2009-12

Sensor interceptor operational policy optimization for maritime interdiction missions

Rozen, Nir

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/4370

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SENSOR-INTERCEPTOR OPERATIONAL POLICY OPTIMIZATION FOR MARITIME INTERDICTION MISSIONS

by

Nir Rozen

December 2009

Thesis Advisor:	Johannes O. Royset
Second Reader:	Moshe Kress

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Sensor-Interceptor Operational Policy Optimization for Maritime Interdiction Missions			**5. FUNDING NUMBERS**
6. AUTHOR(S) Nir Rozen			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT

Maritime Interdiction Missions (MIM) are of great interest and high operational importance to the U.S. Navy, the U.S. Coast Guard, and allied forces. The MIM scenario discussed in this thesis includes an area of interest with multiple neutral and hostile vessels moving through this area, and an interdiction force consisting of an unmanned aerial vehicle (UAV) and an intercepting vessel, whose objectives are to search, identify, and intercept hostile vessels within a given time frame. In this thesis we develop Stochastic Dynamic Programming models, which represent the MIM scenario. While a theoretical method of producing an optimal decision policy for the interdiction force is presented in this thesis, it is shown that such computation is intractable. The models developed in this study are used to analyze and evaluate the performance of a heuristic decision policy that we recommend to be applied by the interdiction force. Based on a numerical case study, which includes several representative MIM scenarios, we show that the number of intercepted hostile vessels following the heuristic decision policy is at least 60% of the number of hostile vessels intercepted following the optimal decision policy. Based on the results of the heuristic performance in the numerical case studies, we recommend the implementation of our suggested heuristic in an operational decision aid for Maritime Interdiction Missions.

14. SUBJECT TERMS Stochastic model, Dynamic programming , Stochastic optimization, Maritime Interdiction Missions, MIM, Maritime Interdiction Operations, MIO			**15. NUMBER OF PAGES** 89
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

SENSOR-INTERCEPTOR OPERATIONAL POLICY OPTIMIZATION FOR MARITIME INTERDICTION MISSIONS

Nir Rozen
Captain, Israeli Air Force
B.Sc., The Hebrew University of Jerusalem, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2009

Author: Nir Rozen

Approved by: Johannes O. Royset
Thesis Advisor

Moshe Kress
Second Reader

Robert F. Dell
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Maritime Interdiction Missions (MIM) are of great interest and high operational importance to the U.S. Navy, the U.S. Coast Guard, and allied forces. The MIM scenario discussed in this thesis includes an area of interest with multiple neutral and hostile vessels moving through this area, and an interdiction force consisting of an unmanned aerial vehicle (UAV) and an intercepting vessel, whose objectives are to search, identify, and intercept hostile vessels within a given time frame. In this thesis, we develop Stochastic Dynamic Programming models, which represent the MIM scenario. While a theoretical method of producing an optimal decision policy for the interdiction force is presented in this thesis, it is shown that such computation is intractable. The models developed in this study are used to analyze and evaluate the performance of a heuristic decision policy that we recommend to be applied by the interdiction force. Based on a numerical case study, which includes several representative MIM scenarios, we show that the number of intercepted hostile vessels following the heuristic decision policy is at least 60% of the number of hostile vessels intercepted following the optimal decision policy. Based on the results of the heuristic performance in the numerical case studies, we recommend the implementation of our suggested heuristic in an operational decision aid for Maritime Interdiction Missions.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
- **A. OVERVIEW....1**
- **B. BENEFITS OF STUDY....1**
- **C. RELATED WORK....1**
- **D. THESIS ORGANIZATION....2**
- **E. SCENARIO....2**
- **F. ASSUMPTIONS....5**
  - **1. Arrival and Movement of Targets and Neutrals....5**
  - **2. Sensors Capabilities....6**
  - **3. Separation of Detected Objects....7**
  - **4. Finite Time Horizon....7**

II. MODELS DEVELOPMENT: MAIN PARTS....9
- **A. DYNAMIC PROGRAMMING FORMULATION....9**
  - **1. State....9**
  - **2. Decision....10**
  - **3. Information....10**
  - **4. State Transition....11**
  - **5. Reward....11**
  - **6. Timeline....12**
  - **7. Bellman Equation....12**
- **B. METHODOLOGY OUTLINE....13**
- **C. SUGGESTED HEURISTIC....14**
- **D. REWARD CALCULATION USING HEURISTICS POLICY....14**
- **E. UPPER BOUND PROBLEM....15**

III. MODELS DEVELOPMENT: DETAILS....17
- **A. ORIGINAL PROBLEM....17**
  - **1. Probabilities and Object Recognition Techniques....17**
    - ***a. Initial Pre-Search Probabilities: Uninterrupted Steady-State....17***
    - ***b. Markov Updates....18***
    - ***c. Recognizer Updates....20***
  - **2. The State-Transition Function....22**
  - **3. Information Probability Mass Function....25**
  - **4. Bellman Equation Calculation....30**
- **B. UPPER BOUND PROBLEM....33**
  - **1. States....33**
  - **2. Decision....34**
  - **3. Information....34**
  - **4. State Transition....34**
  - **5. Reward....35**
  - **6. Bellman Equation....35**

7. Bellman Equation Calculation 35
C. REWARD CALCULATION USING HEURISTIC'S POLICY 37

IV. NUMERICAL CASE STUDY 41
A. OVERVIEW 41
B. SPECIFIC SCENARIOS AND DATA 41
1. Baseline Scenario 41
2. Additional Scenarios 46
C. RESULTS AND INSIGHTS 46
1. Performance of the Heuristic Decision Policy 47
2. No Discounting of Rewards 49
3. Low Quality Signature Recognition 50
4. How Often Should We Call the Interceptor? 52
5. Extended Time Horizon 54
6. 8-by-8 AOI 54

V. CONCLUSIONS AND RECOMMENDATIONS 57
A. CONCLUSIONS 57
B. SUGGESTED WORK AHEAD 58

APPENDIX. MATLAB IMPLEMENTATION OVERVIEW 59

LIST OF REFERENCES 61

INITIAL DISTRIBUTION LIST 63

LIST OF FIGURES

Figure 1. An example of an AOI with multiple neutrals (*N*) and targets (*T*), a Recognizer (*R*) and an Interceptor (*I*). 4
Figure 2. Summary of CONOPS of the interdiction force 5
Figure 3. Timeline of the dynamic programming formulation 12
Figure 4. Timeline of state transitions....................23
Figure 5. A state example ($t = 0$)....................38
Figure 6. A state example ($t = 1$) 39
Figure 7. A state example ($t = 18$) 39
Figure 8. A heuristic run summary example with two intercepted targets ($T = 48$) 40
Figure 9. The baseline scenario AOI....................42
Figure 10. Partial Markov transition prob. of neutrals (right) and targets (left) 43
Figure 11. Baseline scenario results....................48
Figure 12. No-discounting scenario results....................49
Figure 13. Poor signature recognition scenario results 51
Figure 14. Sensitivity to boarding time....................53

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Neutrals Markov transition probabilities matrix *P* 44
Table 2. Targets Markov transition probabilities matrix *Q* 44
Table 3. Recognizer’s all AC pairs transition times 45
Table 4. Interceptor’s all AC pairs transition times...45
Table 5. Baseline scenario results..48
Table 6. No-discounting scenario results...49
Table 7. Poor signature recognition scenario results 51
Table 8. Sensitivity to boarding time (*x* marks scenarios which have not been calculated)...53
Table 9. MATLAB code filenames and descriptions 59

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AC	Area-Cell
AOI	Area of interest
CONOPS	Concept of Operations
MIM	Maritime Interdiction Mission
PMF	Probability Mass Function
R.V.	Random variable
UAV	Unmanned Aerial Vehicle
a	An AC
$\mathcal{A}$	The set of ACs in the AOI
$\mathcal{A}_0$	The area outside of the AOI
$\lVert a - a' \rVert$	The Euclidean metric for distance between AC a and AC a'
B	Original Problem
$\overline{B}$	Upper Bound Problem
c	Reward function
$\overline{c}$	Reward function in the Upper Bound Problem
$\tilde{c}$	Reward function (with necessary arguments only)
d	Random variable representing detection of an object
E	The boundary of the AOI
f	Random variable representing flagging of an object
g	Number of sensor glimpses
$\mathcal{H}$	Probability update operator for tracking and interception phase
$\mathcal{H}^k$	The operator $\mathcal{H}$ applied k times

$(\mathcal{H}\pi)_a$	The component of the $\mathcal{H}\pi$ vector corresponding to AC a
$(\mathcal{H}\theta)_a$	The component of the $\mathcal{H}\theta$ vector corresponding to AC a
i	Interceptor location (part of the state)
$\bar{i}$	Interceptor location in the Upper Bound Problem
i_w	Interceptor location when the decision is fathomed
$\bar{i}_w$	Interceptor location when the decision is fathomed in the Upper Bound Problem
$I_{velocity}$	Interceptor velocity
j	An AC (usually used to denote the AC to which a tracked object has moved)
l	An AC (usually used to denote an AC in the boundary of the AOI)
M	Probability threshold for flagging a tracked object as a likely target
n	Number of sensor glimpses which reports "neutral"
n_j^*	Maximal number of sensor glimpses which reports "neutrals" given a tracked object has moved to AC j for which that object will be flagged as a likely target
N_a	An event representing AC a contains a neutral
O_a	An event representing AC a contains an object (neutral or target)
P	Markov transition matrix for neutrals
$P(a',a)$	Probability of single time step transition of neutral from AC a' to AC a
$P^k(a',a)$	Probability of k time steps transition of neutral from AC a' to AC a
Q	Markov transition matrix for targets
$Q(a',a)$	Probability of single time step transition of target from AC a' to AC a
$Q^k(a',a)$	Probability of k time steps transition of target from AC a' to AC a

r	Recognizer location (part of the state)
r_w	Recognizer location when decision is fathomed (part of the information)
$\overline{r}$	Recognizer location (part of the state) in the Upper Bound Problem
$\overline{r}_w$	Recognizer location when decision is fathomed (part of the information) in the Upper Bound Problem
$R_{velocity}$	Recognizer velocity
$\mathbb{R}$	Real numbers
s	The state
$\mathcal{S}$	The state space
$\overline{s}$	The state in the Upper Bound Problem
$\overline{\mathcal{S}}$	The state space of the Upper Bound Problem
s^M	The state transition function
$\overline{s}^M$	The state transition function in the Upper Bound Problem
$\tilde{s}^M$	The state transition function (with necessary arguments only)
t	Current time (part of the state), also used for general time
$\overline{t}$	Current time (part of the state) in the Upper Bound Problem
Δt_w	Time between making a decision and when that decision is fathomed (part of the information)
$\Delta \overline{t}_w$	Time between making a decision and when that decision is fathomed (part of the information) in the Upper Bound Problem
T	Time horizon of the Maritime Interdiction Mission scenario
$T^R_{r,a}$	Travel time of Recognizer from AC r to AC a
$T^I_{i,a}$	Travel time of Interceptor from AC i to AC a
T_a	An event representing AC a contains a target

$1-u$	Single glimpse false negative probability of identifying a target
$1-v$	Single glimpse false positive probability of identifying a target
V_a	An event representing AC a void of objects
$V(s)$	Value of state s
$\bar{V}(\bar{s})$	Value of state $\bar{s}$ in the Upper Bound Problem
$V^H(s)$	Value of state s following the heuristic decision policy
w	The information
$\bar{w}$	The information in the Upper Bound Problem
$\mathcal{W}$	The information space
x	The decision (which AC the Recognizer visits next)
$\bar{x}$	The decision (which AC the Recognizer visits next) in the Upper Bound Problem
x^H	The decision (which AC the Recognizer visits next) following the heuristic decision policy
Y_i	Random variable representing the state of each AC in the AOI
z_w	A Bernoulli random variable representing whether a target was intercepted following a decision (part of the information)
$\bar{z}_w$	A Bernoulli random variable representing whether a target was intercepted following a decision (part of the information) in the Upper Bound Problem
α_l	Single time step probability of a neutral arrival to AC l in the boundary of the AOI
β_l	Single time step probability of a target arrival to AC l in the boundary of the AOI
γ	Discounting factor for intercepted targets
π	Probability vector representing the probability of a neutral in each AC in the AOI

π_a	The component of probability vector π corresponding to AC a
π^0	Probability vector representing the steady-state probability of a neutral in each AC in the AOI
π_a^0	The component of probability vector π^0 corresponding to AC a
π^t	Probability vector π at time t
π_a^t	The component of probability vector π^t corresponding to AC a
$\hat{\pi}\left(t-t',\pi^{t'}\right)$	A function updating the probability vector $\pi^{t'}$ from time t' to time t
$\pi_x^{t,Det}$	The probability that an object detected in AC x at time t is a neutral
$\pi_j^{t+1,Sig}$	The probability that a tracked object which has moved to AC j at time $t+1$ is a neutral, after taking into account signature recognition
$\pi_j^{t+1,Rec}$	The probability that a tracked object which has moved to AC j at time $t+1$ is a neutral, after taking into account both signature recognition and movement recognition
π^M	A function representing the state-transition component of the probability vector π
$\tilde{\pi}$	The probability vector π at the time the Recognizer reaches the AC it has decided to visit next
$\tilde{\pi}_a$	The component of probability vector $\tilde{\pi}$ corresponding to AC a
$\overline{\pi}^0$	The initial probability vector π used in the Upper Bound Problem each time a decision has to be made
$\overline{\pi}_a^0$	The component of probability vector $\overline{\pi}^0$ corresponding to AC a
θ	Probability vector representing the probability of a target in each AC in the AOI
θ_a	The component of probability vector θ corresponding to AC a
θ^0	Probability vector representing the steady-state probability of a target in each AC in the AOI

θ_a^0	The component of probability vector θ^0 corresponding to AC a
θ^t	Probability vector θ at time t
θ_a^t	The component of probability vector θ^t corresponding to AC a
$\hat{\theta}\left(t-t',\theta^{t'}\right)$	A function updating the probability vector $\theta^{t'}$ from time t' to time t
$\theta_x^{t,Det}$	The probability that an object detected in AC x at time t is a target
$\theta_j^{t+1,Sig}$	The probability that a tracked object which has moved to AC j at time $t+1$ is a target, after taking into account signature recognition
$\theta_j^{t+1,Rec}$	The probability that a tracked object which has moved to AC j at time $t+1$ is a target, after taking into account both signature recognition and movement recognition
θ^{Rec}	A short-hand notation for $\theta_j^{t+1,Rec}$
θ^M	A function representing the state-transition component of the probability vector θ
$\tilde{\theta}$	The probability vector θ at the time the Recognizer reaches the AC it has decided to visit next
$\tilde{\theta}_a$	The component of probability vector $\tilde{\theta}$ corresponding to AC a
$\bar{\theta}^0$	The initial probability vector θ used in the Upper Bound Problem each time a decision has to be made
$\bar{\theta}_a^0$	The component of probability vector $\bar{\theta}^0$ corresponding to AC a

EXECUTIVE SUMMARY

Maritime Interdiction Missions (MIM) are of great interest and high operational importance to the U.S. Navy, the U.S. Coast Guard, and allied forces. The MIM scenario discussed in this thesis includes an area of interest (AOI) with multiple neutral and hostile vessels moving through this area, and an interdiction force consisting of an unmanned aerial vehicle (UAV) and an intercepting vessel, whose objectives are to search, identify, and intercept hostile vessels within a given time frame. The goal of this thesis is to optimize the operational policy used in employment of such interdiction force. We discuss why finding the optimal decision policy for the interdiction force is intractable, while suggesting a sub-optimal yet effective and practical heuristic decision policy, which performance does not fall by much behind the optimal policy.

Scenario and Concept of Operations

The MIM scenario discussed in this thesis includes multiple moving neutral and hostile vessels and an interdiction force comprising two assets: a UAV that detects identifies, and tracks suspected vessels and a navy vessel that intercepts suspected vessels indicated by the UAV. The goal of the interdiction force is to intercept as many hostile vessels as possible in a given time frame. We assume the existence of some prior intelligence regarding the expected movement patterns of both neutral and hostile vessels. The UAV moves around the AOI while searching for vessels. Once a vessel is detected, the UAV tracks that vessel for a given time period, while attempting to identify that vessel as either a neutral or a hostile one. The UAV identification is accomplished by utilizing a sensor (e.g., electro-optical or radar sensors) to recognize the physical signature of that vessel and its movement pattern. Once the UAV has flagged a vessel as a suspicious one, the interception vessel is called in to physically intercept that suspicious vessel.

Methodology and Models used

We developed a Stochastic Dynamic Programming model, which represents a system of neutral and hostile vessels together with an interdiction force operating inside the AOI. This model is used to optimize the operational policy of the interdiction force in this MIM scenario. The model includes the states of the system, the decisions the interdiction force is required to make, the information obtained by the interdiction force following such a decision, the manner by which the system evolves over time according to the decision policy being used and the overall objective function.

While this model can theoretically be used to find the optimal decision policy for the interdiction force, we show why this problem is intractable and cannot be implemented in a real-world operational scenario. Instead, we suggest a heuristic approach to constructing an effective sub-optimal decision policy and present its performance in achieving the interdiction force’s goal. The analysis of this heuristic decision policy is accomplished by means of bounding its expected performance based on a simplified dynamic programming model of the MIM scenario at hand.

Both of the above models—the original dynamic programming model and the simplified one—have been implemented in MATLAB with the aim of analyzing a numerical case study of several representative MIM scenarios. This numerical case study is used to evaluate the performance of the heuristic decision policy.

Results

The analysis of several MIM scenarios as part of the numerical case study, along with a sensitivity analysis and parametric study, has shown that the number of intercepted hostile vessels following the heuristic decision policy is at least 60% of the number of hostile vessels intercepted following the optimal decision policy (which is intractable to produce). Furthermore, we present an approximate optimization for one of the key operational parameters in the employment of the interdiction force: How often should one call the interception vessel to intercept suspicious vessels being tracked by the UAV. It has been found that when the boarding time of an intercepted vessel is negligible, it is best to call the Interceptor even when there is only a small probability that the tracked vessel is an hostile one, while when the boarding time is in the order of an hour, it is best to call the Interceptor only when there is at least 40% chance that the tracked vessel is a hostile one.

Conclusions

Based on the results of the heuristic performance in the numerical case studies, we recommend the use of this heuristic in any MIM scenario which closely resembles the MIM scenario discussed in this thesis. This heuristic can be effectively implemented in an operational decision aid for Maritime Interdiction Missions.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my thesis advisor, Professor Johannes O. Royset, and to my second reader, Professor Moshe Kress, for countless hours of brainstorming and passionate discussions, a lot of great ideas, productive guidance and endless patience. Thank you so much for making this journey together truly a fun one.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. OVERVIEW

Maritime Interdiction Missions (MIM) are of great interest and high operational importance to the U.S. Navy, the U.S. Coast Guard, and allied forces. MIM typically involve search, identification, and interception of suspected vessels and are the operational background to this thesis. We develop a stochastic dynamic-programming model representing a maritime area of interest with multiple moving neutral and hostile vessels and an interdiction force comprising two assets: an unmanned aerial vehicle (UAV) that detects identifies, and tracks suspected vessels and a navy vessel that intercepts suspected vessels. The model developed in this thesis generates optimal and near-optimal policies for MIM that may lead to better utilization of the interdiction force. The model can be implemented in a tactical decision aid and produce real-time recommendations for courses of actions to an interdiction force commander in MIM scenarios.

B. BENEFITS OF STUDY

The near-optimal policy for employing an interdiction force in MIM developed in this thesis will have operational value when implemented in a real-time decision aid. Moreover, the thesis is a first attempt to model combined search-interception operations in a stochastic and dynamic setting.

C. RELATED WORK

The field of classical search theory has been extensively studied, as discussed by Washburn [1] and Stone [2]. Additional work on route optimization for multiple sensors and resource-constrained searchers has been done by Royset and Sato [3], [4]; see also references therein.

Dynamic task allocation and vehicle routing has been studied by Smith [5], who discusses multiple autonomous vehicles employment in dynamic environments.

Optimization of employment of non-reactive sensors has been addressed by Kress, Szechtman and Jones [6]. Probabilistic search optimization and mission assignment for heterogeneous agents is discusses by Chung, Kress and Royset [7]. This last study deals with a similar situation as in this thesis, but adopts a heuristic approach without solution quality estimates in terms of upper and lower bounds. This thesis develops such bounds for the specific situation with one UAV and one navy vessel.

D. THESIS ORGANIZATION

This thesis is organized in the following way: Chapter I presents an overview of the operational scenario and the problem at hand, while listing the main assumption used in this research. Chapter II presents the basic development of the dynamic programming models used in this thesis. In Chapter III we continue to discuss the models presented in Chapter II while completing the detailed development of all necessary models. Chapter IV discusses an analysis of a numerical case study, and presents the results and insights of this analysis. Chapter V summarizes this research and presents the conclusions. Appendix A offers a list of acronyms and symbols used throughout this thesis, while Appendix B presents a brief overview of the MATLAB code implemented in this research.

E. SCENARIO

We consider a maritime area of interest (*AOI*) that contains multiple objects of interest (*objects*) some of which are hostile objects (*targets*) and the remaining are neutral objects (*neutrals*). The number of targets and the number of neutrals are unknown. The AOI is subdivided into a number of area cells (*ACs*). The time is divided into discrete time periods. All targets are dynamic and move independently according to a known Markov chain defined on the set of ACs. The neutrals move likewise, but according to a different Markov chain. Motivated by our discretization of space and time, with resolution that can be arbitrarily high, and assuming that the AOI is relatively large compared to the (unknown) number of objects, we neglect the possibility of more than one object in any specific AC at any given time period. This is an approximation to the real situation, which is quite reasonable in open-sea scenarios, and which makes the model tractable. The interdiction force's UAV is hereafter referred to as a *Recognizer* and the navy vessel as an *Interceptor*. Figure 1 presents an example of such an AOI, with multiple targets and neutrals arriving, leaving and moving about the AOI as well as a Recognizer and an Interceptor.

The Recognizer patrols the AOI in a fashion to be determined, with the goal to correctly identify as many targets as possible. After arriving to a specific AC, the Recognizer searches for the presence of an object in that AC. We assume that the Recognizer has perfect detection capabilities, which means that the Recognizer can determine with certainty whether the AC is empty or contains an object. If there is no object in that specific AC, the Recognizer decides on its next AC to visit. In case there is an object in that AC, the Recognizer follows it for a single time period while trying to identify it as either a target or a neutral. After this single time period of tracking the object, the Recognizer has to decide whether that object is a likely target or a likely neutral. The Recognizer flags a tracked object as a likely target if the probability that the tracked object is a target is greater or equal to a predetermined threshold. If a tracked object is flagged as a likely target, the Recognizer then calls in the Interceptor to intercept the object. While waiting for the Interceptor to arrive and intercept the object, the Recognizer stays with the object. In this thesis we make the simplifying assumption that once flagged as a likely target, the object is forced to remain stationary at its location at the time when the Recognizer identified it as a likely target. This assumption avoids the need of a complicated interception model, which would have made our scenario more realistic, yet would needlessly complicate the model if one is not concerned about the interception "end-game."

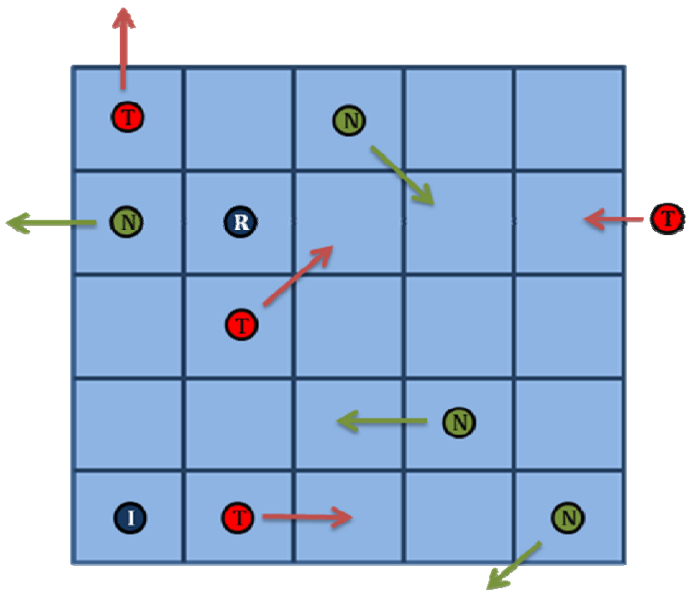

Figure 1. An example of an AOI with multiple neutrals (*N*) and targets (*T*), a Recognizer (*R*) and an Interceptor (*I*).

The Interceptor's only task is to intercept objects identified to be likely targets. The interceptor intercepts the suspicious target (still being tracked by the Recognizer) and identifies it as a (real) target or a neutral. The Interceptor has perfect identification capability; it can distinguish with certainty between a target and a neutral. While waiting for such tasks, the Interceptor is stationary at a certain location inside the AOI (the Interceptor only moves when called in to intercept a likely target). The goal of the interdiction force is to intercept as many targets as possible within a given time horizon, while possibly taking into account discounting of intercepted targets (the discounted value of an intercepted target is called a *reward*). If the Recognizer identifies an object as a likely neutral or if the Interceptor intercepts an object that turns out to be a neutral, then that object is "marked" as such and is of no future interest (e.g., some form of a tag or a beacon is placed on that object to avoid future "redetection" and tracking by the Recognizer). Figure 2 summarizes the concept of operations (CONOPS) used by the interdiction force.

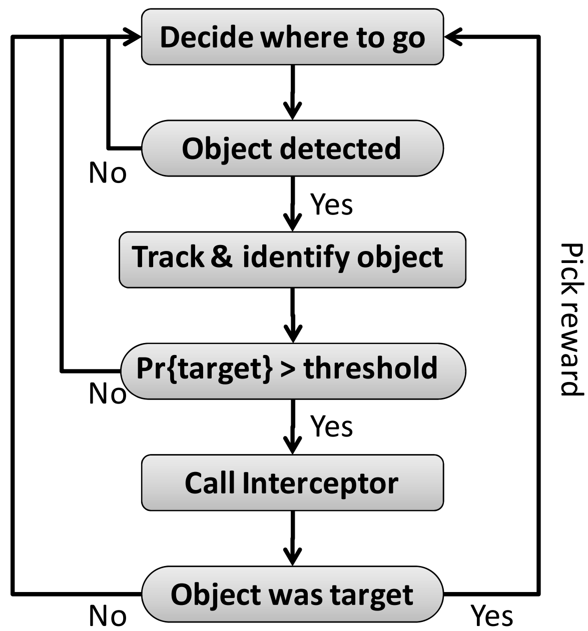

Figure 2. Summary of CONOPS of the interdiction force

F. ASSUMPTIONS

The scenario analyzed in this thesis and the corresponding model includes several key assumptions, which are discussed next.

1. Arrival and Movement of Targets and Neutrals

We assume that all objects, targets and neutrals, independently enter the AOI, move inside the AOI, and then leave the AOI. The objects do not behave strategically and, therefore, their movement is not affected by the actions of the interdiction force, that is, neither the presence of the Recognizer nor the actions of the Interceptor affects the behavior of the objects. We also assume that the AOI is sparse enough, or the spatial and time resolutions are high enough, such that the probability of more than one object in an AC at any given time period is negligible.

At any given time period, new objects may enter the AOI. The set of ACs to which objects may arrive is called the *Boundary* of the AOI. At any given time period, at most one object may enter each of the ACs on the boundary of the AOI with given probabilities. The entry probability may depend on the type of object—target or neutral—as well as the location of the AC. ACs not on the boundary have probability 0 of an arrival from outside the AOI. Once inside the AOI, each object moves according to a known Markov process with different transition probabilities for targets and neutrals. A designated AC represents the area outside the AOI to which objects eventually transit according to the Markov process transition matrix. The arrival process into each of the ACs on the boundary of the AOI is Bernoulli. Note that a key assumption is that the likelihood of more than two simultaneous arrivals to the same AC is negligible.

Absent interruptions by the interdiction force, the overall stochastic process describing the arrivals, transits and exits of objects is a large-scale Markov chain, with states defined by a vector $(Y_1,...Y_{|AOI|})$ where Y_i is 1 if a neutral is present in the corresponding AC, Y_i is 2 if a target is present in the corresponding AC, Y_i is 0 if the AC is empty, and $|AOI|$ is the number of ACs in the AOI. The state space of this Markov chain is very large ($3^{|AOI|}$ states), and so it is not used directly in any calculations in this thesis. Instead of looking at the probability of a state, we consider the marginal probabilities for each AC $P[Y_i = y_i]$. Our assumptions imply that $Y_1,...Y_{|AOI|}$ are independent discrete random variables. We assume that the MIM starts when the system represented by $Y_1,...Y_{|AOI|}$ is at steady-state. In subsequent chapters in this thesis, whenever we refer to a Markov process or a Markov transition matrix, we refer to the movement process of some individual object and not the overall large-scale Markov process discussed above. We assume stationarity in the sense that neither the entry probabilities nor the in-AOI transition or exit probabilities depend on the time period t, and they do not depend on the interdiction force actions.

2. Sensors Capabilities

The Recognizer has perfect detection capability, which means that, once in an AC, the Recognizer detects the presence of an object in that AC with certainty and there are no false positive detections. Following a detection, the Recognizer tracks the detected object for a single time period in an attempt to recognize it and determine whether it is a target or a neutral. The tracking process is fault-free in the sense that the Recognizer never loses contact with the tracked object, however, the recognition capability is imperfect and the identity of the object may be subject to false negative (identifying a target as a neutral) and false positive (identifying a neutral as a target) errors. The Recognizer utilizes two recognition techniques: *movement recognition* and *signature recognition*. Movement recognition is based on the movement pattern of the object, attempting to detect anomalies. Signature recognition is based on the physical characteristics of the object.

The Interceptor is assumed to have perfect recognition capability. Once the Interceptor has intercepted a suspicious object (after being dispatched by the Recognizer, which had flagged that object as a likely target) it can perfectly determine whether this object is indeed a target or a neutral.

3. Separation of Detected Objects

Once an object is flagged as a likely neutral (at the end of the Recognizer's tracking phase) or once an intercepted object turns out to be a neutral, it is taken away from the AOI for the remainder of the time horizon. We assume that in both cases some form of a beacon or other tagging technique is used to prevent this specific object from ever being of interest for the interdiction force.

4. Finite Time Horizon

The time horizon is finite, which means that the interdiction force does not care about events that will take place past that time horizon. In particular, a successful interception of a target past the time horizon is considered to be worthless for the interdiction force. This assumption may be appropriate in an operational scenario in which there exists a strict deadline. In scenarios in which a strict deadline does not exist, this artificial time horizon is known to have some distorting effect on the results of the model. One can try to reduce this end-effect by using a long time horizon.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MODELS DEVELOPMENT: MAIN PARTS

A. DYNAMIC PROGRAMMING FORMULATION

Let $\mathcal{A}$ denote the set of ACs in the AOI, $\mathcal{A}_0$ denote the area outside the AOI to which objects eventually transit when they leave $\mathcal{A}$, and $E \subset \mathcal{A}$ denote the subset of ACs at the boundary of the AOI. Let t denote the index of time periods. Let T denote the finite time horizon of the scenario. As mentioned in the Introduction, we assume that the grid of ACs and time resolutions are fine enough such that at any given time period there may be at most one arrival of a new object into each one of the ACs in E and at most one object present in a certain AC in $\mathcal{A}$. The objects enter the AOI and move about the AOI independently.

The dynamic programming formulation used in this thesis is based on the conventions found in Powell [8], pp. 129–178.

1. State

We define the state as the vector $s = (t, r, i, \pi, \theta)'$ where t is the time period, $r \in \mathcal{A}$ is the Recognizer's location, $i \in \mathcal{A}$ is the Interceptor's location, π is a vector of probabilities with components π_a , $a \in \mathcal{A}$, where π_a is the probability that a neutral is present in AC a , and θ is a vector of probabilities with components θ_a , $a \in \mathcal{A}$, where θ_a is the probability that a target is present in AC a . Let $\mathcal{S} \subset \{1, 2, \ldots T\} \times \mathcal{A} \times \mathcal{A} \times [0,1]^{|\mathcal{A}|} \times [0,1]^{|\mathcal{A}|}$ be the space of all possible state vectors. Although the two probability vectors π and θ are continuous, they only can take finite number of values because the number of detection and interception opportunities in a given time horizon is finite. Thus, our defined state-space $\mathcal{S}$ is of finite (though high) cardinality.

2. Decision

A decision x in the scenario determines the next AC to be visited by the Recognizer, thus $x \in \mathcal{A}$. This decision is made either at $t = 0$ or when the existing decision is *fathomed.* A decision $x \in \mathcal{A}$ is said to be fathomed in one of the following three situations: (i) no object is found by the Recognizer in AC x, (ii) an object is found by the Recognizer but it is recognized as a neutral, or (iii) an object is intercepted. Note that as a decision can only be made in one of the cases presented above (start of the search or when a decision is fathomed), this dynamic programming model slightly differs from standard models where decisions are made at a constant rate at every time step (see Powell [8], pp. 132–135). In fact, the time interval between two consecutive decisions is a random variable, which is defined in the next section. Since a decision is always made based on the current state, we often write $x(s)$. In some cases, we drop the explicit dependency of the decision x on the state s to simplify our expressions.

3. Information

Let $w = (\Delta t_w, r_w, i_w, z_w)'$ denote the random vector representing the information obtained by the interdiction force once a decision is fathomed and the consequences of that decision are realized. The time-interval Δt_w denotes the time duration of the current state-transition, that is, the duration between the time period in which decision x is taken and the time period when x is fathomed. The variables r_w and i_w denote the Recognizer's and Interceptor's locations, respectively, at the end of the current state-transition. The Bernoulli random variable z_w is equal 1 if a target has been intercepted and 0 otherwise. Note that these four random variables are not independent. Let $\mathcal{W}$ denote the space of possible realizations of w. The explicit probability distribution of w, which is a function of both the current state s and the decision $x(s)$, is presented later on. To represent this dependency, we write $w(s, x(s))$ to denote the random vector w with probability distribution depending on the state s and the decision $x(s)$. Our notation does not distinguish between the random vector w or its components $\left(\Delta t_w, r_w, i_w, z_w\right)$ and their realizations. The meaning should be clear from the context.

4. State Transition

The next state is determined using the state-transition function $s^M\left(s, x(s), w\left(s, x(s)\right)\right)$, which is a deterministic function of the current state, of the decision made and of the information obtained following the realization of a decision:

$$s^M : \mathcal{S} \times \mathcal{A} \times \mathcal{W} \to \mathcal{S} \tag{2.1}$$

The explicit function will be presented later.

5. Reward

The reward c is a function of the information w and the state s and is defined in the following way:

$$c : \mathcal{W} \times \mathcal{S} \to \mathbb{R} \tag{2.2}$$

$$c\left(w\left(s, x(s)\right), s\right) = \begin{cases} z_w \cdot (1+\gamma)^{-(t+\Delta t_w)}, & t + \Delta t_w \le T \\ 0, & t + \Delta t_w > T \end{cases} \tag{2.3}$$

where $w = \left(\Delta t_w, r_w, i_w, z_w\right)'$, $s = \left(t, r, i, \pi, \theta\right)'$ and T is the time horizon of the scenario. The reward is 0 if no target is intercepted or if the time of interception is beyond the time horizon, and the reward is 1 if a target is intercepted earlier than the end of the time horizon, while introducing a discounting factor γ for interception at later times. Choosing a discount factor $\gamma = 0$ means that the same reward is collected for intercepting a target now or later. A discount factor $\gamma > 0$ means that the reward for intercepting a target in the near future is higher than intercepting it at a later point in time. Note that the reward is not a function of the state alone, but of the information obtained (after making a decision) as well.

6. Timeline

We assume the following timeline (Figure 3): Once in a state s , and a decision has just been fathomed, a new decision $x(s) \in \mathcal{A}$ is made, after which information $w(s,x)$ is obtained, and the new state s' is derived by the deterministic function $s^M(s,x,w)$. As a decision is not made at every time step, and information is not obtained at every time step, we do not use the common notation of subscript t for s_t, x_t or w_t. In our notation t is treated as any other component of the state vector s.

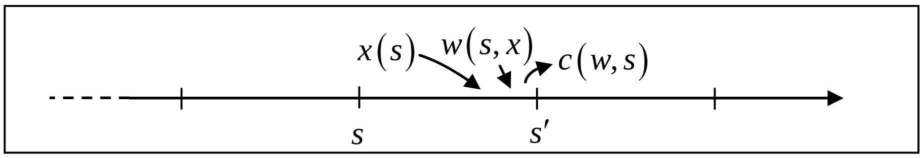

Figure 3. Timeline of the dynamic programming formulation

7. Bellman Equation

We define the value of being in state s as the maximum expected reward from the time of being in state s to the end of the time horizon T. This value $V(s)$ is given by a Bellman equation:

$$V(s) = \begin{cases} \max\limits_{x(s)} \left\{ E\left[c\left(w\left(s,x(s)\right),s\right) + V\left(s^M\left(s,x(s),w\left(s,x(s)\right)\right)\right)\right]\right\}, & t < T \\ 0, & t \geq T \end{cases} \tag{2.4}$$

where t is the time of state $s = (t,r,i,\pi,\theta)'$. This equation can be written in a simplified form:

$$V(s) = \begin{cases} \max\limits_{x(s)} \left\{ E\left[c(w,s) + V\left(s^M(s,x,w)\right)\right]\right\}, & t < T \\ 0, & t \geq T \end{cases} \tag{2.5}$$

The optimal policy in a state s is the optimal solution of the above Bellman equation.

B. METHODOLOGY OUTLINE

Let B denote the problem of finding the optimal policy, which maximizes the reward obtained in a finite time horizon T according to the states, decisions, information, state transition and rewards presented above. This problem can theoretically be solved by backward calculation of its states' values using the Bellman equation (Equation 2.5). A Backward Dynamic Programming algorithm (see Powell [8], p. 50) starts from the end of the time horizon and backtrack to the present while calculating the value of each possible state at each time period. We recall (2.5) that the value of each state is the expected reward obtained in the next state transition plus the expected value of the next resulted state, which already has been calculated since we are moving backwards through time. This sequential process requires the calculation of each state value for each time period in the finite time horizon, and so it is crucial to have a small enough state space to be able to practically implement this algorithm and solve a real-life problem. We encounter a difficulty when we realize that the state space $\mathcal{S}$ has a huge number of possible states, mainly due to the probabilities vectors π and θ, which can take numerous values. This calculation difficulty renders our problem intractable, and so we turn to a different method of producing an operational policy to be applied in the scenario.

In this thesis, we suggest to use an easy-to-calculate heuristic instead of solving for the exact optimal policy, while analyzing the performance of this heuristic by bounding the error between this heuristic's value and the actual optimal policy's value. We construct a new problem $\overline{B}$, named the *Upper Bound Problem*, characterized by the fact that the reward collected in this new problem is an upper bound to the reward collected in our original problem B. Any feasible policy in problem B is a lower bound on the optimal reward collected in B using the optimal policy, and so by calculating the reward collected using our heuristic's policy in problem B and solving $\overline{B}$ for its optimal reward we are able to bound the optimal reward of our original problem B. Furthermore, the difference between the optimal reward collected in $\overline{B}$ and the reward collected using our heuristic's policy is an upper bound for the difference between the optimal reward collected in the original problem B and the reward collected using our heuristic's policy.

This thesis includes the complete developing of B and $\overline{B}$, a method of effectively solving $\overline{B}$, a method of effectively calculating the reward collected using the heuristic's policy, and a numerical analysis of several case-study examples.

C. SUGGESTED HEURISTIC

We construct a simple and greedy heuristic for the Original Problem B, defined as follows:

$$x^H(s) \in \arg\max_{a \in \mathcal{A}} \left\{ \frac{\hat{\theta}_a}{T^R_{r,a} + 1 + T^I_{i,a}} \right\} \tag{2.6}$$

where $s = (t, r, i, \pi, \theta)'$ is the current state, r is the Recognizer's current location, i is the Interceptor's current location, $T^R_{r,a}$ is the time it takes the Recognizer to reach AC a from its current location r, $T^I_{i,a}$ is the time it takes the Interceptor to reach AC a from its current location i, and $\hat{\theta}_a$ is the estimated probability of a target in AC a at the time the Recognizer reaches AC a if the Recognizer decides to visit that AC. We add one time period for the Recognizer's identification process (the time it takes to identify a detected object as either a likely target or a likely neutral while tracking it). This is an attempt to estimate a normalized value of each potential AC to visit next, by dividing the probability of a target being in that AC by a rough estimation of the time it will take to detect and intercept it. There is a chance for a tie between several ACs that maximize the above expression, in which case we can arbitrarily choose among them.

D. REWARD CALCULATION USING HEURISTICS POLICY

This reward is calculated by running forward in time in the original problem B while following our heuristic decision policy. The Bellman equation used in this calculation is:

$$V^H(s) = \begin{cases} E\left[c(w, s) + V^H\left(s^M\left(s, x^H, w \right) \right) \right], & t < T \\ 0, & t \geq T \end{cases} \tag{2.7}$$

which is the Original Problem's Bellman equation without the maximization over the possible decisions x, but instead just choosing x with our heuristic. We use $V^H(s)$ to denote the value of each state following our heuristic decision policy, which is different than the optimal value of each state using the original Bellman equation (2.5).

E. UPPER BOUND PROBLEM

We define an Upper Bound Problem $\overline{B}$ that is a simplified version of our Original Problem B. The main difference between the two problems is that in $\overline{B}$ the interdiction force experiences a "situational awareness memory loss" after each decision is fathomed. Specifically, each time a decision has been fathomed, we "reset" the two probability vectors π and θ to their initial values at time $t = 0$. Because the probability vectors π and θ both remain the same each time a decision is made, we can exclude them from the definition of the state in the Upper Bound Problem $\overline{B}$. This simplified state space makes the Upper Bound Problem tractable and allows us to solve it in reasonable time. The above "memory loss" property also means that we allow the interdiction force to potentially re-collect the same rewards over and over again. In the Original Problem, each time the Recognizer visits an AC, we set the probability that the AC contains a target and the probability that AC contains a neutral to 0 (a comprehensive discussion of these probability updates appears in Chapter III.A.2). Our definition of the state in the Upper Bound Problem does not include any history of all previously visited ACs. Specifically, the Recognizer do not "remember" that the probabilities of a target and a neutral being in each visited AC should be set to 0 at the time of that visit. Explicitly, by not setting to 0 the probability that a visited AC contains another target once it has been visited, all ACs in the AOI have greater or equal probability of containing a target than they should truly have (as in the Original Problem). Instead of keeping track of the most updated π and θ vectors as being done in the Original Problem B, each time we make a decision we assume we are back to the initial steady-state of the probability vectors π and θ. These steady-state probability vectors are discussed later, with explicit formulas appearing in (3.1) and (3.2). Having this "memory loss" property and assuming we always make a decision in steady-state, we are in risk of getting "trapped" in the same

AC (which probably have relatively high probability of containing a target) just because we do not take into account the fact that we have just visited this AC and as such the probability that another target has just entered into this AC is likely to be less than the high steady-state probability of a target being in this AC. In an attempt to avoid these unwanted cases we temporarily update the probability of another object in the AC the Recognizer is currently in at the time of making each decision (i.e., drop both the probability of a target in this AC and the probability of a neutral in this AC down to 0). This temporary update exists only until the current decision is fathomed. Once we complete the current state transition and end up in the next state we "forget" this temporary update and assume we are back to a stead-y-state.

As discussed above, the Upper-Bound Problem's state is slightly different from the Original Problem's state and is defined as follow:

$$\bar{s} = \left(\bar{t}, \bar{r}, \bar{i}\right)' \tag{2.8}$$

where $\left(\bar{t}, \bar{r}, \bar{i}\right)$ are the time, Recognizer's location and Interceptor's location (the same as in the Original Problem B), and the probability vectors π and θ have been omitted. The Upper-Bound Problem's Bellman equation can now be written:

$$\bar{V}\left(\bar{s}\right) = \begin{cases} \max_{\bar{x}(\bar{s})} \left\{ E\left[\bar{c}\left(\bar{w}, \bar{s}\right) + \bar{V}\left(\bar{s}^M\left(\bar{s}, \bar{w}\right)\right)\right]\right\}, & \bar{t} < T \\ 0, & \bar{t} \geq T \end{cases} \tag{2.9}$$

which seems almost identical to the Original Problem Bellman equation (2.5), but of course differ in the actual definitions and meaning of the variables and functions it incorporates (e.g., the definitions of the state $\bar{s}$, the probability distribution of the information $\bar{w}$, etc). We use "bar" to denote these variables and functions in the Upper-Bound Problem $\bar{B}$, which differ from their counterparts in the Original Problem B. Another subtle difference is that the transition function $\bar{s}^M$ is not directly related to $\bar{x}$ but only through $\bar{w}$, as presented later in (3.54). A more comprehensive discussion of the Upper Bound Problem $\bar{B}$ is presented in Chapter III.B.

III. MODELS DEVELOPMENT: DETAILS

A. ORIGINAL PROBLEM

1. Probabilities and Object Recognition Techniques

In this section, we present the notation and formulas describing the calculation and update of several variables and probability distributions needed for our model. The steady-state probabilities of a neutral and a target at each AC comprise the baseline at the beginning of the scenario (though we could start with non steady-state probability vectors just as well). As the scenario progresses, these steady-state probabilities will no longer be relevant and we will have to keep track of an updated probability map of neutrals and targets at each AC in the AOI. These probabilities will change as a result of the sensors' observations and the Markov transition probabilities of the objects.

a. Initial Pre-Search Probabilities: Uninterrupted Steady-State

Let $P = \left[P\left(a', a\right) \right]$, $a', a \in \mathcal{A} \cup \mathcal{A}_0$ be the Markov transition matrix for any neutral in the AOI, with $P\left(a', a\right)$ representing the probability of a single time-step transition from AC a' to AC a. Let $Q = \left[Q\left(a', a\right) \right]$, $a', a \in \mathcal{A} \cup \mathcal{A}_0$ be the Markov transition matrix for any target in the AOI, with $Q\left(a', a\right)$ representing the probability of a single time-step transition from AC a' to AC a. Let α_l denote the single time step probability of a neutral entry to AC $l \in E$, and let β_l denote the single time step probability of a target entry to AC $l \in E$.

Absent any other information (such as sensor's observations), and based on our assumption that the probability of more than one object in a cell is negligible, the steady-state probability of a neutral in AC $a \in \mathcal{A}$ is approximated by

$$\pi_a^0 = \begin{cases} 1-(1-\alpha_a)\prod_{l\in E}\prod_{k=1}^{\infty}\left(1-\alpha_l P^k(l,a)\right), & a\in E \\ 1-\prod_{l\in E}\prod_{k=1}^{\infty}\left(1-\alpha_l P^k(l,a)\right), & a\notin E \end{cases} \tag{3.1}$$

where $P^k(l,a)$ is the (l,a) entry of P^k—the transition matrix P raised to the k^{th} power, and the superscript 0 denotes steady-state probabilities. Similarly, the steady-state probability of a target in AC $a\in\mathcal{A}$ is approximated by

$$\theta_a^0 = \begin{cases} 1-(1-\beta_a)\prod_{l\in E}\prod_{k=1}^{\infty}\left(1-\beta_l Q^k(l,a)\right), & a\in E \\ 1-\prod_{l\in E}\prod_{k=1}^{\infty}\left(1-\beta_l Q^k(l,a)\right), & a\notin E \end{cases} \tag{3.2}$$

Based on our assumptions, we identify three mutually exclusive and collectively exhaustive events associated with an AC $a\in\mathcal{A}$: (i) void of objects, denoted V_a, (ii) contains one neutral, denoted N_a, and (iii) contains one target, denoted T_a. The event O_a denote a situation where there is an object, of any kind, in AC a, that is, $O_a = N_a \cup T_a = V_a^C$. Our spatial and temporal assumptions lead us to the following approximated results regarding the steady-state:

$$\Pr\{N_a\}\approx\pi_a^0 \tag{3.3}$$

$$\Pr\{T_a\}\approx\theta_a^0 \tag{3.4}$$

$$\Pr\{V_a\}\approx\left(1-\pi_a^0\right)\left(1-\theta_a^0\right)\approx 1-\pi_a^0-\theta_a^0 \tag{3.5}$$

b. Markov Updates

When about to make a decision regarding the next AC to investigate, the Recognizer first needs to estimate the probability of a target and the probability of a neutral at each of the ACs in the AOI at the time it would arrive to the designated AC. Let t' denote the time a decision is made and $\pi^{t'}$ the probability vector of a neutral in each AC at time t'. For any time $t\ge t'$ we can calculate the new probability vector due to the progress of time, absent any new information about the AOI (i.e., no sensor observations). Let π^t denote this new probability vector:

$$\pi^t = \hat{\pi}\left(t-t',\pi^{t'}\right) \qquad \forall t \geq t' \tag{3.6}$$

where $\hat{\pi}\left(t-t',\pi^{t'}\right)$ is the function which updates the probability vector of a neutral at each AC. Note that this function does not depend on both time arguments but only on their difference. This function appears in several other places in this thesis. The trivial case is when no time had passed (i.e., $t=t'$), in which case there is no update and so we get that $\pi^t = \hat{\pi}\left(t-t',\pi^{t'}\right) = \pi^{t'}$. Also note that initially, absent any sensor information, the system is in steady-state and therefore $\pi^t = \hat{\pi}\left(t-t',\pi^{t'}\right) = \pi^{t'}$ for all $t \geq t'$. We need to distinguish between four non-trivial cases: (i) single time-step and $a \in E$, (ii) two or more time-steps and $a \in E$, (iii) single time-step and $a \notin E$, (iv) two or more time-steps and $a \notin E$:

$$\pi_a^t = \hat{\pi}_a\left(t-t',\pi^{t'}\right) = \\ = \begin{cases} 1-(1-\alpha_a)\prod_{a'\in\mathcal{A}}\left(1-\pi_{a'}^{t'}P(a',a)\right), & t=t'+1, a\in E \\ 1-\prod_{a'\in\mathcal{A}}\left(1-\pi_{a'}^{t'}P(a',a)\right), & t=t'+1, a\notin E \\ 1-(1-\alpha_a)\left(\prod_{a'\in\mathcal{A}}\left(1-\pi_{a'}^{t'}P^{t-t'}(a',a)\right)\right)\left(\prod_{l\in E}\prod_{k=1}^{t-t'-1}\left(1-\alpha_l P^k(l,a)\right)\right), & t\geq t'+2, a\in E \\ 1-\left(\prod_{a'\in\mathcal{A}}\left(1-\pi_{a'}^{t'}P^{t-t'}(a',a)\right)\right)\left(\prod_{l\in E}\prod_{k=1}^{t-t'-1}\left(1-\alpha_l P^k(l,a)\right)\right), & t\geq t'+2, a\notin E \\ \pi_a^{t'}, & t=t' \end{cases} \tag{3.7}$$

where P^k is the neutrals' Markov transition probabilities matrix raised to the k^{th} power.

Similarly we define $\theta_a^t = \hat{\theta}_a\left(t-t',\theta^{t'}\right)$ for the targets:

$$\theta_a^t = \hat{\theta}_a\left(t-t',\theta^{t'}\right) = \\ = \begin{cases} 1-(1-\beta_a)\prod_{a'\in\mathcal{A}}\left(1-\theta_{a'}^{t'}Q(a',a)\right), & t=t'+1, a\in E \\ 1-\prod_{a'\in\mathcal{A}}\left(1-\theta_{a'}^{t'}Q(a',a)\right), & t=t'+1, a\notin E \\ 1-(1-\beta_a)\left(\prod_{a'\in\mathcal{A}}\left(1-\theta_{a'}^{t'}Q^{t-t'}(a',a)\right)\right)\left(\prod_{l\in E}\prod_{k=1}^{t-t'-1}\left(1-\beta_l Q^k(l,a)\right)\right), & t\geq t'+2, a\in E \\ 1-\left(\prod_{a'\in\mathcal{A}}\left(1-\theta_{a'}^{t'}Q^{t-t'}(a',a)\right)\right)\left(\prod_{l\in E}\prod_{k=1}^{t-t'-1}\left(1-\beta_l Q^k(l,a)\right)\right), & t\geq t'+2, a\notin E \\ \theta_a^{t'}, & t=t' \end{cases} \tag{3.8}$$

c. *Recognizer Updates*

In this section, we discuss the probability updates generated by the Recognizer while detecting and tracking an object. These updates eventually determine whether or not the Recognizer calls in the Interceptor to intercept the suspicious object.

Recall that the Recognizer has perfect detection capabilities (i.e., detecting whether an AC is empty or not). Let $x \in \mathcal{A}$ denote the AC the Recognizer has decided to investigate next. If the Recognizer arrives at time t in AC x and finds no object there, then the current decision (search AC x at time t) is fathomed and a new decision needs to be made. If an object is detected in AC x, then the immediate probabilities update are:

$$\pi_x^{t,Det} = \frac{\pi_x^t}{\pi_x^t + \theta_x^t} \tag{3.9}$$

$$\theta_x^{t,Det} = \frac{\theta_x^t}{\pi_x^t + \theta_x^t} = 1 - \pi_x^{t,Det} \tag{3.10}$$

where $\pi_x^{t,Det}$ and $\theta_x^{t,Det}$ are the probabilities of the detected object being a neutral or a target, respectively, and π_x^t and θ_x^t are the corresponding prior probabilities.

If an object is detected in AC x at time t, the Recognizer continues tracking the object for one more time period ($t+1$) in which the tracked object either moves to AC $j \in \mathcal{A}$, leaves the AOI to AC $\mathcal{A}_0$ or stays at AC x. Note that the object might stay at AC x or leave the AOI to $\mathcal{A}_0$, according to the specific Markov transition matrices P and Q. If the object has left the AOI, the Recognizer ends up back in AC x at the end of the tracking time period and the decision is fathomed. After tracking the detected object (assuming it did not leave the AOI), the Recognizer decides if it is likely to be a neutral or a target. While tracking the object, the Recognizer only focuses on the tracked object and is not searching in other ACs.

During tracking, the Recognizer utilizes two modes for recognizing the tracked object: *signature recognition* (e.g., electro-optical sensor, radar, etc) and *movement recognition*, in which the Recognizer tries to identify the movement pattern of the tracked object (i.e., leaving known shipping lanes or any other suspicious movement).

Both recognition techniques take place within the tracking time period. Without loss of generality, we assume that signature recognition takes place first.

During the signature recognition mode, the Recognizer takes g looks (glimpses) at the tracked object where the glimpses are conditionally independent given the presence of the object. Let $1-u$ denote the single glimpse false negative probability of the Recognizer identifying a target as a neutral, and let $1-v$ denote the single glimpse false positive probability of the Recognizer identifying a neutral as a target. Suppose that n glimpses report "neutral" and $g-n$ glimpses report "target." Recall that $j \in \mathcal{A}$ is the AC to which the tracked object has moved while being tracked. Let $\pi_j^{t+1,Sig}$ denote the posterior probability of a neutral following the g glimpses of the signature recognition process. We have:

$$\pi_j^{t+1,Sig} = \frac{\binom{g}{n} v^n (1-v)^{g-n} \pi_x^{t,Det}}{\binom{g}{n} v^n (1-v)^{g-n} \pi_x^{t,Det} + \binom{g}{n} (1-u)^n (u)^{g-n} \theta_x^{t,Det}} \tag{3.11}$$

and similarly, the posterior probability of a target is:

$$\theta_j^{t+1,Sig} = \frac{\binom{g}{n} (1-u)^n (u)^{g-n} \theta_x^{t,Det}}{\binom{g}{n} v^n (1-v)^{g-n} \pi_x^{t,Det} + \binom{g}{n} (1-u)^n (u)^{g-n} \theta_x^{t,Det}} = 1 - \pi_j^{t+1,Sig} \tag{3.12}$$

Note that the factors $\binom{g}{n}$ cancel out.

Finally, the movement recognition mode takes into account the fact that the object has moved from AC x to AC j. Taking the posteriors of the signature recognition mode as priors for the movement recognition mode we obtain the probability of neutral:

$$\pi_j^{t+1,Rec} = \frac{P(x,j)\pi_j^{t+1,Sig}}{P(x,j)\pi_j^{t+1,Sig} + Q(x,j)\theta_j^{t+1,Sig}}. \tag{3.13}$$

Similarly, the posterior probability of a target is:

$$\theta^{Rec} = \theta_j^{t+1,Rec} = \frac{Q(x,j)\theta_j^{t+1,Sig}}{P(x,j)\pi_j^{t+1,Sig} + Q(x,j)\theta_j^{t+1,Sig}} = 1 - \pi_j^{t+1,Rec}. \tag{3.14}$$

where we introduce θ^{Rec} as a short-hand notation for $\theta_j^{t+1,Rec}$, which we use later.

Once tracking is complete, the Recognizer compares the probability in (3.14) to a predetermined threshold M and decides whether to flag the tracked object as a target or to let it go and decide on the next AC to visit. If this threshold is met, the Interceptor is called for interception while the Recognizer keeps watching the object until the arrival of the Interceptor.

2. The State-Transition Function

We recall that the state-transition $s^M(s,x,w)$ is a deterministic function of the current state s, of the decision to visit AC x, and of the information revealed once x is fathomed. We define the state-transition function in the following way:

$$s^M : \mathcal{S} \times \mathcal{A} \times \mathcal{W} \rightarrow \mathcal{S} \tag{3.15}$$

$$s^M(s,x,w) = \begin{pmatrix} t + \Delta t_w \\ r_w \\ i_w \\ \pi^M \\ \theta^M \end{pmatrix} \tag{3.16}$$

where $w = (\Delta t_w, r_w, i_w, z_w)'$ is the information, $s = (t, r, i, \pi, \theta)'$ is the current state, and π^M and θ^M are components of s^M representing the probability vectors π and θ of the next state. In computing π^M and θ^M we use (3.7) and (3.8). There are three periods of time we potentially need to account for (Figure 4): (i) the time between making the decision to go to AC x and the time of arrival to x, (ii) the tracking time of the detected object, and (iii) the waiting time for the Interceptor to arrive and intercept the object.

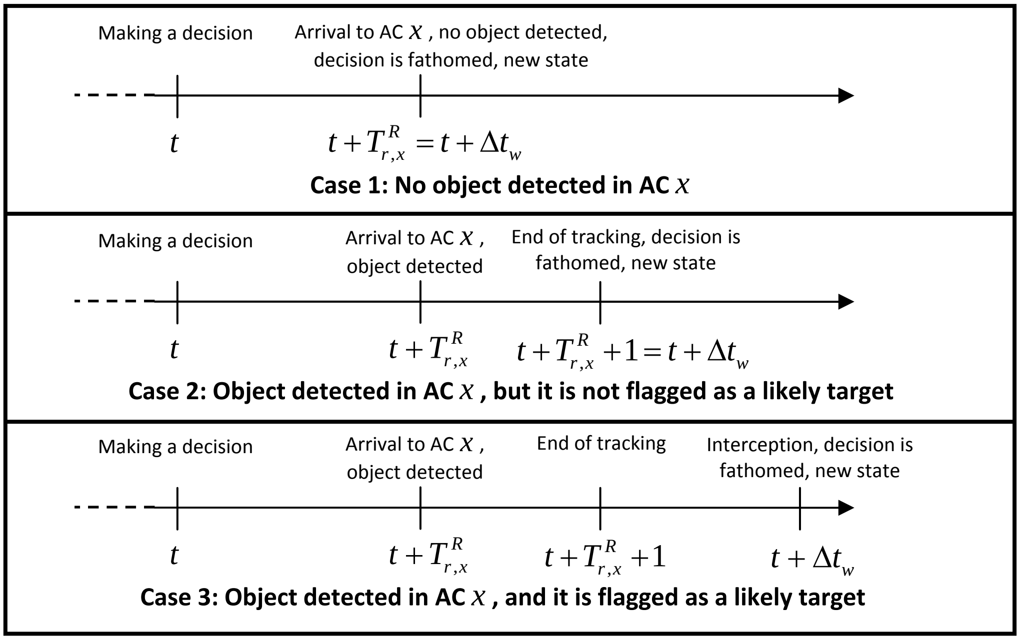

Figure 4. Timeline of state transitions

For any decision $x \in \mathcal{A}$, the time the Recognizer arrive at AC x is $t + T^R_{r,x}$. Let $\tilde{\pi}$ and $\tilde{\theta}$ denote the resulting probability vectors of a neutral and a target, respectively, in each AC in the AOI at time $t + T^R_{r,x}$. For each vector component corresponding to an AC $a \in \mathcal{A}$ we get:

$$\tilde{\pi}_a = \begin{cases} \hat{\pi}\left(T^R_{r,x}, \pi\right), & a \neq x \\ 0, & a = x \end{cases} \tag{3.17}$$

$$\tilde{\theta}_a = \begin{cases} \hat{\theta}\left(T^R_{r,x}, \theta\right), & a \neq x \\ 0, & a = x \end{cases} \tag{3.18}$$

where we set $\tilde{\pi}_x$ and $\tilde{\theta}_x$ to 0 because of two possible reasons: either no object has been detected in AC x, or an object was detected but because it will be tracked and handled separately – we can now assume the probability of another object in AC x is 0.

We now need to account for the time of tracking a detected object, if one had indeed been detected. Let $\mathcal{H}$ denote an operator that operate on a probability vector (either $\tilde{\pi}$ or $\tilde{\theta}$). The operator $\mathcal{H}$ updates the probability vector it operates on to the appropriate value one time step later, and then set to 0 the probability component of that vector which corresponds to the current Recognizer location. Using the operator $\mathcal{H}$ on a probability vector produces another probability vector. Let $(\mathcal{H}\tilde{\pi})_a$ denote the component of the resulting probability vector $\mathcal{H}\tilde{\pi}$ corresponding to AC $a \in \mathcal{A}$. We can formally define the operation of $\mathcal{H}$ in the following way:

$$(\mathcal{H}\tilde{\pi})_a = \begin{cases} \hat{\pi}(1,\tilde{\pi}), & \text{if } a \text{ is not the AC to which the tracked object moved} \\ 0, & \text{if } a \text{ is the AC to which the tracked object moved} \end{cases} \quad (3.19)$$

$$(\mathcal{H}\tilde{\theta})_a = \begin{cases} \hat{\theta}(1,\tilde{\theta}), & \text{if } a \text{ is not the AC to which the tracked object moved} \\ 0, & \text{if } a \text{ is the AC to which the tracked object moved} \end{cases} \quad (3.20)$$

Recall that the duration of a tracking phase is always a single time step. When arriving at AC r_w we need to have probabilities update similarly to the one we had for the time it takes the Recognizer to reach AC x. If the tracked object is eventually flagged as a likely target, we need to have a similar probability update for each time step until the decision is fathomed. For both cases when the tracked object is flagged as a likely target and when it is not flagged as a likely target, the overall remaining time between the time the Recognizer reaches AC x and when the decision is ultimately fathomed, is $\Delta t_w - T^R_{r,x}$. For each time step until the decision is fathomed, we need to update the probabilities of a target and a neutral at each AC using $\mathcal{H}$. We can now explicitly write π^M and θ^M:

$$\pi^M = \mathcal{H}^k \tilde{\pi} \quad (3.21)$$

$$\theta^M = \mathcal{H}^k \tilde{\theta} \quad (3.22)$$

where $\mathcal{H}^k \tilde{\pi}$ denotes operating $\mathcal{H}$ for k times on the probability vector $\tilde{\pi}$, $\mathcal{H}^k \tilde{\theta}$ denotes operating $\mathcal{H}$ for k times on the probability vector $\tilde{\theta}$, and we use $k = \Delta t_w - T^R_{r,x}$. In the case that $k = \Delta t_w - T^R_{r,x} = 0$, the operator $\mathcal{H}^k$ is the identity operator.

3. Information Probability Mass Function

We need the probability mass function of the obtained information w for several reasons. Firstly, in order to calculate the Bellman equation for the Original Problem B (2.5), we need to compute the expected value (with respect to w) of the sum of the reward and the value of the next state. As discussed earlier (Chapter II.B), such a direct calculation is intractable, and so the approach taken in this thesis is to approximate the expected value using lower and upper bounds. Still, we present the formulas for direct calculation of the Original Problem's Bellman equation in Chapter III.A.4. Another reason we need the probability mass function of w is for the simulative calculation of Bellman equation using our suggested heuristic, as later discussed in Chapter III.C. In this simulative calculation we need to generate multiple realizations of the obtained information w according to its probability mass function. Furthermore, the following derivation of the probability mass function of w serves as the basis for the derivation of the probability mass function of the obtained information $\bar{w}$ in the Upper-Bound Problem $\bar{B}$.

Recall that the information vector w describes the tactical consequences of a decision to visit a certain AC x: the time until the decision is fathomed, the locations of the Recognizer and Interceptor when this happens and whether or not a target has been eventually intercepted. While $w = \left(\Delta t_w, r_w, i_w, z_w\right)'$ is an end-result, it can be equivalently described in terms of the events that occur during Δt_w. Following a decision x there are five possible scenarios which may occur: (i) no object is detected in AC x, (ii) an object is detected in AC x and it leaves the AOI (moves to $\mathcal{A}_0$) while being tracked, (iii) an object is detected at AC x, moves to AC $j \in \mathcal{A}$ and is flagged as a neutral, (iv) an object is detected at AC x, moves to AC $j \in \mathcal{A}$, is flagged as a target, and when intercepted is identified as a neutral, and (v) an object is detected at AC x, moves to AC $j \in \mathcal{A}$, is flagged as a target, and when intercepted, it is confirmed as a target.

Based on these five possible scenarios, we can explicitly write the possible values $w = (\Delta t_w, r_w, i_w, z_w)'$ can take:

$$\begin{pmatrix} \Delta t_w \\ r_w \\ i_w \\ z_w \end{pmatrix} = \begin{cases} \left(T^R_{r,x}, x, i, 0\right)', & \text{if no object is detected in AC } x \\ \left(T^R_{r,x} + 1, x, i, 0\right)', & \text{if an object is detected in AC } x\text{, then moves to } \mathcal{A}_0 \\ \left(T^R_{r,x} + 1, j, i, 0\right)', & \text{if an object is detected in AC } x\text{, moves to AC } j \in \mathcal{A} \text{ and is flagged as a likely neutral} \\ \left(T^R_{r,x} + 1 + T^I_{i,j}, j, j, 0\right)', & \text{if an object is detected in AC } x\text{, moves to AC } j \in \mathcal{A}\text{, flagged as a likely target, intercepted but turns out to be a neutral} \\ \left(T^R_{r,x} + 1 + T^I_{i,j}, j, j, 1\right)', & \text{if an object is detected in AC } x\text{, moves to AC } j \in \mathcal{A}\text{, flagged as a likely target, intercepted and is indeed a target} \end{cases} \tag{3.23}$$

To derive the probability mass function of w we need to derive the probability of each of the above scenarios. To do that, we first define two new random variables:

$$d = \begin{cases} -1, & \text{if no object is detected in AC } x \\ 0, & \text{if an object is detected in AC } x \text{ and while being tracked it moves to } \mathcal{A}_0 \\ j, & \text{if an object is detected in AC } x \text{ and while being tracked it moves to AC } j \in \mathcal{A} \end{cases} \tag{3.24}$$

$$f = \begin{cases} -1, & \text{if no object is detected in AC } x \text{ (and so nothing is flagged)} \\ 0, & \text{if an object is detected in AC } x \text{ and after tracking it is not flagged as a likely target} \\ 1, & \text{if an object is detected in AC } x \text{ and after tracking it is flagged as a likely target} \end{cases} \tag{3.25}$$

Note that $f = 0$ can either imply that the tracked object is flagged as a likely neutral or that it has left the AOI and so it is not flagged at all. We also recall (see Chapter III.A.1.a) that T_x denotes the event of AC x containing a target at the time the Recognizer had reached AC x, and that N_x denotes the event of AC x containing a neutral at the time the Recognizer reaches AC x. Rewriting (3.23) we get:

$$\begin{pmatrix} \Delta t_w \\ r_w \\ i_w \\ z_w \end{pmatrix} = \begin{cases} \left(T^R_{r,x}, x, i, 0\right)', & f = -1,\ d = -1 & (I) \\ \left(T^R_{r,x} + 1, x, i, 0\right)', & f = 0,\ d = 0 & (II) \\ \left(T^R_{r,x} + 1, j, i, 0\right)', & f = 0,\ d = j \quad \forall j \in \mathcal{A} & (III) \\ \left(T^R_{r,x} + 1 + T^I_{i,j}, j, j, 0\right)', & f = 1,\ d = j,\ N_x \quad \forall j \in \mathcal{A} & (IV) \\ \left(T^R_{r,x} + 1 + T^I_{i,j}, j, j, 1\right)', & f = 1,\ d = j,\ T_x \quad \forall j \in \mathcal{A} & (V) \end{cases} \tag{3.26}$$

Next we compute the probability of each of the event on the right-hand side of (3.26).

The first event (I) is the simplest to compute (note that $f=-1 \Leftrightarrow d=-1$):

$$\Pr\{f=-1, d=-1\} = 1-\hat{\pi}_x\left(T^R_{r,x},\pi\right)-\hat{\theta}_x\left(T^R_{r,x},\theta\right) \tag{3.27}$$

where $s=(t,r,i,\pi,\theta)'$ is the current state.

The second event (II) is also relatively simple to compute (note that $d=0 \Rightarrow f=0$):

$$\Pr\{f=0, d=0\} = \Pr\{d=0\} = P(x,\mathcal{A}_0)\,\hat{\pi}\left(T^R_{r,x},\pi\right)+Q(x,\mathcal{A}_0)\,\hat{\theta}\left(T^R_{r,x},\theta\right) \tag{3.28}$$

To compute the probabilities of the other three events in (3.26), we first recall that θ^{Rec} denotes the posterior probability, at the end of the tracking phase, that the tracked object is a target. Also recall that a tracked object is flagged as a target if $\theta^{Rec} \geq M$, where M is the flagging threshold.

We defer the computation of event (*III*) after we compute events (IV) and (*V*). The following derivation is true for all $j \in \mathcal{A}$:

$$\begin{aligned}
&\Pr\{f=1, d=j, N_x\} \\
&= \Pr\{d=j, N_x\}\Pr\{f=1 \mid d=j, N_x\} \\
&= \Pr\{d=j, N_x\}\Pr\{\theta^{\text{Rec}} \geq M \mid d=j, N_x\} \\
&= \Pr\{d=j, N_x\}\sum_{n'=0}^{g}\Pr\{\theta^{\text{Rec}} \geq M \mid d=j, N_x, n=n'\}\Pr\{n=n' \mid d=j, N_x\}
\end{aligned} \tag{3.29}$$

where g is the total number of glimpses the Recognizer takes while tracking an object, and $n \leq g$ is the number of glimpses which indicated the tracked object being a neutral. Note that for every $j \in \mathcal{A}$, we can calculate the maximal value of n for which $\theta^{\text{Rec}} \geq M$. Let n^*_j denote this value of n. This means that for every $j \in \mathcal{A}$:

$$\Pr\{\theta^{\text{Rec}} \geq M \mid d=j, n=n'\} = \begin{cases} 1, & \text{if } n' \leq n^*_j \\ 0, & \text{if } n' > n^*_j \end{cases} \tag{3.30}$$

Note that the value of n_j^* depends on the specific AC *j* to which the tracked object moves, because θ^{Rec} takes into account movement recognition (i.e., an object moving to AC *j* may result in flagging it as a neutral while the same object moving to a different AC j' may result in flagging it as a target, even if in both cases the same number of glimpses report "neutral").

Continuing our derivation from (3.29) we get:

$$\begin{aligned}&\Pr\left\{d=j,N_x\right\}\sum_{n'=0}^{g}\Pr\left\{\theta^{\text{Rec}}\geq M\mid d=j,N_x,n=n'\right\}\Pr\left\{n=n'\mid d=j,N_x\right\}=\\&=\Pr\left\{d=j,N_x\right\}\sum_{n'=0}^{n_j^*}\Pr\left\{n=n'\mid d=j,N_x\right\}\end{aligned}\tag{3.31}$$

Using Bayes' formula for the first multiplicative term on the right-hand-side of (3.31) we get:

$$\Pr\left\{d=j,N_x\right\}\sum_{n'=0}^{n_j^*}\Pr\left\{n=n'\mid d=j,N_x\right\}=\Pr\left\{d=j\mid N_x\right\}\Pr\left\{N_x\right\}\sum_{n'=0}^{n_j^*}\Pr\left\{n=n'\mid d=j,N_x\right\}\tag{3.32}$$

The right-hand-side of (3.32) can be explicitly expressed in the following way:

$$\Pr\left\{d=j\mid N_x\right\}\Pr\left\{N_x\right\}\sum_{n'=0}^{n_j^*}\Pr\left\{n=n'\mid d=j,N_x\right\}=P\left(x,j\right)\hat{\pi}_x\left(T_{r,x}^R,\pi\right)\sum_{n'=0}^{n_j^*}\left(\binom{g}{n'}v^{n'}\left(1-v\right)^{g-n'}\right)\tag{3.33}$$

Summarizing (3.29) - (3.33) we get that for every $j\in\mathcal{A}$:

$$\Pr\left\{f=1,d=j,N_x\right\}=P\left(x,j\right)\hat{\pi}_x\left(T_{r,x}^R,\pi\right)\sum_{n'=0}^{n_j^*}\left(\binom{g}{n'}v^{n'}\left(1-v\right)^{g-n'}\right)\tag{3.34}$$

Following a very similar derivation we get the probability of event (V):

$$\Pr\left\{f=1,d=j,T_x\right\}=Q\left(x,j\right)\hat{\theta}_x\left(T_{r,x}^R,\theta\right)\sum_{n'=0}^{n_j^*}\left(\binom{g}{n'}\left(1-u\right)^{n'}\left(u\right)^{g-n'}\right)\tag{3.35}$$

The derivation of the probability of event (III) is similar to (IV) and (V), with two key differences: We need to consider both T_x and N_x events, and we are interested in the cases where the number of glimpses reporting "neutral" is high enough to result in flagging the tracked object as a neutral. This means that in the summation over the possible number of glimpses reporting "neutral" we sum from n_j^*+1 to g. We get that for every $j \in \mathcal{A}$:

$$\Pr\left\{ f=0, d=j \right\} = \\ = P(x,j)\hat{\pi}_x\left(T_{r,x}^R,\pi\right)\sum_{n'=n_j^*+1}^{g}\left(\binom{g}{n'}v^{n'}(1-v)^{g-n'}\right)+Q(x,j)\hat{\theta}_x\left(T_{r,x}^R,\theta\right)\sum_{n'=n_j^*+1}^{g}\left(\binom{g}{n'}(1-u)^{n'}u^{g-n'}\right) \tag{3.36}$$

Combining (3.26) with the probabilities calculated in (3.27), (3.28), (3.34), (3.35) and (3.36) we can completely present the probability mass function of w:

$$\Pr\left\{\begin{pmatrix}\Delta t_w\\ r_w\\ i_w\\ z_w\end{pmatrix}=\begin{pmatrix}T_{r,x}^R\\ x\\ i\\ 0\end{pmatrix}\right\}=1-\hat{\pi}_x\left(T_{r,x}^R,\pi\right)-\hat{\theta}_x\left(T_{r,x}^R,\theta\right) \tag{3.37}$$

$$\begin{aligned}\Pr\left\{\begin{pmatrix}\Delta t_w\\ r_w\\ i_w\\ z_w\end{pmatrix}=\begin{pmatrix}T_{r,x}^R+1\\ x\\ i\\ 0\end{pmatrix}\right\}= &\ P(x,\mathcal{A}_0)\hat{\pi}\left(T_{r,x}^R,\pi\right)+Q(x,\mathcal{A}_0)\hat{\theta}\left(T_{r,x}^R,\theta\right)\\ &+P(x,x)\hat{\pi}_x\left(T_{r,x}^R,\pi\right)\sum_{n'=n_x^*+1}^{g}\left(\binom{g}{n'}v^{n'}(1-v)^{g-n'}\right)\\ &+Q(x,x)\hat{\theta}_x\left(T_{r,x}^R,\theta\right)\sum_{n'=n_x^*+1}^{g}\left(\binom{g}{n'}(1-u)^{n'}u^{g-n'}\right)\end{aligned} \tag{3.38}$$

$$\begin{aligned}\Pr\left\{\begin{pmatrix}\Delta t_w\\ r_w\\ i_w\\ z_w\end{pmatrix}=\begin{pmatrix}T_{r,x}^R+1\\ j\\ i\\ 0\end{pmatrix}\right\}= &\ P(x,j)\hat{\pi}_x\left(T_{r,x}^R,\pi\right)\sum_{n'=n_j^*+1}^{g}\left(\binom{g}{n'}v^{n'}(1-v)^{g-n'}\right)\\ \forall j\in\mathcal{A}/x\qquad &+Q(x,j)\hat{\theta}_x\left(T_{r,x}^R,\theta\right)\sum_{n'=n_j^*+1}^{g}\left(\binom{g}{n'}(1-u)^{n'}u^{g-n'}\right)\end{aligned} \tag{3.39}$$

$$\Pr\left\{\begin{pmatrix}\Delta t_w\\ r_w\\ i_w\\ z_w\end{pmatrix}=\begin{pmatrix}T_{r,x}^R+1+T_{i,j}^I\\ j\\ j\\ 0\end{pmatrix}\right\}=P(x,j)\hat{\pi}_x\left(T_{r,x}^R,\pi\right)\sum_{n'=0}^{n_j^*}\left(\binom{g}{n'}v^{n'}(1-v)^{g-n'}\right) \quad \forall j\in\mathcal{A} \tag{3.40}$$

$$\Pr\left\{\begin{pmatrix}\Delta t_w \\ r_w \\ i_w \\ z_w\end{pmatrix} = \begin{pmatrix}T^R_{r,x} + 1 + T^I_{i,j} \\ j \\ j \\ 1\end{pmatrix}\right\} = Q(x, j)\hat{\theta}_x\left(T^R_{r,x}, \theta\right)\sum_{n'=0}^{n^*_j}\left(\binom{g}{n'}(1-u)^{n'}(u)^{g-n'}\right) \quad (3.41)$$
$\forall j \in \mathcal{A}$

while the probability that w will get any other value than those which appear above is 0.

4. Bellman Equation Calculation

Finding the optimal decision policy for the Original Problem B requires calculating the value $V(s)$ of all states $s = (t, r, i, \pi, \theta)'$ using Bellman equation (2.5). The trivial case is when $t \geq T$, which results in $V(s) = 0$. Computing the value of a state when $t < T$ requires calculating the maximal expected sum of the reward obtained between the time the current decision is made and the time that decision is fathomed, and the value of the next state. Calculating this maximal value is attained by total enumeration of all feasible decisions. In the following section, we discuss the calculation of this expected sum of reward and value of the next state. Recall that the expected value of a sum is always equal to the sum of the expected values:

$$V(s) = \max_{x(s)}\left\{E\left[c(w,s) + V\left(s^M(s,x,w)\right)\right]\right\} = \max_{x(s)}\left\{E\left[c(w,s)\right] + E\left[V\left(s^M(s,x,w)\right)\right]\right\} \quad (3.42)$$

where:

$$E\left[c(w,s)\right] = \sum_{w' \in \mathcal{W}} c(w',s)\Pr\{w = w'\} \quad (3.43)$$

$$E\left[V\left(s^M(s,x,w)\right)\right] = \sum_{w' \in \mathcal{W}} V\left(s^M(s,x,w)\right)\Pr\{w = w'\} \quad (3.44)$$

Recall that $c(w,s)$ is actually only a function of the Δt_w and z_w components of the information random vector w (2.3), and so for the calculation of the first expected value we only need the joint probability distribution of these two random variables, and not the complete joint probability distribution of all four components of w. Similarly, $s^M(s,x,w)$ is actually a function of just Δt_w, r_w and i_w components of w (3.16), and so for the calculation of the second expected value we only need the joint probability distribution of these three random variables and not the complete joint probability distribution of all four components of w. We define two new functions which represent these dependencies:

$$\tilde{c}(\Delta t_w, z_w, s) \equiv c(w, s) \qquad \forall s \in \mathcal{S}, \forall w = (\Delta t_w, r_w, i_w, z_w)' \in \mathcal{W} \tag{3.45}$$

$$\tilde{s}^M(s, x, \Delta t_w, r_w, i_w) \equiv s^M(s, x, w) \qquad \forall s \in \mathcal{S}, \forall x \in \mathcal{A}, \forall w = (\Delta t_w, r_w, i_w, z_w)' \in \mathcal{W} \tag{3.46}$$

Using this functions we can now rewrite the expected values from (3.43) and (3.44):

$$E[c(w,s)] = E[\tilde{c}(\Delta t_w, z_w, s)] = \sum_{z'_w=0}^{1} \sum_{\Delta t'_w=0}^{\infty} \tilde{c}(\Delta t'_w, z'_w, s) \Pr\{\Delta t_w = \Delta t'_w, z_w = z'_w\} \tag{3.47}$$

$$\begin{aligned} &E\left[V\left(s^M(s,x,w)\right)\right] = \\ &= E\left[V\left(\tilde{s}^M(s,x,\Delta t_w, r_w, i_w)\right)\right] = \sum_{\Delta t'_w=0}^{\infty} \sum_{r'_w \in \mathcal{A}} \sum_{i'_w \in \mathcal{A}} V\left(\tilde{s}^M(s,x,\Delta t'_w, r'_w, i'_w)\right) \Pr\{\Delta t_w = \Delta t'_w, r_w = r'_w, i_w = i'_w\} \end{aligned} \tag{3.48}$$

The joint probabilities appearing in (3.47) and (3.48) are marginal probabilities of the full probability mass function of $w = (\Delta t_w, r_w, i_w, z_w)'$:

$$\Pr\{\Delta t_w = \Delta t'_w, z_w = z'_w\} = \sum_{r'_w \in \mathcal{A}} \sum_{i'_w \in \mathcal{A}} \Pr\{\Delta t_w = \Delta t'_w, r_w = r'_w, i_w = i'_w, z_w = z'_w\} \tag{3.49}$$

$$\Pr\{\Delta t_w = \Delta t'_w, r_w = r'_w, i_w = i'_w\} = \sum_{z'_w=0}^{1} \Pr\{\Delta t_w = \Delta t'_w, r_w = r'_w, i_w = i'_w, z_w = z'_w\} \tag{3.50}$$

Using (3.50) and the probability mass function of w in (3.37)–(3.41) we can explicitly present the expected value in (3.48):

$$E\left[V\left(s^M\left(s,x,w\right)\right)\right]=\left(I\right)\left(II\right)+\sum_{j\in\mathcal{A}}\left(\left(III\right)\left(IV\right)\right)+\sum_{j\in\mathcal{A}}\left(\left(V\right)\left(VI\right)\right)+\left(VII\right)\left(VIII\right)$$

where:

$$\left(I\right)=V\left(\tilde{s}^M\left(s,x,T^R_{r,x},x,i\right)\right)$$

$$\left(II\right)=\left(1-\hat{\pi}\left(T^R_{r,x},\pi\right)-\hat{\theta}\left(T^R_{r,x},\theta\right)\right)$$

$$\left(III\right)=V\left(\tilde{s}^M\left(s,x,T^R_{r,x}+1+T^I_{i,j},j,j\right)\right)$$

$$\left(IV\right)=\sum_{n'=0}^{n^*_j}\left(\binom{g}{n'}\left(1-u\right)^{n'}u^{g-n'}Q\left(x,j\right)\hat{\theta}_x\left(T^R_{r,x},\theta\right)+\binom{g}{n'}v^{n'}\left(1-v\right)^{g-n'}P\left(x,j\right)\hat{\pi}_x\left(T^R_{r,x},\pi\right)\right)$$

$$\left(V\right)=V\left(\tilde{s}^M\left(s,x,T^R_{r,x}+1,j,i\right)\right)$$

$$\left(VI\right)=\sum_{n'=n^*_j+1}^{g}\left(\binom{g}{n'}\left(1-u\right)^{n'}u^{g-n'}Q\left(x,j\right)\hat{\theta}_x\left(T^R_{r,x},\theta\right)+\binom{g}{n'}v^{n'}\left(1-v\right)^{g-n'}P\left(x,j\right)\hat{\pi}_x\left(T^R_{r,x},\pi\right)\right)$$

$$\left(VII\right)=V\left(\tilde{s}^M\left(s,x,T^R_{r,x}+1,x,i\right)\right)$$

$$\left(VIII\right)=Q\left(x,\mathcal{A}_0\right)\hat{\theta}_x\left(T^R_{r,x},\theta\right)+P\left(x,\mathcal{A}_0\right)\hat{\pi}_x\left(T^R_{r,x},\pi\right) \tag{3.51}$$

Similarly, we can use (3.49) and the probability mass function of w to explicitly present the expect value in (3.47) (recall that the reward is non-zero only when $z_w=1$):

$$E\left[c\left(w,s\right)\right]= =\hat{\theta}_x\left(T^R_{r,x},\theta\right)\sum_{j\in\mathcal{A}}\left(\left(1+\gamma\right)^{-\left(t+T^R_{r,x}+1+T^I_{i,j}\right)}Q\left(x,j\right)\sum_{n'=0}^{n^*_j}\binom{g}{n'}\left(1-u\right)^{n'}\left(u\right)^{g-n'}\right) \tag{3.52}$$

Calculating the value of the initial state according to the Bellman equation (2.5) using our results in (3.51) and (3.52), requires going over all combinations of states, decisions and possible information realizations. Recall that the state is defined as $s=\left(t,r,i,\pi,\theta\right)'$. Though the two probability vectors θ and π can theoretically take any value in the continuous interval $\left[0,1\right]$, their values are determined by the history of the Recognizer location. The number of different paths the Recognizer can take in the time horizon T is $\left|\mathcal{A}\right|^T$, and therefore the state space size is $\left|\mathcal{S}\right|=T\cdot\left|\mathcal{A}\right|\cdot\left|\mathcal{A}\right|\cdot\left|\mathcal{A}\right|^T=T\cdot\left|\mathcal{A}\right|^{T+2}$. Examination of the information w Probability Mass Function in (3.37) - (3.41) shows that for every state s and decision x, the size of the information space is

$|\mathcal{W}| = 3 \cdot |\mathcal{A}| + 2$ (by counting the number of possible values of w in the explicit PMF formulas). Combining all together, calculating Bellman equation in the Original Problem requires going over all combinations of state, decision and information realization, which results in a run time of $O\left(T \cdot |\mathcal{A}|^{T+3} \cdot \left(3 \cdot |\mathcal{A}| + 2\right)\right)$ for the backward recursion dynamic programming algorithm (see Powell [8], p. 50). This shows why the Original Problem is intractable for realistic situations with more than a few ACs and time periods. As an example, even a small scenario with 10 ACs and time horizon $T = 5$, results in 16 billion calculations.

B. UPPER BOUND PROBLEM

The Upper Bound Problem $\overline{B}$ is very similar to the Original Problem B: A decision is defined in the same way (though the two problems have different optimal decision policies). The information in $\overline{B}$ is defined in the same way as in B, though its probability distribution is different. The state transition functions of the two problems are closely related. The rewards in the two problems are practically the same, except that we formally use different functional notation because they operate on different state spaces. Lastly, The Bellman equations of the two problems appear to be almost identical, but they use slightly different variables as presented earlier in (2.9). In the following sections, we formally define the Upper Bound Problem $\overline{B}$ and discuss the differences with respect to the Original Problem B.

1. States

Recall that we previously defined a state in $\overline{B}$ as $\overline{s} = \left(\overline{t}, \overline{r}, \overline{i}\right)'$ where $\overline{t}$ is the time, $\overline{r}$ is the Recognizer's location and $\overline{i}$ is the Interceptor's location (2.8). The space of all possible states is denoted $\overline{\mathcal{S}}$.

2. Decision

A decision $\bar{x} \in \mathcal{A}$ is the next AC to be visited by the Recognizer. The decision in $\bar{B}$ is defined in the same way as in the Original Problem B.

3. Information

Let the random vector $\bar{w} = \left(\Delta\bar{t}_w, \bar{r}_w, \bar{i}_w, \bar{z}_w\right)'$ denote the information obtained when a decision is fathomed in the Upper Bound Problem $\bar{B}$. The definition of each of these four vector components is exactly the same as in the Original Problem B, but because each of these random variables has different probability mass function than in the Original Problem B, we use different names for these random variables. The space of possible information values $\mathcal{W}$ is the same for $\bar{B}$ and for B (though the probability of obtaining each realization from these possible values is different).

4. State Transition

The state transition function in the Upper Bound Problem differs from that in the Original Problem by the fact that it does not include the decision $\bar{x}$ as an argument. Recall that the state transition function in the Original Problem requires x as an argument in order to calculate the new state's probability vectors π^M and θ^M, which are not a part of the state definition in the Upper Bound Problem, and as such they do not need to be calculated. The decision $\bar{x}$ is of course a key factor in determining the next state, but it does so indirectly by influencing the obtained information $\bar{w}\left(\bar{s}, \bar{x}\right)$:

$$\bar{s}^M : \bar{\mathcal{S}} \times \mathcal{W} \rightarrow \bar{\mathcal{S}} \tag{3.53}$$

$$\bar{s}^M\left(\bar{s}, \bar{w}\left(\bar{s}, \bar{x}\right)\right) = \begin{pmatrix} \bar{t} + \Delta\bar{t}_w \\ \bar{r}_w \\ \bar{i}_w \end{pmatrix} \tag{3.54}$$

where $\bar{s} = \left(\bar{t}, \bar{r}, \bar{i}\right)'$ is the state and $\bar{w} = \left(\Delta\bar{t}_w, \bar{r}_w, \bar{i}_w, \bar{z}_w\right)'$ is the obtained information.

5. Reward

The reward $\bar{c}$ is a function of the information $\bar{w}$ and the state $\bar{s}$, and is defined as follow:

$$\bar{c}:\mathcal{W}\times\bar{\mathcal{S}}\rightarrow\mathbb{R} \tag{3.55}$$

$$\bar{c}\left(\bar{w},\bar{s}\right)=\begin{cases}\bar{z}_w\cdot\left(1+\gamma\right)^{-\left(\bar{t}+\Delta\bar{t}_w\right)}, & \bar{t}+\Delta\bar{t}_w\leq T\\ 0, & \bar{t}+\Delta\bar{t}_w>T\end{cases} \tag{3.56}$$

where $\bar{w}=\left(\Delta\bar{t}_w,\bar{r}_w,\bar{i}_w,\bar{z}_w\right)'$, $\bar{s}=\left(\bar{t},\bar{r},\bar{i}\right)'$ and T is the time horizon used in the scenario. This definition is practically the same as in the Original Problem B, just using the appropriate Upper Bound Problem's variables.

6. Bellman Equation

We define the value of being in state $\bar{s}$ as the maximum expected cumulative reward that can be obtained from the time of being in state $\bar{s}$ to the time horizon T. This value $\bar{V}\left(\bar{s}\right)$ is given by Bellman equation as presented earlier (2.9):

$$\bar{V}\left(\bar{s}\right)=\begin{cases}\max\limits_{\bar{x}(\bar{s})}\left\{E\left[\bar{c}\left(\bar{w},\bar{s}\right)+\bar{V}\left(\bar{s}^M\left(\bar{s},\bar{w}\right)\right)\right]\right\}, & \bar{t}<T\\ 0, & \bar{t}\geq T\end{cases} \tag{3.57}$$

where $\bar{t}$ is the time of state $\bar{s}=\left(\bar{t},\bar{r},\bar{i}\right)'$.

7. Bellman Equation Calculation

The formulas used for the calculation of a state's value in the Upper Bound Problem $\bar{B}$ are similar to those in the Original Problem B. We use the same formulas for the two expected value terms in the Original Problem Bellman equation, appearing in (3.51) and in (3.52), when we replace the current state's θ and π by the steady-state θ^0 and π^0, with the probability components for both a target and a neutral at the current Recognizer's location $\bar{r}$ set to 0. Let $\bar{\theta}^0$ and $\bar{\pi}^0$ denote these updated probability vectors:

$$\bar{\theta}_a^0=\begin{cases}\theta_a^0, & a\neq\bar{r}\\ 0, & a=\bar{r}\end{cases} \tag{3.58}$$

$$\overline{\pi}_a^0 = \begin{cases} \pi_a^0, & a \neq \overline{r} \\ 0, & a = \overline{r} \end{cases} \tag{3.59}$$

Substituting θ and π with (3.58) and (3.59) in (3.51) and (3.52), while explicitly computing the next state using the state transition function in (3.54), we get the following formulas for computing the Bellman equation in the Upper Bound Problem:

$$E\left[\overline{V}\left(\overline{s}^M = \left(\overline{t} + \Delta\overline{t}_w, \overline{r}_w, \overline{i}_w\right)\right)\right] = (I)(II) + \sum_{j\in\mathcal{A}}\left((III)(IV)\right) + \sum_{j\in\mathcal{A}}\left((V)(VI)\right) + (VII)(VIII)$$

where:

$$(I) = \overline{V}\left(\overline{s}^M = \left(\overline{t} + T^R_{\overline{r},\overline{x}}, \overline{x}, \overline{i}\right)\right)$$

$$(II) = \left(1 - \hat{\pi}\left(T^R_{\overline{r},\overline{x}}, \overline{\pi}^0\right) - \hat{\theta}\left(T^R_{\overline{r},\overline{x}}, \overline{\theta}^0\right)\right)$$

$$(III) = V\left(\overline{s}^M = \left(\overline{t} + T^R_{\overline{r},\overline{x}} + 1 + T^I_{\overline{i},j}, j, j\right)\right)$$

$$(IV) = \left(\sum_{n'=0}^{n_j^*}\left(\binom{g}{n'}(1-u)^{n'} u^{g-n'} Q(\overline{x}, j)\hat{\theta}_{\overline{x}}\left(T^R_{\overline{r},\overline{x}}, \overline{\theta}^0\right) + \binom{g}{n'} v^{n'}(1-v)^{g-n'} P(\overline{x}, j)\hat{\pi}_{\overline{x}}\left(T^R_{\overline{r},\overline{x}}, \overline{\pi}^0\right)\right)\right)$$

$$(V) = \overline{V}\left(\overline{s}^M = \left(\overline{t} + T^R_{r,x} + 1, j, \overline{i}\right)\right)$$

$$(VI) = \left(\sum_{n'=n_j^*+1}^{g}\left(\binom{g}{n'}(1-u)^{n'} u^{g-n'} Q(\overline{x}, j)\hat{\theta}_{\overline{x}}\left(T^R_{\overline{r},\overline{x}}, \overline{\theta}^0\right) + \binom{g}{n'} v^{n'}(1-v)^{g-n'} P(\overline{x}, j)\hat{\pi}_{\overline{x}}\left(T^R_{\overline{r},\overline{x}}, \overline{\pi}^0\right)\right)\right)$$

$$(VII) = \overline{V}\left(\overline{s}^M\left(\overline{t} + T^R_{\overline{r},\overline{x}} + 1, \overline{x}, \overline{i}\right)\right)$$

$$(VIII) = \left(Q(\overline{x}, \mathcal{A}_0)\hat{\theta}_{\overline{x}}\left(T^R_{\overline{r},\overline{x}}, \overline{\theta}^0\right) + P(\overline{x}, \mathcal{A}_0)\hat{\pi}_{\overline{x}}\left(T^R_{\overline{r},\overline{x}}, \overline{\pi}^0\right)\right) \tag{3.60}$$

$$E\left[\overline{c}\left(\overline{w}, \overline{s}\right)\right] = \\ = \hat{\theta}_{\overline{x}}\left(T^R_{\overline{r},\overline{x}}, \overline{\theta}^0\right)\sum_{j\in\mathcal{A}}\left((1+\gamma)^{-\left(\overline{t} + T^R_{\overline{r},\overline{x}} + 1 + T^I_{\overline{i},j}\right)} Q(\overline{x}, j)\sum_{n'=0}^{n_j^*}\binom{g}{n'}(1-u)^{n'}(u)^{g-n'}\right) \tag{3.61}$$

As in the Original Problem, computing the value of the initial state requires going over all combinations of state, decision and information realization. The main difference from the corresponding calculation in the Original Problem is the much smaller size of the state space $\left|\overline{\mathcal{S}}\right| = T \cdot \left|\mathcal{A}\right|^2$, due to the fact that the probability vectors π and θ are not a part of the Upper Bound Problem state $\overline{s} = \left(\overline{t}, \overline{r}, \overline{i}\right)'$. This means that the run time of calculating the value of the initial state in the Upper Bound Problem is

$O\left(T \cdot |\mathcal{A}|^3 \cdot (3 \cdot |\mathcal{A}| + 2)\right)$, which is much faster than the Original Problem run time, which is $O\left(T \cdot |\mathcal{A}|^{T+3} \cdot (3 \cdot |\mathcal{A}| + 2)\right)$ (as discussed in Chapter III.A.4).

When there is a small number of possible transitions from each AC in the AOI (i.e., once in a specific AC, an object can only move to a small number of neighboring ACs), there is a way to improve the run time of this algorithm. This is accomplished by calculating the possible realizations of the information w only for those ACs which have a non-zero probability the object has moved to. Let the set of ACs reachable from AC a in a single time step transition be denoted as the *forward star* of AC a (for all ACs $a \in \mathcal{A}$). Let μ denote the maximal of all ACs' forward star sizes. The results run time of $O\left(T \cdot |\mathcal{A}|^3 \cdot (3 \cdot \mu + 2)\right)$ is indeed an improvement when $\mu < |\mathcal{A}|$.

C. REWARD CALCULATION USING HEURISTIC'S POLICY

Theoretically, calculating the reward collected following the heuristic decision policy could be done in a similar way to the backward solution algorithm for solving the Bellman equation in the Original Problem discussed in Chapter III.A.4, using the formulas for the expected values terms in (3.51) and (3.52). This calculation is simpler than calculating the collected reward following the optimal decision policy, because we do not need to compare the possible future rewards given every possible decision at each state, but only the future state resulting from making a decision based on our heuristic. Nevertheless, this simpler calculation is still intractable as the state space is still very large (we still need to solve for each state in the Original Problem state space $\mathcal{S}$).

The method used in this thesis is to estimate the value of each state following the heuristic decision policy using a Monte-Carlo simulation instead of an exact calculation as discussed in the above paragraph. Instead of directly calculating the expected value in the Bellman equation, we simulate the scenario while randomizing the required realizations of the obtained information w using its known probability mass function. We continue to generate these single-runs while keeping track of the mean and variance of the collected rewards until the confidence interval for the expected reward is sufficiently small.

Figures 5 through 7 demonstrate three states taken from a heuristic calculation run. Each figure presents the probability map for neutrals at each AC in the AOI (π) and the probability map for targets at each AC in the AOI (θ), together with the decision x, the Recognizer location r at the time of the decision, and the Recognizer location r_w at the time the decision has been fathomed. Figure 5 shows the state at time $t = 0$, which represents the steady-state. The Recognizer initial location is AC #18, it decides to visit AC #13 which turns out to be empty, and so the Recognizer eventually ends up in AC #13. Figure 6 shows the state at time $t = 1$, at which the Recognizer location is AC #13 (where it ended up in the previous state transition). Note that both the probability of a target and the probability of a neutral in AC #13 were set to 0 (as discussed in Chapter III.A.2). This time the Recognizer decides to visit AC #17, which also turns out to be empty.

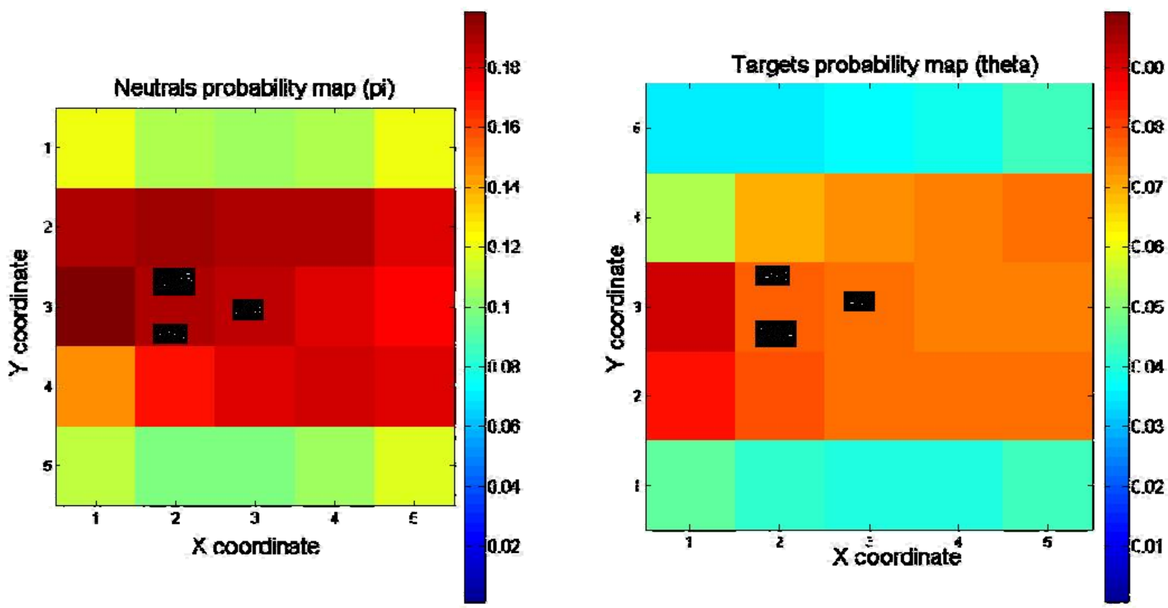

Figure 5. A state example ($t = 0$)

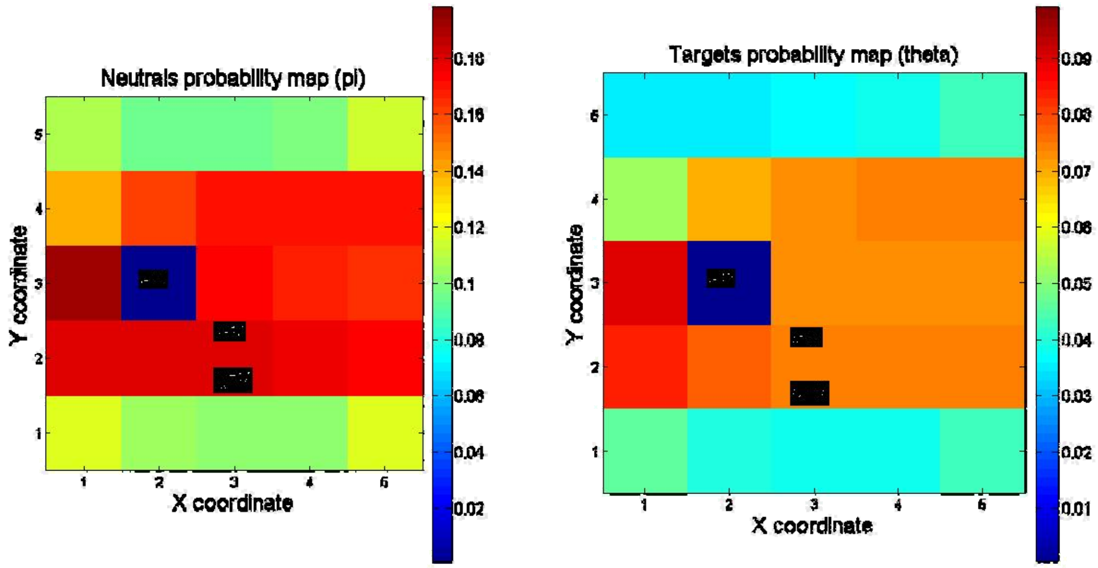

Figure 6. A state example ($t = 1$)

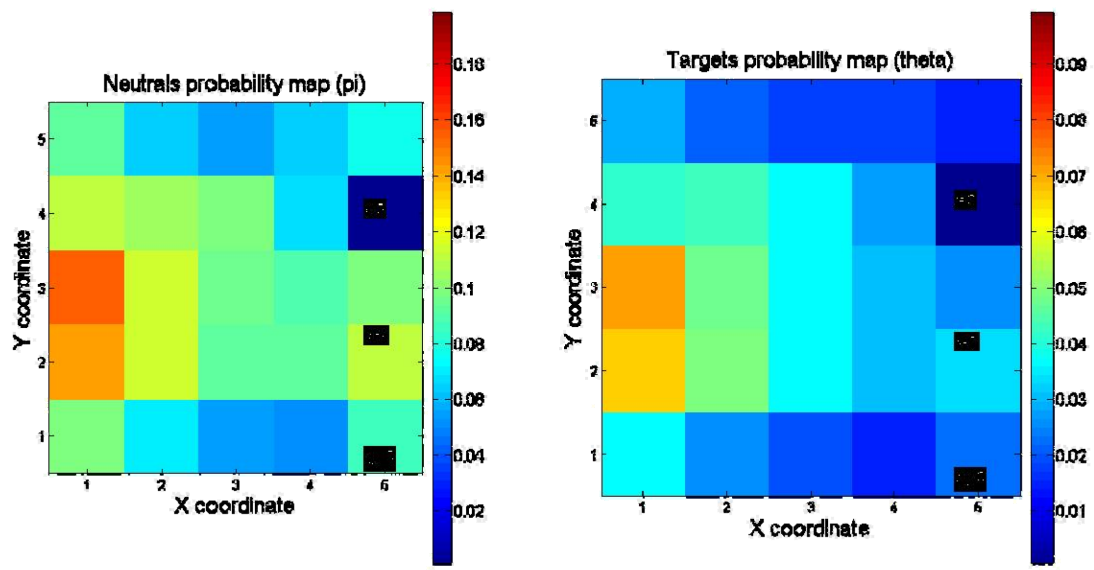

Figure 7. A state example ($t = 18$)

Figure 7 shows the state at time $t = 18$, at which the Recognizer location is AC #9. Note that the overall probabilities of targets and neutrals in each AC in the AOI seem to be much lower than in the beginning of the run (i.e., in steady-state). This is due to the fact that many ACs has been visited and the corresponding probability components of π and θ were set to 0 after each visit, and as time advanced low probabilities "propagated" to the rest of the AOI according to the neutrals and targets Markov chains. The Recognizer has decided to visit AC #7, in which an object was detected and so it was tracked while moving to AC #6 (from this figure we cannot tell whether the object was flagged as a likely target or not, nor if it was eventually intercepted).

Figure 8 demonstrates the result of a heuristic calculation run with time horizon $t = 48$. In the bottom part of the figure we can identify seven events of object detection, out of which three have not resulted in flagging the tracked object as a likely target, and so the Interceptor has not been called for interception ($t = 0,13,17$), while the remaining four detection events resulted in the tracked object being flagged as a likely target and so the Interceptor has been called in for interception ($t = 5, 24, 31, 36$). Out of these four interception attempts, we can identify two successful targets interception which resulted in collecting rewards at times $t = 9, 33$ (the delay is due to the time it takes the Interception to reach the flagged object and intercept it). The two remaining intercepted objects were in fact neutrals.

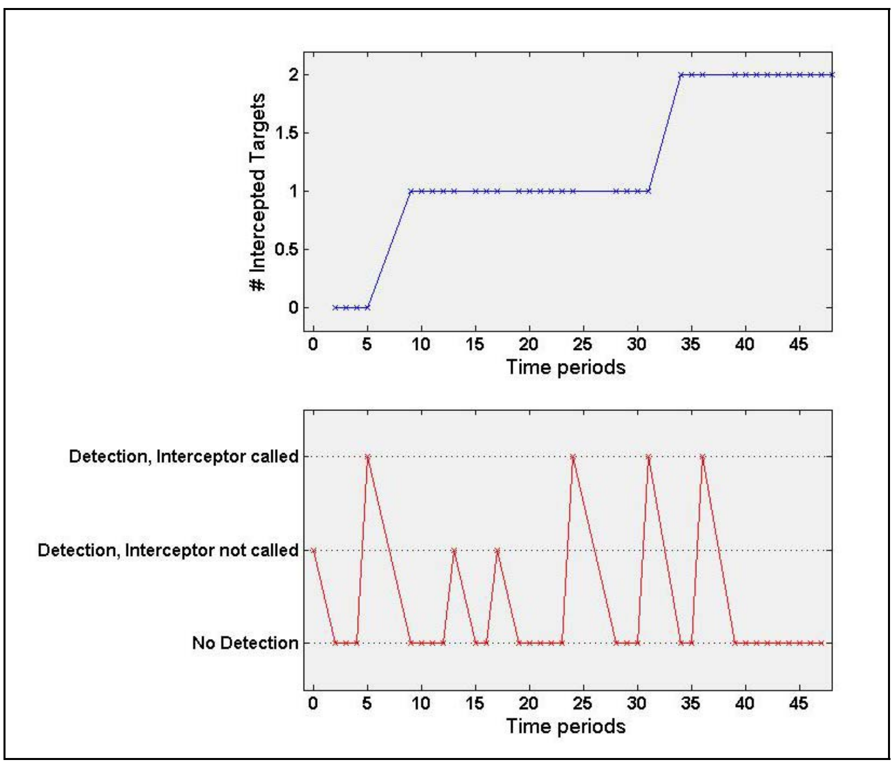

Figure 8. A heuristic run summary example with two intercepted targets ($T = 48$)

IV. NUMERICAL CASE STUDY

A. OVERVIEW

In this chapter, we present a numerical case study that addresses the implementation of the different models, the specific scenarios, the data used, the results of the different runs, and the insights resulting from this analysis. The main scenario chosen to be analyzed is of a 25 ACs strait-like AOI, with slightly different movement patterns for neutrals and targets, and with a time horizon of 48 time steps (representing 12 real-life hours). All scenarios were implemented and analyzed using MATLAB. All together, we have run seven different scenarios, including 47 Upper Bound Problem expected value calculations, 43 heuristic's expected value estimations, and over 50 MATLAB run hours.

B. SPECIFIC SCENARIOS AND DATA

The numerical case study presented in this chapter includes a baseline scenario, and several variations of this baseline scenario for sensitivity analysis and evaluation of the robustness of the baseline scenario's results and insights.

1. Baseline Scenario

The baseline scenario represents a strait-like AOI, with land on the North and South edges of the AOI (i.e., no arrivals from or departures to the North and South of the AOI). The AOI $\mathcal{A}$ is a 5-by-5 square grid of 25 total ACs (see Figure 9). Each AC represents a 5-by-5 nautical-miles area in real-life, with a total are of 625nm^2. The boundary of the AOI, E , is the five ACs on the West edge of the AOI (ACs #1-5) and the five ACs on the East edge of the AOI (ACs #6-10) . AC #26 represents the area $\mathcal{A}_0$ outside of the AOI. The single time-step probability of a neutral and the single time-step probability of a target arriving to each of the ACs in the boundary is 0.05 and 0.01, respectively. The Markov chains representing neutrals movement and targets movements are slightly different to represent an operational situation in which one has some intelligent or other prior knowledge about the differences in the expected movement of the two objects' types. In a single time step, both types of objects can only move to the four immediate neighboring ACs, with no diagonal movement. This results in having a maximal forward star size $\mu = 5$. Generally speaking, we assume neutrals tend to move across the strait (West-East traffic), while targets tend to move perpendicular to the shipping lanes (North-South traffic). This distinction is not absolute, meaning that both a neutral object and a target can move to the exact same neighboring ACs, just with different probabilities (i.e., there is no feasible movement which is unique to either targets or neutrals).

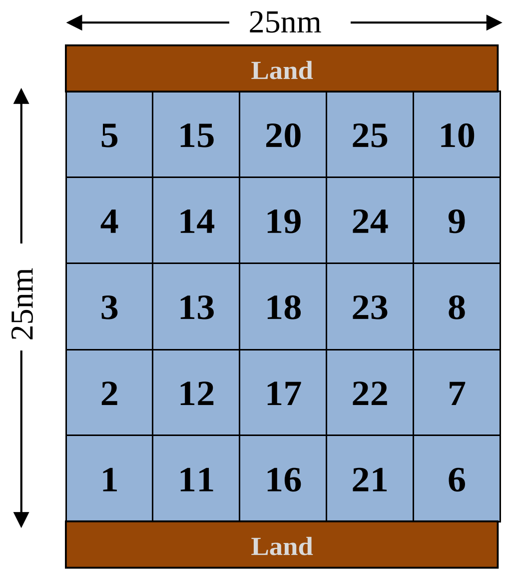

Figure 9. The baseline scenario AOI

For neutrals, the probability of moving on the West-East axis is always double than the probability of moving on the North-South axis. For targets, the situation is flipped, with double the probability of moving on the North-South axis than on the West-East axis. For each of the ACs in $\mathcal{A}$, there is a 0.1 single time step probability of staying in the same AC. The probability of moving back from $\mathcal{A}_0$ to any AC in $\mathcal{A}$ is 0, and an object currently in $\mathcal{A}_0$ will remain there with probability 1 ($\mathcal{A}_0$ is an absorbing state in the individual Markov movement process of each of the objects). Figure 10 shows examples of single time step transition probabilities for targets and for neutrals, in a geographical manner, while Table 1 and Table 2 present the complete Markov transition probability matrices for both neutrals and targets.

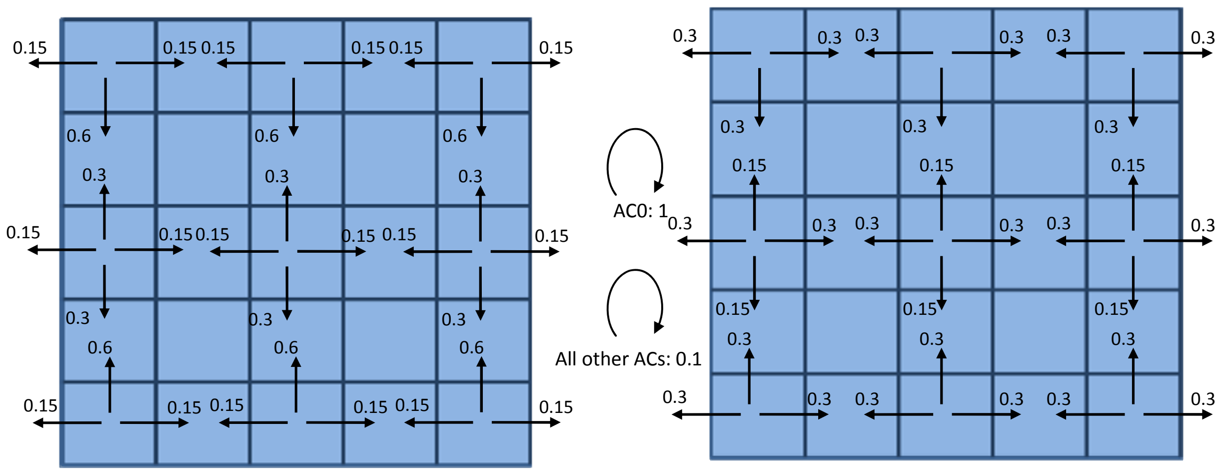

Figure 10. Partial Markov transition prob. of neutrals (right) and targets (left)

Both the Recognizer and the Interceptor in this baseline scenario start in the middle of the AOI in AC #18. A single time step represents 15 minute in real-life. The operational time horizon used is 12 hours, and so we have a time horizon of 48 time steps. We assume the Interceptor has roughly the same velocity as the both neutrals and targets, which is one AC per time step (approximately 20 knots in real-life). The Recognizer velocity is assumed to be four times the velocity of the Interceptor and of the objects (approximately 80 knots in real-life). The Recognizer's and Interceptor's traveling times between each pair of ACs is calculated as following:

$$T^{R}_{a,a'} = \left\lceil \frac{\|a - a'\|}{R_{velocity}} \right\rceil \tag{4.1}$$

$$T^{I}_{a,a'} = \left\lceil \frac{\|a - a'\|}{I_{velocity}} \right\rceil \tag{4.2}$$

where $R_{velocity}$ and $I_{velocity}$ are the Recognizer's and Interceptor's velocities, respectively, and $\|a - a'\|$ is the Euclidean metric representing the geographical distance between AC a and AC a'. Traveling times between all pairs of ACs are presented in Table 3 and Table 4. Recognizer traveling times include detection time and Interceptor traveling times include boarding time.

AC#	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	26
1	0.1	0.3	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3
2	0.15	0.1	0.15	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3
3	0	0.15	0.1	0.15	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0.3
4	0	0	0.15	0.1	0.15	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0.3
5	0	0	0.3	0	0.1	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0.3
6	0	0	0	0	0	0.1	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0.3
7	0	0	0	0	0	0.15	0.1	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.3
8	0	0	0	0	0	0	0.15	0.1	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0.3
9	0	0	0	0	0	0	0	0.15	0.1	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0.3
10	0	0	0	0	0	0	0	0	0.3	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.3
11	0.3	0	0	0	0	0	0	0	0	0	0.1	0.3	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0
12	0	0.3	0	0	0	0	0	0	0	0	0.15	0.1	0.15	0	0	0	0.3	0	0	0	0	0	0	0	0	0
13	0	0	0.3	0	0	0	0	0	0	0	0	0.15	0.1	0.15	0	0	0	0.3	0	0	0	0	0	0	0	0
14	0	0	0	0.3	0	0	0	0	0	0	0	0	0.15	0.1	0.15	0	0	0	0.3	0	0	0	0	0	0	0
15	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0.3	0.1	0	0	0	0	0.3	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0.1	0.3	0	0	0	0.3	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.15	0.1	0.15	0	0	0	0.3	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.15	0.1	0.15	0	0	0	0.3	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.15	0.1	0.15	0	0	0	0.3	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.3	0.1	0	0	0	0	0.3	0
21	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0.1	0.3	0	0	0	0
22	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.15	0.1	0.15	0	0	0
23	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.15	0.1	0.15	0	0
24	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.15	0.1	0.15	0
25	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.3	0.1	0
26	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	1

Table 1. Neutrals Markov transition probabilities matrix *P*

AC#	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	26
1	0.1	0.6	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15
2	0.3	0.1	0.3	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15
3	0	0.3	0.1	0.3	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0	0	0	0.15
4	0	0	0.3	0.1	0.3	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0	0	0.15
5	0	0	0.6	0	0.1	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0	0.15
6	0	0	0	0	0	0.1	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0.15
7	0	0	0	0	0	0.3	0.1	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.15
8	0	0	0	0	0	0	0.3	0.1	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0.15
9	0	0	0	0	0	0	0	0.3	0.1	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0.15
10	0	0	0	0	0	0	0	0	0.6	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0.15
11	0.15	0	0	0	0	0	0	0	0	0	0.1	0.6	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0
12	0	0.15	0	0	0	0	0	0	0	0	0.3	0.1	0.3	0	0	0	0.15	0	0	0	0	0	0	0	0	0
13	0	0	0.15	0	0	0	0	0	0	0	0	0.3	0.1	0.3	0	0	0	0.15	0	0	0	0	0	0	0	0
14	0	0	0	0.15	0	0	0	0	0	0	0	0	0.3	0.1	0.3	0	0	0	0.15	0	0	0	0	0	0	0
15	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0.6	0.1	0	0	0	0	0.15	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0.1	0.6	0	0	0	0.15	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.3	0.1	0.3	0	0	0	0.15	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.3	0.1	0.3	0	0	0	0.15	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.3	0.1	0.3	0	0	0	0.15	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.6	0.1	0	0	0	0	0.15	0
21	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0.1	0.6	0	0	0	0
22	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.3	0.1	0.3	0	0	0
23	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.3	0.1	0.3	0	0
24	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.3	0.1	0.3	0
25	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0	0	0	0	0	0	0.15	0	0	0	0.6	0.1	0
26	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	1

Table 2. Targets Markov transition probabilities matrix *Q*

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

Table 3. Recognizer’s all AC pairs transition times

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

Table 4. Interceptor’s all AC pairs transition times

The Recognizer's sensor is assumed to take three glimpses at the tracked object during the single time step tracking phase ($g = 3$). The false positive and false negative detection probabilities of a target are both 0.2 ($u = v = 0.8$).

The discount factor used for discounting rewards is $\gamma = 0.05$, which means that a target intercepted at the end of the 12 hours time horizon has approximately $1/10$ of the operational value of a target intercepted at $t = 0$.

The value of the probability threshold M , which the Recognizer uses to flag a tracked object as either a likely target or a likely neutral, is systematically varied to examine its effects on the results.

Using a MacBook Pro with Dual-Core 2.53GHz CPU and 4GB of RAM, computing the expected reward of the Upper Bound Problem with this scenario has a run time of approximately 30 minutes, while computing the expected reward following the heuristic decision policy with this scenario has a run time of approximately 6 minutes.

2. Additional Scenarios

To better evaluate our suggested heuristic's performance and to gain a deeper insight into the nature of this problem, we perform a brief parametrical study and sensitivity analysis on several key parameters in the baseline scenario. This is accomplished by running and analyzing several other scenarios based on the baseline scenario. These scenarios include a zero-discounting scenario, a scenario with poor sensor capabilities, a scenario with extended boarding time for the Interceptor, a 48 hours time horizon scenario and an 1600nm^2 8-by-8 AOI scenario.

C. RESULTS AND INSIGHTS

The following sections present the results of the numerical case studies with respect to the different scenarios examined, and discuss some insights derived from these results. The first section discusses the main question the numerical case study is intended to answer: How well does the suggested heuristic perform? The subsequent sections discuss additional insights and interesting results regarding the MIM operational scenario.

1. Performance of the Heuristic Decision Policy

The key motivation for this numerical case study is to evaluate the performance of our suggested heuristic in the context of some representative scenarios. The main result is the gap between the expected reward (i.e., discounted expected number of intercepted targets) achieved by the heuristic decision policy and that achieved by the Upper Bound Problem optimal decision policy. Note that this gap is an upper bound on the true gap between the performance of the heuristic decision policy and the Original Problem optimal decision policy, and so the gap presented here is a "worst case scenario". Table 5 and Figure 11 present the results of both the heuristic and the Upper Bound Problem decision policies in the baseline scenario, using several different values as the probability threshold for deciding whether or not to call the Interceptor at the end of the tracking phase of a detected object. The error bars in Figure 11 (and later in Figures 12–14) represent the confidence intervals of the estimated expected reward following the heuristic policy.

In this baseline scenario, the gap is about 30% on average for different values of the probability threshold M, with relatively little sensitivity to the choice of M. This means that using the heuristic decision policy results with at least ~70% of the Original Problem expected reward.

Additional scenarios, presented in the subsequent sections, support the statement that the heuristic is useful in obtaining a simple and effective decision policy for the MIM operational scenario. By choosing the appropriate value for the probability threshold M, we get that all examined scenarios resulted in a gap of less than 40% between the heuristic expected reward and the optimal expected reward.

Probability threshold M	Upper Bound expected reward	Heuristic expected reward	% gap
0	0.72	0.50	**30.9**
0.01	0.75	0.52	**30.5**
0.05	0.76	0.54	**29.7**
0.1	0.77	0.52	**32.2**
0.15	0.77	0.53	**30.8**
0.25	0.74	0.51	**31.0**
0.35	0.74	0.51	**31.4**
0.5	0.68	0.47	**30.8**
0.75	0.63	0.41	**34.3**
0.9	0.44	0.30	**32.2**

Table 5. Baseline scenario results

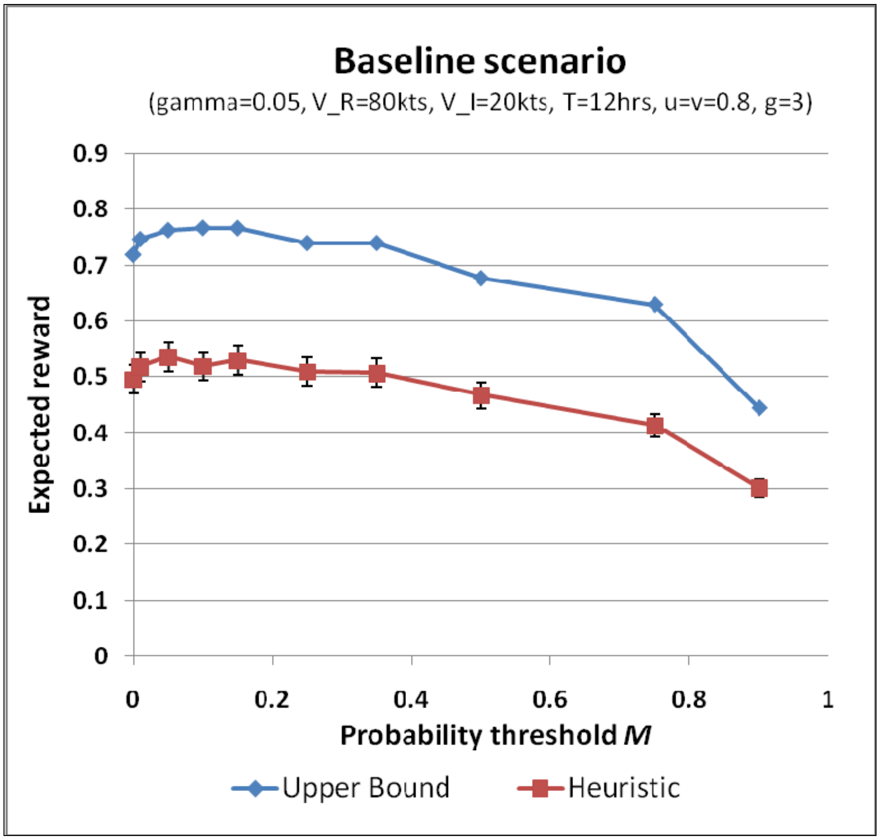

Figure 11. Baseline scenario results

2. No Discounting of Rewards

This scenario is identical to the baseline scenario except we did not use any discounting ($\gamma = 0$).

Probability threshold M	Upper Bound expected reward	Heuristic expected reward	% gap
0	1.92	1.21	**37.1**
0.01	2.02	1.22	**39.4**
0.05	2.10	1.28	**39.1**
0.1	2.10	1.27	**39.2**
0.15	2.10	1.23	**41.1**
0.25	2.03	1.20	**40.6**
0.35	2.03	1.23	**39.4**
0.5	1.81	1.14	**37.0**
0.75	1.72	0.90	**47.8**

Table 6. No-discounting scenario results

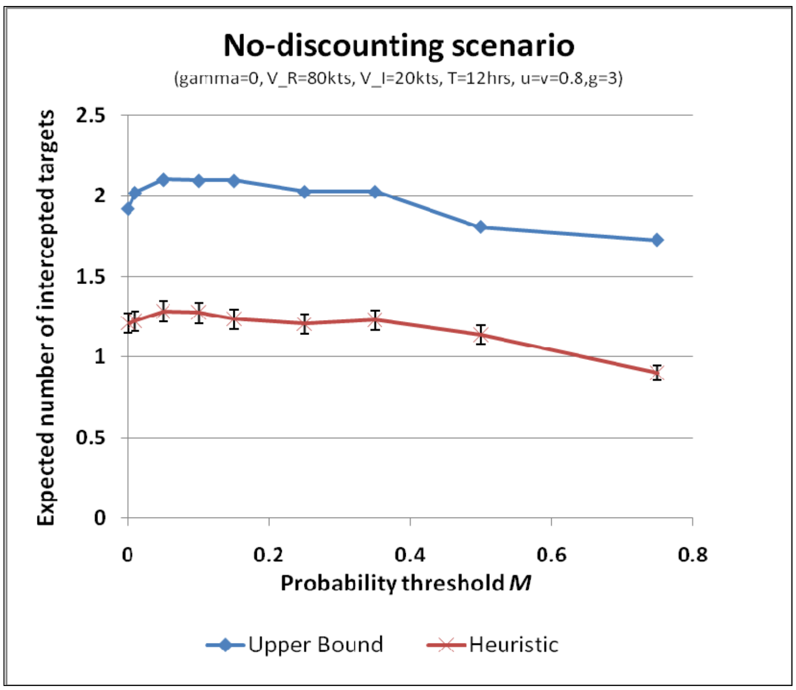

Figure 12. No-discounting scenario results

In this no-discounting scenario, the gap is slightly larger than in the baseline scenario, with about 40% gap on average for different values of the probability threshold M . This means that using the heuristic decision policy (in this no-discounting scenario) results in expected reward that is at least ~60% of that in the Original Problem. The shape of the graph in Figure 12 is very similar to the baseline scenario result, with relatively little sensitivity to the choice of M . A possible explanation to the slightly better results when using discounting over no-discounting is the "greedy" nature of the heuristic. A myopic approach, as used in this heuristic, makes more operational sense when there is higher operational value for intercepting targets in the near future than for targets intercepted later in the future. Nevertheless, even without any discounting at all, the myopic heuristic is useful, with only ~10% worse performance than with a discounting factor of $\gamma = 0.05$ as used in the baseline scenario.

3. Low Quality Signature Recognition

In this scenario, we assumed the Recognizer sensor has poor signature recognition, meaning they can hardly tell between a neutral and a target based on the signature of the tracked object. The false positive and false negative detection probabilities of a target were assumed to both be 0.4 with $u = v = 0.6$ (Note that $u = v = 0.5$ is a useless sensor with no ability to identify the type of a tracked object). This scenario was run without discounting the intercepted targets ($\gamma = 0$). As expected, the overall results (Table 7 and Figure 13) are worse than the corresponding scenario with better sensor quality.

We can easily notice the effect of the poor sensor quality on the results of this run: Except when choosing to intercept every detected target ($M = 0$), as we choose higher values for the probability threshold M we get worse and worse results, meaning less and less targets intercepted. This can be explained by the fact that with such poor quality signature recognition sensor, situations in which a tracked object has a probability of being a target of 0.2 and above is rare (remember that the probability of an object being a neutral prior to any sensing is higher than the probability of that object being a target, that is because of the arrival probabilities of neutrals and targets to the AOI). This results in the fact that any threshold of 0.2 and above is rarely met.

This situation of using poor sensors is obviously unwanted, but if there is no way to avoid it, a very low probability threshold is best. Using a high probability threshold in this scenario does not make any sense, and so we ignore the poor results of the heuristic in these cases (as bad as ~94% gap) when evaluating the overall heuristic performance.

Probability threshold M	Upper Bound expected reward	Heuristic expected reward	% gap
0	1.92	1.23	**36.2**
0.01	1.92	1.20	**37.5**
0.05	1.92	1.22	**36.3**
0.1	1.92	1.19	**38.4**
0.15	1.83	1.15	**37.3**
0.25	1.62	1.01	**37.5**
0.35	1.51	0.79	**47.9**
0.5	0.99	0.55	**45.0**
0.75	0.28	0.02	**93.4**

Table 7. Poor signature recognition scenario results

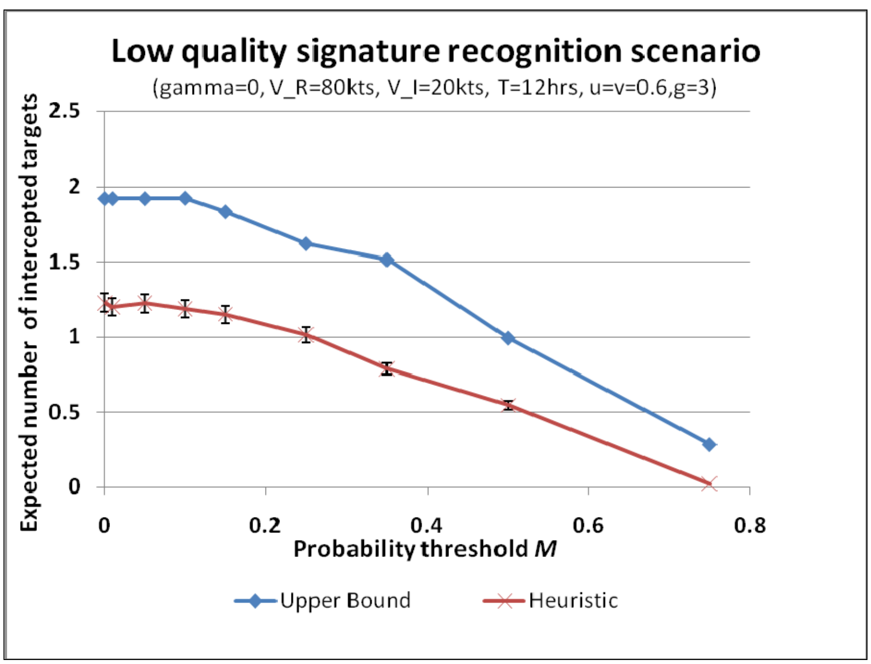

Figure 13. Poor signature recognition scenario results

4. How Often Should We Call the Interceptor?

Inspecting the results of the baseline scenario (Table 5 and Figure 11), we notice that the "best" probability threshold in the heuristic context is $M = 0.05$. This threshold value results in an expected number of 0.76 intercepted targets in the Upper Bound Problem, which is practically the same as the optimal 0.77 targets. A threshold of $M = 0.05$ also results in the lowest observed gap of 29.7%. All of the above can point to the fact that it is good to choose a value of $M = 0.05$ when creating the CONOPS of an interdiction force under the assumptions of the baseline scenario. This result appears to be counter intuitive, as it appears intuitive to choose a higher threshold. It seems to make sense to choose a threshold of at least 0.5, meaning that we should call the Interceptor (and "waste" the time associated with the interception) only when it is at least more likely that a tracked object is a target than it is a neutral. The first guess of an effective threshold value, when this scenario was first implanted and tested, was in fact $M = 0.8$ (the motivation was to set the threshold high enough as to only call the Interceptor when it is most likely that a target will be eventually intercepted). The fact that such high probability threshold values are worse than much lower values as $M = 0.05$ was initially surprising. To better understand these counterintuitive results, we evaluated several scenarios with longer interception times. In the baseline scenario, the time the interdiction force "pays" for calling in the interceptor over letting the tracked object go without intercepting it, is only the travel time of the Interceptor from its current location to the interception location. While speculating that this "time-penalty" for risking an interception is too low, we constructed and analyzed two new scenarios in which we artificially extended the time of interception by introducing a *boarding time* for the interception of objects. This means that every time the Interceptor is called in to intercept a likely target, the time it takes until the interception is complete is the sum of the travel time of the Interceptor to the interception location and the boarding time. A longer boarding time discourages calling the Interceptor "too often." Table 8 and Figure 14 compare the results of these three scenarios: the baseline scenario with 0 boarding time, a scenario with 5 time steps boarding time, and a scenario with 20 time steps boarding time.

Probability threshold M	*Boarding time = 0 time steps*			*Boarding time = 5 time steps*			*Boarding time = 20 time*		
	Upper Bound expected reward	Heuristic expected reward	% gap	Upper Bound expected reward	Heuristic expected reward	% gap	Upper Bound expected reward	Heuristic expected reward	% gap
0	0.72	0.50	**30.9**	0.35	0.29	**17.3**	0.09	0.09	**3.5**
0.01	0.75	0.52	**30.5**	0.39	x	**x**	0.10	x	**x**
0.05	0.76	0.54	**29.7**	0.42	0.34	**19.7**	0.12	0.11	**12.1**
0.1	0.77	0.52	**32.2**	0.45	x	**x**	x	x	**x**
0.15	0.77	0.53	**30.8**	0.45	0.36	**21.1**	x	0.12	**x**
0.25	0.74	0.51	**31.0**	0.46	x	**x**	0.14	x	**x**
0.35	0.74	0.51	**31.4**	0.46	0.35	**23.3**	0.14	0.13	**10.3**
0.5	0.68	0.47	**30.8**	0.43	0.33	**23.2**	0.14	0.12	**13.8**
0.75	0.63	0.41	**34.3**	0.41	0.30	**25.5**	x	x	**x**
0.9	0.44	0.30	**32.2**	0.30	0.22	**26.6**	0.11	0.09	**18.5**

Table 8. Sensitivity to boarding time (*x* marks scenarios which have not been calculated)

The main motivation for the analysis of these scenarios is to examine the shape of the graphs, and so not all data points are computed for both scenarios.

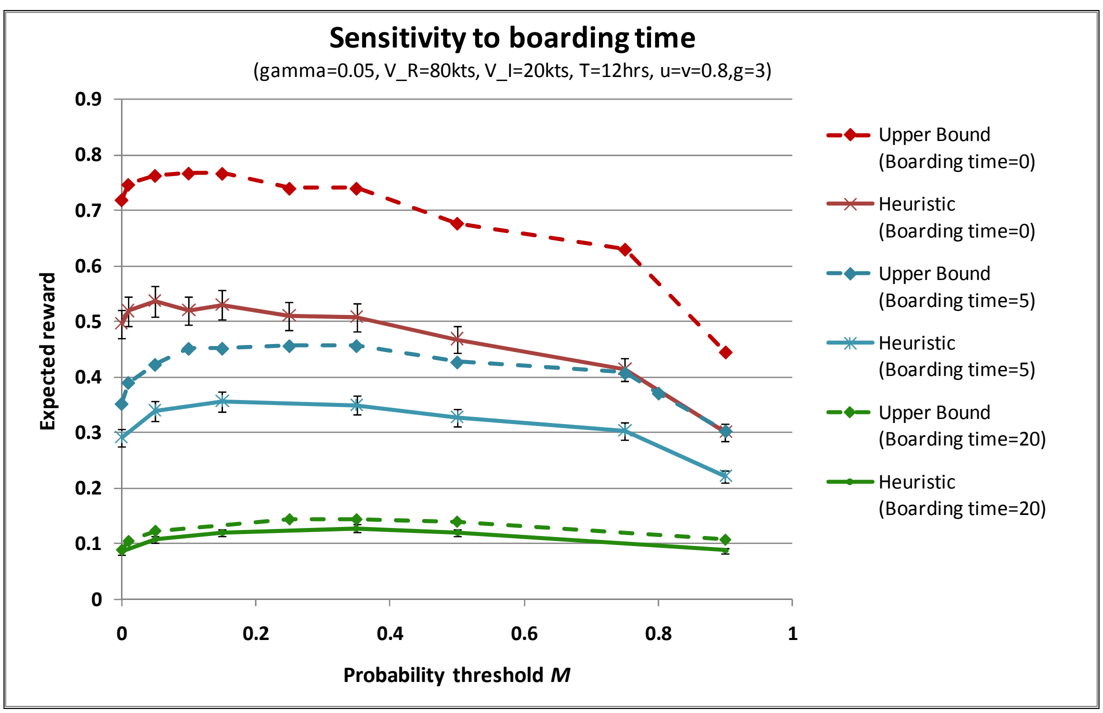

Figure 14. Sensitivity to boarding time

The results of the scenario with five time steps boarding time are indeed more intuitive than the baseline scenario with 0 boarding time. Note the shape of the graphs in the two new scenarios, for both the Upper Bound expected reward and the heuristic expected reward, which suggests it is best to choose higher values for the probability threshold parameter M than in the baseline scenario. A threshold value of approximately 0.2 seems to have the best results in the scenario with boarding times of five time steps, while a value of approximately 0.4 seems to be the best in the scenario with boarding times of 20 time steps. These results make sense since with longer overall interception time it is not that efficient to call the Interceptor too often.

5. Extended Time Horizon

All the scenarios presented so far had a time horizon of 12 hours ($T = 48$ time steps). To confirm the heuristic's performance and insights presented earlier in this chapter, we examine a scenario with 24 hours time horizon ($T = 96$ time steps, with all other parameters as in the baseline scenario). The heuristic's expected reward in this scenario is 0.57 and the Upper Bound expected reward is 0.85, with a gap of 33%. The heuristic performance in this scenario is very close to the observed performance in the baseline scenario with half the length of the time horizon. This encouraging result supports our statement that difference between the heuristic and optimal expected rewards is less than ~30%.

6. 8-by-8 AOI

Another scenario examined for confirming our results and insights is a scenario with a larger AOI: a 1600nm^2 AOI (8-by-8 ACs) instead of the 625nm^2 AOIs (5-by-5 ACs) used in all previous scenarios. The number of ACs in this enlarged AOI is $\frac{64}{25} \approx 2.5$ times the number of ACs in the baseline scenario. Recall that the run time of the Upper Bound Problem backward calculation is $O\left(T \cdot |\mathcal{A}|^3 \cdot (3 \cdot \mu + 2)\right)$, which means that the running time for the Upper Bound calculation of this 8-by-8 scenario is approximately 16 times longer than the 5-by-5 scenario run time. As the baseline scenario

Upper Bound calculation takes approximately 30 minutes, this 8-by-8 scenario indeed takes approximately 8 hours. For this reason, we only examined a single run of this 8-by-8 scenario. The rest of the scenario parameters are as in the baseline scenario.

The heuristic expected reward in this scenario is found to be 0.45, while the Upper Bound expected reward is 0.62, with a gap of 28%. The result of this scenario is in agreement with previously presented results and, therefore, supports our statements regarding the heuristic performance.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The goal of this thesis is to develop a useful tactical decision aid for an interdiction force in a MIM scenario. We discuss the reasons why finding the exact optimal decision policy in this scenario is an intractable problem, and so we suggest a heuristic sub-optimal decision policy instead of the optimal one.

Based on the analysis of several numerical case studies, we conclude that the number of intercepted targets following the heuristic decision policy is at least 60% of the number of targets intercepted following the optimal decision policy. This percentage improves to approximately 70% when discounting intercepted targets with a discount factor of 0.05 with respect to time steps of 15 minutes.

Based on the observed heuristic performance in the numerical case study, we recommend the use of this heuristic in any MIM scenario which closely resembles the MIM scenario discussed in this thesis. While 40% is indisputably a significant gap, the heuristic decision policy calculation can be completed almost instantaneously while the true optimal decision policy is intractable to compute in any plausible operational situation.

Furthermore, we gained additional insights regarding the effect that several operational and technical parameters of the interdiction force have on the expected outcome of the MIM. Such insights include how to better choose the probability threshold for intercepting a tracked object, and what performance should we expect from our heuristic and from any other feasible decision policy (including the optimal one) under different scenarios.

This thesis is a first attempt to obtain an optimal operational policy for a synchronized sensor-interceptor maritime interdiction force or to quantitatively analyze a heuristic approach to this problem. The models, methodology and even the implementation code developed in this thesis, can be easily applied to future research of this problem or similar ones, as briefly discussed in the next section.

B. SUGGESTED WORK AHEAD

There are several interesting research avenues to be extended from this thesis. New heuristic decision policies can be developed and evaluated with minimal modifications to the heuristic decision policy suggested in this thesis.

Improved implementation of the models presented in this thesis with faster run time and more efficient memory use, can enable the analysis of bigger and more realistic operational scenarios that will reinforce our confidence in the heuristic performance presented in this research.

Extensions of the scenarios and models in this thesis can include the introduction of multiple Recognizers and/or Interceptors, the optimization of the Interceptor location throughout the scenario (instead of waiting at its current location until called for by the Recognizer), or a more realistic interception model for the phase between the flagging of a tracked object as a likely target and the actual interception.

Another extension of this thesis can be to introduce “Red-team intelligence”, meaning that we allow the targets to act strategically react to the interdiction force’s actions. Such possible reactions must be taken into account when optimizing the operational policy of the interdiction force, using game-theoretic approaches.

An alternative approach to the numerical analysis of any suggested heuristic is to attempt to analytically prove an approximated sub-optimal decision policy.

APPENDIX. MATLAB IMPLEMENTATION OVERVIEW

The implementation of all models and algorithms in this thesis was coded in MATAB 7 (R14). All runs were done on a MacBook Pro with an Intel Dual-Core 2.53GHz CPU with 4GB RAM, on Windows XP Professional.

The MATLAB code (Table 9) implements the heuristic expected value calculation (*H*) and the Upper Bound Problem expected value calculation (*U*). The code consists of two main run files and 13 supporting functions (some of which are used in both main run files). All together, there are approximately 2000 lines of code.

Filename	**Description**	***H***	***U***
Heuristic_calc.m	Heuristic expected value calculation (main run file)	X	
Upper_Bound_calc.m	Upper Bound Problem expected value calculation (main run file)		X
create_AOI.m	Creates the AOI: Markov transition matrices, arrival probabilities and ACs trace (translation between ACs index and coordinates)	X	X
travel_times.m	Calculates all ACs pairs travel times for both Recognizer and Interceptor	X	X
steady_state.m	Calculates the targets and neutrals steady-state probability vectors	X	X
x_heuristic.m	Calculates the current decision following the heuristic policy	X	
w_realization.m	Get a realization of the information according to the appropriate PMF	X	
reward.m	Calculates the obtained reward	X	
sM.m	Calculates the new state after decision is fathomed	X	
pi_hat.m	Targets probability vector update function	X	
pi_hat_a.m	Single AC target probability update function	X	X
theta_hat.m	Neutrals probability vector update function	X	
theta_hat_a.m	Single AC neutral probability update function	X	X
theta_rec.m	Calculates the probability for a target after the tracking phase	X	X
plot_prob_map.m	Plots the probability vectors as probability color maps	X	X

Table 9. MATLAB code filenames and descriptions

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] A. R. Washburn, *Search and Detection*, 4th Ed., Linthicum, MD: INFORMS, 2002.

[2] L. D. Stone, *Theory of Optimal Search*, 2nd Ed., Linthicum, MD: INFORMS, 2004.

[3] J. O. Royset and H. Sato, "Route Optimization for Multiple Searchers," paper in review, Naval Postgraduate School, Monterey, CA, 2009.

[4] H. Sato and J. O. Royset, "Path Optimization for the Resource-Constrained Searcher," paper in review, Naval Postgraduate School, Monterey, CA.

[5] S. L. Smith, *Task Allocation and Vehicle Routing in Dynamic Environments,* Santa Barbara, CA: University of California, 2009.

[6] M. Kress, R. Szechtman, and J. S. Jones, "Efficient Employment of Non-Reactive Sensors," *Military Operations Research*, vol. 13, No. 4, 2008.

[7] T. H. Chung, M. Kress, and J. O. Royset, "Probabilistic Search Optimization and Mission Assignment for Heterogeneous Autonomous Agents," *Proceedings of the International Conference on Robotics and Automation, Kobe, Japan*, 2009.

[8] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*, Hoboken, NJ: Wiley-Interscience, 2006.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor Johannes O. Royset
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California

4. Professor Moshe Kress
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California

5. Professor Yeo Tat Soon
   Director, Temasek Defense Systems Institute (TDSI)
   National University of Singapore
   Singapore