ON THE EFFECT OF JAMMING ATTACKS ON CYBER PHYSICAL SYSTEMS WITH THE FOCUS ON TARGET TRACKING APPLICATIONS

THESIS

Presented to the Graduate Council of
Texas State University-San Marcos
in Partial Fulfillment
of the Requirements

for the Degree
Master of SCIENCE

by

Emad Guirguis, B.S

San Marcos, Texas
May 2012

ON THE EFFECT OF JAMMING ATTACKS ON CYBER PHYSICAL SYSTEMS WITH THE FOCUS ON TARGET TRACKING APPLICATIONS

Committee Members Approved:

Mina Guirguis, Chair

Hongchi Shi

Qijun Gu

Approved:

J. Michael Willoughby
Dean of the Graduate College

COPYRIGHT

by

Emad Guirguis

2012

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

ACKNOWLEDGEMENTS

I am heartily thankful to my supervisor, Dr. Mina Guirguis, my thesis advisor and thesis committee chair, whose guidance, support and encouragement from the initial to the final level enabled me to develop an understanding of the subject, in addition to providing experiment materials for this study.

I also want to thank Dr. Hongchi Shi and Dr. Qijun Gu, members of my thesis committee, for their availability and participation in making this thesis possible.

To my parents, siblings and friends who have stood so steadfastly by me, always so reliable, throughout this graduate program, I want to say thank you.

Lastly, I offer my regards and blessings to all of those who supported me in any aspect during the completion of this thesis.

This manuscript was submitted on December 5th, 2011.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS V

LIST OF TABLES IX

LIST OF FIGURES X

ABSTRACT XII

CHAPTER

I INTRODUCTION 1

1.1 Motivation 1

1.2 Cyber Physical Systems 1

1.3 Security 3

1.4 Jamming 4

1.5 Thesis Statement 5

II RELATED WORK 7

2.1 Introduction 7

2.2 Jamming Attacks 7

2.3 Anti-Jamming 9

2.4 Cyber-Physical Systems 11

2.5 Thesis Focus 13

III SYSTEM MODEL 15

3.1 Introduction 15

3.2 System Model 16

3.2.1 Attacker's Model 17

3.3 Model Assumptions 18

3.4 Experimental Setup 19

IV ATTACK PARADOX 21

4.1 Introduction 21

4.2 Model Instantiation 22

4.3 Positioning Scenarios 23

4.4 Target Tracking Scenarios 25

4.5 Conclusion 29

V SELECTIVE CONTROL/MEASUREMENT SIGNAL JAMMING…….. ……31

5.1 Introduction 31

5.2 Model Instantiation 31

5.3 Problem Setup 36

5.4 Experimental Setup 38

5.5 Robots Profiling 39

5.6 Experimental Results 40

5.7 Conclusion 51

VI CONCLUSION AND FUTURE WORK 52

6.1 Conclusion 52

6.2 Future Work 53

APPENDIX A BELLMAN EQUATION 54

BIBLIOGRAPHY 58

VITA 62

LIST OF TABLES

Table **Page**

1. Target Transition Matrix 34

2. Experiments' Different Cases 41

3. Coefficients for the Different Cases 42

4. Heuristics Values 43

LIST OF FIGURES

Figure **Page**

1. A Cyber Physical System 15
2. Surveyor SRV-1 Blackfin Robot 19
3. Cyber Physical System – Case Study 22
4. A PI Controller 23
5. Positioning Results under Selective and Random attacks, and no attack 26
6. Target Tracking Results – total distance covered under no attacks 28
7. Target Tracking Results – under no jamming, 5% jamming and 20% jamming attacks 29
8. Cyber Physical System – Case Study 32
9. Case Study Layout 35
10. Target's Possible Moves 35
11. The Agent's Possible Moves 36
12. Approximate Solution Process and Usage 38
13. Attacker's Rewards Using Robot A 44
14. Agent Total Steps Using Robot A 44
15. Attacker's Rewards using Robot B 45
16. Agent Total Steps using Robot B 46
17. Attacker's Rewards - Simulations 47

18. Agent Total Steps - Simulations 47

19. Attacker's Rewards – Case 1 – Error Rates 48

20. Attacker's Rewards – Case 2 – Error Rates 49

21. Attacker's Rewards – Case 3 – Error Rates 49

22. Attacker's Rewards – Case 4 – Error Rates 50

23. Attacker's Rewards – Case 5 – Error Rates 50

ABSTRACT

ON THE EFFECT OF JAMING ATTACKS ON CYBER PHYSICAL SYSTEMS

WITH THE FOCUS ON TARGET TRACKING APPLICATIONS

by

Emad Guirguis, B.S.

Texas State University-San Marcos

May 2012

SUPERVISING PROFESSOR: MINA GUIRGUIS

A Cyber-Physical System (CPS) is a one that features coordination between computational and physical components. CPSs are used in a diverse number of areas such as aerospace, chemical processes, civil infrastructure, healthcare, etc. The principal goal of CPSs is to monitor the behavior of the physical components and adjust their behaviors through the proper control signals. Due to their reliance on wireless technologies with shared mediums, communication between the components is vulnerable to various attacks. Thus, it is important to address such vulnerabilities as this research area grows.

In this thesis, we assess the security of CPSs through experimental evaluation for two different components in target tracking applications. In the first component, we assess the impact of low level jamming attacks on convergence properties in different target tracking scenarios. Surprisingly, this study uncovers some of the benefits realized when systems are subjected to low level of jamming. In the second component, we study intelligent jamming attack techniques whereby an attacker aims to increase his/her reward based on various jamming attack policies and actions. This component raises awareness for the impact of a determined attacker who is bent on inflicting damage with the minimum cost. Our evaluation is conducted through the SRV-1 robots.

CHAPTER I

INTRODUCTION

1.1. Motivation

Many crises occur now and then that critically need the involvement of remotely controlled robots to do dangerous tasks – ex. Fukushima Daiichi nuclear disaster happened in March 2011 in Japan. Such applications and other less critical or daily ones in wide diverse areas such as aerospace, automotive, chemical processes, civil infrastructure, energy, healthcare, manufacturing, transportation, entertainment, and consumer appliances, face considerable challenges that make the mission of Cyber-Physical Systems (CPS) critical. Such systems need to be protected from attacking techniques that are getting more intelligent and hard to detect.

1.2. Cyber Physical Systems

Cyber-Physical Systems refer to a new generation of integrated computational systems with physical capabilities. Expanding the capabilities of these physical systems and the ability to interact with are the keys for the future of technology developments that have a lot of opportunities and research challenges [16]. The principal goal of Cyber-Physical Systems (CPS) is to monitor some physical process behavior they are a part of (through

obtained measurement signals), and make decision and take actions to change its behavior accordingly (through control signals).

Physical components of CPSs' platforms include:

- an amalgamation of electro-mechanical sensors and actuators
- a communication stack
- memory
- a processing unit

Each of these components can be centralized in one entity as we can see in embedded systems to observe its behavior and correct in the event of out of the ordinary behavior [18], or distributed over a group of entities as in the case of an automobile control system, where sensors provide data to a microprocessor dedicated to manage functionality, which then communicates data through a network to a controller that takes it into actions [17]. The Bus System of any University can be an example of a CPS technology – Vehicular Networks – where buses are moving in different parts of the city and are connected together to make a network.

Mobile Cyber Physical Systems is a CPS where the physical components have inherent mobility. One of the important applications of Mobile CPSs is target tracking. Target tracking is an application where base stations are connected to number of mobile agents to track set of targets. There are many applications where target tracking in CPS is used, such as search and rescue, and border control.

1.3. Security

Security is one of the critical attributes of any communication network. Various attacks have been reported over the past many years. At the present time, with the advances in technology, wireless networks are becoming more affordable and easier to build. Many metropolitan areas deploy public WMANs for people to use freely. Moreover, the prevalence of WLANs as the basic edge access solution to the Internet is rapidly becoming the reality. However, wireless networks are accompanied with an important security flaw; they are much easier to attack than any wired network. Although the shared and easy to access medium is great advantage of wireless networks, but it is one of their greatest weaknesses at the same time. In particular, it makes it extremely easy for an adversary to launch an attack. While the goal of traditional DoS attacks is to overflow the user and the kernel domain buffers to deny service, in wireless networks, however, there are many occasions when launching an attack – such as jamming – can be much easier for an adversary [1]. For example, in carrier sensing based networks (e.g. 802.11, sensor networks, etc.) a saboteur might continually transmit electromagnetic energy on the medium, achieving the following two results:

1. Packet transmissions at the sender are deferred because the medium is sensed to be busy
2. Reception at the receiver is interfered with due to the jamming signals

Both of these effects degrade the wireless network performance significantly. With such malicious techniques, it is feasible to block any communication between two wireless capable nodes.

However, such “brute-force” jamming techniques, which mainly exploit physical and MAC layer vulnerabilities, can be detected easily. Jammers have responded by employing more intelligent ways to accomplish jamming task in order to evade detection. They exploit vulnerabilities at the higher layers of the network stack. A typical example is detecting the transmission of specific control packets and preferentially corrupting such packets by injecting interference. In order to address these threats, security experts must deploy more efficient methods for detecting and preventing such “smart” (stealthy) attackers. A fascinating arms-race, thus, begins between adversaries and network administrators [1].

1.4. Jamming

Jamming is the radiation of electromagnetic energy in a communication channel which reduces the effective use of the electromagnetic spectrum for legitimate communication. Jamming results in the loss of the link’s reliability, increased energy consumption, extended packet delays, and disruption of the end-to-end routes [7]. Jamming may be both malicious with the intention to block communication of an adversary or non-malicious in the form of unintended channel interference. In the context of embedded wireless networks for time-critical and safety critical operation such as in Cyber Physical Systems (ex. medical devices and industrial control networks), it is essential that mechanisms for resilience to jamming are easy and fast to detect ongoing attacking on the communication protocol. Resilience to jamming and its avoidance, collectively termed as anti-jamming, are hard practical problem as the jammer has an unfair advantage in detecting legitimate communication activity due to the broadcast nature of the channel.

Communication nodes are unable to differentiate between jamming signals and legitimate ones. They are unable even to change in the communication activity due to node mobility or nodes powering off without some minimum processing at the expense of local and network resources [7]. For our purposes, jamming is any attack to deny service to legitimate users by generating noise or fake protocol packets or legitimate packets but with spurious timing.

1.5. Thesis Statement

Wireless networks are built upon a shared medium that makes it easy for adversaries to launch jamming attacks. These attacks can be easily accomplished by an adversary emitting radio frequency signals that do not follow the underlying MAC protocols. Jamming attacks can severely interfere with the normal operation of the system and, consequently, mechanisms are needed that can cope with jamming attacks. Target tracking in CPS where the agents communicate with the base station via wireless network and the base station receives the measurement signals about the targets through the shared medium, is one of the applications that is vulnerable to such attacks.

In this thesis - through experimental implementation and simulation - we (1) present an attack paradox on Mobile Cyber-physical systems where attacking could be helpful under low-level jamming attacks, and (2) examine the agent and base station behaviors in the presence of smart attacking policy where an attacker is monitoring the communication channel and launching several attacks to drop/delay measurement and/or control packets.

In this thesis, we try to answer three main questions:

- How would the CPS behave when it only receives a subset of the measurement and the control signals?
- Can the attacker exploit the CPS by targeting specific packets?
- What is the overall performance under different jamming scenarios?

CHAPTER II

RELATED WORK

2.1. Introduction

Radio interference attacks are a serious threat to the operation of a wireless network regardless of its type. In order to cope with the threat of jamming attacks, it is important to understand the different threat models that are employed by adversaries, the methods that are needed to diagnose these threats, and the counter-measures that may be employed to defend against jamming attacks.

2.2. Jamming Attacks

Researchers in [9], [10], and [11] discussed the traditional literature on jamming primarily focusing on the design of physical layer technologies, such as spread spectrum, that are resistant to Radio Frequency (RF) jamming. It should be realized that the physical layer technologies needed to reliably resist jamming have not found widespread deployment in commodity wireless devices, such as wireless LANs and sensor networks.

Thuente and Acharya have listed a variety of metrics that can be used to compare various jamming attacks in order to study the intelligent jamming in wireless networks [3]. Clearly, the following metrics are all relevant. However, which of these are the most important ones, will depend greatly on the application addressed.

1. Energy efficiency
2. Stealthy level
3. Maintaining consistent behavior with or close to the protocol standard
4. Dealing with authenticated or unauthenticated users
5. Strength against error correction algorithms
6. Strength against physical layer techniques such as Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), Code Division Multiple Access (CDMA)

There are many different attack strategies that a jammer can perform in order to interfere with other wireless communications:

1. A constant jammer
2. Stealthy jammer
    a. Deceptive jammer
    b. Random jammer
    c. Reactive jammer

The above jamming models try to break down the communication between two nodes. While they can achieve a high degree of denial of service, they exhibit (in general) low energy efficiency and high probability of detection. Intelligent jamming [4] on the other hand tries to exploit upper layer protocol vulnerabilities in order to achieve three main goals:

- Maximizing the jamming gain
- Targeting specific nodes

- Reducing the probability of detection

The major intelligent jamming attacks models are:

1. CTS Corruption Jamming
2. ACK Corruption Jamming
3. DATA Corruption Jamming
4. Narrow-band Jamming
5. Greedy Behavior

2.3. Anti-Jamming

The issue of detecting non-MAC compliancy was studied by [10]. This work showed that a greedy user can increase his share of bandwidth by slightly modifying the driver of his network adapter. The greedy user may try to corrupt the Request To Send (RTS) and Clear To Send (CTS) of other users to prevent packet transmission, or may corrupt ACKs to cause the ACK contention window to increase, leading to larger back off. They proposed DOMINO, a system that has to be implemented only at the AP for detection of such greedy behavior in the MAC layer of IEEE 802.11 public networks. DOMINO algorithm is conducted by collecting the traffic traces and run several tests on them.

Exploiting MAC layer is one of the important attacks that have been addressed by several research works. Through simulations, Kyasanur and N. Vaidya in [13] presented modifications to IEEE 802.11 MAC protocol that simplifies misbehavior detection. These modifications are effective in restricting the throughput of selfish nodes to a fair share. And Bellardo and Savage in [12] presented a stopgap non-cryptographic

countermeasure that can be implemented in the firmware of existing MAC hardware with low overhead on existing hard-ware.

Wood and Stankovic [8] studied briefly the issue of jamming detection in the context of sensor networks. This study posed the issue of jamming detection in the loose context of the utility of the communication channel, and presented several factors that might affect the channel's utility:

- Repeated inability to access wireless channel
- Bad framing
- Checksum failures
- Illegal values for addresses or other fields
- Protocol violations (e.g., missing ACKs)
- Excessive received signal level
- Low signal-to-noise ratio
- Repeated collisions
- Duration of a certain condition

Countermeasures for coping with jammed regions in wireless networks have been studied in [14] by Noubir and Lin, and in [15] by Xu, Wood, Trappe, and Zhang. In the first work, the use of Low Density Parity Check (LDPC) codes is proposed to cope with jamming. Furthermore, an anti-jamming technique is proposed for 802.11b that involves the use of Reed-Solomon codes. In the second study, two countermeasures methods are presented for coping with jamming. The first method, channel surfing, involves a form of an on-demand link-layer frequency hopping technique, where valid participants change

the channel they are communicating on when a denial of service attack occurs. The second method, spatial retreats, involves legitimate network devices moving away from the adversary to reestablish connections.

Finally the research conducted in [7] by Xu, Trappe, Zhang and Wood primarily focuses on the issue of mapping the jammed region as the base for jamming detection decisions rather than a single measurement which might not be sufficient for basing decisions upon. This work takes the viewpoint that rather than replace existing systems with more complicated radio platforms, it is instead desirable to understand the modes of attacks that may be launched against existing platforms, and be able to detect them. Following detection, appropriate countermeasures may be employed. They have explored the inconsistencies that might arise from naively employing decision processes built upon these factors. Their detection algorithms may be viewed as a complement to work presented in [8] and when integrated with their mapping algorithm, can lead to enhanced mapping services for Wireless Sensor Networks (WSN).

2.4. Cyber-Physical Systems

An important aspect of CPSs is that they are networked in nature. This not only allows them to form a network for data fusion, and delivery to back-end entities but also take coordinated response actions (in both the passively active and active operational modes). While it is clear that the security of control systems has become an active area in recent years, Communication Security needs the development of protocols for securing both inter and intra-CPS communication from both active (interferers) and passive (eavesdroppers) adversaries [17].

Frank Mueller has highlighted multiple shortcomings in the current design process of cyber-physical embedded systems with real-time constraints in [20]. He urged for an immediate need for research on the protection of critical infrastructure to counter cyber-physical attacks and distributed control problems. He could find absolute absence of solutions to such problems. Besides the suffering from many issues including scalability, robustness, and performance [19], one of the main causes for the research absence is a lack of adequate simulation infrastructure to foster academic to contribute viable solutions [20].

Cyber-Physical Systems (CPS) require the integration of a heterogeneous physical layer and a virtual global decision and control network, mediated by decentralized and distributed local sensing/actuation structures [19]. And hence, [Mueller, 20] recommends that a software simulation framework for the IEC 61850 - International Electrotechnical Commission - standard be design at the level of substation devices, their interaction and their relation to and communication with control centers. This activity should be coordinated with a concerted effort by industry leaders providing valuable input on practicality and requirements. The resulting framework then needs to be complemented by initiatives to support follow-on research on possible cyber-attacks or distributed control problems at the simulation level, the development of counter-measures at the software level and their integration into future standards as well as commercial deployment.

To mitigate vulnerabilities' effects on the traffic system, the cyber system needs to develop tolerance against various attacks. The above discussion brings out a number of

challenges involved in preventing, detecting and mitigating different attacks in CPS. In the following we outline them:

1. Prevention: The attack space is large to enumerate and develop prevention mechanisms. The present cryptographic techniques may not provide a complete solution due to the additional constraints exhibited by CPS.
2. Detection: In CPS, the mechanism to coordinate information for attack detection needs to be scalable. Designing such a mechanism is challenging. Furthermore, sometimes it is difficult to distinguish between attacks scenarios from an unusual but genuine behavior of the physical system. An effective detection mechanism should be able to make this distinction.
3. Mitigation: Mitigation scheme should coordinate with the physical system. Depending on its state, the scheme should choose an appropriate degree of attack isolation. Mitigating an attack while still keeping the system operational is essential for few critical applications.

Ravi Akella, Han Tang, Bruce M. McMilli in [28] presented a semantic model for information flow analysis in a CPS and describes an approach to perform the analysis, including both trace-based analysis and automated analysis through process algebra specification.

2.5. Thesis Focus

Our work in this thesis is built upon these works and study how the system could potentially be at advantage of being attacked to sustain better performance in a networked cyber physical control system and how to mitigate intelligent attacking techniques. We

here study a general case that has a base station which receives measurements signals and sends control signals to agents to adjust their behavior when the communication medium is vulnerable to attacks.

CHAPTER III

SYSTEM MODEL

3.1. Introduction

In this chapter we present the system model that is used in the following chapters to study the effect of different jamming scenarios on a CPS system. Figure 1 represents a CPS composed of a Base Station and a number of Agents. Each agent receives a stream of control signals from the base station to adjust its operation. A stream of measurement signals is fed-back from the agents to the base station. The goal is to adjust the operation to meet specific goals (e.g., reaching equilibrium, tracking targets, etc.). We assume that the measurement and control signals traverse wireless networks with the possibility of them being lost due to the presence of adversaries.

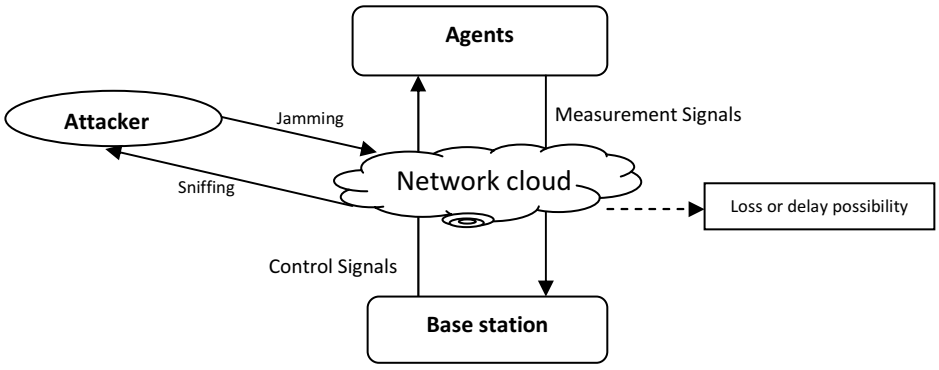


Fig 1: A Cyber Physical System

Our study is performed through our case study of “Target Tracking” with robots (agents). In a target tracking system, the base station receives measurements from the agent about its location and sends control commands to inform the agent with the target’s latest acknowledged state. All communication is done through wireless channels.

3.2. System Model

The block diagram presented in Figure 1 is a CPS system composed of a number of agents (mobile devices or robots), a controller (base station) and a number of targets (mobile devices). Each plant receives a stream of control signals (ui for plant i) from the controller to adjust its operation. A stream of measurement signals (yi from plant i) is fed-back from each plant to the controller to update its control rules. The goal is to adjust the operation of the plants to meet a particular function (case study here is tracking targets). We assume that the measurement and control signals traverse network segments that may be jammed by adversaries.

We consider a Linear Time-Invariant (LTI) system that can be described by Equations (1) and (2), where x represents the state of the system (plants and controller), y is the output vector (measurements from the plants to the controllers) and u is the control vector to all plants. We omit the plant’s number and refer to u and y as the general control and measurement signals, respectively.

$$\dot{x} = Ax + Bu + \omega \quad (1)$$

$$y = Cx + z \quad (2)$$

Matrices A, B, and C represent the plants coefficient matrix, the control matrix, and the output matrix, respectively. ω represents a Gaussian random variable with a zero mean and a covariance matrix Q. We refer to Q as the process noise covariance matrix and it is independent from x. Similarly, z represents a Gaussian random variable with a zero mean and a covariance matrix R. We refer to R as the measurement noise covariance matrix and it is independent from x. Since the control/measurement signals are typically continuous but they are transmitted in packets over the network, we assume the presence of a sampler and a holder. We assume that the measurement signal $y(t)$ is sampled at times t_k, so we have

$$y_k = y(t_k) \qquad (3).$$

Similarly, we assume that the control signal from the controller can be held by the plant for a duration τ, so we have

$$u_\tau = u(t_\tau) \qquad (4)$$

3.2.1 Attacker's Model

Jamming wireless signals has been one of the most effective attack techniques against any system that relies on wireless communications and cause degradation of quality and systems instability. Based on the above model, we consider an adversary that can jam a subset of the measurement and control signals, y and u, respectively. We assume that jammed packets are dropped and not retransmitted. This assumption is realistic since these packets either carry important control signals or will be used in applying the control rules. If we allow them to be retransmitted by the sender, then the system would always be lagging behind in its control (due to the additional delay in packet loss detection and

retransmission). We consider a stealthy adversary, which tries to drop a small number of packets to avoid being detected.

Consider an instantiation from the model above in which a base station controls the speed of the robot through sending control signals that carry the voltage value to be applied at the robot's circuits that controls the motors. For simplicity, we assume that the speed of the robot is a linear function of the voltage signal applied. The robot reports its measured speed back to the base station. Due to different slopes and types of surfaces, the voltages need to be adjusted for the robot to maintain a specific speed. Based on the model above, the control signal u is given by:

$$u = K\,y - r \quad (5)$$

where K is the control matrix, r is the reference point (robot target speed) and y is the measured speed.

3.3. Model Assumptions

We consider a system that have these assumptions:

- we have only one agent
- we have only one target
- attacker have the ability to jam control and measurement signals and decides when to attack
- packet retransmission is useless as decision data carried by control signals depend on the system's current status, and delayed packets would be outdated

3.4. Experimental Setup

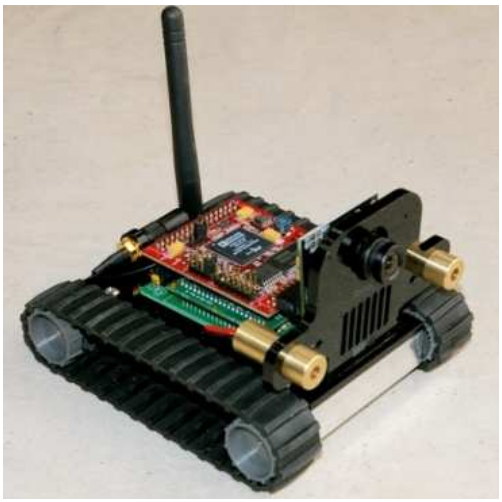


Fig. 2: Surveyor SRV-1 Blackfin Robot

Our experimental setup is composed of the Surveyor SRV-1 Blackfin robots [27] and a base station. The SRV-1 is an open source robot and uses 802.11b/g for communication. Each robot has two laser pointers for detecting distances to obstacles as well as an onboard camera. Since the robots can be configured to be remotely controlled by the base station, we were able to develop two target-tracking related experiments (by having the robots act as plants connected wirelessly to laptop computer as base station as shown in Figure 1). The measurement signal can be a report on either target's or agent's status. The control signal is an array composed of three parameters

Control Signal: $\begin{bmatrix} L_s \\ R_s \\ T \end{bmatrix}$

where L_s is the speed of the left set of wheels, R_s is the speed of the right set of wheels, and T is the duration of movement in milliseconds. In case of negative control values of the speed parameters, the robot moves backwards. Using these parameters, the base

station can control the robot to go backward or forward with specific speed for specific duration of milliseconds (which translates into distance). In addition, the ability of setting the speed of the right set of wheels to a value different from the speed value of the left set of wheels gives the ability to rotate the robot.

Although, we have noticed inaccuracies in the actions taken by the robot (mainly due to physical characteristics of the surface, the power level of the battery and the resolution of the measured distance, etc.), our experiments reflect common difficulties that an agent would experience in real-world scenarios. In other words, the inaccuracy of distances calculated by the robot resembles difficulties that an agent can always experience in real world tasks.

CHAPTER IV

ATTACK PARADOX

4.1. Introduction

Smart attackers tend to increase their benefits, which can be achieved by manipulating different factors that affect their gain in the jamming technique used. One of those factors is the cost incurred for mounting the attack. In this chapter we assess the impact of low level jamming attacks on (Cyber Physical Systems) CPS and their effect on convergence. A system is called “converged” when it reaches a stable state and/or accomplishes its tasks. For example in the system explained in Chapter III, the Base Station controls the Agents to track Targets. In such a case, the system is converged when the Agent reaches the Target. The Attacker’s goal is to prevent the Agent from reaching the Target with the least cost possible. Cost can be captured by several metrics such as the power consumed by Attacker or the number of attempts done by the Attacker to jam the measurement or the control signals.

A case study – we consider Target Tracking in CPS with mobile devices under jamming attack to study the effect of low level jamming (that is a result of an intelligent attack).

4.2. Model Instantiation

Consider an instantiation from the system model defined in Chapter III. We use it to implement our case study of target tracking. As shown in Figure 3, we have a Base Station which is aware of Target's location and communicates with an Agent (robot) through a wireless network to send control signals in order to track the Target and receive measurements of the Agent's location. The measurement signal is composed of the distance reported from the robot to the first object in front of it (this is collected through specific commands sent to the robots). Meanwhile the Attacker is trying to jam the communication between the Base Station and the Agent. We consider attacks on measurement signals only and no attacks on the control signals.

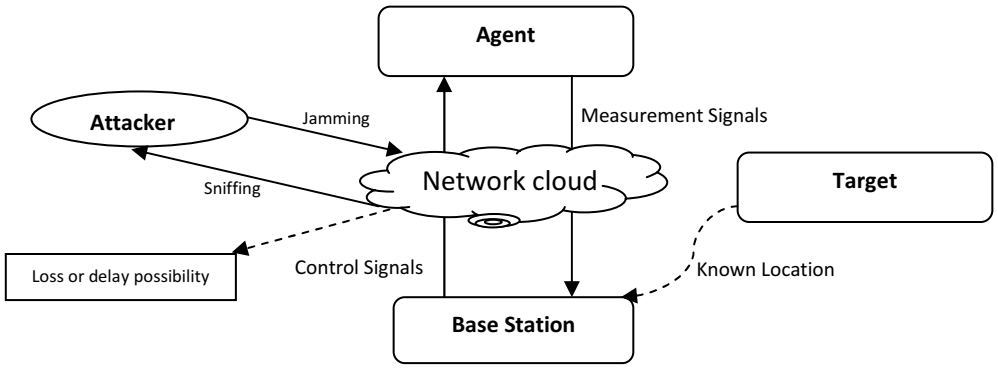


Fig 3: Cyber Physical System – Case Study

In many target tracking scenarios, a controller is needed by the Base Station to make decisions. Consider a Proportional Integral (PI) controller – Figure 4 – employed by the Base Station that adjusts the voltage based on the deviation between the current measurement and the reference point (target value). This deviation arises due to factors,

such as different slopes/surface textures in the environment, manufacturing imperfections, etc. The discrete equation for the PI controller is given by equation 4.

$$\mu(i) = \mu(i-1) + k_\rho e(i) + k_I\big(e(i) - e(i-1)\big) \qquad (4)$$

Where $e(i)$ is the deviation (error signal) between the measurement $y(i)$ and the reference point *r* at time i. While $k_\rho$and $k_I$are the controllers' constants. As the integral parameter - k_I – has a little effect in our model, it is set to zero and so our model can be represented by equation 5.

$$\mu(i) = \mu(i-1) + k_\rho e(i) \qquad (5)$$

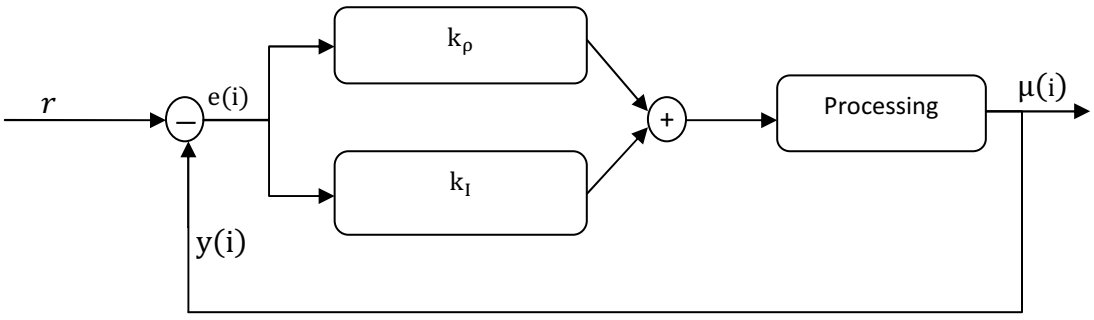


Fig 4: A PI Controller

4.3. Positioning Scenarios

The goal of this experiment is to locate the robot at a specific location from a fixed object. In this experiment the robot measures the distance from the fixed object and reports it to the Base Station. Then the Base Station calculates the error between robot's location and the desired location, and applies the PI controller on the measurements and sends command to the robot to move forward or backward till it reaches the desired distance. In Figure 5, we present a comparison between three graphs presenting three different jamming cases. The first is robot's behavior under no attacks, i.e. without

jamming any of the measurements' packets. The second is generated by dropping one specific packet in each run to watch the effect caused by dropping a single packet on convergence. The third is generated by dropping randomly only one packet. The x-axis represents value of k_ρ used in the PI controller and the y-axis represents the number of iterations the robot takes till convergence. Convergence is achieved when the robot fulfills its taks – i.e. being located at a specific distance from the obstacle within an accepted range of error. Each value is the average of five runs on the same initial distance. We can see how jamming attacks achieve better performance indicated by faster convergence. In addition, looking at the graph, we can conclude three important observations: first, dropping one specific packet usually results in faster convergence than random packet dropping technique as the randomly selected packet to be dropped can be the last one to convergence, which adds unnecessary extra iterations. Second observation is that this principle does not apply if the convergence is so quick – for example, in case of $k_\rho = 5$ – and this can be explained by the high percentage one measurement packet represents in few packets scenarios. Third observation we noticed while executing random packets dropping experiments is that when the random selection jammed more than one packet, it resulted in worse convergence. This can again be explained by the significant effect of the percentage of attacked packets. The low percentages of packets dropped help the system to converge faster than the cases of high dropped packets' percentages and no attacks. All observations highlight the surprising effect of jamming a fine margin of measurements packets.

4.4. Target Tracking Scenarios

This set of experiments is a more dynamic one that can be applicable to a larger number of real life applications. In this implementation, we have a virtual Target that moves at speed that is half the robot's speed. The Agent is supposed to follow the Target keeping a specific distance between them, in other words, the Agent is tracking a moving Target. Let's define an iteration to be the single movement of the robot towards the goal. In a case where no attacks are mounted, moving the virtual Target half the distance that is moved by the Agent would simulate the desired scenario. However, the robots – as explained in Chapter III – may not necessarily move the exact distance they are supposed to. Applying the PI controller with different values of k_ρ and the error is calculated as the difference between the measurement and the reference point which is the updated goal distance after moving the virtual Target. The robot represents the Agent where its measurements are vulnerable to attacking. The virtual Target represents the Target and the Base Station is always aware of its location.

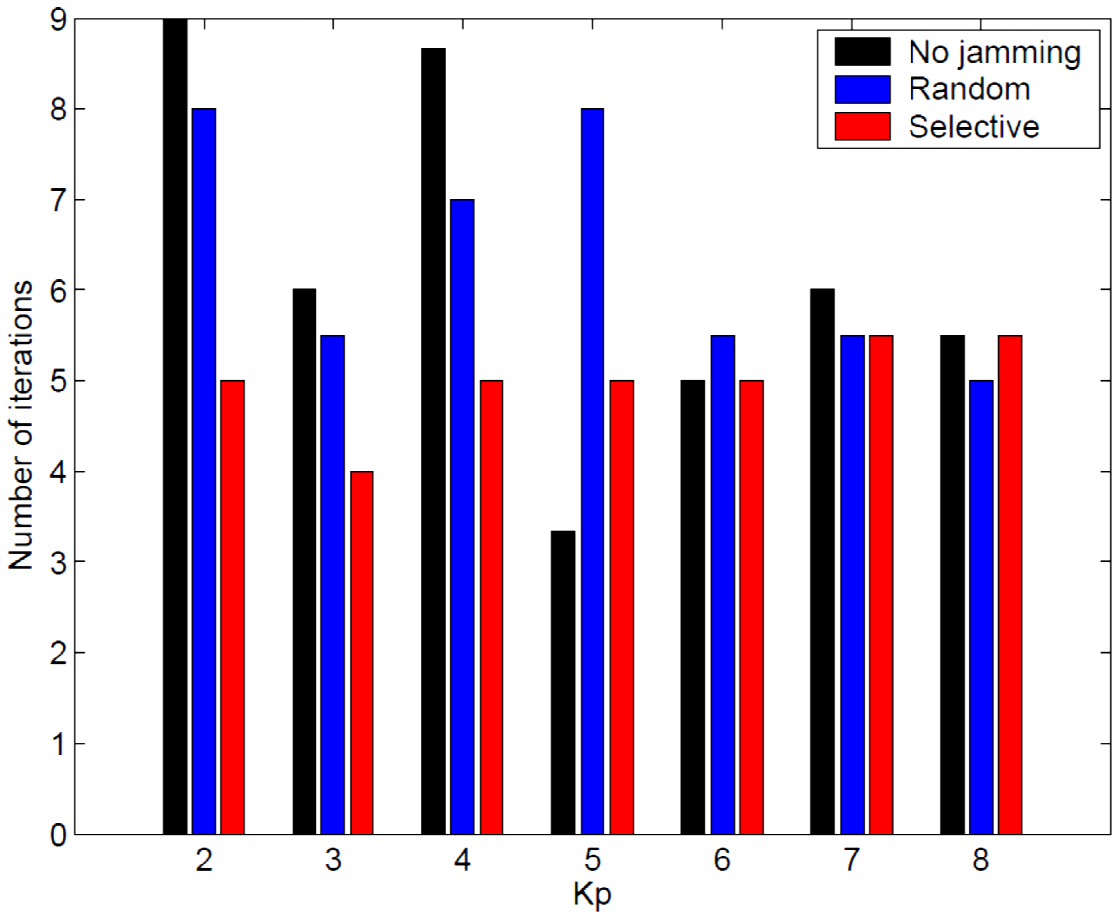


Fig. 5: Positioning Results under Selective and Random attacks, and no attack

We run three different cases similar to the Positioning Scenarios. The first case is without any jamming attacks, the second one is under jamming 5% of the signals uniformly at random, and the third one is under jamming 20% of the signals uniformly at random. Figures 6 and 7 present our results averaged over five runs. Under no attacks – first case – Figure 6 presents the impact of different k_ρ values on the system's convergence. As we can see, small values of k_ρ result in large distances covered by the Agent as well as large values. The small values decrease the aggressiveness degree of the controller, while the large values result in too aggressive controller and overshooting results than the target.

This shows the impact that the value of k_ρ has on PI controller and so on the system's convergence.

A comparison graph shown in Figure 7 presents the impact of three different jamming levels on the convergence of the system. The effect of the level of packet dropping is more obvious than results of Positioning Scenarios, and we could find similar observations to the ones we found in positioning scenario. However, the significance effect of fine margin jamming can be seen in Target Tracking Scenarios more obviously. The number of iterations resulted in cases of 5% and 20% jamming is less than the number of iterations in case of no jamming.

In addition, the effect of random packet dropping is so close to the effect of dropping specific packet. To explain this observation, let's emphasize that random packet dropping did not give a good performance as specific packet dropping in Positioning Scenarios because the dropped packet was sometimes the last or the first packet and because the likelihood of this to happen in Target Tracking Scenarios is less as convergence needs more iterations than convergence in Positioning Scenarios.

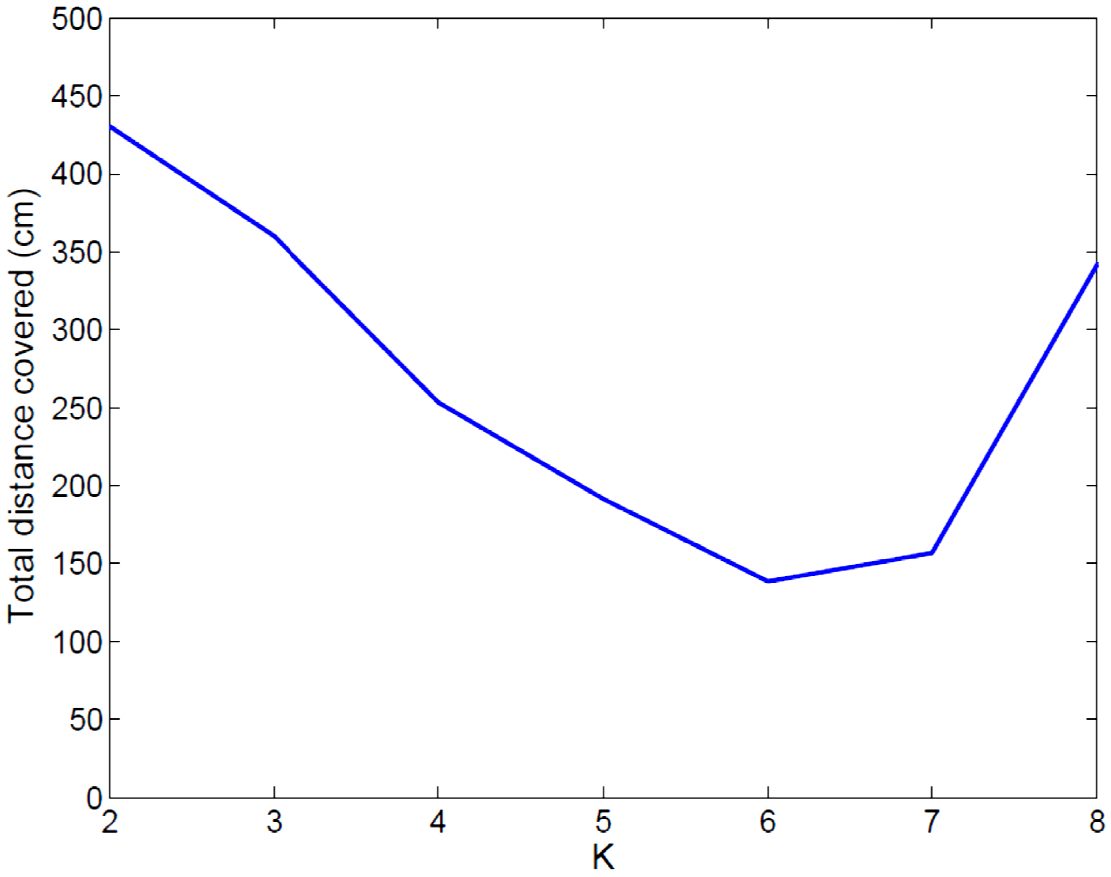


Fig. 6: Target Tracking Results – total distance covered under no attacks

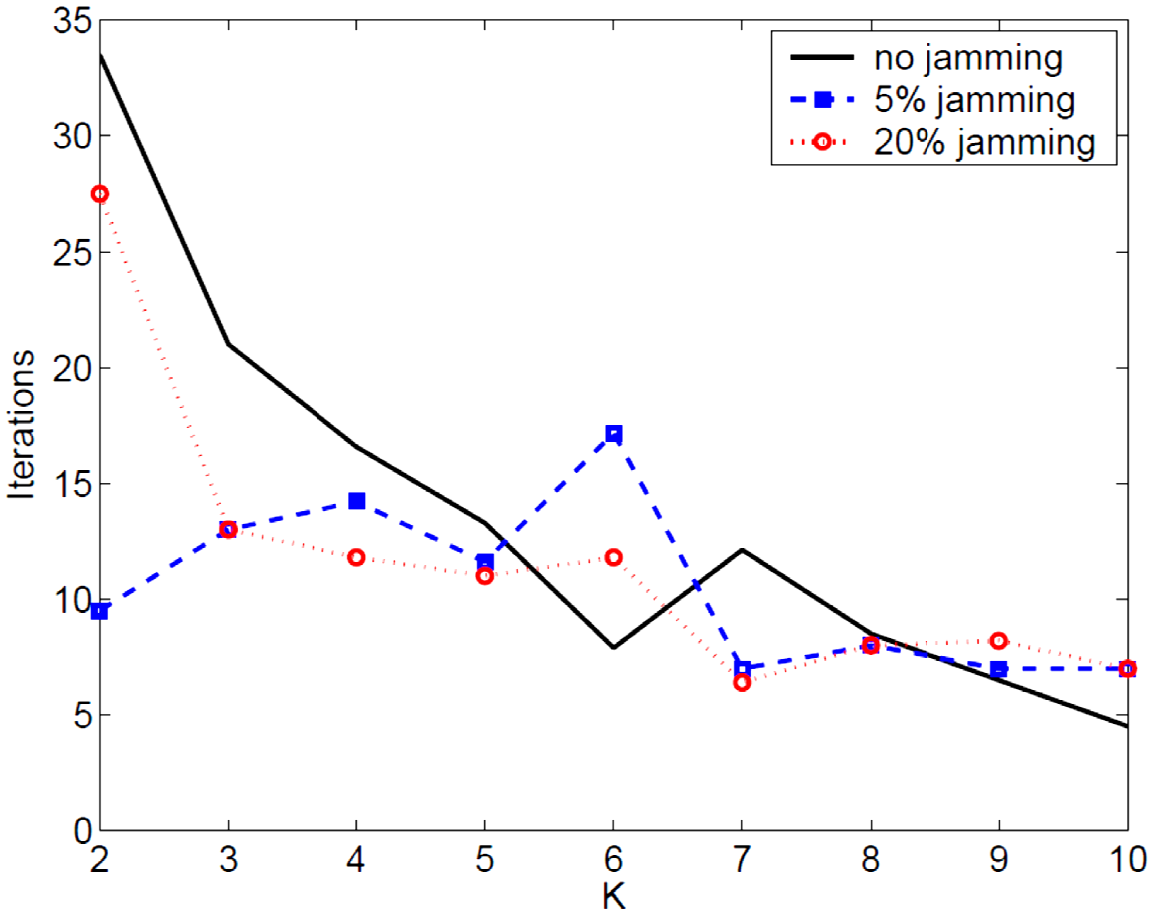


Fig. 7: Target Tracking Results – under no jamming, 5% jamming and 20% jamming attacks

4.5. Conclusion

In this chapter we presented surprising results for systems that can benefit from being subjected to low level of jamming attack. Although jamming wireless signals have been one of the most effective attack techniques against any system that relies on wireless communications and cause degradation of quality and systems instability. In this study, some of the benefits are realized in which low-level jamming attacks help the system to converge in shorter times. Two case studies – Positioning and Target Tracking - in Cyber Physical Systems (CPS) with mobile devices are applied to study these surprising results. This issue because the low percentage of dropped packets cause the controller to react

more aggressively towards the reference value (due to larger error signal) which results in faster convergence. We believe this study can be applied to other controllers.

CHAPTER V

SELECTIVE CONTROL/MEASUREMENT SIGNAL JAMMING

5.1. Introduction

In the previous chapter, we have shown how some systems can benefit from being subjected to low level jamming attacks compared to systems that are not subjected to attacks. In this chapter, we present an evaluation of smart attacking policies that aim to maximize Attacker's gain. We consider an Attacker that is monitoring the control and measurement signals and can interfere with them, taking full advantage of the wireless communication vulnerabilities. Attacking the control and measurement signals can be accomplished by emitting radio frequency signals that do not follow the underlying MAC protocols.

5.2. Model Instantiation

Like the Target Tracking case study explained in Chapter III, the case study here is an environment in which a Target is to be tracked by an Agent (robot) which receives commands from and sends location reports to a Base Station over a wireless network that is vulnerable to attacks. Unlike in the previous chapter, the measurement signal is for Target's location. The environment is considered to be vulnerable to adversaries who can interfere with the communication signals. We consider a grid environment in which the

Target moves independently (randomly) in any of the four directions - forward, backward, left, or right – with equal probabilities. A Base Station detects the Target location (measurement signal) and sends control signal to an Agent to track the Target.

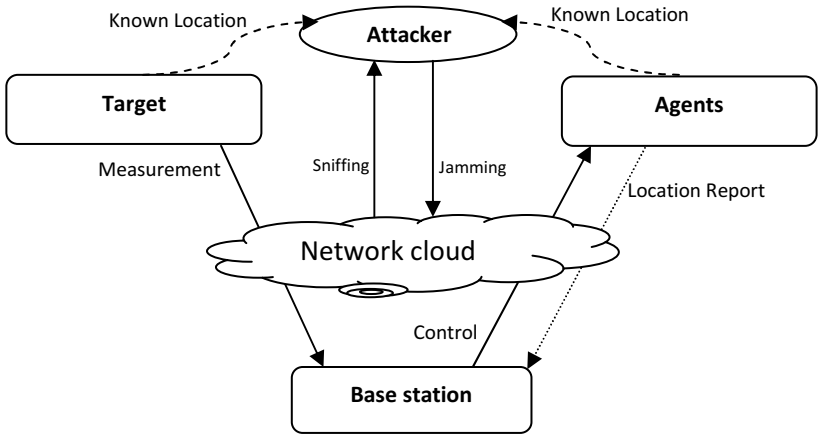


Fig 8: Cyber Physical System – Case Study

Figure 8 illustrates a general block diagram for a CPS composed of a number of mobile devices and a controller (Base Station). As can be seen from Figure 8, the Attacker is aware of the Target's location in this study. However, the Target's location might not be updated to the Base Station if the Attacker jams the measurement signal. Figures 9 through 11 explain the layout of our case studies and the possible moves for the Target and the Agent from each location. As it can be seen from Figures 10 and 11, the Agent in our model has more options to move than the Target. For example, if the Target is at location 1, possible moves can be to locations 2 or 6 while the Agent can move to locations 2, 6, 5, or 21. Agent and Target can move either horizontally or vertically only. Locations 12, 13, 14, 17 and 19 are obstacles that neither the Agent nor the Target can

move to. The Agent receives control signals from the Base Station that specify the Agent's destination at next time slot.

Table 1 presents the Target Transition Matrix (TTM), where first column is the start location and the first row is the destination location. At any time slot t and Target at location x, cell value TTM(x, y) where x is row number and y is column number, is the probability of the Target moving to location y at time t+1. For example, TTM(3, 2) = 25 means that the Target can move from location 3 to location 2 with probability of 0.25, while TTM(1, 10) = 0 means that Target cannot reach location 10 from location 1 in one time slot. Given the Target's location x at time t, we can calculate the possibility to be at location y after n time slots by multiplying TTM by itself n times, which means that TTM^n(x, y) represents the probability of Target being at location y at time t+n, from location x at time t. The Base Station utilizes TTM^n in deciding the Agent's move that minimizes the distance between the Agent and the Target.

Table 1: Target Transition Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	.33	.33	0	0	0	.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	.25	.25	.25	0	0	0	.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	.25	.25	.25	0	0	0	.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	.25	.25	.25	0	0	0	.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	.33	.33	0	0	0	0	.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	.25	0	0	0	0	.25	.25	0	0	0	.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	.25	0	0	0	.25	.25	.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	.25	0	0	0	.25	.25	.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	.25	0	0	0	.25	.25	.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	.25	0	0	0	.25	.25	0	0	0	0	.25	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	.33	0	0	0	0	.33	0	0	0	0	.33	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	.33	0	0	0	0	.33	0	0	0	0	.33	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	.33	0	0	0	0	.33	0	0	0	0	.33	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.5	0	0	0	0	.5	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.33	0	0	0	0	.33	0	0	0	0	.33
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.33	0	0	0	0	.33	.33	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.33	.33	.33	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.33	.33	.33	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.33	.33	.33
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.33	0	0	0	.33	.33

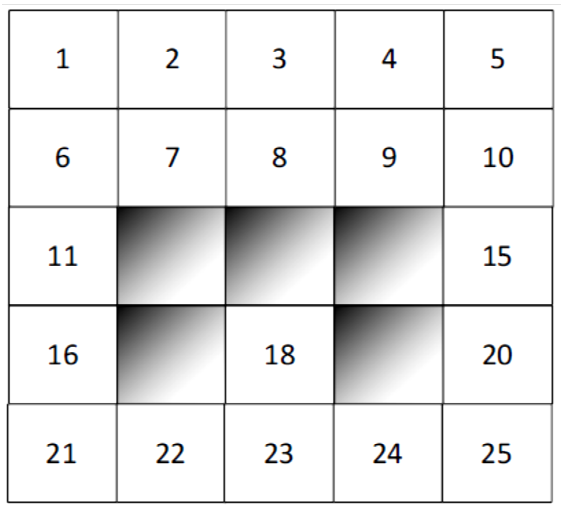


Fig 9: Case Study Layout

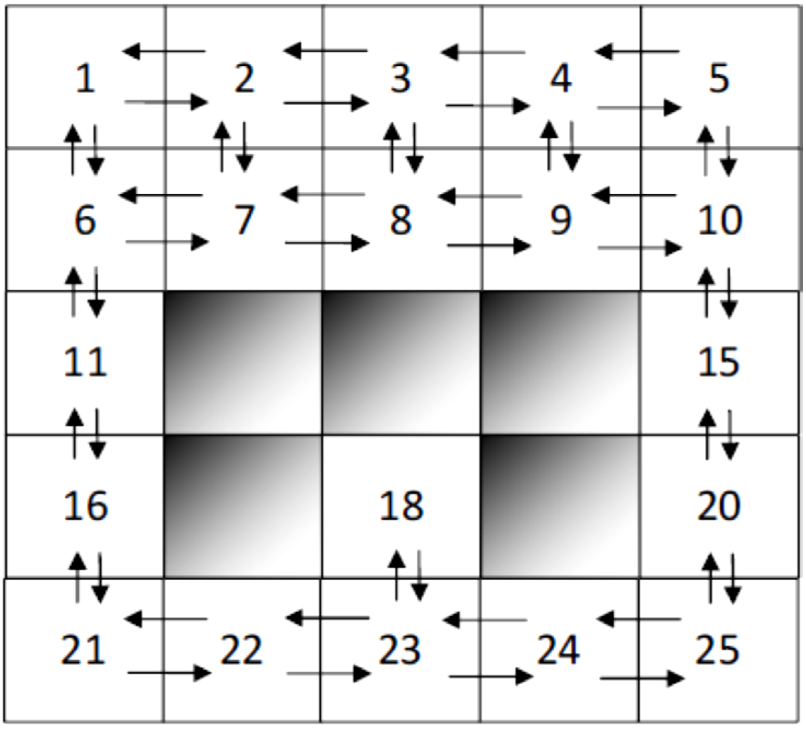


Fig 10: Target's Possible Moves

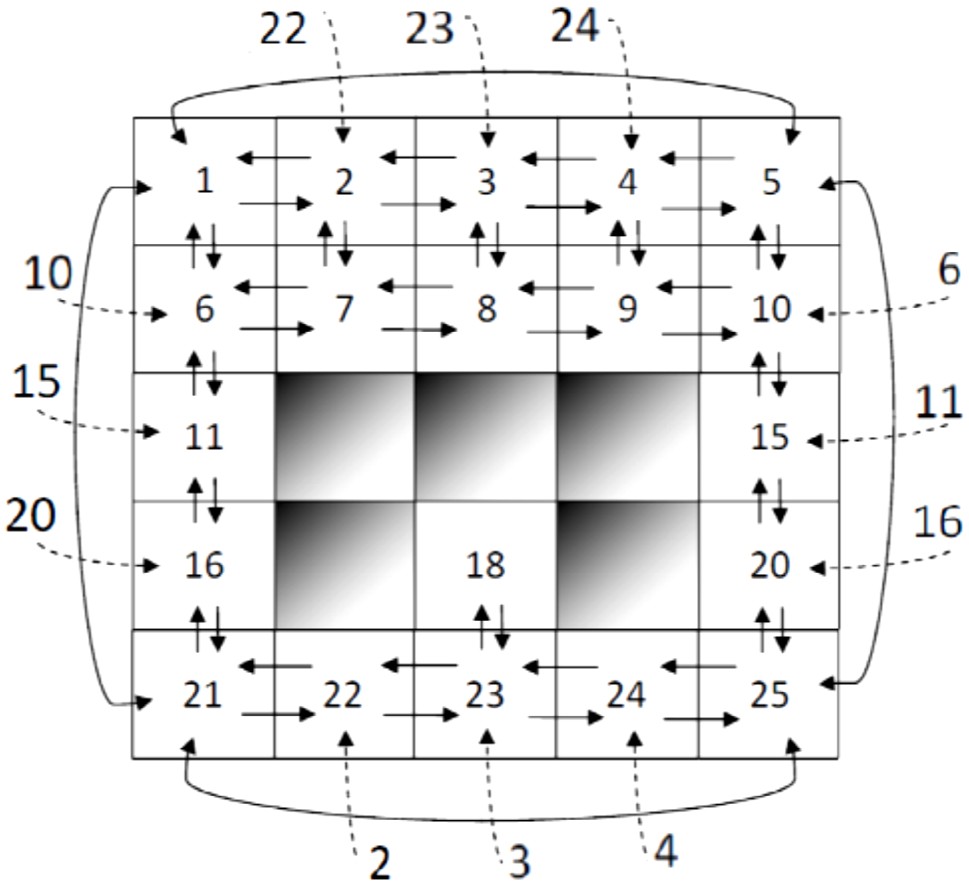


Fig 11: The Agent's Possible Moves

5.3. Problem Setup

In order to understand Attacker's method to achieve best results, lets define $S_G(k)$ to be the state of the Agent at time k. Similarly, $S_T(k)$ is the tracked Target state at time k. While $y(k)$ be the measurement received at time k at the Base Station about the state of the Target T. If we do not consider the Attacker's actions, Base Station would know Target's state $y(k)$ and can calculate Target's possible states at $k+1$ using the Target transition matrix TTM shown in Table 1. However, in case of measurement signal jamming existence, the Agent can estimate Target's possible states by $\text{TTM}^{\Delta k}$ where Δk denotes the difference between k for the current time and k_l, the last time that the Base

Station received Target's state $y(k_l)$. Based on that, the Agent decides the best next move. If the Attacker jammed control signal, the Agent will not move.

The actions available for the Attacker are:

- Jam both the measurement and the control signal
- Jam the measurement signal only
- Jam the control signal only
- No attack

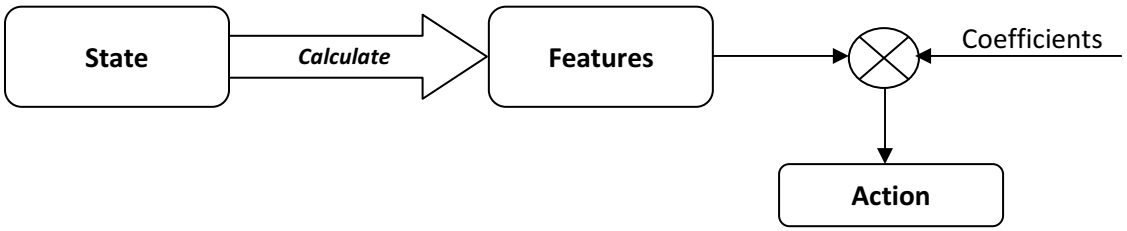
The goal is to maximize the gain, given that jamming comes with costs $c_m and\ c_c$ for jamming the measurement and the control signal, respectively. The Attacker aims to maximize the gain in the future not just at time slot k. A possible solution to calculate the maximum gain with future considerations is to explore all the system's possible states and predetermine the best decision for each state. Given the explosion of the number of the system's possible states, it is so expensive both time and space wise for the Attacker to implement such method. A proposed applicable solution is to construct Bellman equation (see Appendix A) that describe the maximum gain given the system state, the target transition matrix, and the jamming costs. The system state at time k can be described by a vector $s(k)$:

$$s(k) = [S_T(k) \quad S_G(k) \quad k_l \quad \lambda(k)]^T$$

where $S_T(k)$ is the Target state, $S_G(k)$ is the Agent's state and k_l is the elapsed time since the last attack action not Targeting the BS measurement signal, and $\lambda(k)$ is the Target state k_l time slots in the past, i.e. $S_T(k - k_l)$.

The proposed approximate solution can be simplified as in Figure 12 where we aim to conclude the Attacker's best action, given current the system's state. So the approximate solution starts from a set of states, applies different actions, and then evaluates the gain and concludes a set of features and their coefficients – Figure 12-a. These features and their coefficients can be then used at any time slot k to help the Attacker make the best decision – Figure 12-b.

(a) Approximate Solution: Deducing Features and Coefficients



(b) Applying Approximate Solution Policy

Fig 12: Approximate Solution Process (a) and Usage (b)

5.4. Experimental Setup

In this study, we evaluate four different attacking policies based on the benefit that the Attacker might gain using each policy besides no-attack policy. The four policies are:

- Heuristic policy: the Attacker decides which action to take based on the distance between the Agent and the Target. We have tested many different heuristics but we are only going to include the best ones in our results

- Random policy: where the Attacker simply decides randomly which action to take regardless of the state of the system
- Myopic policy: the Attacker pays more effort in calculating the maximum gain by comparing all possible system's states under each action after one time slot and pick the one that has best expected gain
- Improved policy: which uses the approximate solution mentioned in the previous section and is explained in more details in Appendix A. This policy aims to consider future steps and calculate the maximum gain

We utilized the robots described in Section 3.3 to run our experiments to resemble real world factors that might affect the results. In order to run a lot of experiments, we profiled two robots and ran experiments using those two profiles. We also included simulations with different robot error rates.

5.5. Robots Profiling

Experimental evaluation is different from simulations due to physical factors that we need to consider in evaluating algorithms. Simulations are powerful in covering more test cases. In our evaluation, we made a mix to use the advantage of each. We profiled two robots acting as Agents and used those profiles in running a lot of simulations to evaluate the proposed "Improved Policy" versus other policies. In order to create such profile, we have run many experiments for possible moves for each robot individually like the ones shown in figure 11. To explain possible moves, assume that the robot's direction is upwards in figure 11. Possible moves are:

- Short forward (ex. location 6 to location 1)

- Short backward (ex. location 6 to location 11)
- Short left (ex. location 2 to location 1)
- Short right (ex. location 2 to location 3)
- Long forward (ex. location 21 to location 1)
- Long backward (ex. location 1 to location 21)
- Long left (ex. location 5 to location 1)
- Long right (ex. location 1 to location 5)

Each move of the above is prone to errors because of the surface texture and the robot's manufacturing defects. The accuracy of each robot is different from one another. For example, out of 120 Long Right moves, the first robot reached the desired destination successfully only 83 times while missed the desired destination 37 times. Similarly we profiled each of the 8 possible steps for each robot and used these profiles in running many experiments for each case to evaluate the attack policies.

5.6. Experimental Results

As mentioned in the previous sections, there are many factors that play important roles in the results. However, we can categorize experiments into four different sets based on profiling. The first set of experiments is conducted without considering any errors. The second set uses the first robot's profile - R_A. The third one is similar to the second set but uses the other robot's profile - R_B. Finally, the last set of experiments is conducted by error rates in the Agent's moves from 5% to 50% with step 5 in between.

Table 2: Experiments' Different Cases

	Measurement Signal jamming cost	Control Signal jamming cost	Gamma
Case 1	4	3	.90
Case 2	4	3	.95
Case 3	4	3	.99
Case 4	5	3	.95
Case 5	6	3	.95

Table 2 lists the different cases we use to evaluate the five different policies with different cost values for measurement and control signals. An additional parameter that differentiates between the cases is the discount factor γ. γ represents the higher significance of benefits at current step (from Attacker's perspective) than future. Although γ doesn't affect Attacker's decision at any of our subjective policies (at the time of the Attacker taking decisions), but it's an important parameter in defining the system while developing Improved Policy. The total benefit B_t for the Attacker in Improved Policy is according to $B_t = \sum_i \gamma^i * B_i$, where B_i denotes the reward gained by the Attacker at step i.

The features and the coefficients deduced from the approximate solution are related to the system model definition. Table 3 lists the features and coefficients for each case. The first

column is the set of features and each column is the corresponding coefficients in each case. Where$\emptyset(\mathrm{i})$ is the vector of features of length s, i.e. $\emptyset(\mathrm{i}) = [\emptyset_1(\mathrm{i}), \ldots, \emptyset_s(\mathrm{i})]^T$, $\mathrm{d}(\mathrm{x}, \mathrm{y})$ denotes the distance between two locations x and y, and $\mathrm{f}(\mathrm{x})$ denotes to the count of possible moves an Agent or a Target can make given current state.

Table 3: Coefficients for the Different Cases

Features	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 5**
$\mathbf{1}$	-36.62	-39.94	-49.32	-47.7	-47.8
$\mathbf{d}(\mathbf{S_T}(\mathbf{k}), \mathbf{S_G}(\mathbf{k}))$	6.99	4.54	2.32	3.87	3.88
$\mathbf{k_l}$	0.72	0.06	0.09	0.37	0.42
$\emptyset_1(\mathbf{i}) * \emptyset_2(\mathbf{i})$	-0.28	-0.05	-0.04	-0.17	-0.18
$\mathbf{d}(\mathbf{S_T}(\mathbf{k}) - \boldsymbol{\lambda}(\mathbf{k}))$	0.82	0.72	0.38	0.87	0.84
$\mathbf{f}(\mathbf{S_T}(\mathbf{k}))$	0.95	1.03	0.39	0.43	0.48
$\mathbf{f}(\mathbf{S_G}(\mathbf{k}))$	-2.87	-3.34	-0.31	0.02	0.03

We also used different heuristics for the Heuristic algorithms. The results presented in the following sections are the average of 15 different Target's scenarios. A Target scenario is a sequence of 300 moves that we randomly generated offline. When the Attacker uses Heuristic policies to decide which action to take, he depends on the distance between the Agent and the Target. Therefore, we developed eight different heuristics listed in Table 4, where the first row represents the Attacker's decisions. Each row is a different case where values in the row represent the distances between the Agent and the Target that are

the base for the Attacker's actions. We then picked the best heuristic result at each case and added it to the graphs. For example, in Heuristic (1), the Attacker would jam the measurement and the control signals if the distance between the Agent and the Target is less than or equal to 2, while it jams the control signal only if the distance equals 3 or 4 and jams the measurement signal when the distance equals 5 or 6, and does no attack otherwise.

Table 4: Heuristics Values

	Attack Measurement Signal	**Attack Control Signal**	**Attack Both Signals**
Heuristic (1)	6	4	2
Heuristic (2)	4	6	2
Heuristic (3)	4	3	1
Heuristic (4)	3	2	1
Heuristic (5)	2	3	1
Heuristic (6)	4	2	1
Heuristic (7)	5	4	2
Heuristic (8)	4	5	2

a. Robot A - Experiments

Figures 13 and 14 show a comparison between the five algorithms using Robot A (R_A) profile. The rewards graph represents the summation of the Attacker's gain through the whole experiment, which the Attacker is aiming to maximize. The steps graph represents the total number of time slots it took the Agent to find the Target. Although Figure 14 show that an Attacker can delay the Agent from finding the Target but this comes at a

high cost; this can be seen in figure 13. From figure 14, the Agent find the Target in the same count of steps under Heuristic (7) policy as the Attacker mounts jamming attacks at large distances between the Agent and the Target. Figure 13 show that an Attacker using the Improved Policy gets the best rewards across algorithms.

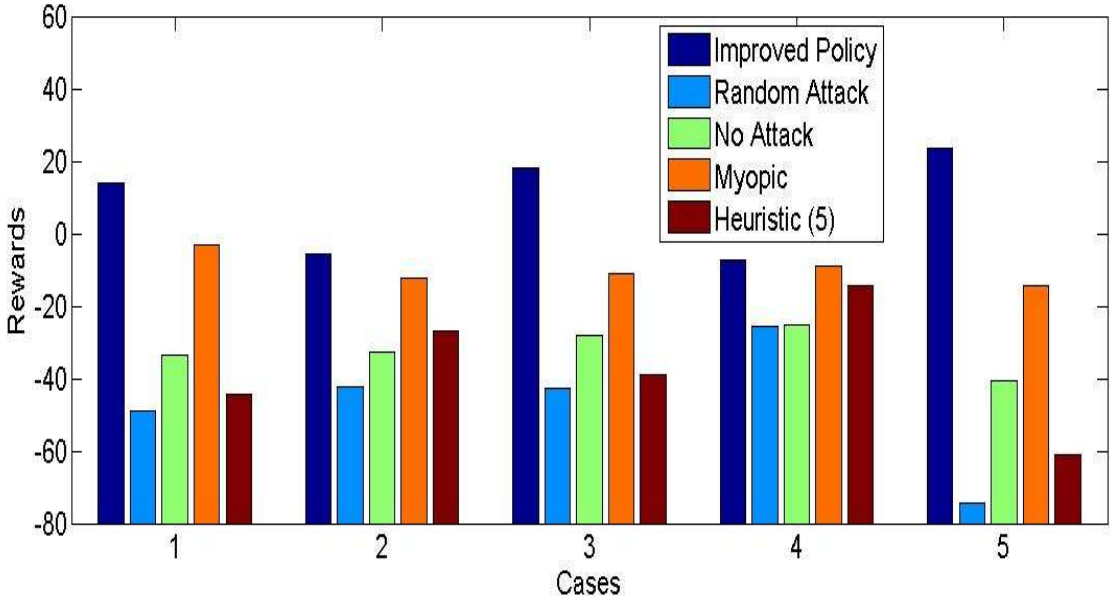


Fig 13: Attacker's Rewards Using Robot A

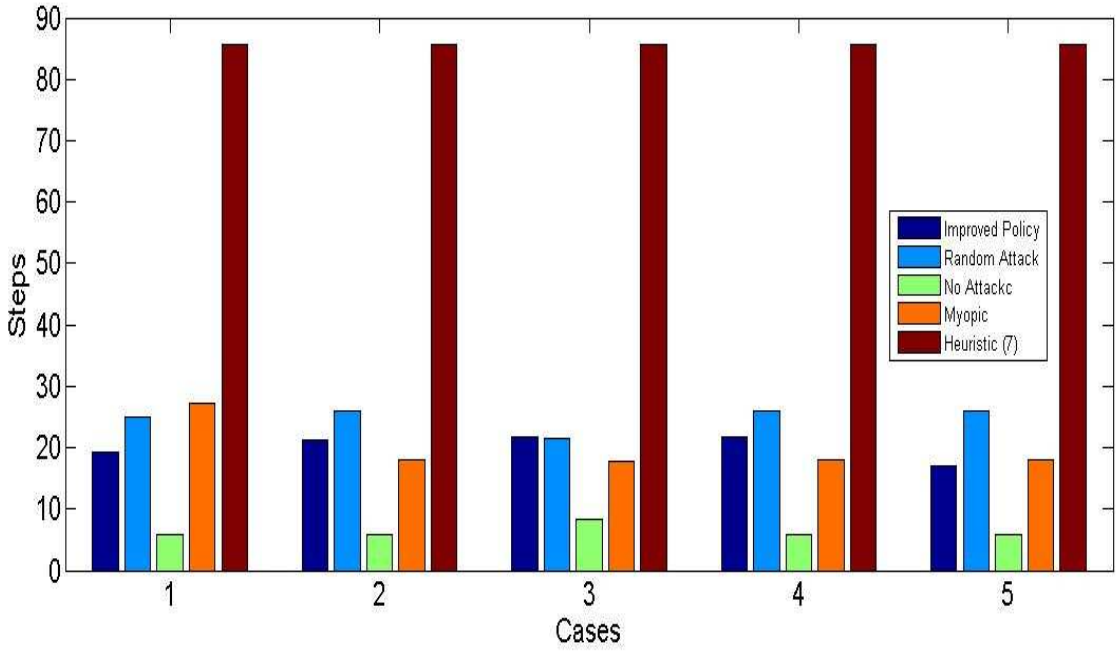


Fig 14: Agent Total Steps Using Robot A

b. Robot B – Experiments

Figures 15 and 16 show a comparison between the five algorithms using Robot B (R_B) profile. We can see that Improved Policy is the best choice for the Attacker. Myopic Policy slightly better rewards in the second and fourth cases imply that features and/or coefficients deduced from approximate solution are not optimal.

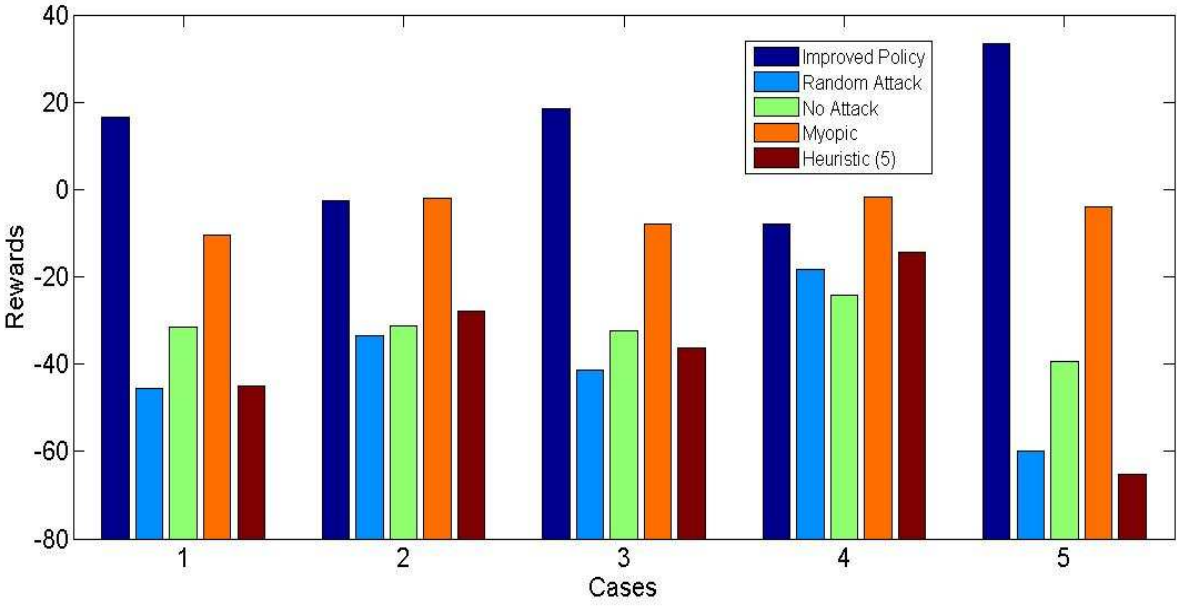


Fig 15: Attacker's Rewards using Robot B

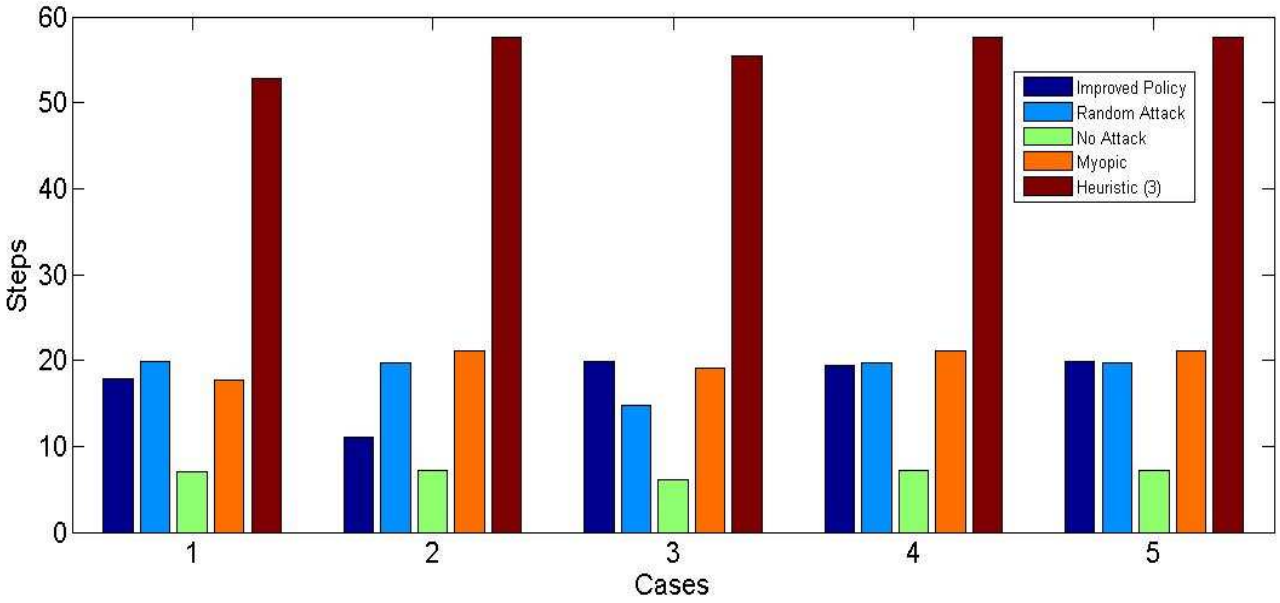


Fig 16: Agent Total Steps using Robot B

c. Simulation Experiments – No Errors

Figures 17 and 18 show the simulation results (rewards and steps) for the five algorithms. The rewards in cases 1,3, and 5 in Figure 17 show the significant effect that the Attacker gains from using Improved Policy. Myopic Policy competency with Improved Policy in cases 2 and 4 in simulation results assures that the provided features and coefficients can be improved.

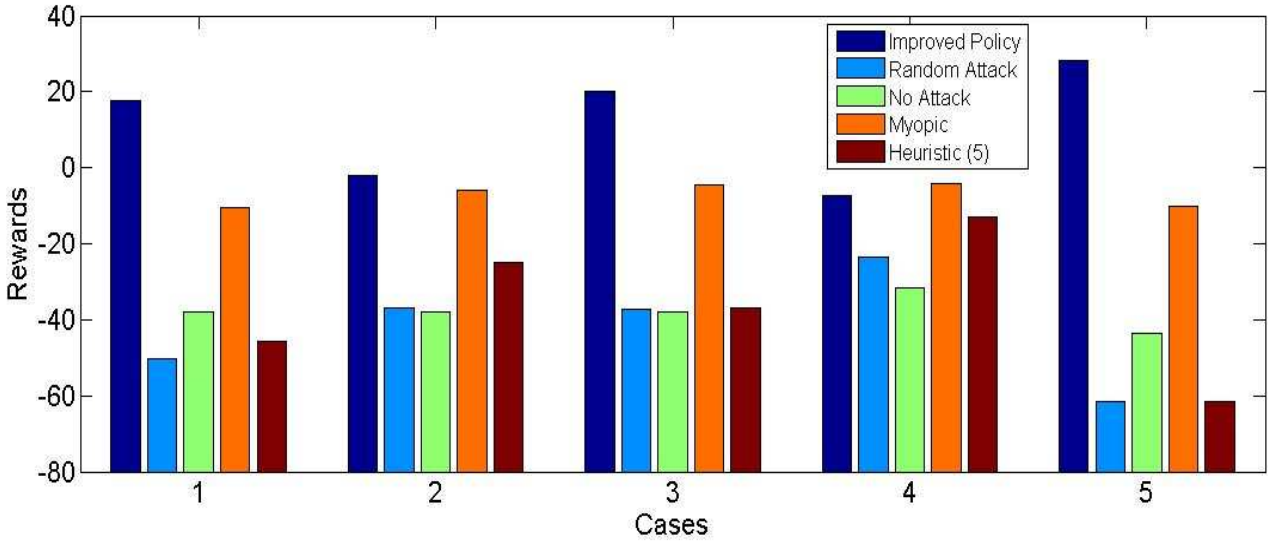


Fig 17: Attacker's Rewards - Simulations

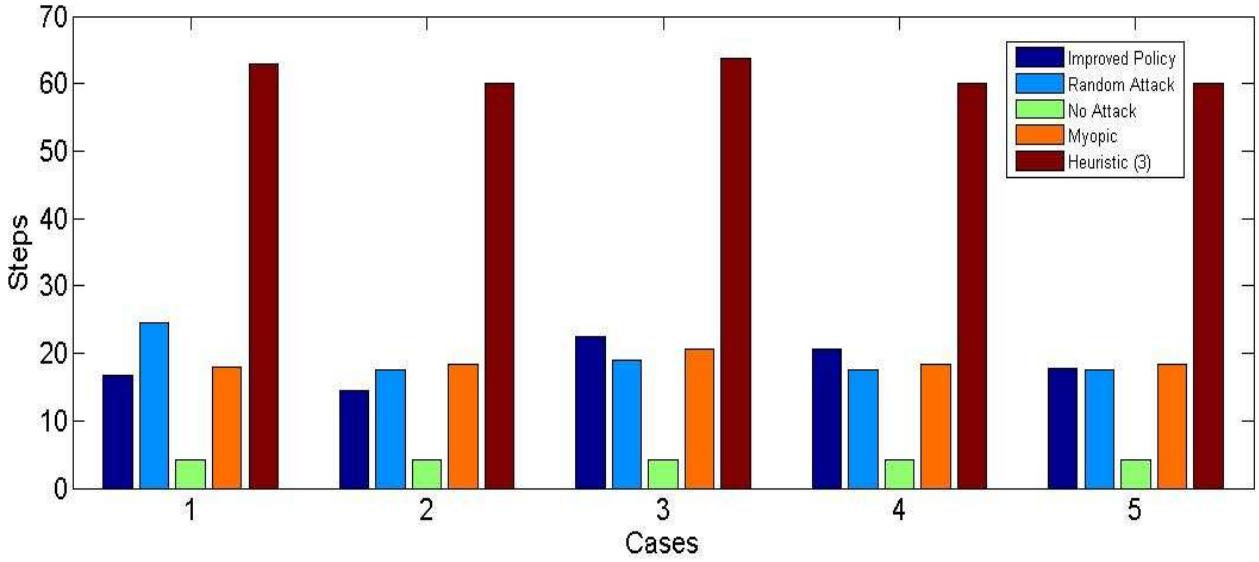


Fig 18: Agent Total Steps - Simulations

d. Simulation Experiments – Different Error Rates

In these experiments we applied ten different error rates on the Agent's moves from a location to another. East graph in Figures 19 through 23 presents the Attacker's rewards

in one case from table 2 under the five policies. We can see that there is no big discrepancy between the Agent's error rates.

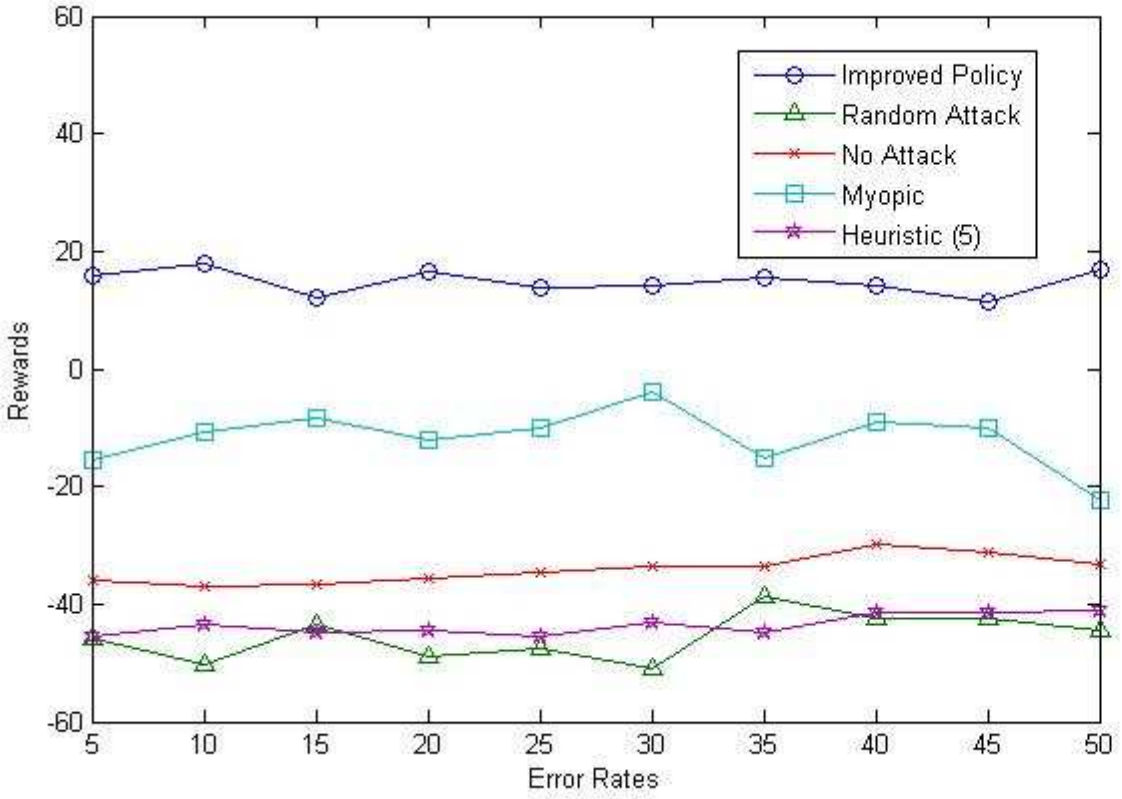


Fig 19: Attacker's Rewards – Case 1 – Error Rates

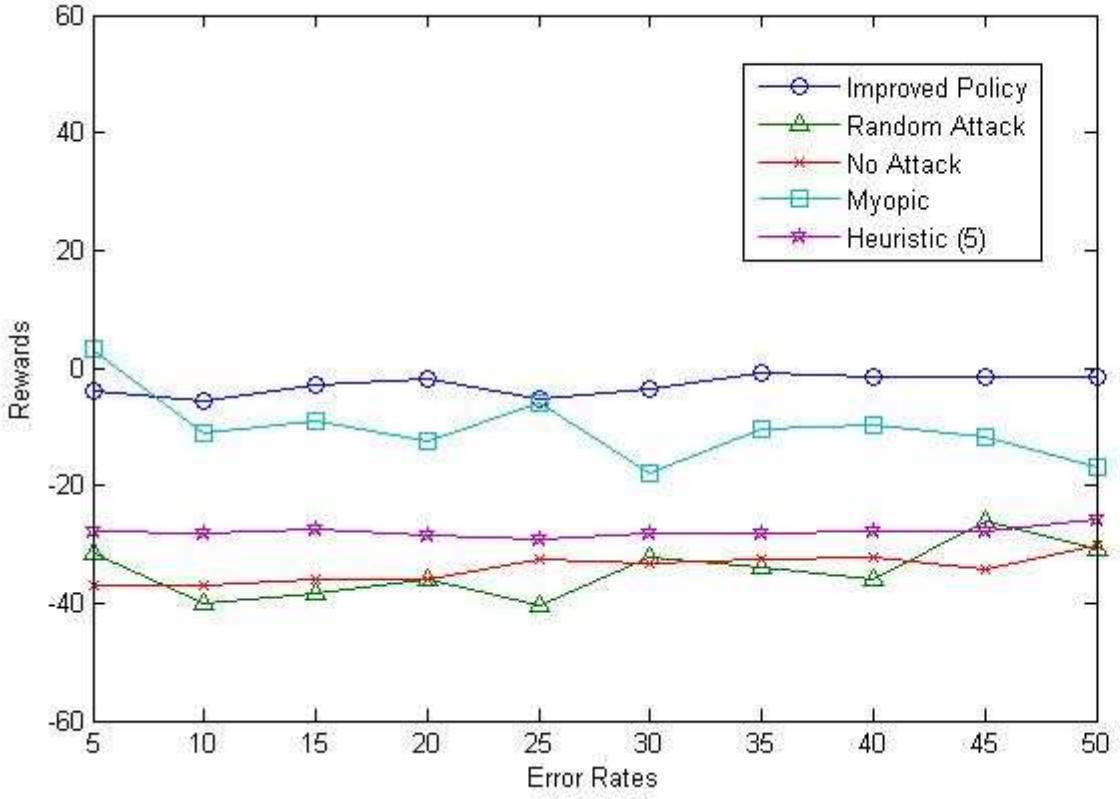


Fig 20: Attacker's Rewards – Case 2 – Error Rates

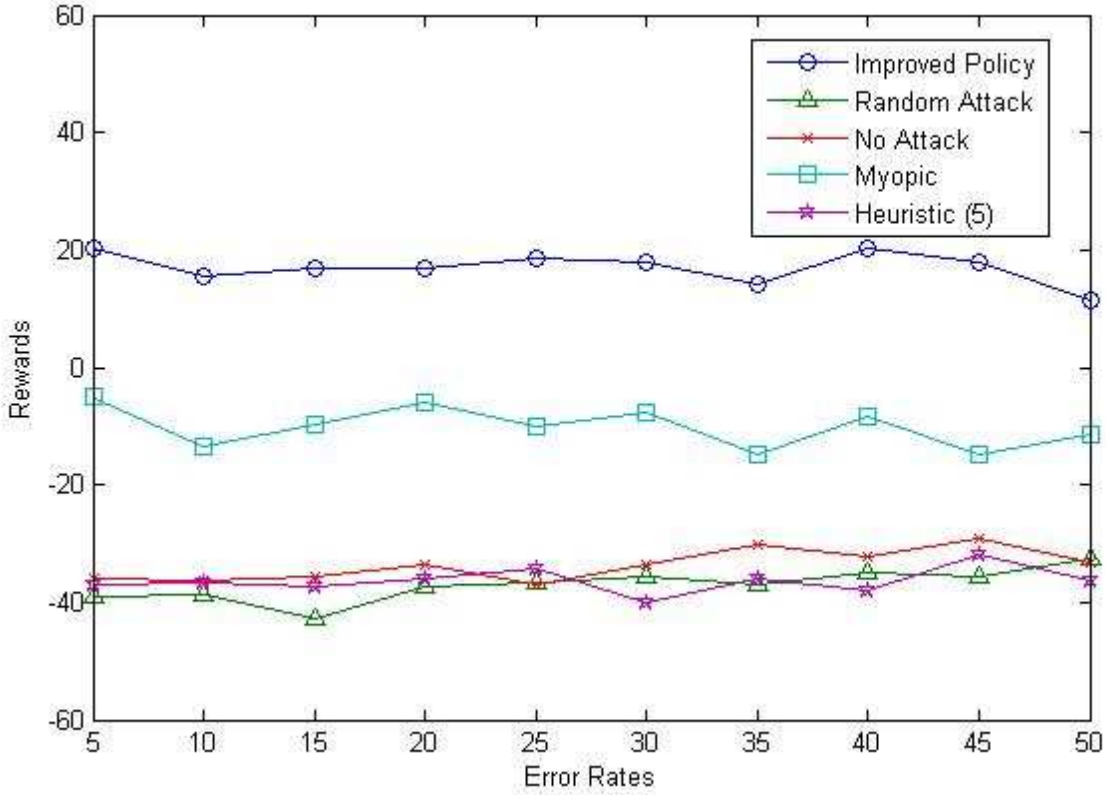


Fig 21: Attacker's Rewards – Case 3 – Error Rates

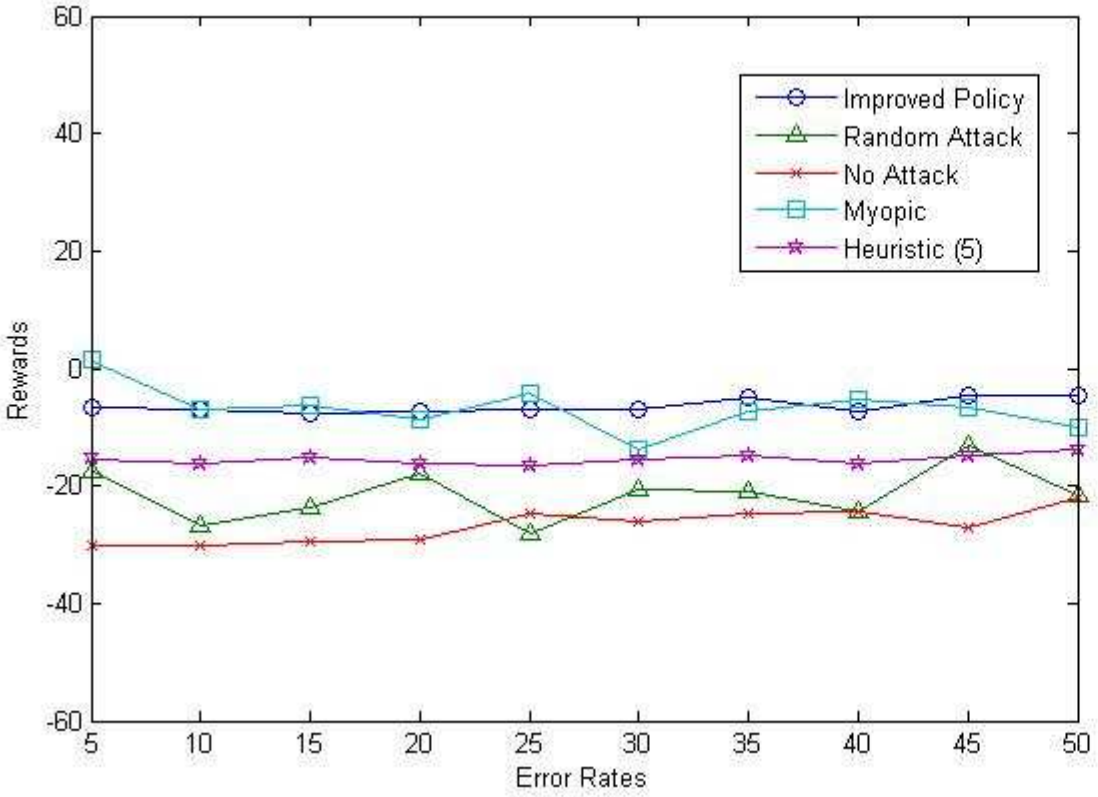


Fig 22: Attacker's Rewards – Case 4 – Error Rates

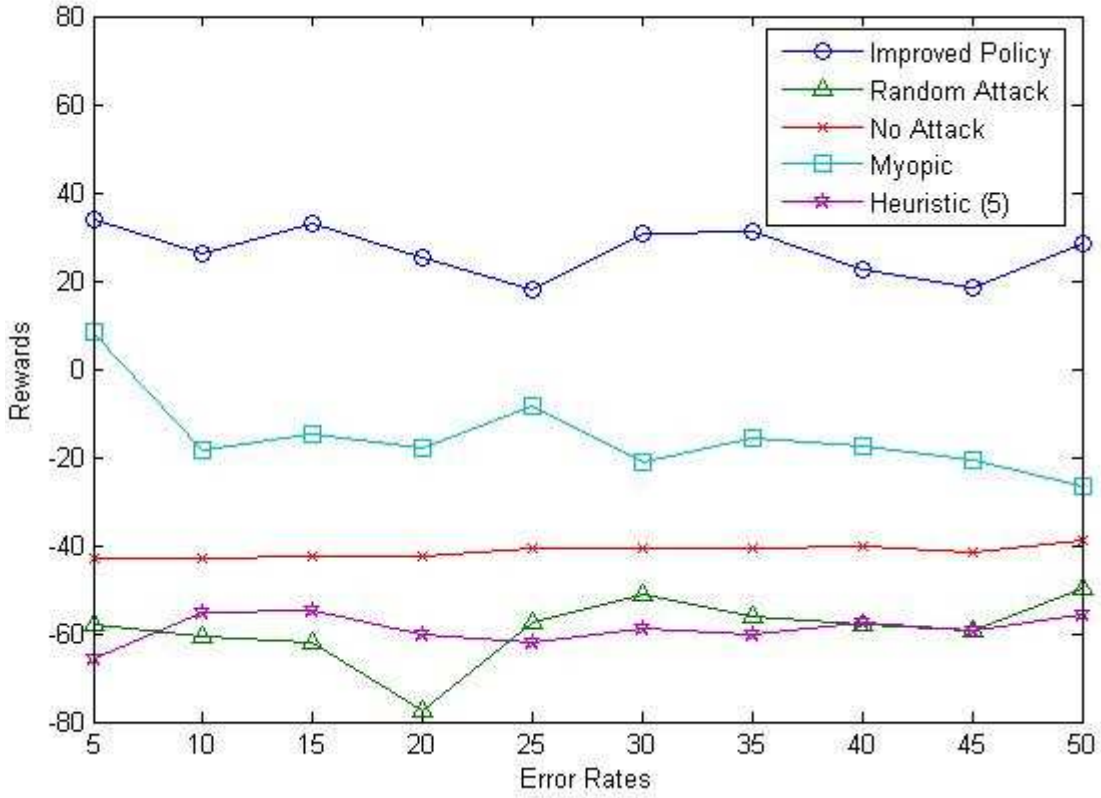


Fig 23: Attacker's Rewards – Case 5 – Error Rates

5.7. Conclusion

This study raises the awareness of possible smart attacking algorithms that might be applied to Cyber Physical Systems applications. From the Attacker's perspective, we applied the Improved Policy that utilizes the approximate solution that makes Attacker's best action obtainable through solving a linear equation (features and coefficients) of the system's current state. Through experiments and implementation, we have shown that an Attacker that uses the improved policy gains higher rewards. However, it's so important to have the correct features and coefficients. Three other policies have been presented include Myopic, Random, and different Heuristics.

CHAPTER VI

CONCLUSION AND FUTURE WORK

6.1. Conclusion

In this thesis, we explored security issues for the next generation of Cyber-physical systems with a focus on target tracking applications. Target tracking applications in CPSs introduce a level of automation in managing physical processes that has previously not been achieved. An important characteristic of CPSs is the level of detail of information they collect (about the physical process) in order to carry out their tasks. This work focused on assessing the impact of low-level jamming attacks and new intelligent jamming attack policies on target tracking applications in CPSs to uncover vulnerabilities. Our studies were conducted through experimental implementation, using SRV-1 robots.

We evaluated the system's behavior when it receives a subset of the measurement and the control signals. This assessment reveals surprising results that the system using some controllers – such as PI controller – might benefit from low level jamming attacks. However, this has a legit explanation that these dropped packets result in making the controller greedier to achieve the goal.

We assessed the system's overall performance under different jamming scenarios that an intelligent attacker might develop. This study has the assumption that the attacker has the ability to monitor the communication channel and launch several attacks to drop/delay control and/or measurement packets. We applied different policies on target tracking case study. The provided Improved Policy is proven to be better than other policies (Myopic, Heuristic, and Random) in terms of maximizing the attacker's gain.

6.2. Future Work

Security of Cyber-Physical Systems is an ongoing research area that needs more research efforts as considerable security challenges still exist. In this research work, we want to apply the attacking paradox shown in the first study on different controllers (eg. PFD) and utilize it in more applications. While we need to work on assessing other possible threatening policies besides improved policy presented in the second study. Meanwhile, efforts are required in finding ways that mitigate the effect of smart attacking on CPSs.

APPENDIX A

BELLMAN EQUATION

A.1. Bellman’s Principle of Optimality

The term 'Bellman equation' usually refers to the dynamic programming equation associated with discrete-time optimization problems. Bellman equation is also known as dynamic programming equation as it is a necessary condition for dynamic programming mathematical optimization method. It’s based on Bellman’s Principle of Optimality: An optimal policy has the property that, whatever the initial state and decision (i.e., control) are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision [6].

Bellman equation writes the value of a decision problem at a certain point in time in terms of the payoff from some initial choices and the value of the remaining decision problem that result from those initial choices. This breaks a dynamic optimization problem into simpler sub-problems, as Bellman's Principle of Optimality prescribes.

A.2. Optimization Problem Solution

Optimization problem is the problem of achieving an objective in optimal way. Each optimization problem has some objective such as minimizing travel time, minimizing cost, maximizing profits, or maximizing utility, etc. The objective is usually written in

mathematical form to describe it. This mathematical function is called the objective function.

Dynamic programming breaks a multi-period planning problem into simpler steps at different points in time. Therefore, it requires keeping track of how the decision situation is evolving over time. The information about the current situation which is needed to make a correct decision is called the state.

The Bellman equation can be solved by backwards induction, either analytically in a few special cases, or numerically on a computer. Numerical backwards induction is applicable to a wide variety of problems, but may be infeasible when there are many state variables, due to the curse of dimensionality.

By calculating the first-order conditions associated with the Bellman equation, and then using the envelope theorem to eliminate the derivatives of the value function, it is possible to obtain a system of difference equations or differential equations called the 'Euler equations'. Standard techniques for the solution of difference or differential equations can then be used to calculate the dynamics of the state variables and the control variables of the optimization problem.

The method of undetermined coefficients, also known as 'guess and verify', can be used to solve some infinite-horizon, autonomous Bellman equations.

A.3. Approximate Solution

The proposed solution is to solve Bellman equation using an approximate policy iteration method that adopts a parametric cost-to-go approximation, and then attacker can solve a

system of linear equation to evaluate the cost function. Developed linear equation's variable may include – but not limited to – distance between agent and target, k_l, and distance between target's current location $S_T(k)$ and its location k_l time slots ago $S_T(k-k_l)$. The last example variable represents the distance between target's current location and last target's location that BS is aware of.

The system used to obtain the coefficients is beyond the scope of this thesis and is developed by George Atia and Mina Guirguis. However, the algorithm can be described in few steps.

1. Select some set of representative states *I*
2. Start from an arbitrary suboptimal policy μ
3. For each state $i \in I$ evaluate the policy μ using simulated trajectories. Let $M(i)$ denote the number of trajectories used for starting state *i*. Let the *m*-th sample of the cost function that corresponds to the policy μ be denoted $c(i,m)$
4. Approximate the cost function using parametric representation $J(i,r)$. We obtain r by solving a least squares problem, i.e.

$$\sum_{i \in I} \sum_{m=1}^{M(i)} \emptyset(i)\big(\emptyset(i)^T r - c(i,m)\big) = 0$$

   where $\emptyset(i)$ is the vector of features of length *s*, i.e. $\emptyset(i) = [\emptyset_1(i), \dots, \emptyset_s(i)]^T$
5. Do policy improvement then repeat from step 2

For the different cases listed in Table 2, the above algorithm was applied and the vector of features *r* was provided and therefore attacker can easily calculate the cost for each decision and choose the best one. In our study, we have seven features (s = 7):

$$\emptyset(i) = \begin{bmatrix} 1 \\ d\left(\mathrm{S_T(k)}, \mathrm{S_G(k)}\right) \\ k_l \\ \emptyset_1(i) * \emptyset_2(i) \\ d(S_T(k) - \lambda(\mathrm{k})\,) \\ f\left(\mathrm{S_T(k)}\right) \\ f\left(\mathrm{S_G(k)}\right) \end{bmatrix}$$

where $d(\mathrm{x}, y)$ denotes the distance between two locations x and y, and $f(x)$ denotes to the count of possible moves available to the Agent or the Target at a given state.

BIBLIOGRAPHY

[1] Pelechrinis, Iliofotou, *Denial of Service Attacks in Wireless Networks: The case of Jammers*, 2006

[2] Hespanha, Naghshtabrizi, *A Survey of Recent Results in Networked Control Systems*, Proceedings of the IEEE | Vol. 95, No. 1, January 2007

[3] David Thuente, and Mithun Acharya, *Intelligent Jamming in Wireless Networks with Applcations to 802.11b and other Networks*, In Proceedings of the 25th IEEE, MILCOM 2006

[4] M. Acharya and D. Thuente, *Intelligent Jamming Attacks, Counterattacks and (Counter)2 Attacks in 802.11b Wireless Networks*, in Proceedings of the OPNETWORK-2005 Conference, Washington DC, USA, August 2005

[5] Pajic, Mangharam, *WisperNet: Anti-Jamming for Wireless Sensor Networks*, University of Pennsylvania, 2008

[6] Thomas J. Sargent, *Dynamic Macroeconomic Theory*

[7] W. Xu, W. Trappe, Y. Zhang, and T. Wood, *The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks*, MobiHoc, 2005

[8] Wood, Stankovic, Son, *JAM: A jammed-area mapping service for sensor networks*, In 24th IEEE Real-Time Systems Symposium

[9] Proakis, *Digital Communications*, McGraw-Hill, 4th edition, 2000

[10] Raya, Hubaux, Aad, *Domino: a system to detect greedy behavior in ieee 802.11 hotspots*, MobiSYS '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services, ACM Press, 2004

[11] D. Curtis Schleher, *Electronic Warfare in the Information Age*, MArtech House, 1999

[12] Bellardo, Savage, *802.11 denial-of-service attacks: Real vulnerabilities and practical solutions*, In Proceedings of the USENIX Security Symposium, 2003

[13] Kyasanur, Vaidya, *Detection and handling of mac layer misbehavior in wireless networks*, In Proceedings of the IEEE International Conference on Dependable Systems and Networks, 2003

[14] Noubir, Lin, *Low-power DoS attacks in data wireless lans and countermeasures*, SIGMOBILE Mobile Computer Communication, 2003

[15] W. Xu, T. Wood, W. Trappe, and Y. Zhan, *Channel Surfing and Spatial Retreats: Defenses Against Wireless Denial of Service*, in Proc. 2004 ACM Wksp. Wireless Security, 2004

[16] T. Samad and A.M. Annaswamy, *The Impact of Control Technology*, 2011

[17] Krishna Venkatasubramanian. *Security Solutions for Cyber-Physical Systems*, 2009. PhD Thesis

[18] Q. Tang. *Thermal-aware scheduling in environmentally coupled cyber-physical distributed systems*, 2008. PhD Thesis

[19] N. Gaddam, G. Kumar, and A. Somani. *Securing Physical Processes against Cyber Attacks in Cyber-Physical Systems*, National Workshop for Research and High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation & Rail, Washington DC, November 2008

[20] F. Mueller. *Challenges for cyber-physical systems: Security, timing analysis and soft error protection*. In National Workshop on High Confidence Software Platforms for Cyber-Physical Systems: Research Needs and Roadmap (HCSP-CPS), 2006

[21] B. Krebs. *Cyber Incident Blamed for Nuclear Power Plant Shutdown*. Washington Post, http://www.washingtonpost.com/wp-dyn/content/article/2008/06/05/AR2008060501958.html, June 2008

[22] P. P. Tsang and S. W. Smith. *YASIR: A low-latency high-integrity security retrofit for legacy SCADA systems,* In 23rd International Information Security Conference (IFIC SEC), pages 445–459, September 2008

[23] A. K. Wright, J. A. Kinast, and J. McCarty. *Low latency cryptographic protection for SCADA communications*, In Applied Cryptography and Network Security (ACNS), pages 263–277, 2004

[24] S. Hurd, R. Smith, and G. Leischner. *Tutorial: Security in electric utility control systems*. In 61st Annual Conference for Protective Relay Engineers, pages 304–309, April 2008

[25] Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, and S. Shankar Sastry. *Challenges for Securing Cyber Physical Systems*. Workshop on Future Directions in Cyber-physical Systems Security, DHS, 23, July, 2009

[26] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes. *Using model-based intrusion detection for SCADA networks.* In Proceedings of the SCADA Security Scientific Symposium, Miami Beach, FL, USA, 2007 2007

[27] Surveyor, *Surveyor SRV-1 Open Source Mobile Robot,* http://www.surveyor.com/SRV info.html

[28] Ravi Akella, Han Tang, and Bruce M. McMillin, *Analysis of information flow security in cyber–physical systems*, Elsevier – International Journal of Critical Infrastructure Protection, Volume 3, Issues 3-4, December 2010

VITA

Emad Guirguis was born in Alexandria, Egypt, on January 17, 1983, the son of Nabil Guirguis and Netris Iskandar. He grew up in Alexandria, Egypt where he studied Computer Science and Automatic Control at Alexandria University. After completing his Bachelor's degree in Engineering at 2004, he worked at ITWorx as a Software Engineer for three months then joined the army between 2004 and 2007 to fulfill the military service. During the following years he was employed as a Software Engineer at Bibliotheca Alexandrina – renewed Library of Alexandria – in Alexandria, Egypt. In August 2009, he enrolled into the Graduate College program in Texas State University–San Marcos seeking Master's degree in Computer Science.

Publications

- Emad Guirguis, Chris Page, and Mina Guirguis, "SKWeak Attacks on Path Splicing: Vulnerability Assessment and Defense Mechanisms", IEEE GLOBECOM, Miami, Florida, December 2010
- Chris Page, Mina Guirguis, and Emad Guirguis, "Performance Evaluation of Path Splicing on the GEANT and the Sprint Networks", Computer Networks - Elsevier, September 2011

Permanent email address: emad.attalla@gmail.com

This thesis was typed by Emad Guirguis.