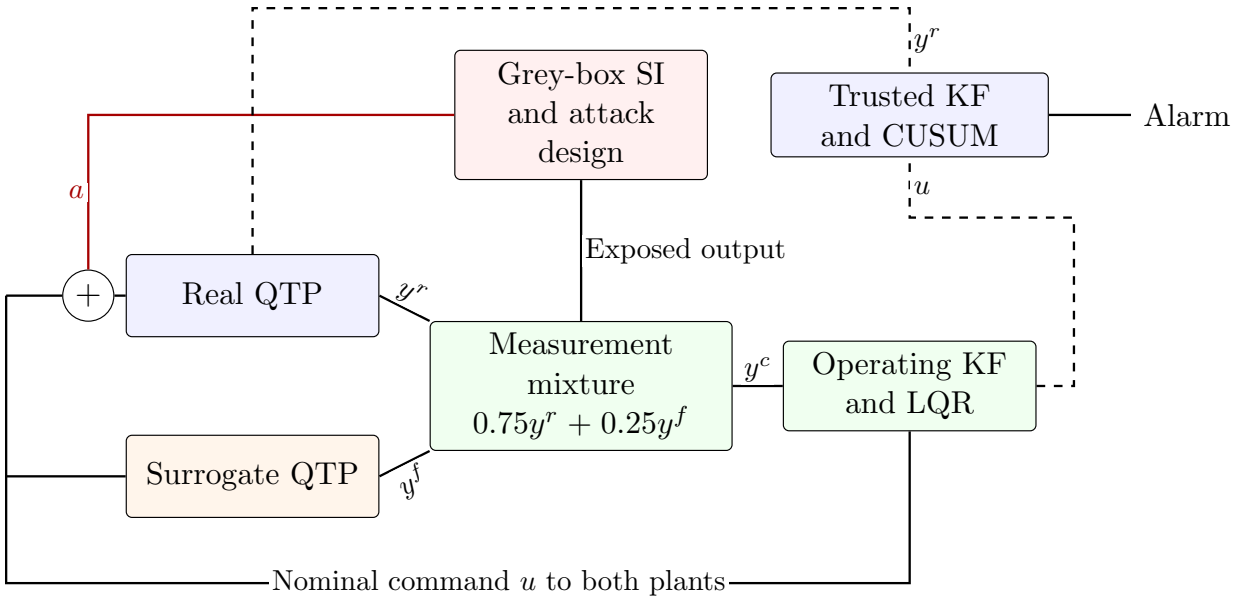


Figure 5.1: Mixed-measurement architecture. The operating Kalman filter (KF) and linear quadratic regulator (LQR) receive $y^c = 0.75y^r + 0.25y^f$, while the trusted KF and cumulative sum (CUSUM) detector receive real measurements and the known command. Both plants receive u, but the attack a reaches only the real plant. The attacker learns from exposed input–output records without access to the trusted channel.

gain, and u_k^{ff} the reference feedforward term. The command is

$$u_k = -K\widehat{x}_k^{\mathrm{op}} + u_k^{\mathrm{ff}}. \tag{5.11}$$

The operating observer receives y_k^c, while the first-move gain from the fixed-horizon control plan remains unchanged. The controller and observers retain the true nominal model throughout the comparison, so they are not redesigned for each surrogate. Only Tanks 1 and 2 have nonzero tracking weights; consequently, the upper-tank reference lines are not direct tracking objectives. The final experiment uses $r_k = 0$, corresponding to the physical-height reference h_0.

Detection, however, relies on a separate trusted observer that receives the real measurements and nominal command. Its estimate is used only for detection and does not correct the operating observer, and the unknown attack is not supplied as an input. Let $\widehat{x}_{k|k-1}^{\mathrm{mon}}$ and

$P^{\text{mon}}_{k|k-1}$ be its predicted state and covariance. Its innovation and innovation covariance are

$$e_k = y^r_k - C_T \widehat{x}^{\text{mon}}_{k|k-1}, \tag{5.12}$$

$$\Sigma_k = C_T P^{\text{mon}}_{k|k-1} C_T^\top + R_v. \tag{5.13}$$

The innovation is then whitened as $\widetilde{e}_k = L_k^{-1} e_k$, where $\Sigma_k = L_k L_k^\top$ is a Cholesky factorization.

The monitor accumulates this signal through eight directional CUSUM statistics based on Gaussian mean shifts [73]. For direction $\ell = 0, \ldots, 7$, the unit vector is $d_\ell = [\cos(\ell\pi/4), \sin(\ell\pi/4)]^\top$. With design shift δ, the statistic is

$$G_{\ell,k} = \max\left\{0, G_{\ell,k-1} + \delta d_\ell^\top \widetilde{e}_k - \frac{\delta^2}{2}\right\}. \tag{5.14}$$

An alarm is recorded when $\max_\ell G_{\ell,k}$ exceeds the threshold b after the warmup period. Because the trusted measurement route and detector design remain fixed across configurations, every learned attack is evaluated against the same detection arrangement. The security endpoint is the first eligible alarm.

5.3.4 Learned zero-dynamics attack

Although the detector retains real measurements, the attacker has access only to the exposed input–output records. It knows the QTP equation structure and a ±10% parameter prior centered on the true nominal vector, and it fits a model using bounded grey-box system identification (SI) as done by the authors in [61]. The real-only records contain real outputs. In a mixed configuration, however, the same nominal input drives both branches; their independently noisy outputs are then combined, and a single QTP model is fitted to the resulting mixture. The attacker is not given an additional unmixed real-measurement channel or the trusted detector's residuals.

Once identification is complete, the fitted matrices $\widehat{A}, \widehat{B}, \widehat{C}$ are used to construct an

actuator zero-dynamics attack (ZDA). The ideal zero-dynamics relation is

$$\begin{bmatrix} \widehat{A} - \lambda I & \widehat{B} \\ \widehat{C} & 0 \end{bmatrix} \begin{bmatrix} q \\ g_a \end{bmatrix} = 0, \tag{5.15}$$

where I is an identity matrix, λ is a transmission zero, q is a nonzero internal-state direction, and g_a is its corresponding input direction. Under this model relation, the internal trajectory can evolve while the model output is canceled [16]. However, finite-time initialization and differences between the learned and real plants can prevent exact cancellation. The attack must therefore be evaluated on the real plant to determine whether it causes damage before detection.

5.3.5 Damage before alarm

To measure that outcome, let t_f be the first physical-limit violation and t_a the first detector alarm in an episode. A missing failure or alarm is assigned $+\infty$. For an eligible episode, DBA is

$$I_{\mathrm{DBA}} = \mathbf{1}\{t_f < t_a\}, \tag{5.16}$$

where $\mathbf{1}\{\cdot\}$ equals one when its condition holds. Because the inequality is strict, a tie is not DBA. A zero outcome consequently describes the observed horizon rather than a guarantee about later behavior.

The eligibility rule excludes false alarms in which the detector detects an event prior to an attack, so the observed rate is the ratio of eligible DBA events to eligible attack episodes. Reporting this conditional rate requires retaining its denominator. DBA serves as the security objective, while normal-operation tracking, physical safety, and disturbance recovery constrain the surrogate design.

Table 5.1: Settings for measurement-weight selection and final validation. The surrogate-design bound and attacker prior serve different purposes.

Quantity	Setting
Sample interval and horizon	0.5 s and 750 s
Real nominal heights	$[12.6, 13.0, 4.8, 4.9]^\top$ cm
Real tank areas	$[28, 32, 28, 32]^\top$ cm^2
Process-noise covariance	$Q_w = 10^{-4} I_4$ cm^2 per step
Sensor-noise covariance	$R_v = 10^{-4} I_2$ V^2 per sample
Control weights	$Q_c = 0.25\,\mathrm{diag}(1, 1, 0, 0)$; $R_c = 10^{-3} I_2$
Controller	Finite-horizon LQR with Kalman estimation; fixed nominal design
Trusted detector	Eight-direction CUSUM; real measurements; 50-s warmup
Physical operating range	Strictly between 0 and 20 cm in all four tanks
Attacker parameter prior	$\pm 10\%$ around the true nominal parameters
Final mixed-feedback weight	$\alpha = 0.75$
Surrogate-design bound	$\rho = 0.50$, with $z \in [-1, 1]^{17}$
Tracking limit	$J_{\max} = 0.177009$ cm
Recovery test	Separate ± 4-cm jumps in Tank 1 or Tank 2 at 250 s
Recovery runs and dwell	50 runs per scenario; 30-s dwell in the error band
Final validation	5 banks, 100 SI models per bank, 5 attacks per model

5.4 Experimental Setup

The design requires two choices: how much real information to retain in the measurement mixture, and how to perturb the surrogate parameters. Because the first choice determines the controller's response to real disturbances, we select the measurement weight before optimizing the surrogate. We then fix both choices and evaluate the resulting configuration against two baselines. Supporting trajectory studies explain the control limitations behind this sequence, but their runs are separate from the final DBA comparison.

5.4.1 Plant, controller and detector settings

All final configurations use the linearized QTP model from Section 5.3, a 0.5-s sample interval, and a 750-s horizon. The reference is zero in deviation coordinates, corresponding to $h_0 = [12.6, 13.0, 4.8, 4.9]^\top$ cm, and the controller assigns tracking weights to Tanks 1 and 2. Table 5.1 gives the remaining settings.

The real and surrogate branches receive the same nominal command, although their

process and measurement noises are independent within each run. Across configurations, the real plant, controller design, trusted measurement route, detector, attacker prior, and attack construction remain fixed. Therefore, the comparison changes the exposed measurements and surrogate dynamics within a common operating and attack protocol.

5.4.2 Disturbance recovery and measurement-weight selection

To select the mixture before introducing parameter mismatch, the recovery experiment uses the unperturbed surrogate, $z = 0$, and tests $\alpha \in \{0, 0.10, 0.25, 0.50, 0.75, 1\}$. Four scenarios are evaluated separately: a -4-cm or $+4$-cm jump in Tank 1, and a -4-cm or $+4$-cm jump in Tank 2. Each scenario contains 50 runs, with the real state disturbed at 250 s while the surrogate remains undisturbed. This arrangement directly tests whether the controller can respond to information originating in the real plant.

Recovery time is measured from the disturbance to the first return to the $J_{\max}$ tracking-error band that is maintained for 30 s. For each scenario, we take the 95th percentile, abbreviated p95, across its runs. We then retain the larger p95 time across the positive and negative disturbances for each tank, so the slower-disturbance direction determines whether a weight passes.

Let $T_i(\alpha)$ denote this worst p95 recovery time for a disturbance in Tank i, where $i \in \{1, 2\}$. The selection rule is

$$\alpha^* = \min \{\alpha \in \{0.10, 0.25, 0.50, 0.75\} : T_i(\alpha) \leq 1.25 T_i(1),\ i = 1, 2\}. \tag{5.17}$$

The endpoints provide fake-only and real-only references, whereas selection is restricted to interior values to retain both measurement sources. The factor 1.25 allows recovery to take at most 25% longer than the matched real-only response; it does not define a percentage height-error band. Among the weights that meet both limits, the smallest real-measurement contribution is selected. Thus, the choice is based on a control requirement, independently of DBA ranking, and remains fixed throughout optimization and final validation.

5.4.3 Supporting disturbance and perturbation studies

The formal selection rule quantifies recovery, while supporting trajectories help explain why real feedback is needed. These studies compare real-only, fake-only and mixed feedback for height jumps in measured Tank 1, then examine disturbances in unmeasured Tank 3 and a plant with smaller outlet areas. Together, they illustrate the difference between passive drainage and a response informed by real measurements. Their reference lines and disturbance schedules belong to those separate studies, rather than to the four-scenario selection experiment.

A further screen examines the effect of parameter mismatch under fake-only feedback. It uses one fixed random direction and the same noise realization at $\rho \in \{0, 0.10, 0.20, 0.30\}$, checking whether every real tank remains within the physical range. Holding the direction fixed makes the effect of increasing its magnitude visible. However, because only one direction and one feedback arrangement are tested, the screen does not establish a maximum safe bound for other surrogate designs.

5.4.4 Surrogate optimization and final configurations

Once recovery-based selection fixes $\alpha = 0.75$, the remaining design variable is the surrogate vector z. We use CMA-ES, a derivative-free method that adapts a distribution of candidate parameter vectors [62], to search over $z \in [-1, 1]^{17}$ at $\rho = 0.50$. The design problem is

$$\min_{z \in [-1,1]^{17}} \widehat{P}_{\mathrm{DBA}}(z) \quad \text{subject to tracking, safety, and recovery requirements,} \tag{5.18}$$

where $\widehat{P}_{\mathrm{DBA}}(z)$ is the estimated eligible-attack DBA probability used during candidate evaluation.

Because a low DBA rate is useful only when the plant remains operable, candidate evaluation also checks normal-operation tracking against $J_{\max}$, trajectories against the physical limits, and recovery under the dedicated disturbances. These are design constraints, while DBA remains the security objective. Furthermore, the componentwise parameterization al-

Table 5.2: Configurations compared in final validation. The trusted detector receives real measurements in every configuration.

Configuration	Controller and SI measurements	Surrogate parameters
Real only	y^r	No surrogate
Mixed, $z = 0$	$0.75y^r + 0.25y^f$	Same parameters as the real plant; independent noise
Mixed, CMA	$0.75y^r + 0.25y^f$	One fixed vector z^* with $\rho = 0.50$

lows different signs and magnitudes within the same 50% envelope, so the search need not move every parameter equally.

The resulting vector, z^*, is fixed before final validation and is used in all five optimized validation banks. Here, *optimized* refers to the solution obtained by the search and does not imply a global minimum. To assess the value of this vector, Table 5.2 defines two baselines alongside the optimized configuration. Real-only feedback establishes performance without a surrogate, while mixed $z = 0$ tests the addition of a surrogate without changing its nominal dynamics. Comparing the two mixed configurations then measures the additional effect of parameter selection at the same weight.

Although the defender searches a 50% parameter envelope, the attacker's identification prior remains fixed at $\pm 10\%$. The two bounds therefore describe different parts of the threat model: the defender's design freedom and the attacker's assumed parameter range. Table 5.5 gives the selected surrogate values.

5.4.5 Final identification and attack evaluation

With the configurations fixed, each is evaluated across five validation banks. A bank contains 100 independently identified attacker models, and each learned model is used in five attack simulations. Consequently, each configuration has 500 fitted models and 2,500 raw attack simulations.

An attack is initialized at 250s and enters the real-plant input at the next state update, while the surrogate continues to receive only the nominal command. DBA is then determined

by the strict event ordering in Eq. (5.16). After excluding false alarms according to the evaluation rule, the eligible counts are 2,499 for real only, 2,498 for mixed $z = 0$, and 2,494 for mixed CMA. The corresponding exclusions for 1, 2, and 6 are retained in the reporting so that the DBA rates use their actual denominators.

5.4.6 Normal-operation utility

Attack evaluation measures security, but it does not establish whether a configuration tracks the reference during ordinary operation. We therefore evaluate utility in separate no-attack episodes. For episode j, let N be the number of evaluated samples and $r_{k,i}$ the reference deviation for lower Tank i. Its root mean square tracking error is

$$J_j = \sqrt{\frac{1}{2N} \sum_{k=1}^{N} \sum_{i=1}^{2} (x^r_{j,k,i} - r_{k,i})^2}. \tag{5.19}$$

The overall tracking value is the reported average episode RMSE for the configuration. To keep the acceptance criterion fixed, its limit is set from the reference evaluation:

$$J_{\max} = 1.5 J_0 = 0.1770089245 \text{ cm}, \qquad J_0 = 0.1180059497 \text{ cm}. \tag{5.20}$$

The limit is therefore not recalculated from the final real-only result. Real and surrogate trajectories are also checked against the physical range, because an acceptable average tracking error alone cannot establish safety.

5.4.7 Statistical analysis

For configuration m, let D_m be its eligible DBA count and n_m its eligible attack count. The pooled estimate is $\widehat{p}_m = D_m/n_m$. Following the final-results analysis, its 95% rate interval is

$$\widehat{p}_m \ \pm \ 1.96 \sqrt{\frac{\widehat{p}_m(1 - \widehat{p}_m)}{n_m}}. \tag{5.21}$$

This is a Wald interval. To compare configurations i and j, the two-sided pooled two-proportion statistic is

$$Z_{ij} = \frac{\widehat{p}_i - \widehat{p}_j}{\sqrt{\widehat{p}(1-\widehat{p})(1/n_i + 1/n_j)}}, \qquad \widehat{p} = \frac{D_i + D_j}{n_i + n_j}. \tag{5.22}$$

The uppercase Z_{ij} distinguishes the test statistic from the design vector z. The unadjusted 5% decision rule is $|Z_{ij}| > 1.96$. Mixed CMA against each baseline forms the two primary comparisons, while mixed $z = 0$ against real only is secondary.

These endpoints answer different questions: the pooled rate describes the frequency of DBA across eligible attacks, while the SI-level rate describes how widely observed vulnerability is distributed across learned models. Counting each model once avoids treating its five repetitions as independent in the secondary analysis, although its tests still assume independence between SI models. Bank-level results provide further detail on variation across the validation groups, and all reported tests are two-sided and unadjusted for multiple comparisons.

For the SI-level vulnerability rate in Table 5.8, we report a 95% Wilson score confidence interval using one binary outcome per identified model and 500 models per configuration. A model is counted as vulnerable when at least one of its five attack repetitions produces an eligible DBA event. These intervals assume independence between identified models and describe vulnerability under the fixed five-attack evaluation protocol.

For mean tracking RMSE and held-out prediction metrics, we report two-sided 95% Student-t confidence intervals,

$$\bar{x} \pm t_{0.975,n-1}\frac{s}{\sqrt{n}},$$

where $\bar{x}$ and s are the sample mean and sample standard deviation, and $t_{0.975,n-1}$ is the 97.5th percentile of the Student-t distribution with $n-1$ degrees of freedom. Tracking intervals in Table 5.10 and Figure 5.6 use $n = 250$ no-attack episodes (50 per bank), with one RMSE value per episode. Prediction intervals in Table 5.12 use $n = 500$ identified models, with one held-out metric value per model. These intervals assume independent evaluation episodes or

models; they are computed from the individual observations, rather than from the five bank means.

5.5 Results

The results follow the design sequence described in Section 5.4. We first establish the measurement weight from disturbance recovery, then examine the selected perturbation and its DBA outcomes. Finally, tracking and bank-level results show the control cost and variation associated with the security result.

5.5.1 Disturbance recovery and the selected measurement weight

Table 5.3 and Figure 5.2 report the formal recovery experiment. The real-only reference has worst p95 times of 154.5 s for Tank 1 and 170 s for Tank 2. Therefore, applying the 25% allowance gives limits of 193.125 s and 212.5 s, respectively.

Table 5.3: Measurement-weight selection with the unperturbed surrogate. Times are the worse p95 values across positive and negative disturbances. The limits are 193.125 s for Tank 1 and 212.5 s for Tank 2.

α	Tank 1 p95 (s)	Tank 2 p95 (s)	Recovery decision
0	196.5	284	Fake-only reference
0.1	193	276.5	Does not pass
0.25	187.5	264	Does not pass
0.5	176	240	Does not pass
0.75	161	209	Selected
1	154.5	170	Real-only reference

Among the tested interior weights, $\alpha = 0.75$ is the smallest that satisfies both limits. Its Tank-1 time is 161 s, or 4.2% above the real-only value, while its Tank-2 time is 209 s, or 22.9% above the real-only value. By comparison, $\alpha = 0.50$ meets the Tank-1 limit but requires 240 s for Tank 2 and therefore fails the joint rule. Consequently, both final mixed configurations retain 75% real and 25% surrogate measurements, with the weight chosen independently of the attack results.

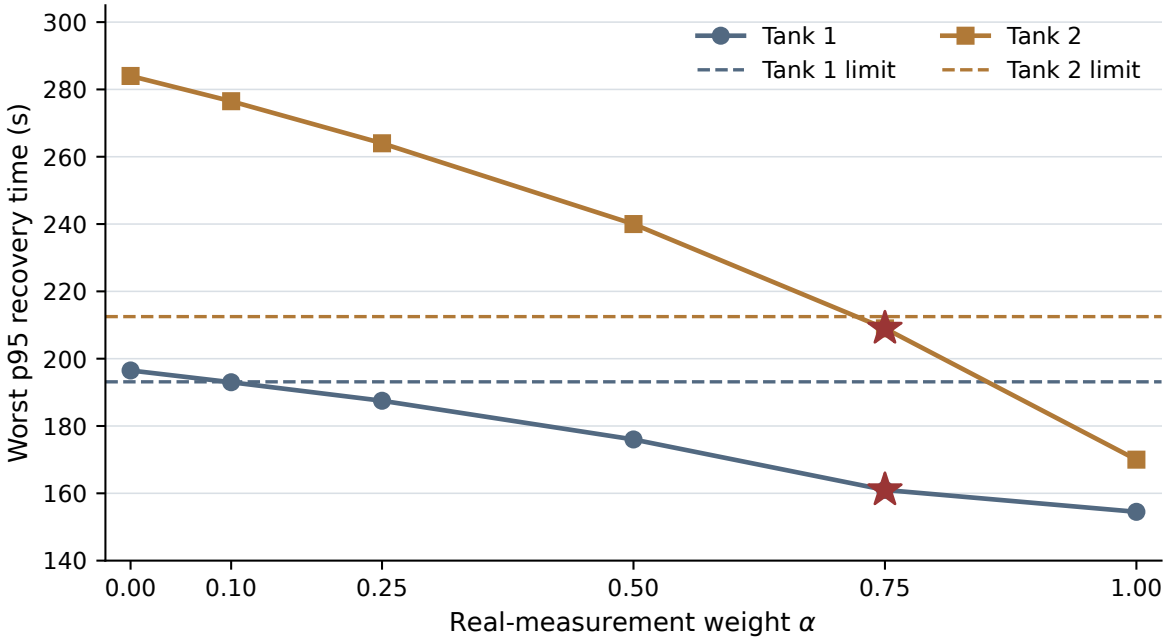


Figure 5.2: Worst p95 recovery times across separate positive and negative disturbances in each measured tank. Dashed lines are 25% above the matched real-only values, while stars identify the selected weight, $\alpha = 0.75$.

The fake-only endpoint helps explain this choice. Although recovery occurs at $\alpha = 0$ through the stable plant dynamics, the real disturbance does not directly enter the controller, and the p95 times reach 196.5 s for Tank 1 and 284 s for Tank 2. Increasing the real-measurement weight reduces these times across the tested grid. Thus, retaining real feedback improves the response, while the selection rule determines how much of that feedback is required for the declared recovery margin.

5.5.2 Physical constraints and arbitrary perturbations

Selecting α addresses feedback visibility, but it does not establish that every surrogate parameter vector is usable. The true plant and optimized surrogate share the same structure; nevertheless, these structural checks must be considered alongside the actual physical trajectories.

The fixed-direction screen in Table 5.4 illustrates this distinction. Under fake-only feedback, the tested $\rho = 0$ and $\rho = 0.10$ trajectories remain inside the physical range. However,

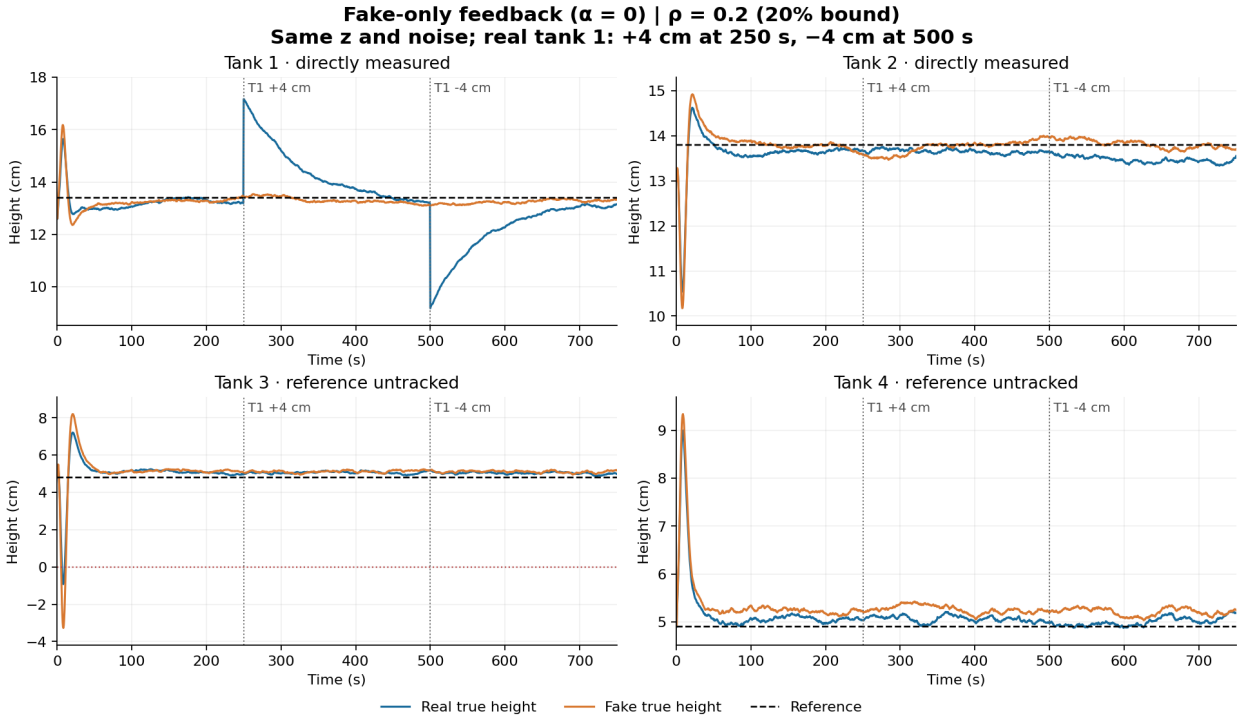


Figure 5.3: Supporting fixed-direction screen with fake-only feedback and $\rho = 0.20$. Real Tank 3 crosses below 0 cm at 7 s, before the Tank-1 disturbance. The screen uses a different feedback arrangement from the final mixed configuration.

at $\rho = 0.20$, real Tank 3 first crosses below zero at 7 s and reaches approximately -0.922 cm, before the external disturbance at 250 s. Figure 5.3 shows the resulting trajectory, while the $\rho = 0.30$ screen crosses the boundary even earlier.

Table 5.4: Supporting fake-only configuration for one fixed perturbation direction. Crossing times and minima concern the real plant; negative heights denote physical-limit violations in the linear simulation.

ρ	Within limits	First crossing (s)	Minimum height (cm)
0.00	Yes	–	4.800
0.10	Yes	–	4.800
0.20	No	7	-0.922
0.30	No	3.5	-85.706

These failures show why physical constraints must be checked during parameter selection. They do not establish a universal 10% bound because the screen uses one-directional fake-only feedback. A different direction can produce a different response, especially when real

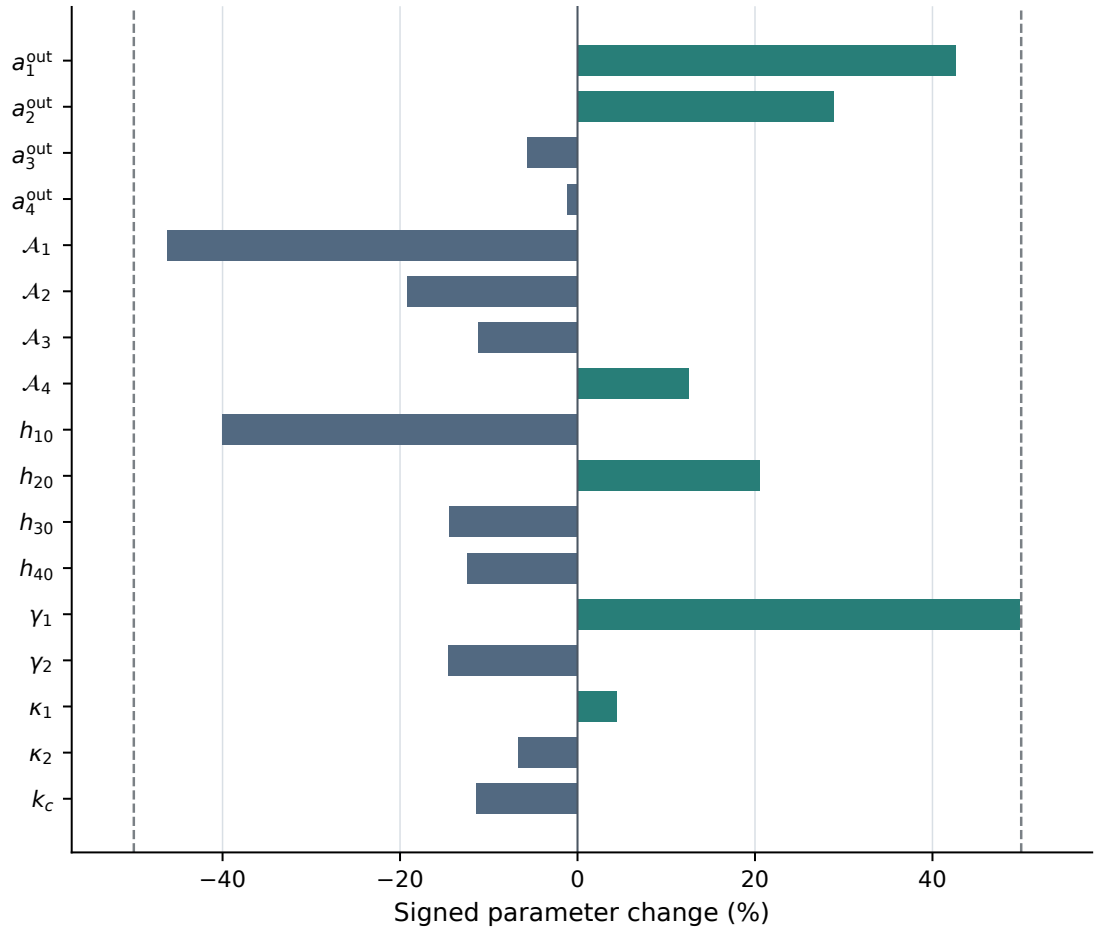


Figure 5.4: Signed relative changes in the optimized surrogate parameters. The components vary within the common ±50% envelope, whose limits are shown by dashed lines.

measurements are retained. The final design, therefore, searches within $\rho = 0.50$ at $\alpha = 0.75$ while evaluating the combined changes in parameters against tracking, safety, and recovery requirements.

5.5.3 Optimized surrogate parameters

The selected vector in Figure 5.4 illustrates how the search uses that design freedom. Its components have different signs and magnitudes: the smallest absolute change is a 1.182% decrease in the fourth outlet area, a_4^{out}, whereas the largest is a 49.875% increase in the flow fraction γ_1. Table 5.5 gives the corresponding nominal values, components of z^*, and surrogate values.

Table 5.5: Nominal and optimized surrogate parameters. Values are rounded for display; the full-precision vector is retained in the accompanying data. Gravity remains fixed at 981 $\mathrm{cm/s^2}$.

Parameter	Unit	Nominal	z_i^*	Change (%)	Optimized
a_1^{out}	cm^2	0.07123333	0.853136	+42.657	0.1016192
a_2^{out}	cm^2	0.05724806	0.577372	+28.869	0.07377476
a_3^{out}	cm^2	0.07102227	-0.114236	-5.712	0.06696561
a_4^{out}	cm^2	0.05711372	-0.023643	-1.182	0.05643857
$\mathcal{A}_1$	cm^2	28	-0.924692	-46.235	15.05431
$\mathcal{A}_2$	cm^2	32	-0.384642	-19.232	25.84574
$\mathcal{A}_3$	cm^2	28	-0.224883	-11.244	24.85164
$\mathcal{A}_4$	cm^2	32	0.250387	+12.519	36.0062
h_{10}	cm	12.6	-0.802314	-40.116	7.545423
h_{20}	cm	13	0.411622	+20.581	15.67554
h_{30}	cm	4.8	-0.289732	-14.487	4.104643
h_{40}	cm	4.9	-0.249348	-12.467	4.289097
γ_1	–	0.43	0.997507	+49.875	0.6444641
γ_2	–	0.34	-0.291366	-14.568	0.2904678
κ_1	$\mathrm{cm}^3/(\mathrm{V\ s})$	3.14	0.089773	+4.489	3.280943
κ_2	$\mathrm{cm}^3/(\mathrm{V\ s})$	3.29	-0.134323	-6.716	3.069039
k_c	V/cm	0.5	-0.227848	-11.392	0.443038

The common bound therefore allows substantial changes in some parameters and small changes in others. However, the evaluation concerns the selected vector as a whole; it does not isolate the effect of one parameter or establish that every change is individually necessary. To assess the resulting design, this same vector is retained across all five optimized validation banks.

5.5.4 Pooled damage before alarm

With the weight and parameter vector fixed, Table 5.6 compares the final eligible-attack outcomes. Real-only feedback produces 131 DBA events among 2,499 eligible attacks, giving 5.2421%. Adding an unperturbed surrogate increases the observed rate to 7.8863%, with 197 events among 2,498 eligible attacks. In contrast, the optimized mixed configuration produces 14 events out of 2,494 eligible attacks, corresponding to 0.5613

Compared with real-only feedback, the optimized configuration reduces the observed rate by 4.681 percentage points, or 89.29%. Relative to mixed $z = 0$, the reduction is 7.325

Table 5.6: Final pooled DBA counts, rates and 95% Wald intervals.

Configuration	DBA events	Eligible attacks	DBA rate	95% CI
Real only	131	2,499	5.2421%	4.368–6.116%
Mixed, $z = 0$	197	2,498	7.8863%	6.829–8.943%
Mixed, CMA	14	2,494	0.5613%	0.268–0.855%

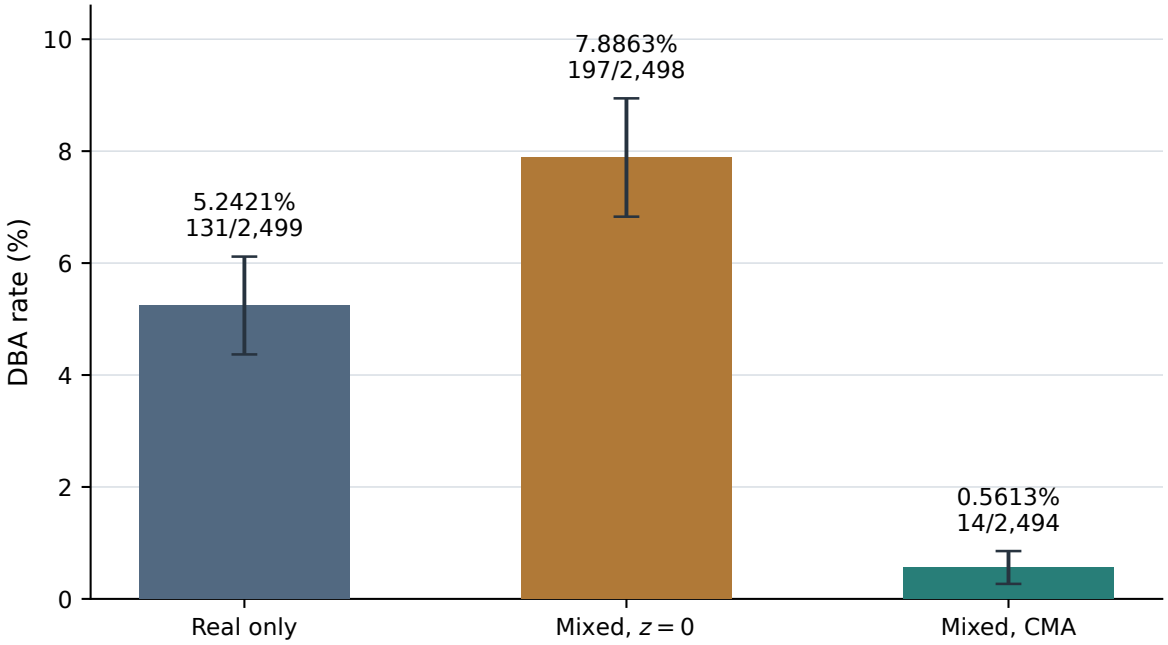


Figure 5.5: Pooled DBA rates with 95% Wald intervals. Labels show DBA events divided by eligible attacks, and both mixed configurations use $\alpha = 0.75$.

percentage points, or 92.88%. The corresponding pooled statistics, −9.85 and −12.86, both exceed the unadjusted 1.96 decision threshold under the independent-episode approximation. Table 5.7 summarizes these comparisons.

Table 5.7: Pooled two-proportion comparisons. A difference is the first named configuration minus the second, in percentage points (pp). All three cross the unadjusted $|Z| > 1.96$ threshold. The first two comparisons are primary.

Comparison	Difference (pp)	Z	Test decision
Mixed, CMA − Real only	-4.681	-9.85	Reduction
Mixed, CMA − Mixed, $z = 0$	-7.325	-12.86	Reduction
Mixed, $z = 0$ − Real only	+2.644	+3.77	Increase

The secondary comparison also crosses the decision threshold, but in the opposite direction: mixed $z = 0$ is 2.644 percentage points above real only, with $Z = 3.77$. Thus, introducing an unperturbed surrogate does not reduce DBA in this experiment. The lower rate appears after optimizing its dynamics within the same mixed-feedback architecture.

5.5.5 SI-level vulnerability

The pooled rate describes attack outcomes but does not indicate how many learned models produce them. Therefore, we also count an SI model as vulnerable when at least one of its five attack repetitions produces an eligible DBA event. Each model contributes one binary result, as defined in Section 5.4.

Table 5.8: SI-level vulnerability in final validation. A model is counted as vulnerable when at least one of its five attack repetitions produces an eligible DBA event. Brackets give 95% Wilson confidence intervals for the vulnerable-model rate, based on 500 SI models per configuration.

Configuration	Vulnerable SI models	Rate (%)
Real only	41/500	8.2 [6.10, 10.94]
Mixed, $z = 0$	64/500	12.8 [10.15, 16.01]
Mixed, CMA	5/500	1.0 [0.43, 2.32]

Table 5.8 shows that 41 of 500 learned models are vulnerable under real-only feedback, compared with 64 under mixed $z = 0$ and five under mixed CMA. Accordingly, the observed

vulnerable-model rate falls from 8.2% and 12.8% in the two baselines to 1.0% with the optimized surrogate. The reductions are 7.2 and 11.8 percentage points, corresponding to relative reductions of 87.80% and 92.19%, respectively.

Table 5.9: SI-level two-proportion comparisons. Differences are the first configuration minus the second; pp denotes percentage points. All three cross the unadjusted two-sided threshold $|Z| > 1.96$.

Comparison	Model count	Rate (pp)	Z	Test decision
Mixed, CMA − Real only	-36	-7.2	-5.43	Reduction
Mixed, CMA − Mixed, $z = 0$	-59	-11.8	-7.36	Reduction
Mixed, $z = 0$ − Real only	+23	+4.6	+2.37	Increase

The model-level comparisons in Table 5.9 agree with the direction of the pooled results. Their Z statistics are -5.43 and -7.36 for mixed CMA against the two baselines, while mixed $z = 0$ against real only gives $+2.37$. Thus, the optimized configuration has fewer DBA events and fewer learned models that produce any DBA. Because each model is counted once, the secondary analysis does not treat its five repetitions as independent observations. Nevertheless, it remains an assessment under the fixed five-attack budget: a model with no observed DBA may still produce one under additional repetitions or different conditions.

5.5.6 Normal-operation tracking and recovery

The security reduction must be considered together with its control cost. Table 5.10 reports overall normal-operation RMSE values of 0.118192 cm for real only, 0.118860 cm for mixed $z = 0$, and 0.175451 cm for mixed CMA. Therefore, optimization adds about 0.056591 cm relative to the unperturbed mixed configuration. Although the resulting value remains below $J_{\max} = 0.177009$ cm, its margin is only about 0.001558 cm, or 0.88%. The 95% confidence interval for the optimized mean is [0.173773, 0.177128] cm and therefore extends slightly above $J_{\max}$. Thus, the observed mean satisfies the stated utility criterion, although the entire two-sided confidence interval does not lie below the limit.

Normal-operation real and surrogate trajectories remain inside the 0–20 cm physical range in the final results summary. In addition, the optimized configuration satisfies the four

Table 5.10: Overall normal-operation tracking and worst p95 recovery times from the dedicated disturbance experiments. Brackets below each mean RMSE give its 95% Student-t confidence interval, based on 250 no-attack episodes per configuration. The RMSE limit is 0.177009 cm; Tank-1 and Tank-2 recovery limits are 193.125 s and 212.5 s.

Configuration	Overall RMSE (cm)	Tank 1 p95 (s)	Tank 2 p95 (s)
Real only	0.118192 [0.116107, 0.120277]	154.5	170
Mixed, $z = 0$	0.118860 [0.116996, 0.120724]	161	209
Mixed, CMA	0.175451 [0.173773, 0.177128]	154	192.5

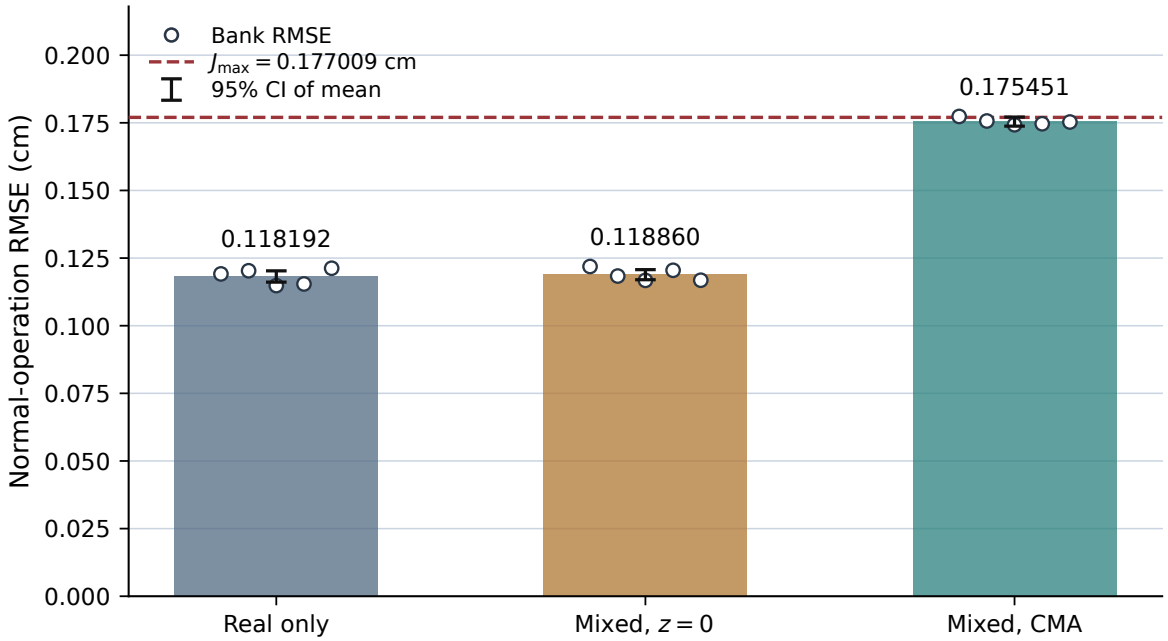


Figure 5.6: Overall normal-operation RMSE with 95% Student-t confidence intervals for the mean, calculated from 250 no-attack episodes per configuration. Open circles show the five bank means, each based on 50 episodes, and the dashed line marks $J_{\max}$. The optimized mean is below the limit, but its upper confidence bound and the Bank-0 mean are slightly above it.

declared disturbance-recovery scenarios: its worst p95 times are 154 s for Tank 1 and 192.5 s for Tank 2, below the respective limits of 193.125 s and 212.5 s. These times are also lower than the 161s and 209s obtained with the unperturbed surrogate at the same measurement weight. Thus, the higher normal-operation RMSE is accompanied by acceptable recovery under the tested disturbances.

The aggregate tracking result, however, conceals a small bank-level exceedance. Optimized Bank 0 has an RMSE of 0.177350 cm, approximately 0.000341 cm above the limit, whereas the other four optimized banks are below it. Therefore, the utility conclusion applies to the overall result; it does not mean that every validation bank passes the RMSE threshold.

5.5.7 Variation across validation banks

To show this variation alongside the security outcomes, Table 5.11 reports all five banks for each configuration. Real-only DBA rates range from 1.8% to 6.814%, while mixed $z = 0$ rates range from 6.613% to 9.018%. The optimized banks range from 0% to 1.406%, with two recording no DBA events in 500 eligible attacks each.

The low optimized rates therefore appear across the validation groups rather than arising solely from one favorable bank. However, a bank with zero observed events does not imply an underlying probability of zero. Pooling all banks retains this variation in the aggregate count, while the separate bank results show the conditions under which that aggregate was obtained.

5.6 Discussion

5.6.1 Effect of optimizing the surrogate

The baseline comparisons show why the surrogate dynamics are part of the defense. At the same weight, $\alpha = 0.75$, the observed DBA rate falls from 7.8863% with the unperturbed surrogate to 0.5613% with the optimized vector, below the 5.2421% real-only rate. The

Table 5.11: Final bank-level DBA and normal-operation tracking results. Banks are numbered 0–4. Optimized Bank 0 is slightly above the RMSE limit; the optimized aggregate is below it.

Configuration	Bank	DBA/eligible	DBA rate (%)	RMSE (cm)
Real only	0	9/500	1.800	0.119149
Real only	1	34/499	6.814	0.120315
Real only	2	32/500	6.400	0.114827
Real only	3	33/500	6.600	0.115429
Real only	4	23/500	4.600	0.121240
Mixed, $z = 0$	0	36/500	7.200	0.121890
Mixed, $z = 0$	1	45/499	9.018	0.118360
Mixed, $z = 0$	2	42/500	8.400	0.116757
Mixed, $z = 0$	3	33/499	6.613	0.120481
Mixed, $z = 0$	4	41/500	8.200	0.116812
Mixed, CMA	0	0/500	0.000	0.177350
Mixed, CMA	1	0/500	0.000	0.175678
Mixed, CMA	2	2/498	0.402	0.174275
Mixed, CMA	3	7/498	1.406	0.174650
Mixed, CMA	4	5/498	1.004	0.175300

SI-level analysis follows the same pattern, with fewer learned models producing any DBA. Taken together, these results support parameter selection within the mixed architecture and show that measurement mixing alone is insufficient in the tested setting.

This pattern is consistent with the intended mechanism. When $z = 0$, the surrogate retains the real plant's nominal dynamics, although independent noise changes the exposed records. Once its parameters are perturbed, the mixture also contains a different response, which can change the fitted model and the attack constructed from it. However, the controller receives that same mixture, so its closed-loop behavior changes as well. Therefore, the DBA comparison measures the combined effect on identification and control, and cannot attribute the reduction solely to identification error.

5.6.2 Identification accuracy with an unperturbed surrogate

The higher DBA rate with the unperturbed surrogate can be examined through the accuracy of the attacker's identified models. When $z = 0$, the real and surrogate plants have the same nominal dynamics and receive the same input during identification. Their noise-free

Table 5.12: Mean held-out prediction accuracy against the real plant, averaged over 500 identified models per configuration. Brackets below each mean give its 95% Student-t confidence interval across models. Prediction RMSE is reported in sensor-output units.

Configuration	Prediction fit (%)	Prediction RMSE
Real only	90.30	0.06089
	[90.03, 90.58]	[0.05936, 0.06241]
Mixed, $z = 0$	91.51	0.05414
	[91.27, 91.74]	[0.05275, 0.05553]

responses are therefore identical, while their process and measurement noise realizations are independent. Consequently, mixing their measurements preserves the underlying response while averaging the noise.

For one output channel, let $\eta_{\mathrm{r},k}$ and $\eta_{\mathrm{f},k}$ denote the noise-induced deviations from the common noise-free response at time k. Since these deviations are independent and have the same variance, denoted by σ_k^2, the variance after mixing is

$$\begin{aligned} \mathrm{Var}(\alpha\eta_{\mathrm{r},k} + (1-\alpha)\eta_{\mathrm{f},k}) &= \left[\alpha^2 + (1-\alpha)^2\right]\sigma_k^2 \\ &= 0.625\,\sigma_k^2, \qquad \alpha = 0.75. \end{aligned} \tag{5.23}$$

Thus, the expected noise variance in the identification output is 37.5% lower than in the real-only case. Under these conditions, the unperturbed mixture can provide cleaner observations of the real plant's dynamics.

The saved identification results support this explanation. For each identified model, prediction accuracy is evaluated by applying the same held-out input to that model and the real-plant model. These inputs are not used for fitting, and both evaluation responses are simulated without noise. Prediction fit measures the output error relative to the variation in each real output, with a higher percentage indicating a better match. Prediction RMSE measures the root mean square output error in the original measurement units, with a lower value indicating more accurate predictions. These identification measures differ from tracking RMSE, which compares the controlled tank heights with their references.

As shown in Table 5.12, the unperturbed mixed configuration improves mean prediction

fit from 90.30% to 91.51% and reduces mean prediction RMSE by approximately 11.1%. This provides evidence that the attacker learns a more accurate representation of the real plant's input–output response on average. Such an improvement may help the attacker construct attacks that more closely follow the real plant's zero dynamics, allowing damage to occur before detection.

This interpretation concerns the observations used for identification. The detector continues to receive trusted real measurements, so measurement mixing does not directly introduce surrogate noise into its input. The findings therefore support improved identification as a possible contributor to the higher DBA rate in Case 2. However, output-prediction accuracy alone does not establish the accuracy of the learned zero dynamics, and the experiments change both identification data and controller feedback. Accordingly, these results do not isolate the contribution of identification accuracy to the observed DBA increase.

5.6.3 Recovery-based selection and tracking cost

The recovery criterion connects the measurement weight to the reason for retaining real feedback. A fully controllable plant can still respond slowly when the controller receives little information about a disturbance, and stable drainage can obscure this limitation under fake-only feedback. The measured-tank, unmeasured-tank, and smaller-outlet studies illustrate that distinction. For this reason, the weight is selected from the formal recovery experiment rather than from apparent convergence in a single trajectory.

The selected value, $\alpha = 0.75$, is the smallest tested interior weight for which the Tank-1 and Tank-2 p95 recovery times remain within 25% of their real-only references. It therefore satisfies the declared control requirement while retaining a surrogate contribution. However, the selection applies to the tested grid and disturbances; it does not establish the smallest feasible weight over the continuous interval or a suitable weight for every plant. With this weight fixed, optimization lowers DBA and meets the aggregate utility limit, but increases normal-operation RMSE, all while being under the limit.

5.6.4 Statistical and experimental scope

Both security endpoints are tied to their evaluation conditions. The pooled rate concerns eligible attacks over a 750-s horizon and records whether damage occurs before the first eligible alarm. An episode without DBA may therefore still experience damage after an alarm, while a zero result cannot establish safety beyond the observed horizon. The SI-level endpoint adds a second condition: vulnerability is observed after five attack repetitions for each learned model. Increasing that budget could identify additional vulnerable models.

The two statistical analyses also use different observation units. Pooled tests and Wald intervals approximate attack episodes as independent, although repetitions share a learned model. The SI-level analysis addresses this grouping by assigning a single binary result to each model, but its tests still assume independence among models. Moreover, the comparisons are unadjusted for multiple testing. Their agreement provides supporting evidence across two endpoints without removing these assumptions.

The bank results provide an additional check on the distribution of the observed effect: optimized DBA rates remain low across all five groups. Nevertheless, the banks evaluate one frozen vector, so they do not measure how consistently repeated CMA-ES searches would find a comparable solution. A separate study of repeated optimization runs would be needed to assess that question.

Finally, the experiments use a linearized QTP, a fixed controller and detector, and a trusted real-measurement channel. Recovery under the four tested disturbances and safety during normal operation establish performance within that scope. Extending the conclusion to nonlinear operation, actuator constraints, other disturbances, or hardware would require additional evaluation, as would removing the trusted-channel assumption.

5.7 Conclusion

This chapter examined how surrogate measurements can limit the effectiveness of attacks based on learned plant models. Although surrogate-only feedback changes the information

exposed to the attacker, it also prevents the controller from observing real-only disturbances. The QTP's passive recovery can conceal that limitation, which motivates retaining real measurements and choosing their weights based on an explicit recovery requirement.

The resulting design uses $\alpha = 0.75$ and selects the surrogate parameters through CMA-ES within a 50% envelope. Under the evaluated conditions, its pooled DBA rate falls to 0.5613%, compared with 5.2421% for real-only feedback and 7.8863% for the unperturbed mixed configuration. The SI-level analysis supports the same conclusion, because fewer learned models produce any DBA after surrogate optimization. Thus, the benefit is observed in both attack frequency and the spread of vulnerability across learned models.

This security benefit has a control cost. The optimized overall RMSE is 0.175451 cm, below the 0.177009 cm limit, but above both baselines and close to the acceptance boundary. Therefore, the results support surrogate-parameter selection under explicit control constraints for the stated plant, detector, and attacker.

Building on these findings, future work can examine stricter tracking criteria, repeated optimization runs, and attackers that adapt to the measurement mixture. Evaluation with nonlinear dynamics, actuator limits, and hardware would then help determine how far the observed security–control trade-off extends beyond the present simulation.

Chapter 6

Conclusion

6.1 Overall Findings

Across the first three studies, the effectiveness of the supervisory defense depended on more than its performance against previously recorded attacks. Attack-generated measurements provided useful data to strengthen the detectors, while training new attackers against the expanded defense exposed remaining weaknesses. Therefore, the results support evaluating improvements against further adversarial adaptation rather than assuming that recognition of existing attacks establishes protection.

The studies also showed that timely detection and correct localization are separate requirements. An alarm raised after the specified physical condition does not prevent that condition, even when a later intervention limits further disruption. Likewise, correct evidence from an individual detector does not necessarily lead to a correct ensemble decision. The rule used to combine detector outputs can preserve or suppress that evidence.

The fourth study shows that the information used for attack construction is also part of the defense. Optimizing a surrogate within a mixed-measurement architecture reduces damage before alarm under the stated attacker model, whereas adding an unperturbed surrogate does not. These findings link attack resistance to the need to retain real feedback and to satisfy explicit control requirements.

The overall conclusion is that cyber-physical defenses should be evaluated by considering the relationships among physical behavior, available measurements, and supervisory decisions. This perspective connects the three studies without treating their attack objectives or success criteria as interchangeable.

6.2 Contributions

The four contributions concern learning-based attack construction, adaptive detection in a microgrid, and transient source mislocalization. Each is supported by the chapter-specific findings summarized below.

6.2.1 Learning-Based Zero-Dynamics-Like Attacks

Chapter 2 establishes a learning-based alternative to direct model-based attack construction [5]. Under the assumed digital-twin access, the attacker learned to drive unmeasured tank states beyond the specified limits without directly using the plant's state-space matrices. Withholding those matrices alone was therefore not sufficient protection in the studied setting.

The study also connects the learned behavior to the physical mechanism of a theoretical zero-dynamics attack. Comparing the directions of induced state changes shows that successful policies can produce state evolution that aligns with the model-based construction. This comparison provides evidence beyond the reward or attack success rate. It supports the interpretation that the agent learns a related attack mechanism.

A further part of this contribution is the iterative construction of attackers and measurement-based detectors. Attack-generated histories are used to train Long Short-Term Memory (LSTM) detectors, and attackers later face the accumulated defenses. Across the successive attacker–defender configurations in Table 2.1, the reported attack success rate decreases from 70% for the initial attacker to 0.9% for the final attacker. These values describe the evolving interaction between the two sides, rather than a comparison of a single fixed attacker against

different detectors. The remaining successful attacks also show that a substantial reduction in success is not the same as complete prevention.

6.2.2 Iterative Adversarial Training for Microgrid Detection

Chapter 3 extends iterative adversarial evaluation to a three-inverter grid-forming microgrid [6]. Its contribution is to connect detector outcomes with the coupled physical response of the inverters, including the response after the compromised inverter is disconnected.

The iterative experiments demonstrate that newly generated attacks provide useful data for strengthening the defense. Later attackers must search for behavior that avoids the accumulated detectors, and the final evaluated defense suppresses successful attacks in the reported final tests. The sequence in Figure 3.2 shows why evaluating each new defense against an adaptive attacker is important: recognizing a previously generated attack does not establish resistance to a newly trained policy.

This study also clarifies the difference between detecting an attack and preventing its physical objective. Some illustrated attacks reach the specified condition before detection and disconnection. A later response may limit the subsequent disturbance, but it cannot change the ordering of those earlier events. The contribution, therefore, includes an evaluation of detection in relation to the physical process, rather than treating an eventual alarm as sufficient evidence of protection.

6.2.3 Transient Mislocalization and Detector Aggregation

The third contribution, presented in Chapter 4, is the evaluation of sustained source mislocalization during load-induced transients and the demonstration that the detector aggregation rule can materially change the security outcome.

Under the tested final configurations, strict aggregation preserved evidence of the true compromised inverter and produced zero observed successful mislocalizations for all three targets. Consensus aggregation, in contrast, retained non-zero success for some targets because confidence averaging could suppress correct minority evidence.

The result identifies a weakness that cannot be understood from detector count or individual accuracy alone. It also requires careful interpretation: strict aggregation can retain multiple candidate sources, so preserving the true source is not the same as always providing a single unambiguous localization decision. The comparison supports an adversarial evaluation of the complete aggregation procedure, rather than a claim that a single rule is best for every operating objective.

6.2.4 Utility-Constrained Mixed-Measurement Defense

Chapter 5 develops a mixed-measurement defense against zero-dynamics attacks constructed from identified plant models. The controller and attacker identification process uses a combination of real and surrogate measurements, while a separate trusted detector continues to receive real measurements. This changes the response exposed to the attacker without removing all feedback from the real plant.

A central contribution is the separation of the two design choices. The real-measurement weight is selected using disturbance recovery, independently of attack outcomes. CMA-ES then searches for surrogate parameters that reduce damage before alarm under tracking, physical-limit, and recovery requirements. The supporting experiments also distinguish passive recovery through stable plant dynamics from a controller response informed by real disturbances.

Under the final evaluated conditions, the results support surrogate-parameter selection rather than measurement mixing alone. The contribution therefore establishes a measured security–control trade-off under the stated attacker model and trusted detection arrangement.

6.2.5 Scope of the Contributions

The contributions are supported by experiments on the modeled quadruple-tank process and three-inverter microgrid. Their outcomes depend on the available measurements, attack channels, controller settings, noise assumptions, training resources, and evaluation condi-

tions. The studies identify weaknesses and demonstrate improvements within those conditions; they do not establish one universal security guarantee.

The terminology also follows the evidence. The tank study compares learned state changes with a theoretical zero-dynamics construction. The transient study establishes persistent wrong-inverter attribution, not the mathematical existence of zero dynamics. Similarly, crossing a specified simulation limit does not directly indicate irreversible hardware damage.

Finite experiments impose further limits. Zero observed successful attacks does not prove that the success probability is exactly zero, and empirical stabilization of training does not prove that the attacker or defender is globally optimal. The transient experiments also do not include a matched steady-state baseline, so they do not establish that all transient conditions are more vulnerable. Larger networks, physical implementations, and broader uncertainty settings remain directions for extending the evidence developed here.

6.3 Future Work

The results suggest several extensions that follow directly from the limitations of the studied systems and decision rules. These extensions concern the attacker's information, the defender's measurements, the way detector outputs are combined, and the physical response following a decision.

6.3.1 Imperfect System Knowledge and Model Mismatch

An important extension is to evaluate attackers whose training environment differs from the target system. The present work gives the attacker useful simulation access, but a broader evaluation should vary plant parameters, controller gains, operating conditions, and measurement availability between training and testing. This would separate sensitivity to different noise realizations from sensitivity to an inaccurate model of the process.

A related direction is to extend the system-identification setting in Chapter 5 to broader

model uncertainty and less restrictive assumptions about the attacker's identification procedure. The resulting experiments should examine both prediction quality and attack outcomes. A model that predicts normal operation well may still represent an attack-relevant part of the dynamics inaccurately; conversely, an imperfect model may retain enough information to support a successful attack. The relationship should therefore be measured rather than assumed.

The attacker's observations should also be restricted progressively. Comparing access to internal states, estimated states, and measured outputs alone would help identify which information is important for each learned strategy. Such experiments would clarify how the reported outcomes change as the informed-attacker assumption is relaxed.

6.3.2 Measurement Selection for Detection and Localization

The studies motivate examining whether a small number of additional measurements can improve the defender's decisions. For the microgrid, a useful comparison would retain the existing voltage and frequency measurements at the point of common coupling and then add selected local measurements, such as current or power from individual inverters. The aim would be to identify measurements that help distinguish the source of an event, rather than assuming that every available signal must be collected.

Evaluation should include both recorded attacks and new attackers trained against the expanded observation set. The first comparison would show whether additional measurements expose existing attack patterns. The second would test whether an adaptive attacker can find another way to avoid or mislead the improved defense.

The benefit should be assessed alongside sensing, communication, and processing requirements. An additional signal is useful when it improves the required decision under the tested conditions, not merely when it increases the input dimension of the detector.

6.3.3 Detector Diversity and Robust Aggregation

The aggregation results motivate alternatives to both strict class-preserving decisions and simple confidence averaging. Possible extensions include adjusting detector confidence using validation data, weighting detectors according to their performance under different operating conditions, and accumulating evidence over time. When the evidence is inconsistent, a system could retain several candidate sources or request additional measurements instead of immediately committing to one inverter.

These alternatives should be assessed against false alarms, localization errors, and decision delay. Preserving every isolated prediction may produce unnecessary candidates, while suppressing disagreement may remove the only correct prediction. Any delay used to obtain more evidence must also be evaluated against the time available for a safe response.

Two complementary comparisons would be useful. Applying several aggregation rules to the same frozen detector outputs would isolate the immediate effect of the rule. Training new attackers against each complete configuration would then evaluate its behavior under adaptation. The same approach could compare ensembles containing different detector families, including sequence models, statistical tests, and model-based residual detectors. A more diverse ensemble should be treated as a hypothesis to test, not as an automatic improvement.

Bibliography

[1] Stephanie Harshbarger, Mohsen Hosseinzadehtaher, Bala Natarajan, Eugene Vasserman, Mohammad Shadmand, and George Amariucai. (a little) ignorance is bliss: The effect of imperfect model information on stealthy attacks in power grids. In *2020 IEEE Kansas Power and Energy Conference (KPEC)*, pages 1–6. IEEE, 2020.

[2] Fabio Pasqualetti, Florian Dörfler, and Francesco Bullo. Attack detection and identification in cyber-physical systems. *IEEE transactions on automatic control*, 58(11): 2715–2729, 2013.

[3] André Teixeira, Iman Shames, Henrik Sandberg, and Karl H Johansson. Revealing stealthy attacks in control systems. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1806–1813. IEEE, 2012.

[4] Karl Henrik Johansson. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on control systems technology*, 8(3):456–465, 2000.

[5] Bipin Paudel and George Amariucai. Reinforcement learning approach to generate zero-dynamics attacks on control systems without state space models. In *European Symposium on Research in Computer Security*, pages 3–22. Springer, 2023.

[6] Bipin Paudel, George Amariucai, Alireza Zare, and Mohammad Shadmand. Robust detection in power systems: Iterative reinforcement learning based adversarial training. In *IECON 2024-50th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–7. IEEE, 2024.

[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic:

Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

[10] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 388:1–29, 2016.

[11] Stephanie Harshbarger. The impact of zero-dynamics stealthy attacks on control systems: stealthy attack success probability and attack prevention, 2022. URL https://krex.k-state.edu/dspace/handle/2097/42853.

[12] Souhila Aoufi, Abdelouahid Derhab, and Mohamed Guerroumi. Survey of false data injection in smart power grid: Attacks, countermeasures and challenges. *Journal of Information Security and Applications*, 54:102518, 2020.

[13] Ruilong Deng, Gaoxi Xiao, Rongxing Lu, Hao Liang, and Athanasios V Vasilakos. False data injection on state estimation in power systems—attacks, impacts, and defense: A survey. *IEEE Transactions on Industrial Informatics*, 13(2):411–423, 2016.

[14] Pritam Dash, Mehdi Karimibiuki, and Karthik Pattabiraman. Out of control: stealthy attacks against robotic vehicles protected by control-based techniques. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 660–672, 2019.

[15] André Teixeira, Daniel Pérez, Henrik Sandberg, and Karl Henrik Johansson. Attack models and scenarios for networked control systems. In *Proceedings of the 1st international conference on High Confidence Networked Systems*, pages 55–64, 2012.

[16] Andre Teixeira, Kin Cheong Sou, Henrik Sandberg, and Karl Henrik Johansson. Secure

control systems: A quantitative risk management approach. *IEEE Control Systems Magazine*, 35(1):24–45, 2015.

[17] Ruochi Zhang and Parv Venkitasubramaniam. Stealthy control signal attacks in linear quadratic gaussian control systems: Detectability reward tradeoff. *IEEE Transactions on Information Forensics and Security*, 12(7):1555–1570, 2017.

[18] Ali Sayghe, Junbo Zhao, and Charalambos Konstantinou. Evasion attacks with adversarial deep learning against power system state estimation. In *2020 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2020.

[19] Cheng Feng, Tingting Li, Zhanxing Zhu, and Deeph Chana. A deep learning-based framework for conducting stealthy attacks in industrial control systems. *arXiv preprint arXiv:1709.06397*, 2017.

[20] Zengji Liu, Qi Wang, Yujian Ye, and Yi Tang. A gan-based data injection attack method on data-driven strategies in power systems. *IEEE Transactions on Smart Grid*, 13(4): 3203–3213, 2022.

[21] Sergei K Alabugin and Alexander N Sokolov. Applying of generative adversarial networks for anomaly detection in industrial control systems. In *2020 Global Smart Industry Conference (GloSIC)*, pages 199–203. IEEE, 2020.

[22] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

[23] Sangjun Kim and Kyung-Joon Park. A survey on machine-learning based security design for cyber-physical systems. *Applied Sciences*, 11(12):5458, 2021.

[24] Jiajun Duan, Di Shi, Ruisheng Diao, Haifeng Li, Zhiwei Wang, Bei Zhang, Desong Bian, and Zhehan Yi. Deep-reinforcement-learning-based autonomous voltage control for power grid operations. *IEEE Transactions on Power Systems*, 35(1):814–817, 2019.

[25] Chong Li and Meikang Qiu. *Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies*. Chapman and Hall/CRC, 2019.

[26] Ying Chen, Shaowei Huang, Feng Liu, Zhisheng Wang, and Xinwei Sun. Evaluation of reinforcement learning-based false data injection attack to automatic voltage control. *IEEE Transactions on Smart Grid*, 10(2):2158–2169, 2018.

[27] Brian DO Anderson. Output-nulling invariant and controllability subspaces. *IFAC Proceedings Volumes*, 8(1):337–345, 1975.

[28] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[29] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[30] Saeedreza Jadidi, Hamed Badihi, and Youmin Zhang. Active fault-tolerant and attack-resilient control for a renewable microgrid against power-loss faults and data integrity attacks. *IEEE Transactions on Cybernetics*, 54(4):2113–2128, 2024. doi: 10.1109/TCYB.2023.3305240.

[31] Subham Sahoo, Sukumar Mishra, Jimmy Chih-Hsien Peng, and Tomislav Dragičević. A stealth cyber-attack detection strategy for dc microgrids. *IEEE Transactions on Power Electronics*, 34(8):8162–8174, 2018.

[32] Subham Sahoo, Jimmy Chih-Hsien Peng, Annavaram Devakumar, Sukumar Mishra, and Tomislav Dragičević. On detection of false data in cooperative dc microgrids—a discordant element approach. *IEEE Transactions on Industrial Electronics*, 67(8):6562–6571, 2019.

[33] Ali Jafarian Abianeh, Yihao Wan, Farzad Ferdowsi, Nenad Mijatovic, and Tomislav Dragičević. Vulnerability identification and remediation of fdi attacks in islanded dc

microgrids using multiagent reinforcement learning. *IEEE Transactions on Power Electronics*, 37(6):6359–6370, 2021.

[34] Mohammad Reza Habibi, Hamid Reza Baghaee, Tomislav Dragičević, and Frede Blaabjerg. Detection of false data injection cyber-attacks in dc microgrids based on recurrent neural networks. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 9(5):5294–5310, 2020.

[35] Stephanie Harshbarger, Balasubramaniam Natarajan, Eugene Y. Vasserman, Muhammad Farooq Umar, Mohammad Shadmand, and George Amariucai. On zero-dynamics stealthy attacks with learned state space models. In *2024 4th International Conference on Smart Grid and Renewable Energy (SGRE)*, pages 1–6, 2024. doi: 10.1109/SGRE59715.2024.10428796.

[36] Stephanie Harshbarger, Mohsen Hosseinzadehtaher, Bala Natarajan, Eugene Vasserman, Mohammad Shadmand, and George Amariucai. (a little) ignorance is bliss: The effect of imperfect model information on stealthy attacks in power grids. In *2020 IEEE Kansas Power and Energy Conference (KPEC)*, pages 1–6, 2020. doi: 10.1109/KPEC47870.2020.9167599.

[37] Claudio Burgos-Mellado, Claudio Zuñiga-Bauerle, Diego Muñoz-Carpintero, Yeiner Arias-Esquivel, Roberto Cárdenas-Dobson, Tomislav Dragičević, Felipe Donoso, and Alan Watson. Reinforcement learning-based method to exploit vulnerabilities of false data injection attack detectors in modular multilevel converters. *IEEE Transactions on Power Electronics*, 38(7):8907–8921, 2023. doi: 10.1109/TPEL.2023.3263728.

[38] Hossein Mohammadi Rouzbahani, Hadis Karimipour, and Lei Lei. Multi-layer defense algorithm against deep reinforcement learning-based intruders in smart grids. *International Journal of Electrical Power & Energy Systems*, 146:108798, 2023.

[39] Hu-Sheng Wu. A survey of research on anomaly detection for time series. In *2016 13th*

International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), pages 426–431. IEEE, 2016.

[40] Kamran Shaukat, Talha Mahboob Alam, Suhuai Luo, Shakir Shabbir, Ibrahim A Hameed, Jiaming Li, Syed Konain Abbas, and Umair Javed. A review of time-series anomaly detection techniques: A step to future perspectives. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 1*, pages 865–877. Springer, 2021.

[41] Hassan Ismail Fawaz. Deep learning for time series classification. *arXiv preprint arXiv:2010.00567*, 2020.

[42] Stephanie Harshbarger, Mohsen Hosseinzadehtaher, Alireza Zare, Amin Y. Fard, Mohammad B. Shadmand, and George Amariucai. Preventing stealthy attacks on power electronics dominated grids. *IEEE Access*, 11:126312–126322, 2023. doi: 10.1109/ACCESS.2023.3331314.

[43] Seyedamirabbas Mousavian, Melike Erol-Kantarci, Lei Wu, and Thomas Ortmeyer. A risk-based optimization model for electric vehicle infrastructure response to cyber attacks. *IEEE Transactions on Smart Grid*, 9(6):6160–6169, 2017.

[44] Seyedamirabbas Mousavian, Melike Erol-Kantarci, and Thomas Ortmeyer. Cyber attack protection for a resilient electric vehicle infrastructure. In *2015 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2015.

[45] Andre Teixeira, Kin Cheong Sou, Henrik Sandberg, and Karl Henrik Johansson. Secure control systems: A quantitative risk management approach. *IEEE Control Systems Magazine*, 35(1):24–45, 2015. doi: 10.1109/MCS.2014.2364709.

[46] Mete Ozay, Iñaki Esnaola, Fatos Tunay Yarman Vural, Sanjeev R. Kulkarni, and H. Vincent Poor. Machine learning methods for attack detection in the smart grid. *IEEE Transactions on Neural Networks and Learning Systems*, 27(8):1773–1786, 2016. doi: 10.1109/TNNLS.2015.2404803.

[47] Liqun Yang, Yuancheng Li, and Zhoujun Li. Improved-elm method for detecting false data attack in smart grid. *International Journal of Electrical Power & Energy Systems*, 91:183–191, 2017.

[48] Ieee standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces. *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, pages 1–138, 2018. doi: 10.1109/IEEESTD.2018.8332112.

[49] James McCarthy, Jeffrey Marron, Don Faatz, Daniel Rebori-Carretero, Johnathan Wiltberger, and Nikolas Urlaub. Cybersecurity for smart inverters: Guidelines for residential and light commercial solar energy systems. Technical report, National Institute of Standards and Technology, 2024.

[50] Tareq Hossen, George Amariucai, and Behrooz Mirafzal. Integrating an analytical risk factor into a neural network framework for self-protective inverters. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 12(2):1535–1544, 2023.

[51] Kumar R James Ranjith, Deepa Kundur, and Biplab Sikdar. Transient model-based detection scheme for false data injection attacks in microgrids. In *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2019.

[52] Mehdi Ganjkhani, Mostafa Gilanifar, Jairo Giraldo, and Masood Parvania. Integrated cyber and physical anomaly location and classification in power distribution systems. *IEEE Transactions on Industrial Informatics*, 17(10):7040–7049, 2021.

[53] Yu Wang and Bikash C Pal. Destabilizing attack and robust defense for inverter-based microgrids by adversarial deep reinforcement learning. *IEEE Transactions on Smart Grid*, 14(6):4839–4850, 2023.

[54] Amr S Mohamed and Deepa Kundur. On the use of reinforcement learning for attacking and defending load frequency control. *IEEE Transactions on Smart Grid*, 15(3):3262–3277, 2023.

[55] Ezzeldin Shereen, Kiarash Kazari, and György Dán. A reinforcement learning approach to undetectable attacks against automatic generation control. *IEEE Transactions on Smart Grid*, 15(1):959–972, 2023.

[56] Yihao Wan, Tomislav Dragičević, and Qianwen Xu. On cybersecurity of dc microgrids: An adversarial learning approach. *IEEE Transactions on Industrial Electronics*, 2026.

[57] Jeewon Choi, Behshad Roshanzadeh, Manel Martínez-Ramón, and Ali Bidram. An unsupervised cyberattack detection scheme for ac microgrids using gaussian process regression and one-class support vector machine anomaly detection. *IET Renewable Power Generation*, 17(8):2113–2123, 2023.

[58] Amir Basati, Josep M Guerrero, Juan C Vasquez, Najmeh Bazmohammadi, and Saeed Golestan. A data-driven framework for fdi attack detection and mitigation in dc microgrids. *Energies*, 15(22):8539, 2022.

[59] Yasin Abdolahi, Behrooz Mirafzal, and Fariba Fateh. Dl-predictive control of gfm inverters for inrush current mitigation and overload protection. In *2025 IEEE Kansas Power and Energy Conference (KPEC)*, pages 1–6. IEEE, 2025.

[60] IEEE. Ieee guide for cybersecurity of distributed energy resources interconnected with electric power systems. *IEEE Std 1547.3-2023 (Revision of IEEE Std 1547.3-2007)*, pages 1–183, 2023. doi: 10.1109/IEEESTD.2023.10352402.

[61] Stephanie Harshbarger, Bala Natarajan, Eugene Y. Vasserman, Muhammad Farooq Umar, Mohammad Shadmand, and George Amariucai. On zero-dynamics stealthy attacks with learned state space models. In *2024 4th International Conference on Smart Grid and Renewable Energy (SGRE)*, 2024. doi: 10.1109/SGRE59715.2024.10428796.

[62] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi: 10.1162/106365601750190398.

[63] Gyunghoon Park, Chanhwa Lee, Hyungbo Shim, Yongsoon Eun, and Karl H Johansson. Stealthy adversaries against uncertain cyber-physical systems: Threat of robust zero-dynamics attack. *IEEE Transactions on Automatic Control*, 64(12):4907–4919, 2019.

[64] Giulio Bottegal, Farhad Farokhi, and Iman Shames. Preserving privacy of finite impulse response systems. *IEEE control systems letters*, 1(1):128–133, 2017.

[65] Jerome Le Ny and George J Pappas. Differentially private filtering. *IEEE Transactions on Automatic Control*, 59(2):341–354, 2013.

[66] Paul Griffioen, Sean Weerakkody, and Bruno Sinopoli. A moving target defense for securing cyber-physical systems. *IEEE Transactions on Automatic Control*, 66(5):2016–2031, 2020.

[67] Yilin Mo, Sean Weerakkody, and Bruno Sinopoli. Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Systems Magazine*, 35(1):93–109, 2015.

[68] Bharadwaj Satchidanandan and Panganamala R Kumar. Dynamic watermarking: Active defense of networked cyber–physical systems. *Proceedings of the IEEE*, 105(2): 219–240, 2016.

[69] Arunava Naha, André Teixeira, Anders Ahlén, and Subhrakanti Dey. Sequential detection of replay attacks. *IEEE Transactions on Automatic Control*, 68(3):1941–1948, 2022.

[70] Jairo Giraldo, David Urbina, Alvaro A Cardenas, and Nils Ole Tippenhauer. Hide and seek: An architecture for improving attack-visibility in industrial control systems. In *International Conference on Applied Cryptography and Network Security*, pages 175–195. Springer, 2019.

[71] David I Urbina, Jairo A Giraldo, Alvaro A Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. Limiting

the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1092–1105, 2016.

[72] Karl Henrik Johansson. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, 8(3):456–465, 2000. doi: 10.1109/87.845876.

[73] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954. doi: 10.1093/biomet/41.1-2.100.

Appendix A

Chapter 1

A.1 Proof for Attack Generation

Since matrix D is 0, we replace the solution in [27] as shown below. We need a value of F_2 that satisfies

$$\begin{bmatrix} V & B \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} V = AV. \tag{A.1a}$$

From this, we can easily obtain

$$\begin{bmatrix} V & B \end{bmatrix}^{+} \begin{bmatrix} V & B \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} VV^{+} = \begin{bmatrix} V & B \end{bmatrix}^{+} AVV^{+}, \tag{A.1b}$$

where, $\begin{bmatrix} V & B \end{bmatrix}^{+}$ and V^{+} represent the Moore-Penrose pseudo-inverses of matrices $\begin{bmatrix} V & B \end{bmatrix}$ and V, respectively.

If we choose $\begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$ to be the right-hand side of (A.1b), that is,

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} V & B \end{bmatrix}^{+} AVV^{+}, \tag{A.2}$$

then using one of the properties of the pseudo-inverse, that states that for a general matrix M we have $M^+MM^+ = M^+$, we see that this choice of $\begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$ satisfies (A.1b). Now substituting the same in the left-hand side of (A.1a), and using the property that for a general matrix M we have that $MM^+M = M$, we get

$$\begin{bmatrix} VB \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} V = \begin{bmatrix} V & B \end{bmatrix} \begin{bmatrix} V & B \end{bmatrix}^+ AV, \tag{A.3}$$

the right-hand side of which is equal to AV (meaning that our choice of $\begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$ satisfies (A.1a)) whenever $\begin{bmatrix} VB \end{bmatrix}$ has linearly independent rows. What remains is to set $F = -F_2$.

A.2 Architectures and Hyperparameters

Table A.1 shows the architecture of neural network models associated with the attackers. The table displays the number of nodes in each subsequent layers of the model, activation function in hidden and output layers, and learning rate and optimizer used to update the model's parameters. Similarly, the table A.2 consists of hyperparameter settings for the neural network models. The target network is updated using a soft update method whose coefficient is 0.005 for each attacker. Similarly, the output of these attackers is scaled using the Tanh activation function. Hence, to get the desired attack signals, we scaled the output by some constant, depicted in the same table. In addition to this, the value of the parameters a and b from the reward function in section 2.4.2 for each attacker are provided in the table. In addition, table A.3 consists of architectures and hyperparameters of LSTM detectors. All the detectors have the same architecture but differ in learning rate parameter, which is given in the table.

Table A.1: Architecture of the neural network models associated with the attackers. Attackers (ATT_1, ATT_2) and (ATT_3, ATT_4 and ATT_5) share the same architecture

	ATTACKERS (1 and 2)			ATTACKERS(3, 4 and 5)		
	Actor	Critic	Value	Actor	Critic	Value
Nodes in each layers	60, 256, 256, 2	60, 256, 256, 1	62, 256, 256, 1	60, 256, 256, 2	60, 256, 256, 1	62, 256, 256, 1
Output layer activation	Tanh	Tanh	Tanh	Tanh	Tanh	Tanh
Hidden layer activation	RELU	RELU	RELU	RELU	RELU	RELU
Learning rate	0.00008	0.0008	0.0008	0.00008	0.0008	0.0008
Optimizer	ADAM	ADAM	ADAM	ADAM	ADAM	ADAM

Table A.2: Hyper Parameter Setting for the attackers based on SAC algorithm

Name	ATT_1	ATT_2	ATT_3	ATT_4	ATT_5
Replay buffer size	1,000,000	1,000,000	1,000,000	1,000,000	1,000,000
Discount factor	0.99	0.99	0.99	0.99	0.99
Mini-batch size	256	256	512	512	512
Training episodes(N)	50000	50000	50000	50000	250000
Soft update coefficient	0.005	0.005	0.005	0.005	0.005
Actions scaled by	10	4	3	3	3
reward scale factor	2	2	2	2	2
a in reward function	2	2	10	10	10
b in reward function	2	2	7	7	7

Table A.3: Architecture of the LSTM-based detectors

Feature	**Value**
Number of inputs	2
Number of features in hidden states	512
Number of recurrent layers	1
Feed-forward layers	2 layers with size 256 and 128 respectively
Hidden layer activation	RELU
Output layer activation	Sigmoid
Optimizer	Adam
Batch size	1024
$DETECTOR_1$ Learning rate	0.004
$DETECTOR_2$ Learning rate	0.004
$DETECTOR_3$ Learning rate	0.003
$DETECTOR_4$ Learning rate	0.001

A.2.1 Design Rationale and Limitations

The settings in Tables A.1 and A.2 were selected manually during attacker training. Training runs were assessed by inspecting their reward trends. When the observed progress was unsatisfactory, hyperparameters were adjusted according to the considerations described below, and training was repeated. This manual process aimed to improve the attacker's ability to cause physical disruption before detection. At iteration i, attacker ATT_i must evade the accumulated detector set $\{DET_0, \ldots, DET_{i-1}\}$ while driving an unobserved tank state outside its physical limits. The adjustments addressed this increasingly constrained objective, while the network architecture, learning rates, replay-buffer capacity, and other settings reported as constant in the tables were retained across attackers.

The action-scaling factor was reduced from 10 to 4 and then to 3 as the detector ensemble expanded. Additional detectors recognize attack patterns that earlier detectors missed, motivating a narrower perturbation range. Reducing the scale limits the amplitude of the injected actions and directs exploration toward smaller perturbations that may produce physical disruption with less observable deviation. The objective was to achieve disruption before detection; avoiding alarms without affecting the unobserved states would not constitute a

successful attack.

The reward coefficients were adjusted to place greater emphasis on stealth as evading the accumulated detectors became more difficult. In Section 2.4.2, a penalizes measured-state deviations, whereas b rewards unobserved-state deviations. The initial values $(a, b) = (2, 2)$ were changed to $(10, 7)$ for ATT_3–ATT_5. This gives more weight to staying undetected while still encouraging disruption of the unobserved states. The adjustment was intended to discourage physical disruption that triggers an alarm before the attack objective is reached. Explicit detector penalties remain necessary because small measured-state deviations alone do not guarantee stealth.

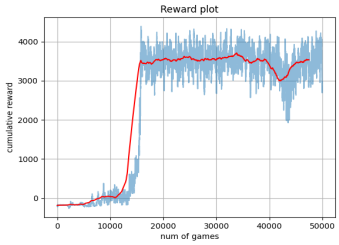
The mini-batch size was also increased manually from 256 to 512 for ATT_3–ATT_5. A benefit is that averaging over more replayed transitions reduces sampling variability in gradient estimates. This may support more stable updates when training combines physical-deviation rewards with detector penalties. However, this explanation is a rationale for the empirical choice; the experiments did not separately establish that the larger batch improved training stability or attack success.

Training duration was determined by inspecting the reward curves and continuing training until the reward trend flattened. The reported runs used 50,000 episodes for ATT_1–ATT_4, whereas the reward for ATT_5 flattened at approximately 250,000 episodes. This observed plateau motivated the longer training duration for the final attacker. The plateau served as a practical stopping criterion and does not establish convergence to an optimal policy.

These settings therefore reflect manual tuning under progressively stronger detection constraints. Because the detector set and several training parameters changed across iterations, the experiments do not isolate the contribution of each adjustment to attack performance or training duration.



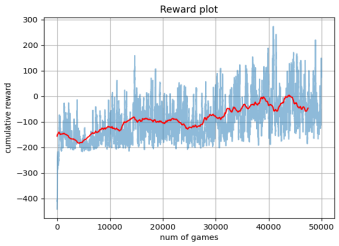
((a)) $ATTACKER_1$



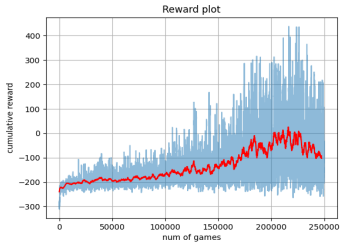
((b)) $ATTACKER_2$



((c)) $ATTACKER_3$

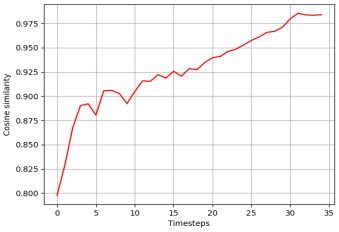


((d)) $ATTACKER_4$

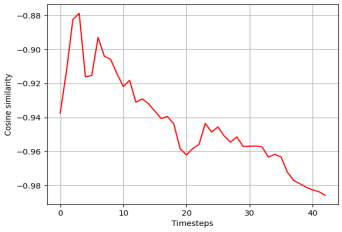


((e)) $ATTACKER_5$

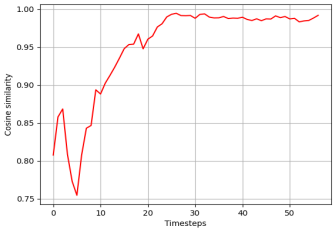
Figure A.1: Reward plots of five consecutive attackers



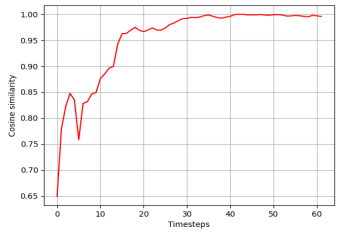
((a)) $ATTACKER_1$



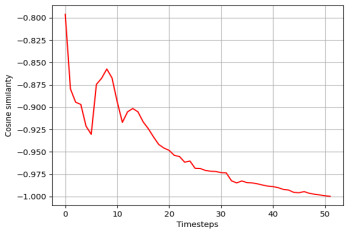
((b)) $ATTACKER_2$



((c)) $ATTACKER_3$



((d)) $ATTACKER_4$



((e)) $ATTACKER_5$

Figure A.2: Cosine similarity as described in 2.5.4.

A.3 Attacker Rewards and Performance

A.4 Individual-Detector Classification Diagnostics

Evaluation scope Figure A.3 presents the saved confusion matrices for D2, trained using the first attacker's data, and the later D4 and D6 checkpoints obtained under strict and consensus training, respectively. D2 is not assigned to either aggregation setting. These results describe individual detectors evaluated on different saved datasets containing normal-operation, fault, and attack windows; they do not represent a common-dataset comparison of the two ensembles. The recorded overall classification accuracies are 94.12% for D2, 93.04% for D4, and 92.91% for D6.

Missed events and wrong-inverter assignments Non-normal entries in the Normal column represent missed events: an inverter-labelled window is classified as normal operation. Across all non-normal windows, the reported missed-event proportions are 7.62% for D2, 5.43% for D4, and 9.17% for D6. D4 additionally assigns 3.26% of non-normal windows to the wrong inverter. These pooled percentages use all non-normal windows as the denominator, whereas the displayed confusion-matrix cells are normalized separately within each true-class row.

Relation to the attack success criterion A missed event is distinct from successful mislocalization under the criterion used in this paper: predicting Normal does not count as attacker success, which requires sustained attribution to an incorrect inverter under the relevant aggregation rule. Furthermore, these matrices summarize individual window-level predictions, whereas Table 4.4 reports ensemble outcomes across complete evaluation trials. The matrices therefore document residual classification errors but are not used to rank the aggregation rules or infer their ASRs. They also do not establish whether a correct individual prediction was overridden by consensus voting on the same input.

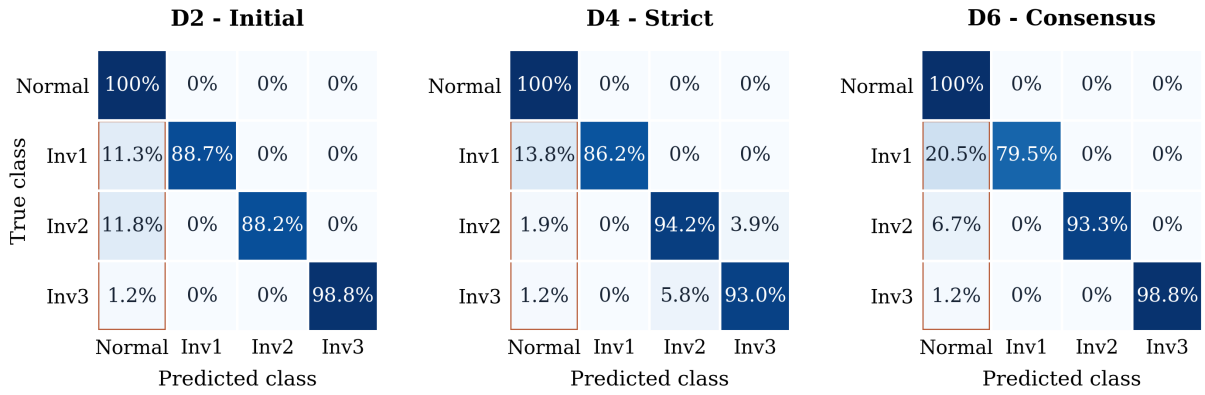


Figure A.3: Individual-detector confusion matrices for D2 (initial attack-trained checkpoint), D4 (later strict checkpoint), and D6 (later consensus checkpoint), evaluated on their respective saved datasets. Rows denote true classes and columns denote predictions. Values are percentages within each true-class row. Outlined cells indicate inverter-labelled windows classified as Normal. These matrices describe individual classifiers, not the aggregated ensemble decisions.

Appendix B

Chapter 4

B.1 Supporting Disturbance Studies

These studies illustrate the responses behind the design. Because they use separate reference heights and disturbance schedules, they support their motivation without adding samples to the formal recovery-selection or DBA analyses.

B.2 Measured and unmeasured disturbances

The trajectories apply +4 cm at 250 s and −4 cm at 500 s to real Tank 1 or Tank 3. While Tank 1 is measured directly, Tank 3 affects feedback indirectly through its outflow.

B.3 Fixed-direction perturbation screen

Having compared feedback routes, the next figures show how increasing one fixed perturbation affects fake-only operation. They complement the $\rho = 0.20$ trajectory in Figure 5.3, using matched noise across the tested bounds. Because the direction and feedback arrangement are fixed, these results illustrate that the particular screen rather than a universal limit on surrogate parameters.

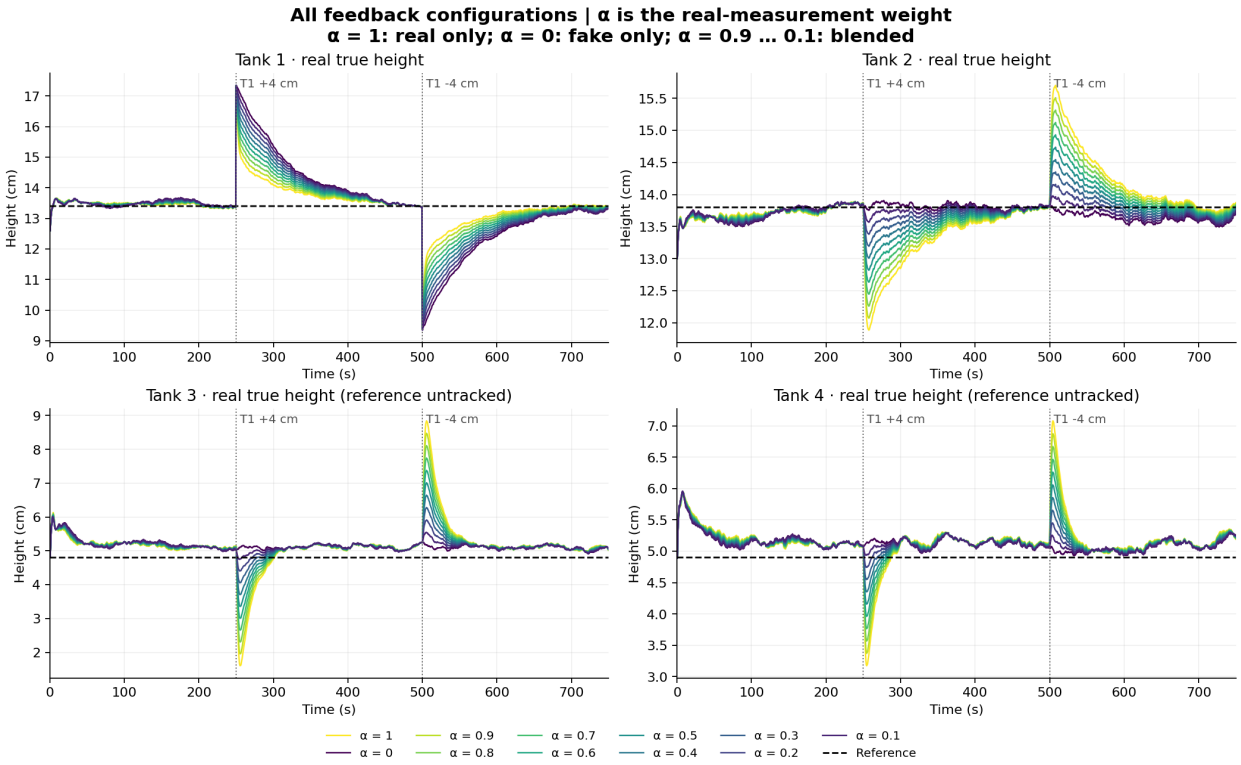


Figure B.1: Illustrative Tank-1 disturbance trajectories across real-only, fake-only and mixed feedback. The weight grid and two-jump schedule differ from the formal recovery-selection experiment.

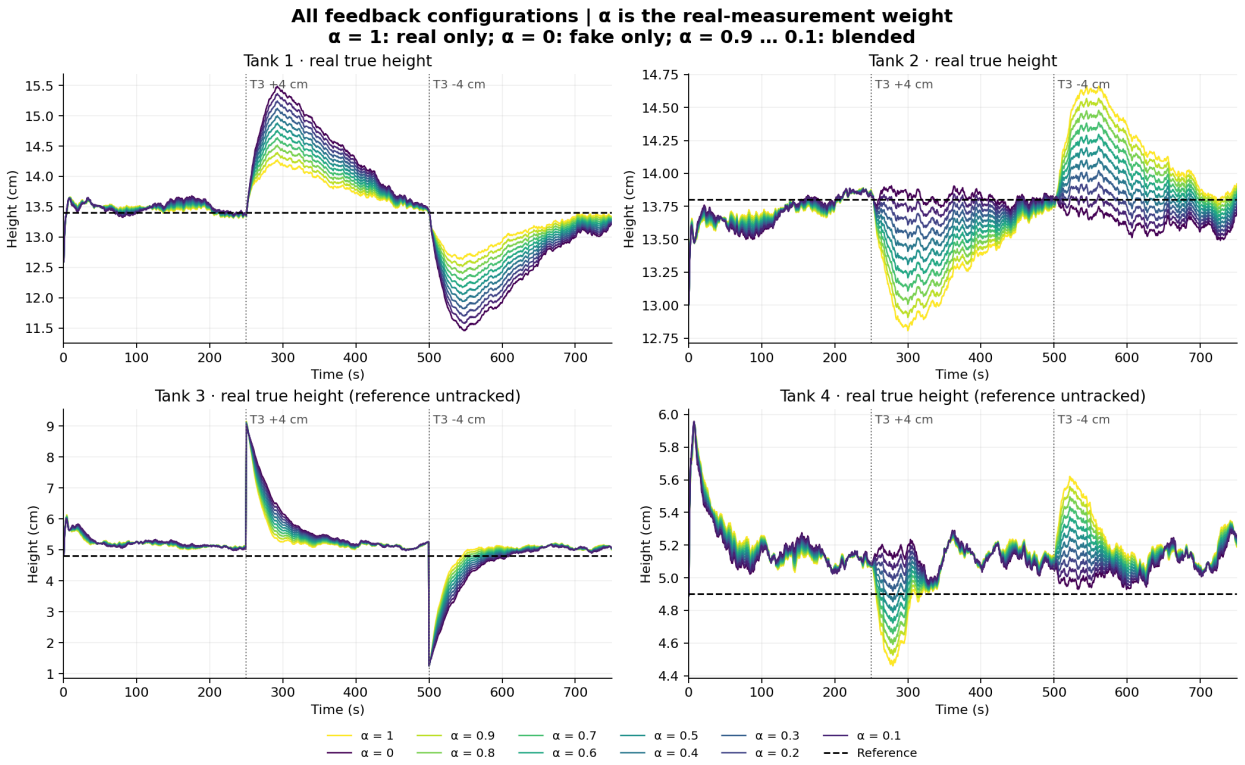


Figure B.2: Illustrative trajectories with disturbances in unmeasured Tank 3. Changes in the lower tanks provide an indirect, real-time feedback path.

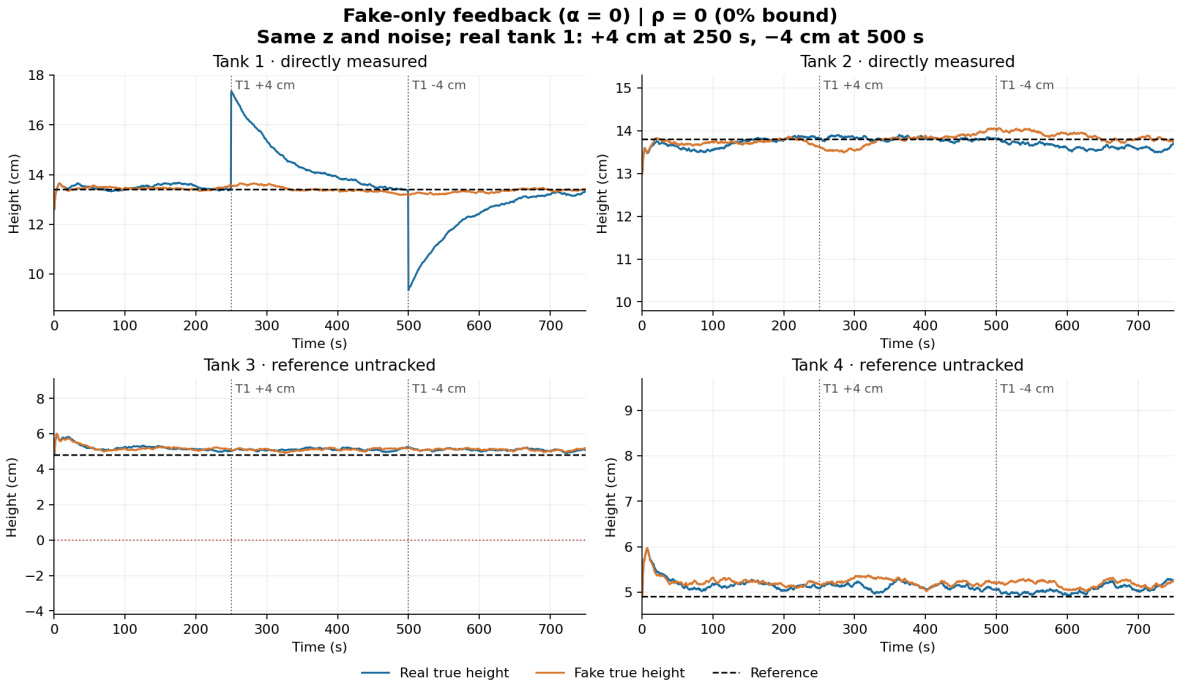


Figure B.3: Fixed-direction screen at $\rho = 0$, with identical real and surrogate parameters and fake-only feedback.

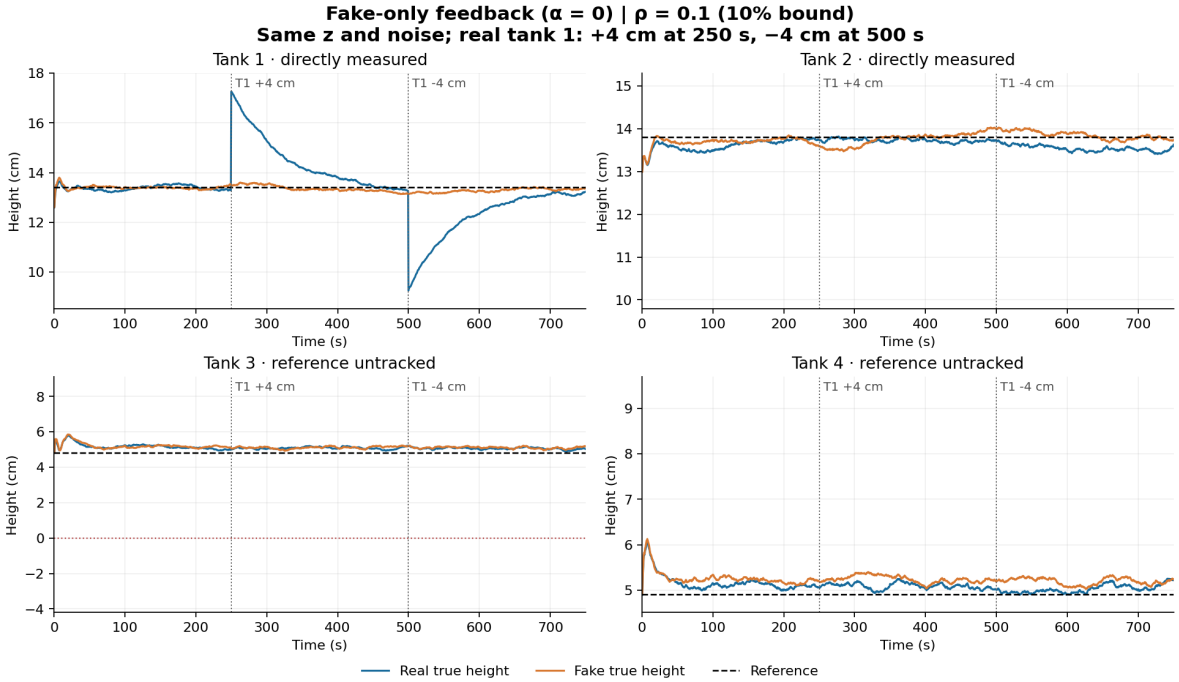


Figure B.4: Fixed-direction screen at $\rho = 0.10$. The real trajectory remains within the physical range for this run.

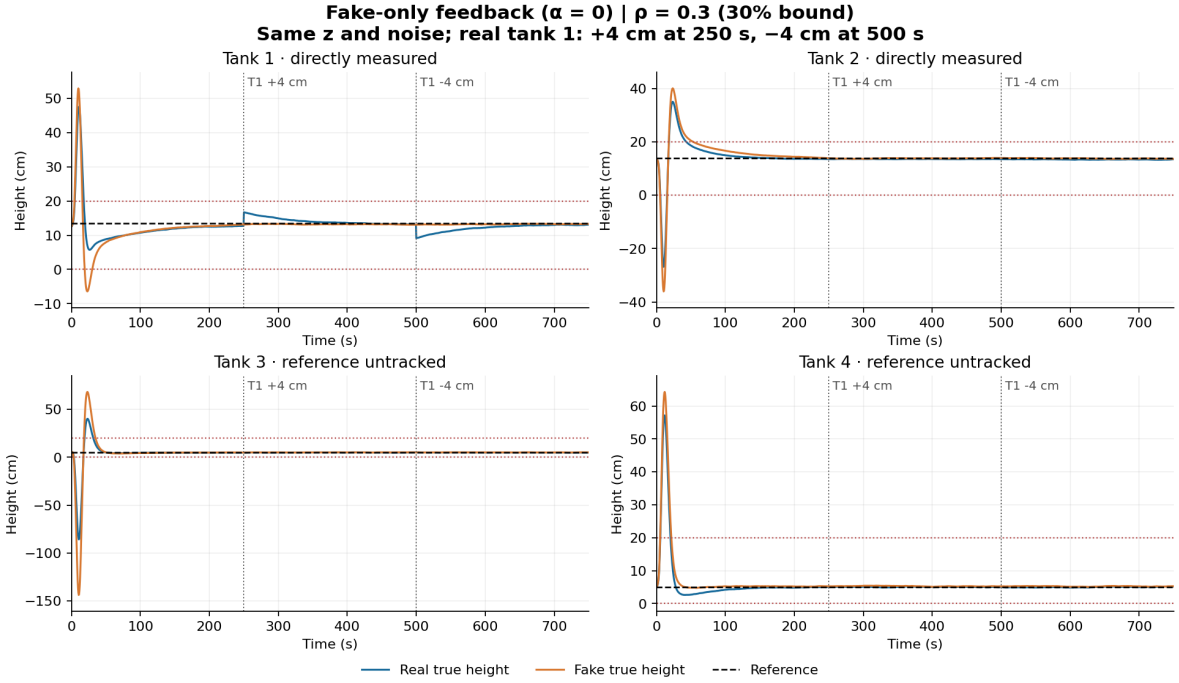


Figure B.5: Fixed-direction screen at $\rho = 0.30$. The real trajectory crosses the physical boundary during startup. Values outside the operating range indicate failure of the linear simulation's physical validity check.

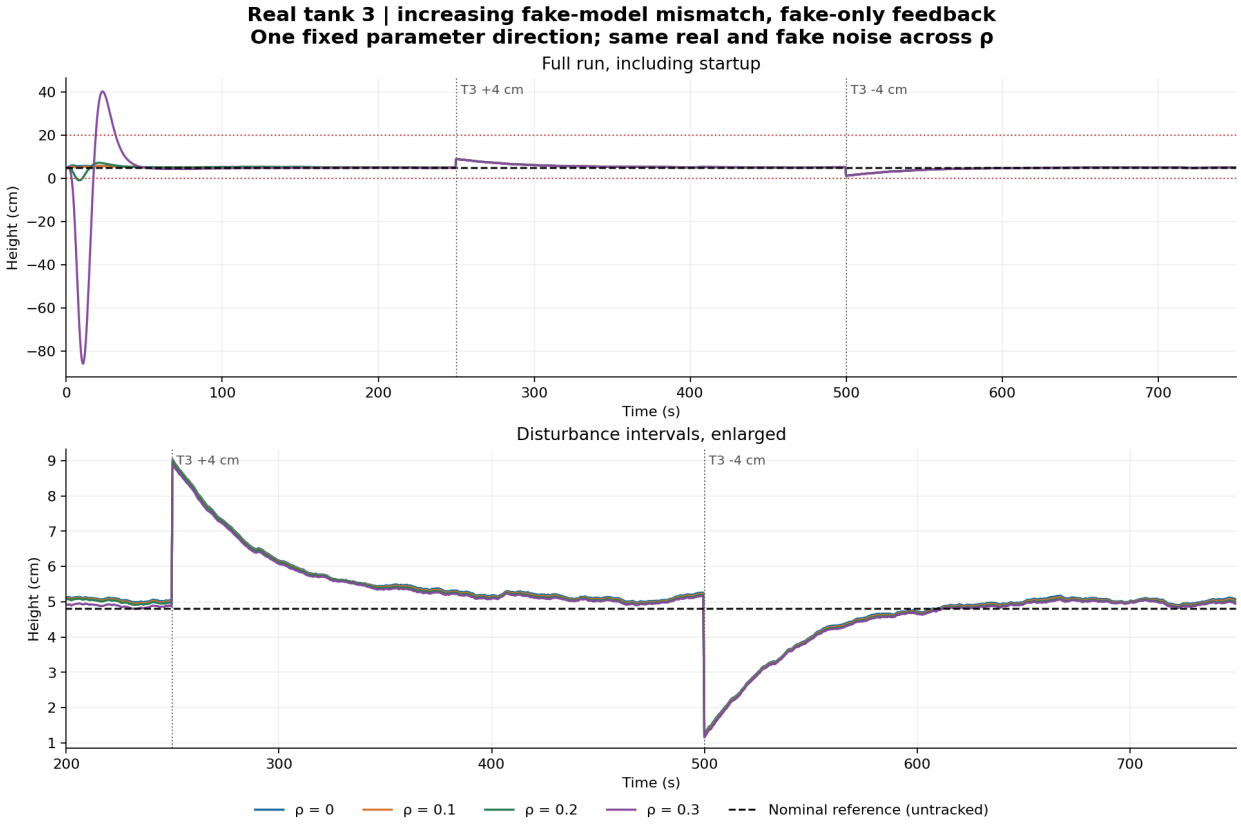


Figure B.6: Additional Tank-3 comparison across perturbation bounds. These illustrative trajectories are not additional samples in the primary DBA analysis.

B.4 Smaller-outlet sensitivity

The smaller-outlet study provides an additional check on passive recovery by halving the outlet areas, thereby slowing natural drainage. Each disturbed run is compared with a no-jump run under matched noise, so subtracting the two isolates the height change caused by the intervention. The plots then show how that contribution evolves as real feedback is retained. Their late-window mean absolute error describes the remaining tracking error and is distinct from the overall lower-tank RMSE used in final validation.

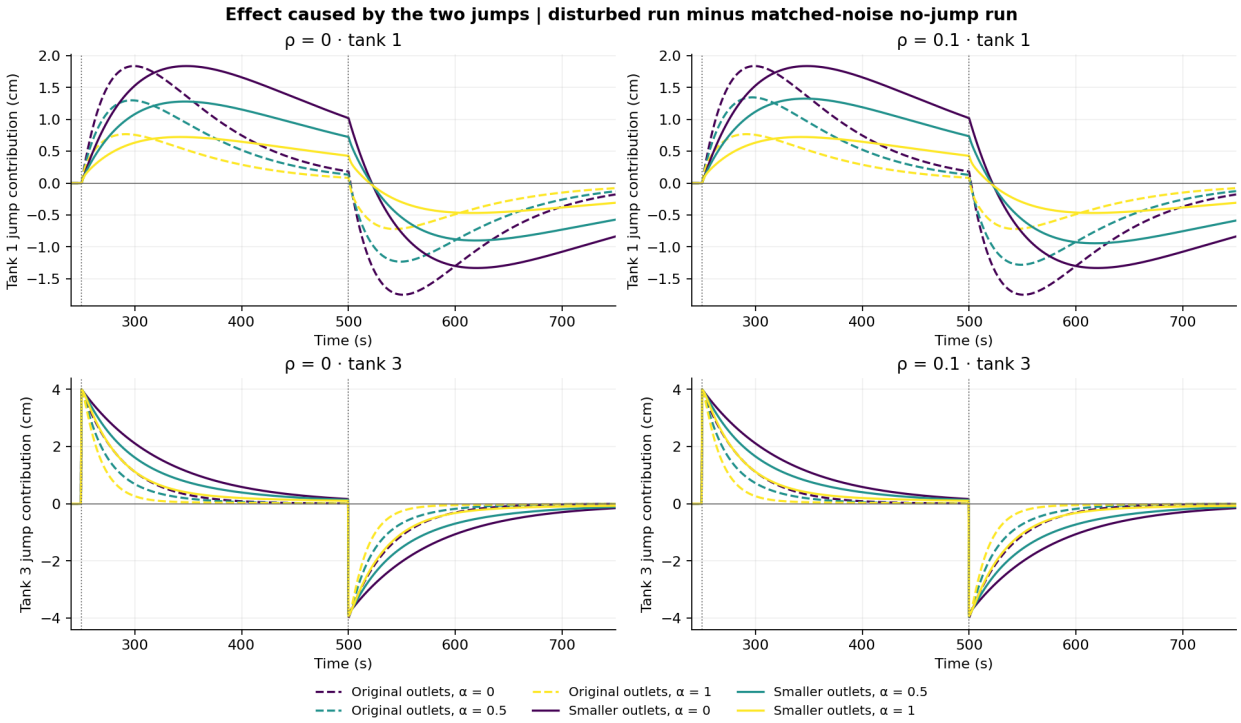


Figure B.7: Supporting comparison of original and halved outlet areas. Subtracting the matched no-jump trajectory isolates the height changes caused by the disturbances.