NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMAL DEPLOYMENT OF UNMANNED AERIAL VEHICLES FOR BORDER SURVEILLANCE

by

Volkan Sözen

June 2014

Thesis Advisor:	Emily Craparo
Second Reader:	Thomas Lucas

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2014	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
OPTIMAL DEPLOYMENT OF UNMANNED AERIAL VEHICLES FOR BORDER SURVEILLANCE

5. FUNDING NUMBERS

6. AUTHOR(S) Volkan Sözen

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. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Border surveillance is an important concern for most nations wanting to detect and intercept intruders that are trying to trespass a border. These intruders can include terrorists, drug traffickers, smugglers, illegal immigrants, and others who represent a threat to national interests. Unmanned aerial vehicles (UAVs) allow for modernization and improvement of border surveillance. There are a number of advantages to using UAVs. Many UAVs can be controlled by a single operator, which reduces personnel costs; they are very fast and can patrol large regions; and they have wider regions of visibility than conventional surveillance methods, which increases the probability of detecting intruders. This thesis formulates mathematical models designed to find the best way to utilize a given fleet of UAVs by deciding their routes, altitudes, and speeds in order to maximize the probability of detecting intruders trying to trespass a given border. These models will enable decision makers to effectively acquire and employ a UAV fleet for border surveillance.

14. SUBJECT TERMS Border patrol, barrier search, UAV

15. NUMBER OF PAGES 107

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	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

OPTIMAL DEPLOYMENT OF UNMANNED AERIAL VEHICLES FOR BORDER SURVEILLANCE

Volkan Sözen
First Lieutenant, Turkish Army
B.S., Middle East Technical University, Ankara, Turkey, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2014

Author: Volkan Sözen

Approved by: Emily M. Craparo
Thesis Advisor

Thomas W. Lucas
Second Reader

Robert F. Dell
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Border surveillance is an important concern for most nations wanting to detect and intercept intruders that are trying to trespass a border. These intruders can include terrorists, drug traffickers, smugglers, illegal immigrants, and others who represent a threat to national interests. Unmanned aerial vehicles (UAVs) allow for modernization and improvement of border surveillance. There are a number of advantages to using UAVs. Many UAVs can be controlled by a single operator, which reduces personnel costs; they are very fast and can patrol large regions; and they have wider regions of visibility than conventional surveillance methods, which increases the probability of detecting intruders. This thesis formulates mathematical models designed to find the best way to utilize a given fleet of UAVs by deciding their routes, altitudes, and speeds in order to maximize the probability of detecting intruders trying to trespass a given border. These models will enable decision makers to effectively acquire and employ a UAV fleet for border surveillance.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

Border surveillance is an important problem for most nations in order to detect and intercept intruders that are trying to trespass a border. These intruders can include terrorists, drug traffickers, smugglers, illegal immigrants, and others who represent a threat to national interests.

Unmanned aerial vehicles (UAVs) allow a modernization and improvement to border surveillance. There are a number of advantages to using UAVs. Many UAVs can be controlled by a single operator, reducing personnel cost. UAVs are fast and thus can patrol large regions; and they have wider regions of visibility than conventional surveillance methods, which increase the probability of detecting intruders.

We study the problem of monitoring a straight line border over a flat terrain without any line-of-sight issues. In other words, the detection probability only depends on the range of the target. The target tries to pass through the border. We assume that the target cannot see the searcher and moves at a constant speed without changing direction.

For simplicity, we assume that the searcher moves in a straight line and can change direction instantaneously. We generally assume the searcher carries a "cookie-cutter" sensor, although we also analyze imperfect sensors.

First, we study a simple border patrol problem by analytical and Monte Carlo simulation methods. We improve the detection probabilities computed in the literature by using highly accurate geometrically-based calculations.

After verifying the models for the simple border patrol problem, we add some complexity to it by adding another searcher. We propose two different searcher paths for the multiple-searcher problem: the disjoint path and the common path. We develop both analytical and Monte Carlo simulation models for the disjoint path problem by building upon the single searcher case. We develop a Monte Carlo simulation model for the common path problem.

When we compare the results of the two multiple-searcher cases, we notice the importance of allocating the border to the two searchers. Therefore, we study the optimal

allocation, which is the allocation resulting in the maximum detection probability. We study two ways to determine the optimal allocation: by analytical methods and by Monte Carlo simulation. We conclude that we should choose the disjoint path rather than the common path and allocate the border to the searchers optimally to maximize the detection probability.

We analyze the effect of degrading detection performance with increasing searcher speed. We perform our analysis for a single searcher and observe that the maximum detection probability occurs at a certain searcher speed and hence detection radius. We extend this analysis to multiple-searcher problems. Besides analyzing the speeds that result in maximum detection probability, we determine the optimal allocation to maximize the detection probability.

We add complexity to the simple border patrol problem by studying geometric considerations in which we vary the angles of the border and searcher's path. We develop both analytical and Monte Carlo simulation models for the geometric considerations on the searcher's path. We note that the detection probability is nearly independent of the searcher's path's angle before introducing *d*, the distance from the border that a detected object can be classified as a target. After introducing *d*, we note that the detection probability decreases considerably after a certain angle, which depends on *d*.

After noting a similar behavior when we change the angle of the border without varying the searcher's angle, we vary both angles and observe the maximum detection probability when the searcher's path is aligned with the border. We also note that we still have some flexibility in this case, and we have higher flexibility when the angles are lower or when *d* is higher.

We suggest approximating non-straight borders as straight borders by using the result that it is not the border but the region in which we can detect the targets is important. If a border cannot be approximated by a single straight border, it can be divided into multiple portions, each of which can be approximated as a linear border. We can perform separate analyses on these portions.

Finally, we introduce the concept of lateral range curves of imperfect sensors. For this case, we study the single searcher problem in which the searcher has an imperfect sensor. We discuss possible approximations to an imperfect sensor and study their performance on a sample imperfect sensor. We find the most appropriate approximation for the lateral range curve we choose. For a different sensor model, we can perform a similar study and select the best approximation.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First, I would like to thank the Turkish Army for giving me the opportunity to earn a master's degree at the Naval Postgraduate School.

I would like to thank my wife, Mehtap, for her support and patience during my education at the Naval Postgraduate School. I also thank my family for their support and trust in me throughout my life.

I would like to thank my advisor, Professor Craparo, for her guidance, support, and knowledge. I would also like to thank Professor Lucas for his work as a second reader.

I would also like to thank Cardy Moten III, a great friend throughout my time here at the Naval Postgraduate School, for letting me know about the High Performance Computing (HPC) network. I would also like to thank Professor Buttrey for helping me to submit my first job to HPC.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Border surveillance is an important problem for most nations that need to detect and intercept intruders trying to trespass a border. These intruders can include terrorists, drug traffickers, smugglers, illegal immigrants, and others who represent a threat to national interests.

Unmanned aerial vehicles (UAVs) allow a modernization and improvement to border surveillance. There are a number of advantages to using UAVs. Many UAVs can be controlled by a single operator, reducing personnel cost. UAVs are fast and thus can patrol large regions; and they have wider regions of visibility than conventional surveillance methods, which increase the probability of detecting intruders.

As noted by Haddah and Gertler (2010), there are two types of UAVs: drones and remotely piloted vehicles (RPVs). Both types are unmanned, but drones are preprogrammed for their flight and mission, whereas RPVs are actively controlled by an operator at the ground station.

RPVs are more appropriate for border surveillance than drones, since the operator can actively control the UAV in a dynamic environment. Considering their long loiter times, it is highly possible that while a UAV is performing a mission, another mission may take precedence and the UAV would need to switch tasks. Also, the operator can maneuver through a rough terrain multiple times to increase detection probability.

B. LITERATURE REVIEW

Since the introduction of UAVs for border surveillance is a relatively new subject, there are not many publications discussing this subject matter. However, Wagner, Mylander, and Sanders (1999) and Washburn (2002) study a similar problem with their main goals being surveillance problems at sea. They both study complex search and detection algorithms. Wagner et al. (1999) use the term “barrier patrol,” and Washburn

(2002) uses the term "patrolling a channel" for the case that is similar to our simple border patrol problem with a single searcher.

Ozcan (2013) examines the effectiveness of UAVs using simulation in a particular scenario and which parameters are important for this particular scenario. She concludes that the UAV's detection and classification performance, as well as target's counter detection capabilities are the most important factors.

Soza & Company (1996), Wagner et al. (1999), Washburn (2002), and Haddah & Gertler (2010) study lateral range curves and possible approximations to them. In addition, Wagner et al. (1999) explains how lateral range curves are determined for a particular sensor. They perform preliminary analyses on the approximations, mentioning the differences between the actual sensor model and its approximations.

C. SCOPE, LIMITATIONS, AND ASSUMPTIONS

This thesis gives guidance to UAV mission planners on how to optimally employ their UAVs while conducting surveillance on a border. The research question we answer is if a fixed number of heterogeneous UAVs are available, how should they be employed in order to maximize the probability of detecting intruders?

Throughout the thesis, we refer to UAVs as searchers and intruders as targets. We do not define specific units of measure; rather, we use generic distance units, speed units, etc., so that our results apply to any platform. Thus, our results provide guidelines for any moving searcher.

We use MATLAB 2012b in all our computations.

We study the problem of monitoring a straight line border over a flat terrain without any line-of-sight issues. In other words, the detection probability depends only on the range of the target. The target tries to pass through the border. We assume that the target cannot see the searcher and moves at a constant speed without changing direction.

For simplicity, we assume that the searcher moves in a straight line and can change direction instantaneously. We generally assume the searcher carries a "cookie-cutter" sensor, although we also analyze imperfect sensors.

Our measure of performance is the probability of detecting a single target.

D. CONTRIBUTIONS AND OUTLINE

In Chapter II, we study the simple border patrol problem by analytical and Monte Carlo simulation methods. This problem is also referred to as the barrier patrol problem by Wagner et al. (1999) and patrolling a channel by Washburn (2002). We improve the detection probabilities computed by Wagner et al. (1999) and Washburn (2002). We also study the multiple-searcher problem.

In Chapter III, we study the optimal allocation problem for multiple searchers. We study the change in optimal allocation depending on the differences in the searchers' characteristics. We also study the optimal allocation when the performance of the searchers' sensors degrades with increasing searcher speed. We calculate both the allocation and speeds required to achieve the maximum detection probability.

In Chapter IV, we study the border patrol problem by adding some geometric considerations. We observe that atypical border geometries do not change the problem considerably, unless the searcher leaves the region in which the objects detected can be classified as targets.

In Chapter V, we introduce imperfect sensors and pick a sample sensor model to study. We perform analysis on approximating this sensor model to compare possible approximations.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BORDER PATROL

This chapter analyzes the border patrol problem. In the border patrol problem, we consider a scenario in which an intruder tries to pass through a border bounded by two barriers that are perpendicular to the border. Note that these barriers may represent actual (physical) barriers or imaginary barriers denoting a region of interest. We have one or more searchers available and would like to employ them in such a way as to maximize our likelihood of detecting the target. We assume that the target moves with constant speed perpendicular to and towards the border, while the searcher patrols the border by moving back and forth at constant speed.

The searcher has a sensor with a finite detection radius, meaning that, if the distance between the target and the searcher is greater than the detection radius, detection will not occur. In this chapter we only consider a "cookie-cutter" sensor, meaning that if the distance between the target and the searcher is less than the detection radius, detection occurs with probability 1. We study non-cookie-cutter sensors in Chapter V.

A. BORDER PATROL WITH SINGLE SEARCHER

In order to obtain useful insights for the general search problem, we start by considering the simplest case: the border patrol problem with one target and one searcher. Figure 1 illustrates our problem setup, which uses the following notation and assumptions:

1. The length of the border is L units. The searcher moves back and forth along the border with constant speed v. When the searcher is at a distance of R units away from either edge of the border, it turns and moves in the opposite direction.
2. The target moves with constant speed u towards the border with its direction of movement perpendicular to the border.
3. If the distance between the target and searcher is less than or equal to R, the searcher detects the target.

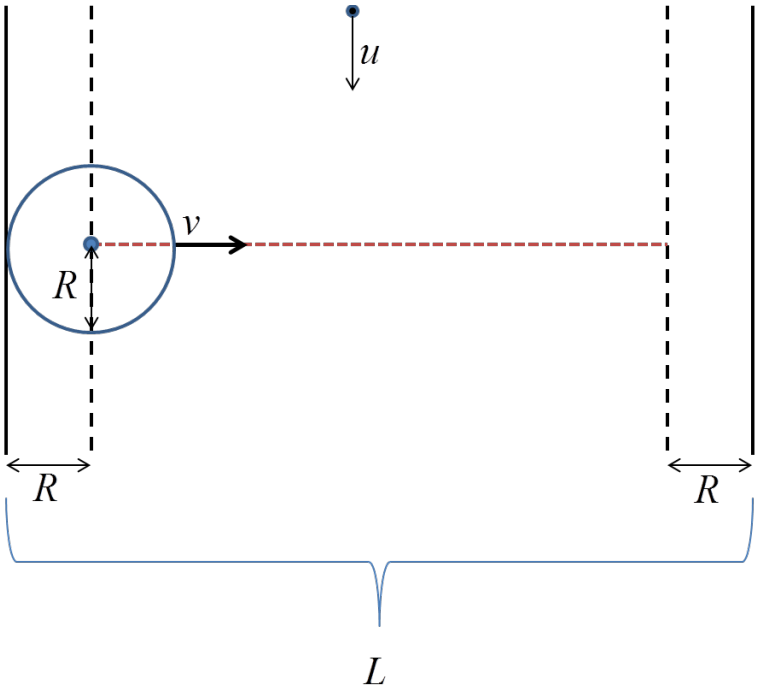


Figure 1. Simple border patrol.

We now analyze the basic border patrol problem both analytically and by Monte Carlo simulation. We compare our results from both methods in order to verify our Monte Carlo simulation, which will form a basis for the later chapters.

1. Analytical Model

In order to calculate the probability of detection we use "target-stationary geometry," which means that we use a coordinate system that moves with the target (Eagle, 2013) rather than a stationary coordinate system. To use target-stationary geometry, we simply add a vector $-u$ to every speed vector in our problem. That is, we add a vector with the same magnitude as the target's speed vector, but in the opposite direction. After performing the reference geometry transformation, the transformed speed of the target ($\tilde{u}$) is 0, and the transformed speed of the searcher ($\tilde{v}$) changes depending on the direction of the searcher's movement. In terms of our problem setup shown in Figure 1, if the searcher is moving to the right, its corresponding speed vector will be the vector shown in Figure 2(a). Likewise, it will be the vector shown in Figure 2(b) if the searcher is moving to the left.

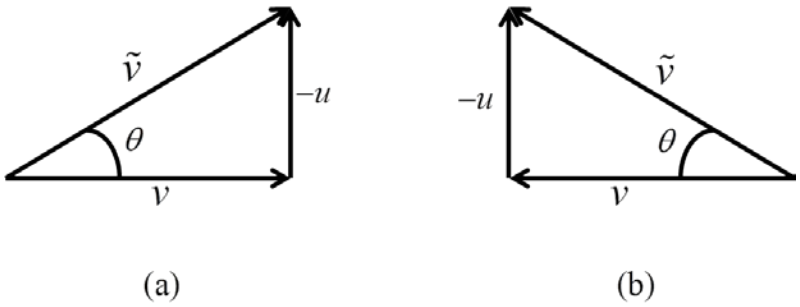


Figure 2. Transformed speed of the searcher in target-stationary geometry.

Following this coordinate transformation, the border patrol problem shown in Figure 1 can be visualized as shown in Figure 3(a). In this figure, the searcher follows the green dashed-dotted lines according to the speed vectors shown in Figure 2, and its detection radius is indicated by solid black lines. Two targets, depicted as red dots, are stationary. The searcher moves in the infinitely long region bounded by the two barriers. Thus, when using target-stationary geometry, the border patrol problem is transformed into a channel search problem in which the searcher looks for stationary targets in the infinitely long channel bounded by the barriers.

Figure 3(b) shows the detection region in the target-stationary case. If a target is in the shaded region, it is detected. Likewise, if it is not in the shaded region, it is not detected. In this example, target 2 from Figure 3(a) is detected and target 1 is not.

Assuming a uniform target density, the probability of detection can be calculated as the ratio of the shaded area in the infinitely long channel to the area of the channel itself. As Figure 3(b) indicates, the shaded area follows a consistent pattern. In particular, between each of the searcher's turning points, the areas of the shaded regions are equal to each other. Moreover, the vertical distance between any two consecutive turning points is the same. Thus, we can calculate the detection probability by considering a region like that shown in Figure 4(a), which is simply the region between two consecutive turning points. We can find the probability of detection by computing the ratio of the shaded area to the area of the blue rectangle in Figure 4(a).

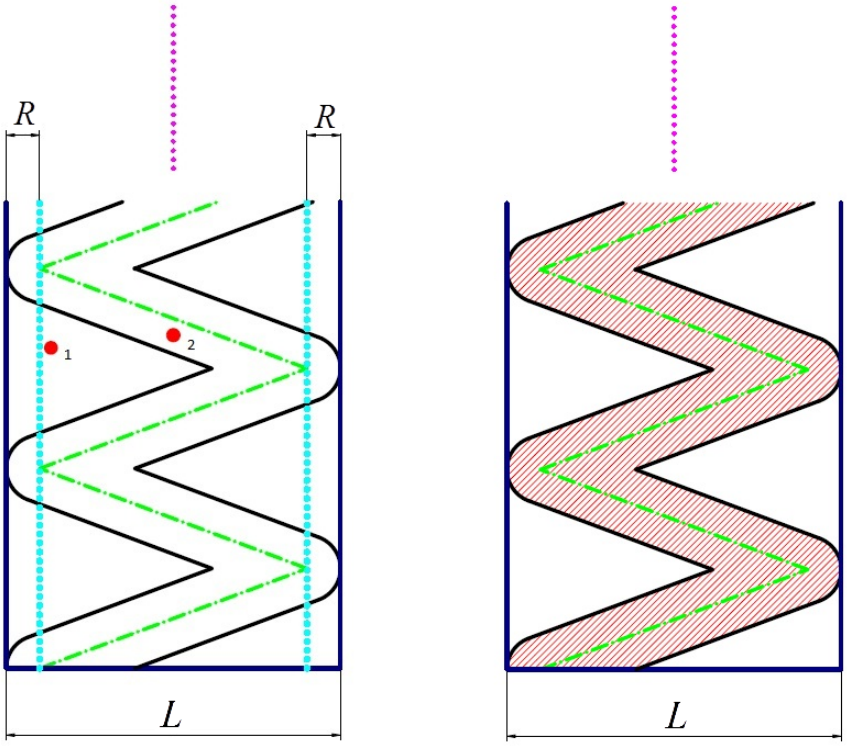


Figure 3. Border patrol in target-stationary geometry.

We calculate the area of the shaded region in Figure 4(a) by dividing it into separate regions as shown in Figure 4(b). The area of the red shaded region in Figure 4(b) can be found by calculating the areas of the two wedge-shaped regions on the ends of the region and the inner rectangle (shaded with both red and blue). Adding these two areas and subtracting areas of the blue triangular regions give us the area of the red shaded region.

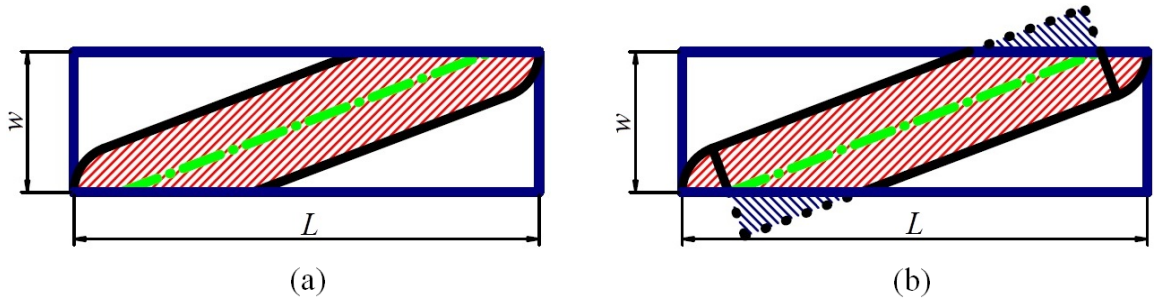
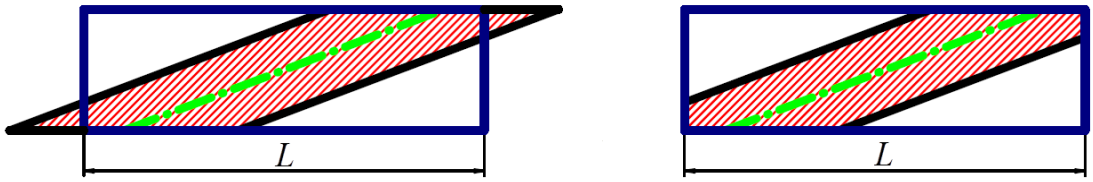


Figure 4. Area of coverage between two turning points in target-stationary geometry.

Washburn (2002) determined an upper bound on this area by calculating the area shown in Figure 5(a). Wagner et al. (1999) arrived at a different approximation of this area by removing the regions lying outside the rectangle, as shown in Figure 5(b).



(a) (b)

Figure 5. Approximations of the area to be calculated.

Using his approximation, Washburn (2002) obtains the following upper bound on the detection probability:

$$p_d \le \min\left\{1, \frac{2R\sqrt{v^2+u^2}}{Lu}\right\}. \tag{1}$$

Likewise, Wagner et al. (1999) approximate the detection probability as:

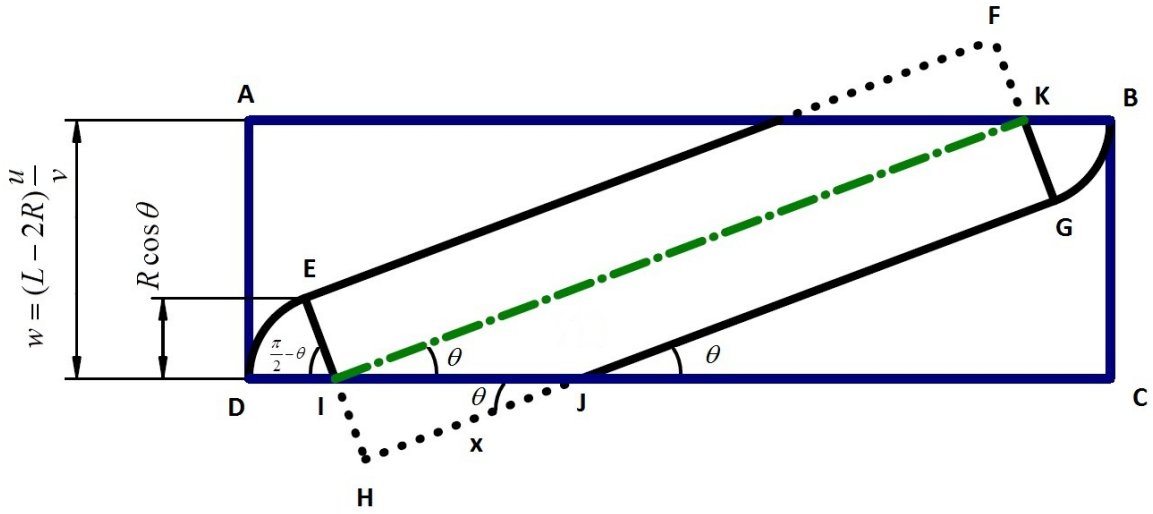
$$P_d = \begin{cases} 1-\left(\frac{L}{R}-\sqrt{\left(\frac{v}{u}\right)^2+1}-1\right)^2 \frac{R^2}{L(L-2R)} & \text{if} \quad R^2 \le \sqrt{L(L-2R)} \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

By calculating the area of the shaded region exactly, we are able to calculate the detection probability without any approximations. The width w of the rectangles in Figure 4 is the vertical distance the searcher travels between turns in target-stationary geometry, or, equivalently, the distance the target travels between the searcher's turns in the original coordinate system. Hence;

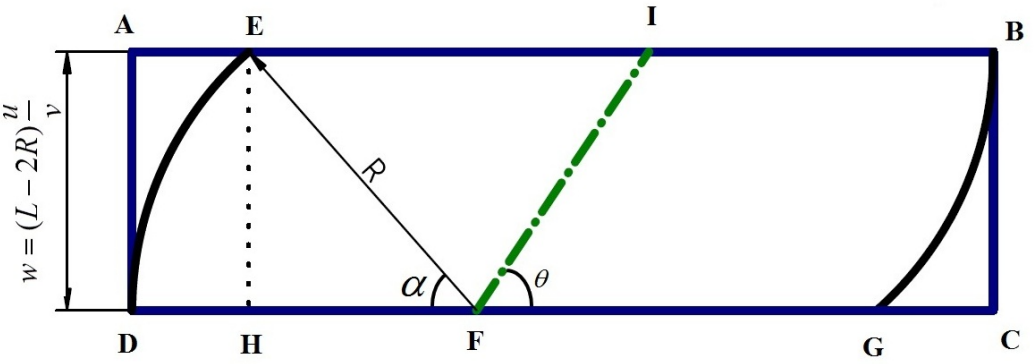
$$w = u\frac{L-2R}{v} = (L-2R)\frac{u}{v}. \tag{3}$$

Depending on the width w, we may have different shapes of the areas to be calculated. Figure 6(a) shows the geometry for large w, i.e., $w > R\cos(\theta)$, where θ is the

angle whose tangent is the ratio of the target's speed to the searcher's speed (Figure 2). Figure 6(b) shows the geometry for small w, i.e., $w \leq R\cos(\theta)$. For this case, we introduce angle α, which is the arcsine of the ratio of the width of the rectangle to the detection radius of the searcher; that is, $\alpha = arcsin(w/R)$.



(a)



(b)

Figure 6. Geometry used in computations.

For large *w*, we can find the area of rectangle EFGH in Figure 6(a) by multiplying its width (EH) by its length (EF, which equals IK). Its width is 2*R*, and its length is $\sqrt{w^2+(L-2R)^2}$; so its area is $2R\sqrt{w^2+(L-2R)^2}$.

The area of the two wedge-shaped regions is $2\left(\pi R^2 \frac{\frac{\pi}{2}-\theta}{2\pi}\right) = R^2\left(\frac{\pi}{2}-\theta\right)$.

The area of the triangular areas (e.g., triangle HIJ in Figure 6(a)) is $2\left(\frac{xR}{2}\right) = xR = R\cot(\theta)R = R^2\cot\theta$.

By combining these areas, we compute the probability of detection as

$$\frac{2R\sqrt{w^2+(L-2R)^2} - R^2\cot\theta + R^2\left(\frac{\pi}{2}-\theta\right)}{wL}. \qquad (4)$$

For the case when $w \leq R\cos(\theta)$, it is simplest to compute the areas ADE and BCG (which are equal by symmetry) and subtract these areas from the area of the rectangle ABCD. The area ADE can be computed by calculating the area of the trapezoid AEFD and subtracting the wedge-shaped area DEF from it.

The area of trapezoid AEFD is $\frac{R+(R-R\cos\alpha)}{2}w = \frac{2R-R\cos\alpha}{2}w$.

The area of semicircle DEF is $\pi R^2\frac{\alpha}{2\pi} = R^2\frac{\alpha}{2}$.

By using these, we calculate the area of region ADE as $\frac{2R-R\cos\alpha}{2}w - R^2\frac{\alpha}{2}$.

By combining these areas, we compute the probability of detection as

$$\frac{wL-2\left(\frac{2R-R\cos\alpha}{2}w - R^2\frac{\alpha}{2}\right)}{wL} = 1-\frac{(2R-R\cos\alpha)w - R^2\alpha}{wL}. \qquad (5)$$

Finally, in order to obtain the probability of detection for the simple border patrol problem, we combine Equations (4) and (5). Furthermore, we express each of the terms w, α, and θ as a function of the border length L, detection radius R, target speed u, and searcher speed v as shown in Equation (6).

$$P_d = \begin{cases} \dfrac{2R}{L}\sqrt{\left(\dfrac{v}{u}\right)^2}+1+\dfrac{R^2 v\left(\dfrac{\pi}{2}-\arctan\left(\dfrac{u}{v}\right)-\dfrac{v}{u}\right)}{(L-2\mathrm{R})uL} & \text{if } \dfrac{Rv^2}{(L-2R)u\sqrt{u^2+v^2}}<1 \\ 1+\dfrac{R^2 v \arcsin\left(\dfrac{(L-2R)u}{Rv}\right)}{(L-2R)uL}-\dfrac{2R}{L}+\dfrac{\sqrt{R^2v^2-(L-2R)^2u^2}}{Lv} & \text{otherwise.} \end{cases} \quad (6)$$

Figure 7 compares the detection probabilities given by Equations (1), (2), and (6) for various searcher speeds. In this figure, the border length L is 50 distance units, the detection radius R is 6 distance units, and the speed of the target u is 5 speed units (distance units/time unit). We observe that all three formulas are nearly the same for low searcher speeds. The upper bound Washburn (2002) obtained starts to differ slightly when the searcher speed is approximately twice the speed of the target. The formula Wagner et al. (1999) obtained starts to differ when the searcher's speed exceeds four times the target's speed.

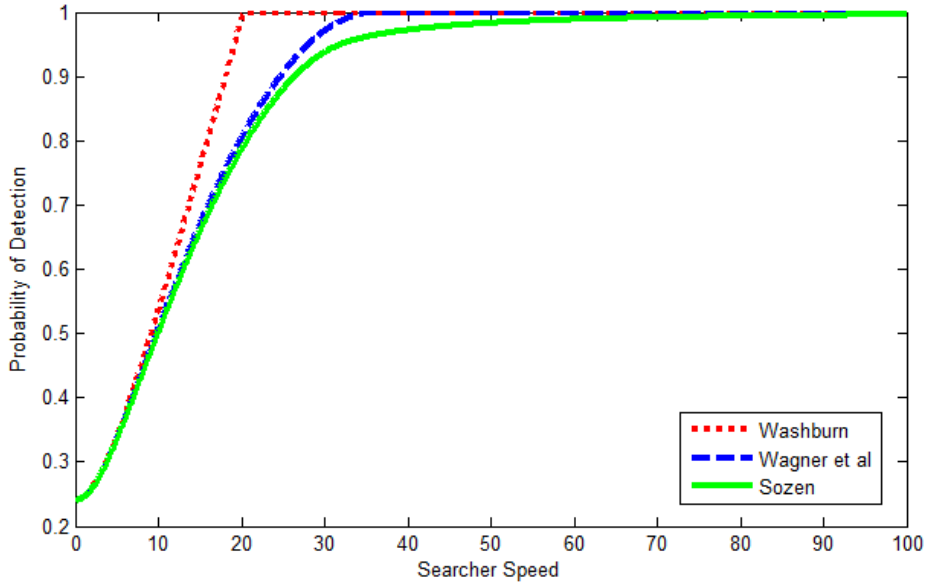


Figure 7. Comparison of formulas.

2. Monte Carlo Simulation

We now examine the border patrol problem by means of a Monte Carlo simulation model. Our simulation setup is as follows:

1. The searcher's initial position is distance R from the left edge of the border, and its initial direction of movement is to the right.
2. We use a time-step model for our simulation. In the time-step model, we calculate the position of the searcher and the target and make necessary computations to see if the target is detected after each time step increments. In time-step simulations we have some error if detection does not occur at the time steps, but occurs between the time steps. We choose the time step as $\Delta t = \frac{R}{25\sqrt{u^2 + v^2}}$; this time step reduces the error below $6.7\text{x}10^{-3}$ percent. We explain how we choose the time-step and the calculation for the error in Appendix A. We use the same time step throughout the time-step simulations in this thesis.
3. The simulation is run until the searcher makes a full cycle, i.e., comes back to its initial position and initial direction of travel. This simulation end time is referred to as $t_{\max}$.
4. Targets are generated randomly using a two dimensional uniform distribution over a rectangle between the two barriers. Each target's initial vertical position is randomized so as to ensure that it passes through the horizontal axis of the searcher's movement before the end of the simulation. Their initial horizontal axis position is between 0 and L.
5. We replicate the number of targets n times and record the number of targets that are detected by the searcher as k. This approach generates the same results instead of generating one target and running n different simulations, however, the simulation runtime improves considerably. We then use MATLAB's "binofit" function to fit the Monte Carlo simulation's results to a binomial distribution, and we compute the estimated probability of detection along with its 95% confidence interval.

Figure 8 and Figure 9 show the probability of detection versus the speed of the searcher when the border length L is 200 distance units, the detection radius is 6 distance units, and the speed of the target u is 5 speed units. Figure 8 shows the results of the Monte Carlo simulation with 1,000 replications, and Figure 9 shows the results of the Monte Carlo simulation with 1 million replications. In both figures, the blue solid line shows the probability of detection obtained from Equation (6). The black dashed line shows the estimated probability of detection from the Monte Carlo simulation, and the

red dashed lines show the upper and lower 95% confidence interval of the estimated probability of detection.

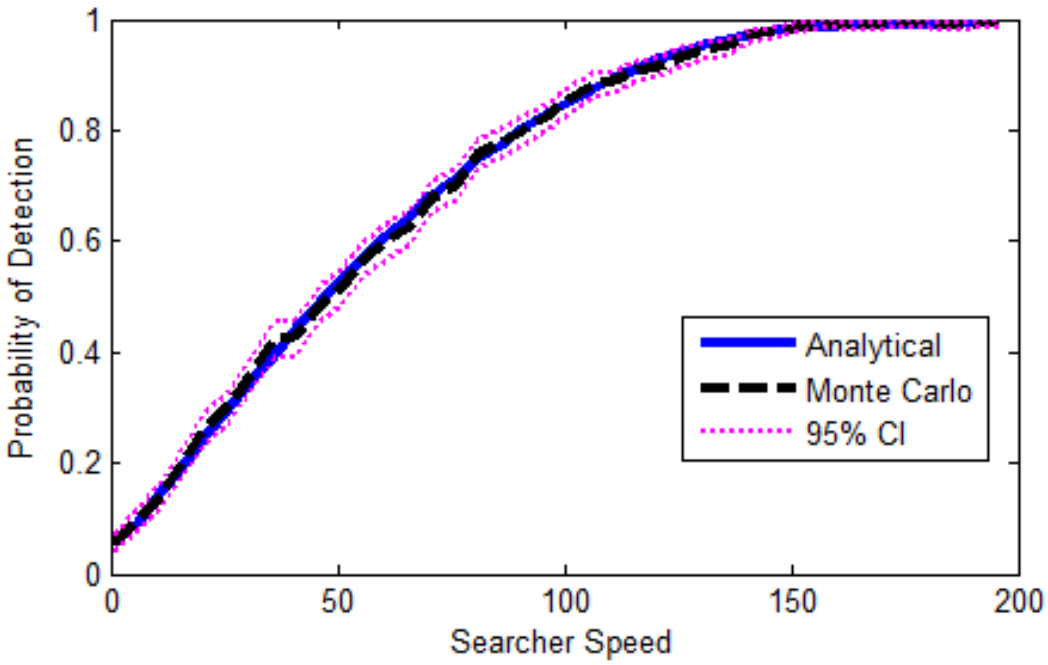


Figure 8. Monte Carlo simulation with 1000 replications.

From Figure 9 we can see that the estimated probability of detection obtained from the Monte Carlo simulation is very close to the probability of detection obtained from Equation (6); hence, we conclude that we have verified the Monte Carlo simulation and checked Equation (6) for the simple border patrol problem. We also show that Equation (6) is more accurate than Washburn's (2002) or Wagner et al.'s (1999). The Monte Carlo simulations in the following sections are all based on this simple case.

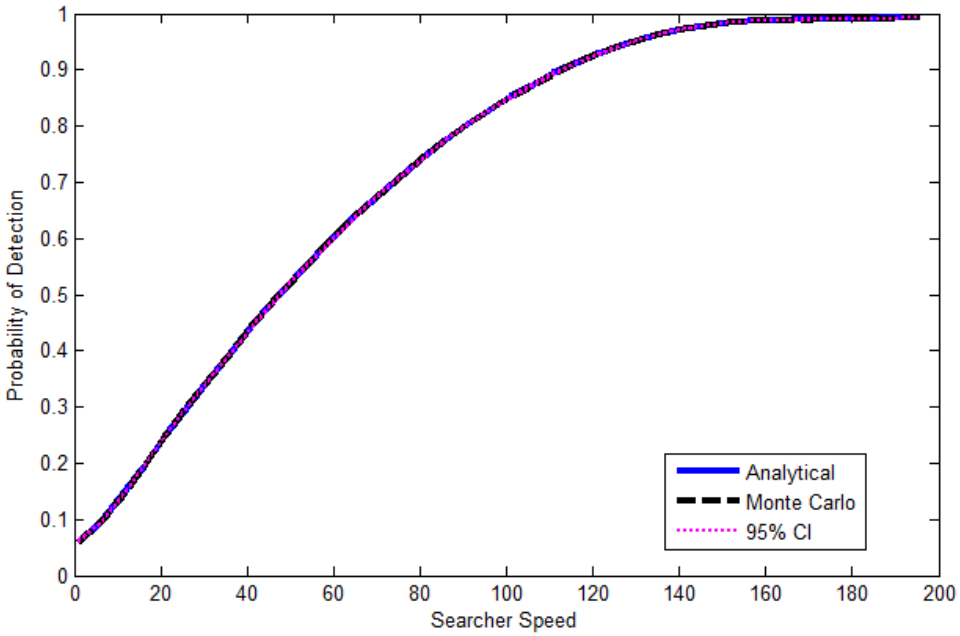


Figure 9. Monte Carlo simulation with 1 million replications.

3. Turning Distance

Recall that we assume when the searcher is *R* distance units away from any edge of the border, it turns in the opposite direction. In this section we study the effect of changing this turning distance.

We consider ten different scenarios defined by the border length. We begin with a border length of 100 units and incrementally increase our border length by 100 units until we reach a length of 1000 units. In doing so, we are able to see the effect of the turning distance for several values of probability of detection.

In all ten scenarios, we fixed the detection radius *R* at 6 units, the target's speed *u* at 5 speed units, and the searcher's speed *v* at 100 speed units. We varied the turning distance from 0 to 12 units in 0.06 unit increments. We performed the analysis by running Monte Carlo simulations with one billion replications for each scenario.

In order to run the simulations with one billion replications, we need to generate two billion random numbers; half of them are used for generating the targets' random vertical position and the remaining half are used for their horizontal position. To ensure

that we did not obtain misleading results from the outcome of the simulations, we need to make sure that these two billion random numbers do not start repeating themselves after a certain point. Although we noted that MATLAB's documentation states the random numbers have an approximate period of 2^{19937}, which is much larger than two billion, we also compared the two billion random numbers and found that among these two billion random numbers, no two of them were equal to each other. We also faced memory errors in MATLAB due to the high dimension of the random numbers generated; however, we solved the memory problem easily by requesting 128 GB of memory from the High Performance Computing (HPC) network at the Naval Postgraduate School for each simulation.

Figure 10 shows the results of the turning point analysis. Each subfigure shows one of the scenarios, with the corresponding barrier length stated on each subfigure. In each subfigure, the horizontal axis shows the turning distance and the vertical axis shows the estimated probability of detection $\widehat{P_d}$ in the blue straight line, along with its 95% confidence interval in red dotted lines.

In the case of low border length (i.e., when $\widehat{P_d}$ is high) in Figure 10(a) through Figure 10(c), the maximum value of $\widehat{P_d}$ occurs when the turning distance is less than the detection radius. This occurs because the searcher covers the area in regions ADE and BCG in Figure 6(b). But in other cases, either the maximum $\widehat{P_d}$ is observed when the turning distance is very close to the detection radius (e.g., Figure 10(d) through Figure 10(h)), or $\widehat{P_d}$ increases with increasing turning distance up to detection radius, and then it stays nearly constant (e.g., Figure 10(i) and Figure 10(j)).

In all ten scenarios, the estimated detection probability $\widehat{P_d}$ is either insensitive to the turning distance, or it is maximized or nearly maximized at the detection radius *R*. Based on these results, we fix the turning distance to *R* for the remainder of our simulations.

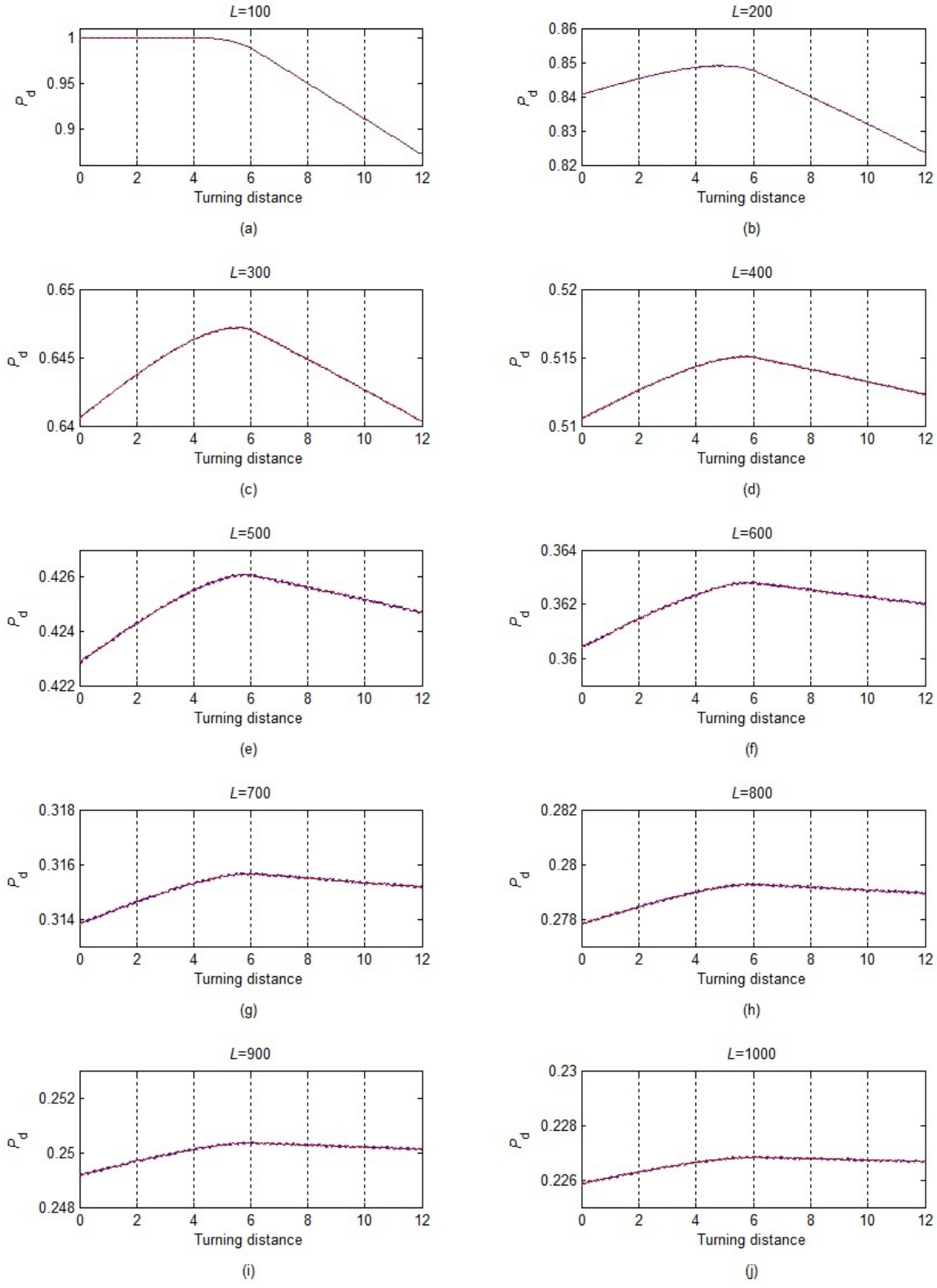


Figure 10. Analysis on turning distance.

B. BORDER PATROL WITH MULTIPLE SEARCHERS

When multiple heterogeneous searchers are available to patrol a border, an operator must decide how best to employ them. For the case of two searchers, the operator may either employ both searchers to patrol the entire border, or the operator may divide the border into two disjoint segments and assign each searcher to patrol one segment. We refer to the first option as the "common path" allocation and the second option as the "disjoint path" allocation.

We examine both the disjoint and common path cases in the following subsections. For simplicity, we consider only two searchers in both cases, but similar logic applies to cases with more than two searchers.

1. Disjoint Path

Figure 11 shows the border patrol problem with two searchers and one target. The searchers allocate the border into two disjoint regions with lengths L_1 and $L_2 = L - L_1$.

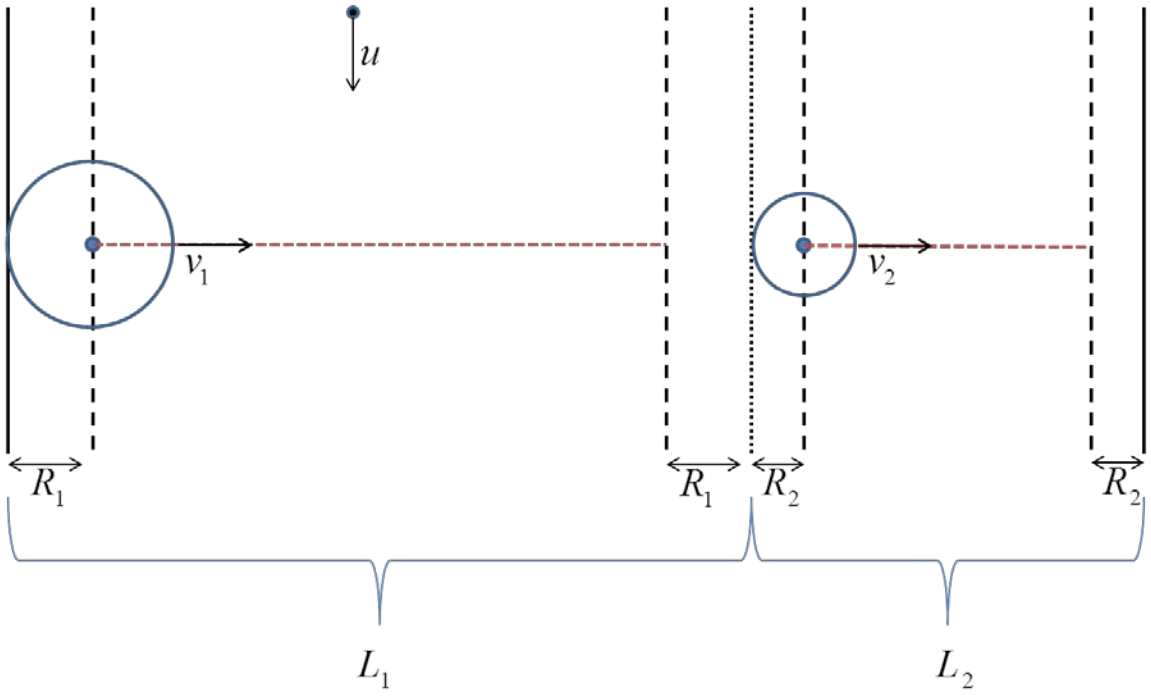


Figure 11. Disjoint path.

a. Analytical Solution

We can divide the disjoint path problem in Figure 11 into two separate border patrol problems of the type considered in Section A. Based on this simplification, we can then compute the probability of detecting the single searcher for this disjoint path case by the law of total probability:

$$P_d = P_{ds1}P_{tr1} + P_{ds2}P_{tr2} \tag{7}$$

where P_{dsi} is the conditional probability that searcher i detects the target, given the target is in its region, and P_{tri} is the probability that the target is in the region of searcher i.

We can compute P_{dsi} by substituting the appropriate values for searcher i into Equation (6):

$$P_{dsi} = P_d(L_i, R_i, u, v_i). \tag{8}$$

Since we assume that the horizontal position of the target is uniformly distributed along the border, we have $P_{tri} = \frac{L_i}{L}$. Thus, we have

$$P_d = P_{ds1}\frac{L_1}{L} + P_{ds2}\frac{L_2}{L} = P_d(L_1, R_1, u, v_1)\frac{L_1}{L} + P_d(L_2, R_2, u, v_2)\frac{L_2}{L}. \tag{9}$$

In general, with n searchers and one target, we have

$$P_d = \sum_{i=1}^{n} P_{dsi}\frac{L_i}{L} = \frac{1}{L}\sum_{i=1}^{n} P_d(L_i, R_i, u, v_i)L_i. \tag{10}$$

b. Monte Carlo Simulation

We also evaluate the disjoint path problem by means of a Monte Carlo simulation. The following setup applies to our simulation:

1. The searchers' initial positions depend on their detection radii and their allocated regions. The initial position of the i^{th} searcher is distance R_i away from the edge of its allocated region. Its direction of movement is towards the opposite edge, and its turning distance is R_i distance units from the end of its allocated region. If $R_i \geq \frac{L_i}{2}$ for any searcher, the searcher remains stationary at the midpoint of its allocated region.
2. Recall that our simulations in Section A.2 were run until the searcher arrived at its initial starting position. Depending on the allocated regions and characteristics of the searchers, in the disjoint path case such a policy may result in a very long runtime. Thus, we limited our simulation end time to 25 times the maximum time required for any searcher to make a full cycle in its allocated region. Appendix B explains the process behind this choice of multiplier.
3. Targets are generated and estimated probabilities of detection are calculated as in Section A.2.

Figure 12 shows the probability of detection P_d as calculated analytically using Equation (9) and the estimated value for the probability of detection $\widehat{P_d}$, determined via Monte Carlo simulation with one million replications, as a function of the ratio of the border length allocated to the first searcher, L_1, to the total border length L. This ratio is varied from 0 to 1 with 0.01 increments. In this specific case L is 200 distance units, detection radii R_1 and R_2 are 12 and 6 distance units, respectively; the target speed u is 5 speed units, and the speeds of the searchers, v_1 and v_2, are each 20 speed units.

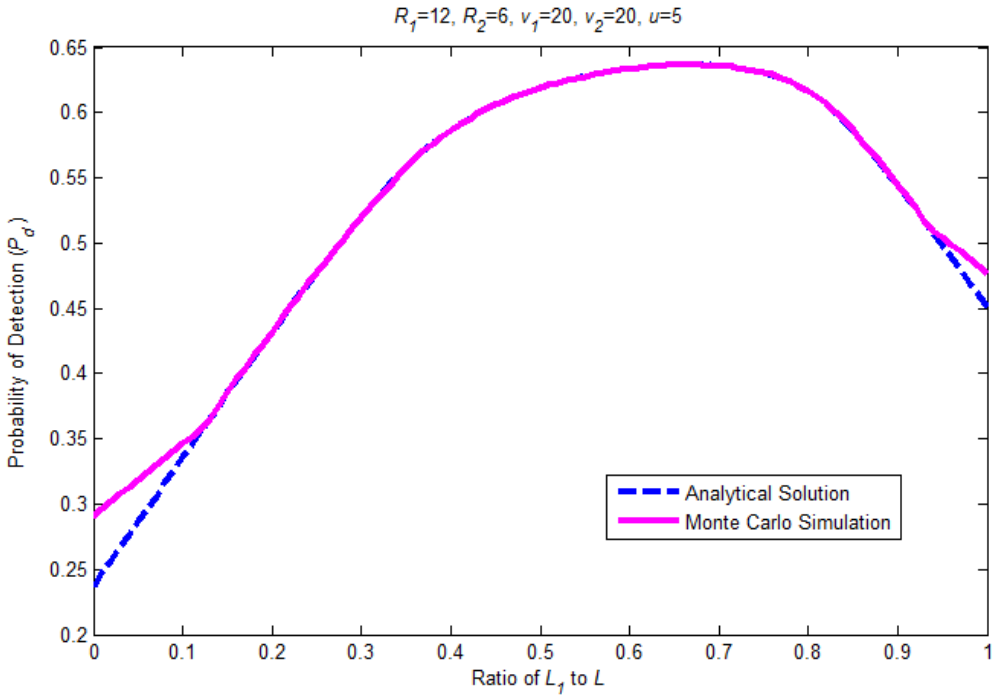


Figure 12. Disjoint path.

Comparing the analytical solution to the Monte Carlo simulation, we see that they are nearly the same except at very high or low values of L_1/L. When L_1/L is very large or very small, this means that the region allocated to one of the searchers is very small. Figure 13 shows the case when the region allocated to the first searcher is very small. In this case we have $R_1 > \frac{L_1}{2}$, so the first searcher remains in the middle of its allocated region and covers some portion of the area allocated to the second searcher. This extra region is shaded with red stripes in Figure 13 and is not accounted for in the analytical solution. Because of this extra region, the Monte Carlo simulations produce higher estimates than the analytical solution when the allocation to any searcher is less than two times the detection radius of that searcher. In this case, the Monte Carlo simulation provides a better estimate since it takes into account the region that the analytical solution does not.

In Figure 12 we see that the analytical solution and the Monte Carlo simulation results differ when L_1/L is less than 0.12. Note that L_1=24 at this point, and that $2R_1$=24. The results also differ when L_1/L is greater than 0.94; L_2=12 at this point, and $2R_2$=12.

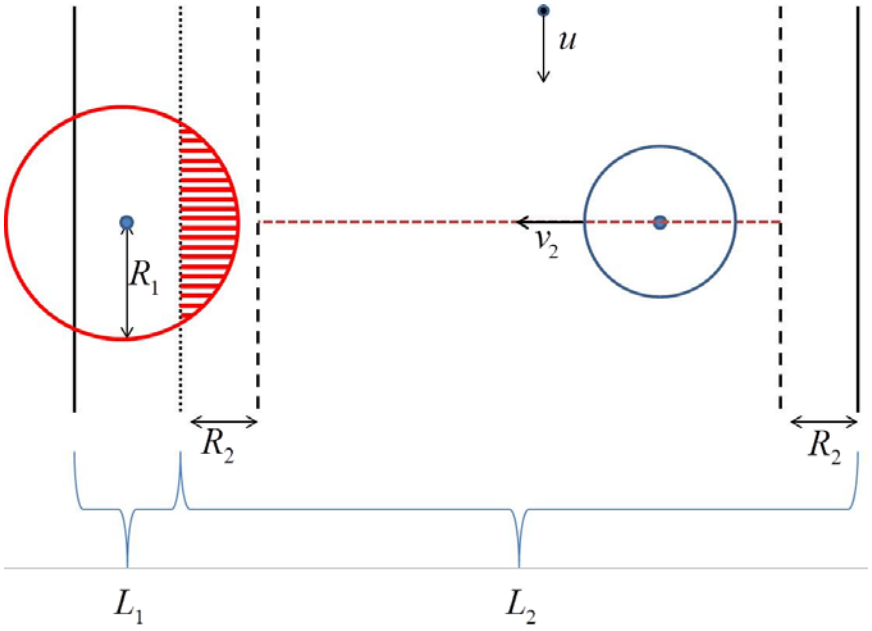


Figure 13. Disjoint path extreme case.

2. Common Path

Figure 14 shows the border patrol problem with two searchers sharing the same path and attempting to detect a single target. Although the searchers share the same path, their turning points may differ due to their differing detection radii. For clarity, we introduced a slight vertical displacement in the paths that the searchers follow, although in reality we assume that there is no vertical displacement. Each searcher moves over the border.

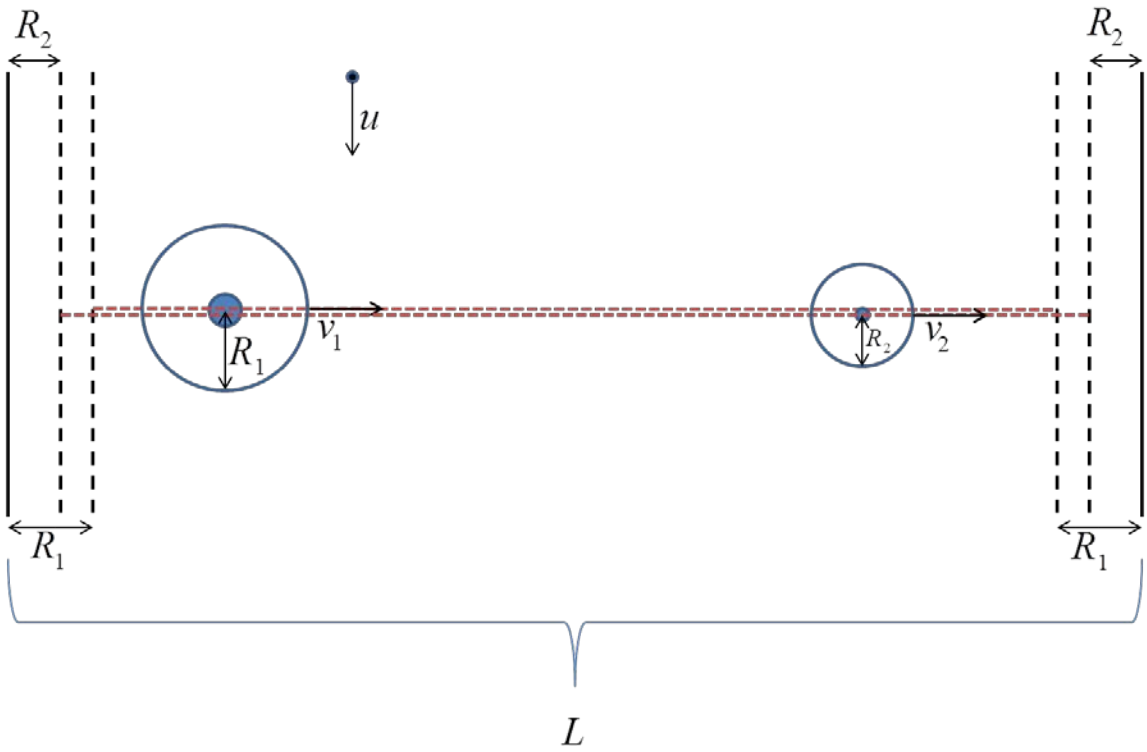


Figure 14. Common path.

We did not perform any analytical computations on the common path problem; rather, we performed Monte Carlo simulations based on the Monte Carlo simulation for the disjoint path case with some minor changes. The only differences are the searchers' initial positions and their turning points.

The initial starting point of the searchers could be chosen to be the same point, i.e., they could initially start their movement from exactly the same location. However, if their detection radii and speeds were identical, then they would move together as one searcher (assuming a "cookie cutter" sensor model). For this reason, we choose to initialize searcher *i*'s location as $\frac{L(i-1)}{n}+R_i$, where *n* is the number of searchers. The searchers' initial direction of movement is towards the right. As before, each searcher's turning distance is simply its detection radius. Figure 15 shows our initial setup for the common path simulations; dashed lines show the turning points of the searchers.

The simulation is run until all searchers come back to their initial starting point. In order to reduce the runtime, the simulation end time is limited to 100 times the maximum time it takes for any searcher to make a full cycle.

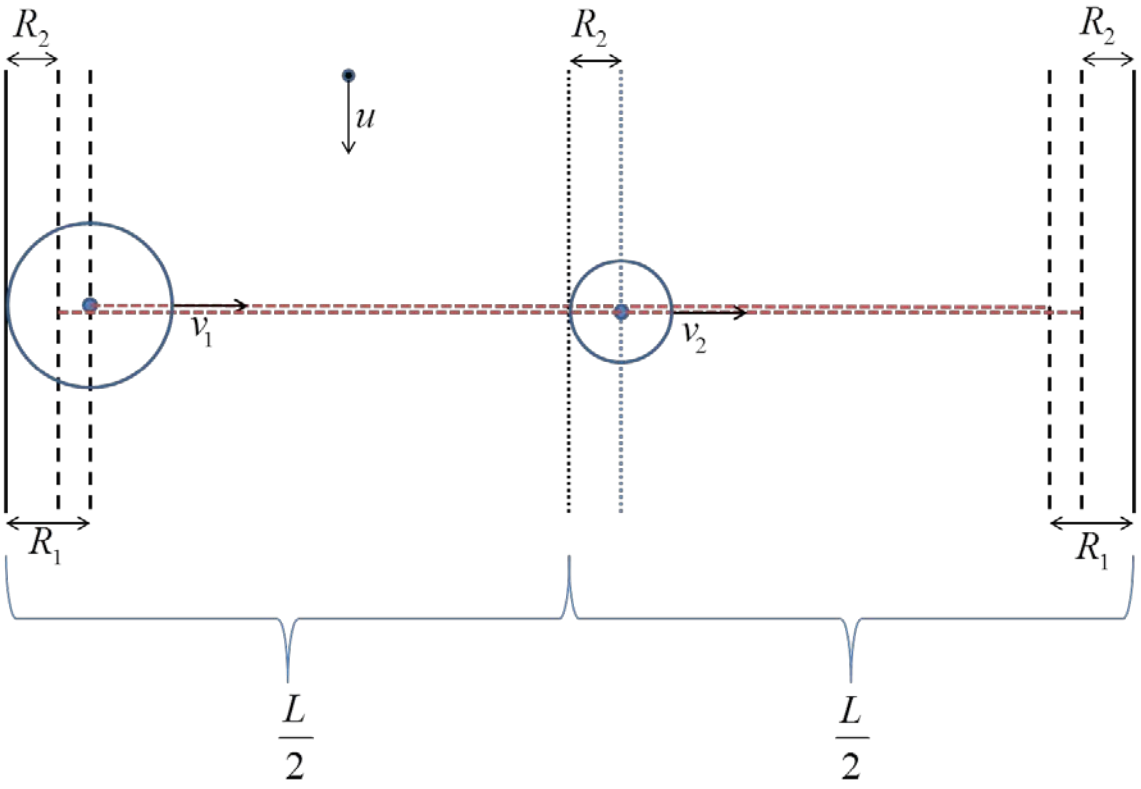


Figure 15. Searchers' initial positions and movement in the common path simulations.

We perform one million Monte Carlo simulations for the common path case with the same searcher and target variable settings we considered in the disjoint path problem (Figure 12). We obtain an estimate and 95% confidence interval for the probability of detection, shown in Figure 16, along with the results previously obtained for the disjoint path case. (Due to the large number of replications, the confidence interval is difficult to detect in the figure.)

Since there is no allocation of borders in the common path problem, the estimated probability of detection is constant. It is plotted over the results from Figure 12 in order to be able to make a comparison between the disjoint and common path problems.

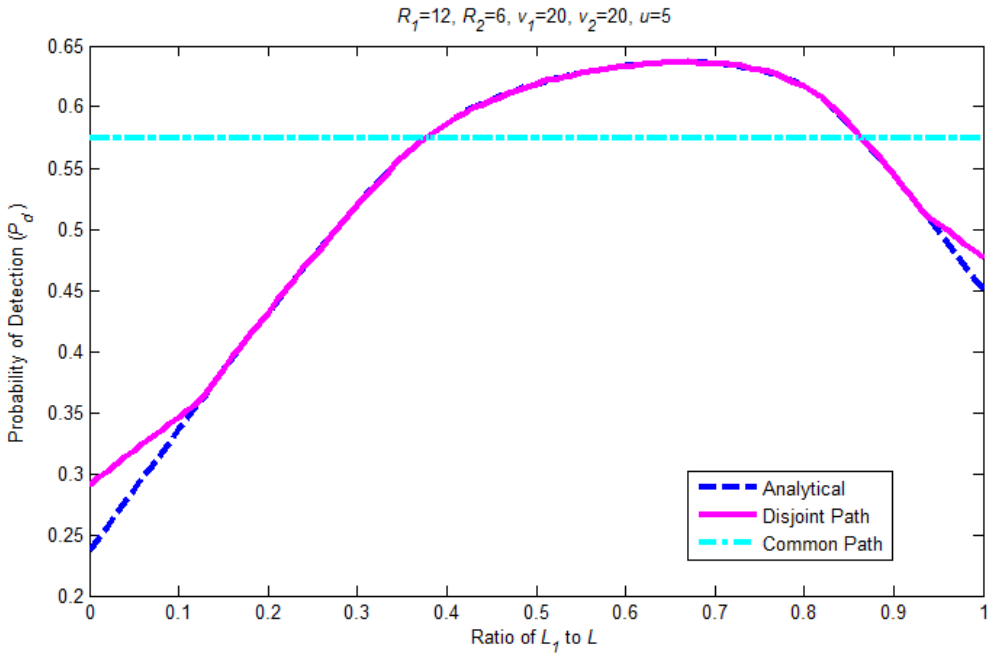


Figure 16. Common path Monte Carlo simulation results.

Figure 16 shows that the common path detection probability is higher than the disjoint path probability for nearly half of the L_1/L range. This means that if we do not allocate the border to the two searchers properly for the disjoint path case, we may end up with a worse probability of detection than we would obtain without allocating the region at all and simply setting the searchers free to search the entire border. However, by intelligently allocating the border, we are able to obtain a higher detection probability than is possible with the common path model. Thus, we now consider the problem of allocating the border to the searchers optimally so as to maximize detection probability in Chapter III.

THIS PAGE INTENTIONALLY LEFT BLANK

III. OPTIMAL ALLOCATION OF REGIONS

In this chapter we study the problem of optimally allocating a border among two searchers in order to maximize detection probability. We utilize both analytical and Monte Carlo simulation methods. Although we consider only two searchers, our approach can be extended to scenarios with $n > 2$ searchers.

A. ANALYTICAL METHOD

In order to find the optimal allocation of regions for two searchers we can use the function for the probability of detection shown in Equation (9).

In the optimal allocation problem, our goal is to determine L_1 and L_2 such that the overall probability of detection is maximized. That is, given values for L, R_1, R_2, u, v_1, and v_2, we wish to determine L_1 and L_2 such that $L_2 = L - L_1$ and the probability of detection is maximized:

$$\begin{aligned} \max_{L1} \quad & P_d(L_1, R_1, u, v_1)\frac{L_1}{L} + P_d(\mathrm{L} - L_1, R_2, u, v_2)\frac{L - L_1}{L} \\ st \quad & 0 \le L_1 \le L \end{aligned} \tag{11}$$

This is a constrained optimization problem with one decision variable and a convex feasible region. Thus, if the objective function is concave, we can find a globally optimal solution by finding a stationary point of the objective function, i.e., a value of such L_1 that

$$\frac{\partial\left(P_d(L_1, R_1, u, v_1)\frac{L_1}{L} + P_d(L - L_1, R_2, u, v_2)\frac{L - L_1}{L}\right)}{\partial L_1} = 0. \tag{12}$$

If the solution to Equation (12) is less than 0 or greater than L, the maximum probability of detection occurs either at $L_1 = 0$ or $L_1 = L$.

The concavity of the function in Equation (9) is difficult to evaluate analytically. However, we performed an extensive computational evaluation and were unable to find a counterexample showing that the function in Equation (9) is non-concave. Our computational evaluation is described in Appendix C.

B. MONTE CARLO SIMULATION

For our Monte Carlo simulation analysis, we generated several scenarios by varying the detection radii and speeds of the searchers while fixing the length of the border and the speed of the target to 200 and 5 units, respectively. In each scenario we varied the allocation of border to the first searcher in 1% increments, each with one million replications, in order to observe the change in the probability of detection.

Figure 17 illustrates the outcome of six representative setups. Each subfigure in Figure 17 is generated with the same logic as Figure 16, where the subfigures show the probability of detection versus the fraction of the border allocation to the first searcher. The blue dashed line shows the analytical disjoint path detection probability (Equation (9)), the purple line shows the estimated probability of detection for the disjoint path case obtained from the Monte Carlo simulation, and the light blue line shows the estimated probability of detection for the common path case obtained from the Monte Carlo simulation. We also plotted 95% confidence intervals for the estimated probabilities of the common path and disjoint path cases, but due to the large number of replications, they are difficult to detect in the figures.

In Figure 17(a) we see that in the optimal allocation, the allocation to the first searcher is very low since its detection radius is inferior to that of the second searcher, while their speeds are equal to each other. It is also interesting to note that the optimal detection probability is very close to the common path probability of detection. In this case, the optimal allocation to the second searcher is nearly the whole border. In other words, if we just allocate the whole border to the second searcher and let the first searcher move freely along the same border, we do not lose much in terms of probability of detection compared to the optimal allocation.

In Figure 17(b), the first searcher's capabilities are still inferior to those of the second searcher, but not as much so as in Figure 17(a). Thus, more of the border is allocated to the first searcher in Figure 17(b) than in Figure 17(a). The difference between the common path probability and optimal allocation probability has also increased. Increasing the speed of the first searcher (Figure 17(c)), results in a further increase in the allocation to the first searcher, and the gap between the common path and optimal allocation probability widens.

When we increase the speed of the first searcher such that the properties of both searchers become equal, we expect to have the same border allocation to both searchers since neither of the searchers is superior to the other. We might also expect to see a larger gap between the common path and optimal allocation probabilities compared to the previous cases (Figure 17(a) through Figure 17(c)) since decreasing the superiority of the superior searcher resulted in an increase in the gap in the previous cases. In Figure 17(d) we see that we are right about our first expectation, but not about the second one. This occurs because, in the common path case when the two searchers have the same characteristics, they operate in a cyclical fashion. Specifically, the two searchers cross each other at the same places every time they transit the border. These crossings occur when they are moving in opposite directions (since they have the same speed, neither searcher can pass the other), causing them to have short crossing times. Shorter crossing times means less coincident area covered by the searchers, resulting in a smaller gap between the common path and optimal allocation probabilities.

In Figure 17(e) and Figure 17(f) the second searcher is inferior, but not as much as the cases in Figure 17(a) though Figure 17(c) when the first searcher was inferior, which causes a considerable amount of the border to be allocated to the second searcher in the optimal allocation. The differences between the common path and optimal allocation probabilities are higher than the other four scenarios.

In general, we observe that if one of the searchers is superior to the other, the allocation of the border to the inferior searcher is low, and the difference between the common path and optimum allocation probabilities is low. Decreasing the difference in the capabilities of the searchers results in a higher allocation to the inferior searcher and

can widen the gap between the common path and optimal allocation probabilities. However, due to the perfect coordination between the searchers when the searchers have the same characteristics, the gap between the common path and optimal allocation probabilities can be low.

We also observe that if we do not allocate the border properly to the searchers, we may lose a good proportion of the probability of detection.

Based on our results, we conclude that, in order to maximize the detection probability, we should choose the disjoint path approach, rather than the common path and allocate the border to the searchers optimally.

However, if a user cannot determine the optimal allocation, he or she can choose the common path approach, since poorly allocating the border in disjoint path may result in a worse detection probability than choosing the common path. If this option is selected, the UAVs should be employed in such a way as to reduce the time when they pass or cross each other to further increase the detection probability.

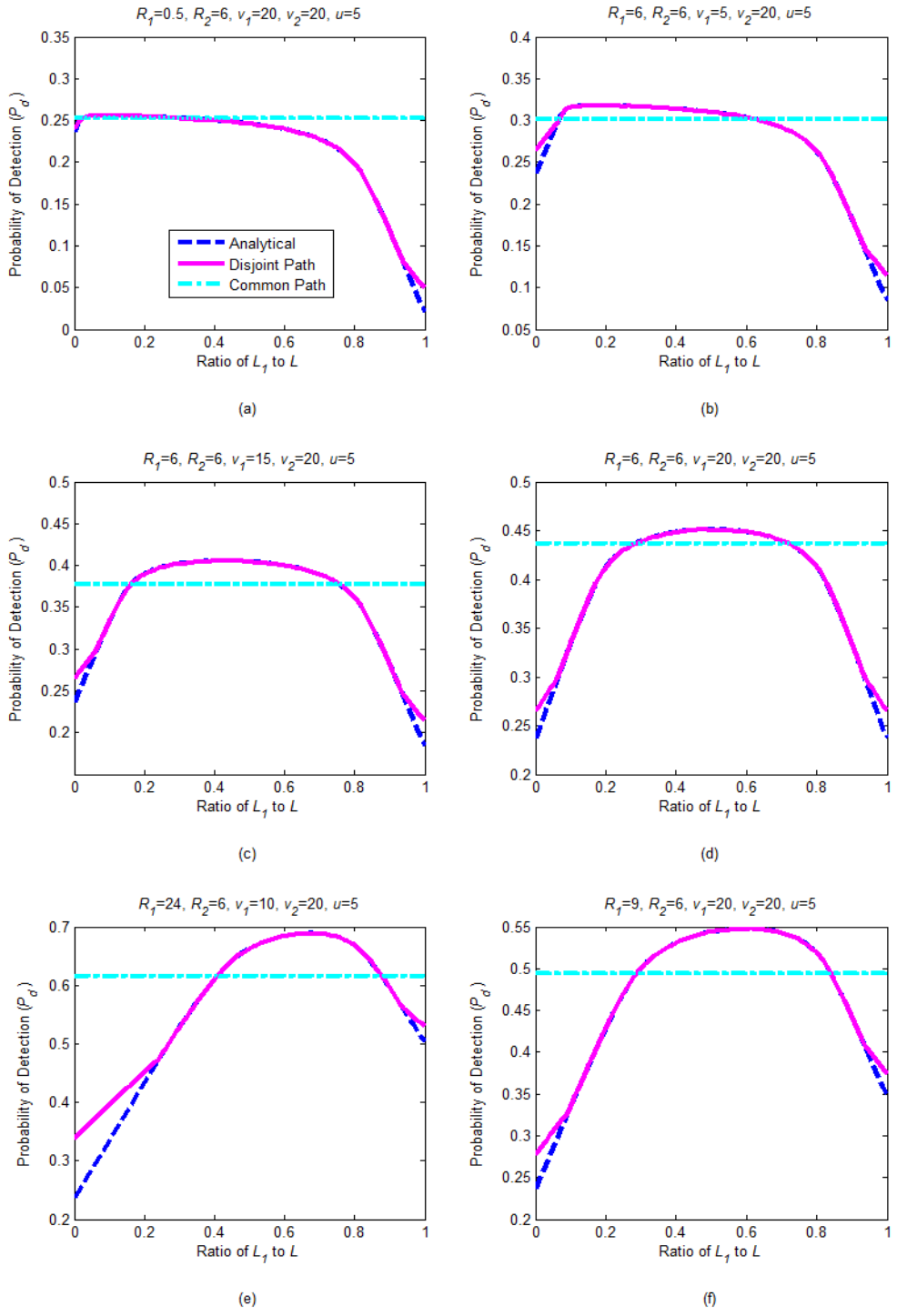


Figure 17. Monte Carlo simulation on several cases.

C. ANALYSIS ON DETECTION RADIUS WHEN IT VARIES AS A FUNCTION OF SPEED

In practice, it is likely that the detection capability of a searcher depends on its speed. For example, if the searcher is a UAV and if its sensor is a camera, the quality of the video would be reduced if the UAV were to travel at a faster speed. In this section, we model this degradation in sensor quality as a reduction in detection radius. We will study other sensor models in Chapter V. We now examine the effect of the detection radius, which is a function of each searcher's speed, on the optimal allocation and probability of detection.

1. Effect on the Single Searcher Problem

We first study the effect of the detection radius being a function of searcher speed on the simple border patrol problem. In order to make a comparison with the constant detection radius case (Figure 9), we utilize the same parameters as before, with the exception of the detection radius. We model the detection radius, as a monotonically decreasing function of the searcher speed; in particular, we arbitrarily choose $R = 6e^{-v/60}$. The same analysis can also be performed for a different sensor model. We choose an arbitrary function because our goal is simply to gain insight about the effect of the detection radius depending on the searcher speed.

Figure 18 shows the detection radius as a function of speed (left) and the corresponding probability of detection (right). The probability of detection first increases with increasing speed; then it starts to decrease after reaching the maximum.

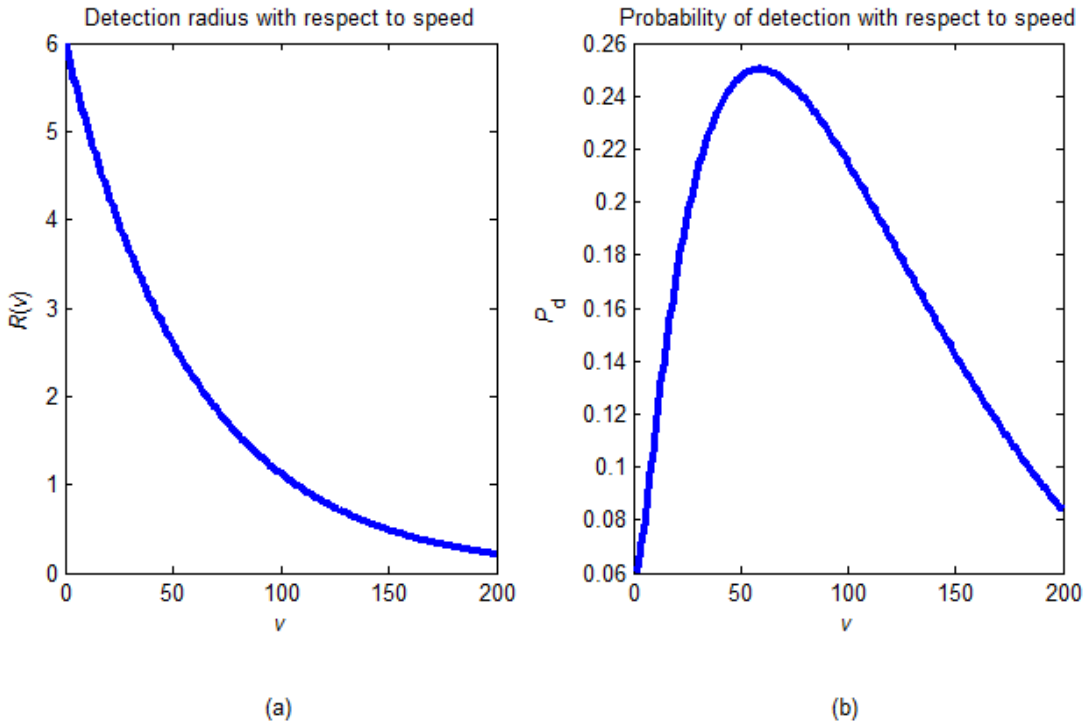


Figure 18. Detection radius as a function of speed.

In general, the probability of detection increases with both increasing speed and increasing detection radius. However, in this case the increase in speed causes a decrease in detection radius. For lower speeds, the probability of detection increases with increasing searcher speed. After the maximum probability of detection point, detection radius dominates the effect of higher speeds.

In all instances we considered, the detection probability curve took on the form shown in Figure 18(b). Thus, we can find the maximum probability of detection by modifying Equation (6) to accommodate detection radius as a function of searcher speed, then taking derivative of the detection probability function with respect to the speed of the searcher, and solving for the searcher speed that causes the derivative to equal zero.

$$\frac{\partial P_d(L, R(v), u, v)}{\partial v} = 0 \tag{13}$$

Solving Equation (13) using MATLAB's symbolic toolbox with the values stated in this section resulted in a speed of 59.06 speed units. This value matches well with the peak of the speed versus probability of detection curve in Figure 18(b).

If the function $P_d(L,R(v),u,v)$ is concave, solving Equation (13) would give us the unique solution. We were not able to show analytically the concavity of the function $P_d(L,R(v),u,v)$. If the function is not concave, solving Equation (13) may give us multiple points, and we need to pick the one which is appropriate for our problem.

2. Effect on the Multiple Searcher Problem

Next, we study the effect of the detection radius being a function of the searcher speed in the two-searcher problem. We have the same border length and target speed as in the previous case; in particular, we have L=200 and u=5 units. For simplicity, we hold one searcher's detection radius constant at 6 units while allowing the other searcher's detection radius to vary as a function of its speed. In particular, we let $R_1 = 6e^{-v_1/60}$ and set the speed of the second searcher to 100 speed units. Then, we vary the speed of the first searcher and observe the optimal allocation and the probability of detection at the optimal allocation.

We can see in Figure 19(a) that the allocation to the first searcher starts with 0.06 when its speed is 0; in other words, when it is stationary. An allocation of 0.06 means that 0.06 x 200 = 12 units are allocated to the first searcher, which is double its detection radius when it is stationary. In this case, the probability of detection of the first searcher in its region is 1.

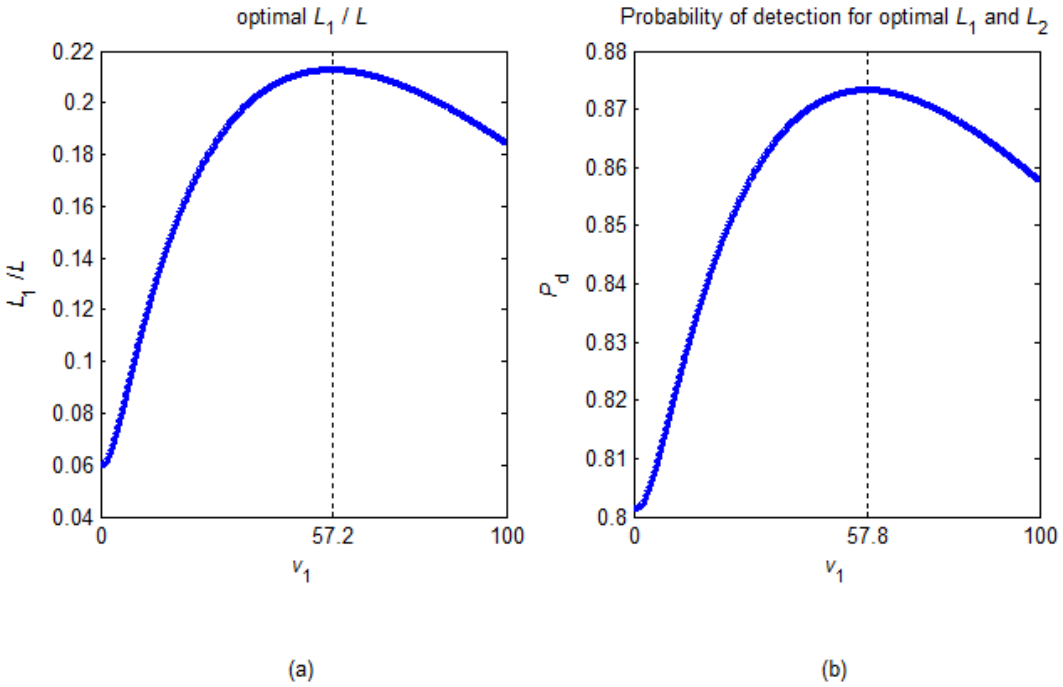


Figure 19. Detection radius as a function of speed for two searchers.

Although increasing v_1 decreases R_1, the optimal allocation to the first searcher nevertheless increases as v_1 increases from zero. After some point (v_1=57.2 in Figure 19(a)) allocation to the first searcher starts to decrease.

We have a similar relationship for the probability of detection as a function of the speed of the first searcher. When we start with the first searcher being stationary and increase v_1 infinitesimally, we notice a reduction in the probability of detection. Further increasing v_1 results in an increase. This behavior shows that the probability of detection for this problem is non-concave. Further increasing v_1 causes a reduction in probability of detection after some point (v_1=57.8 in Figure 19(b)).

Note that the peaks of two curves in Figure 19(a) and Figure 19(b) do not have to occur at the same v_1.

3. Multiple Searchers with Varying Detection Radius Depending on Speed

After studying the effect of the detection radius being a function of speed only for the first searcher, we now study the case in which the detection radii of both searchers are functions of their corresponding speeds. We use the same function for the detection radius of the first searcher that we used in the previous section, $R_1 = 6e^{-v_1/60}$, and use a slightly different function for the second searcher in order to observe the difference in the results. We use $R_2 = 6e^{-v_2/90}$. We can see the difference between the two detection radii functions in Figure 20.

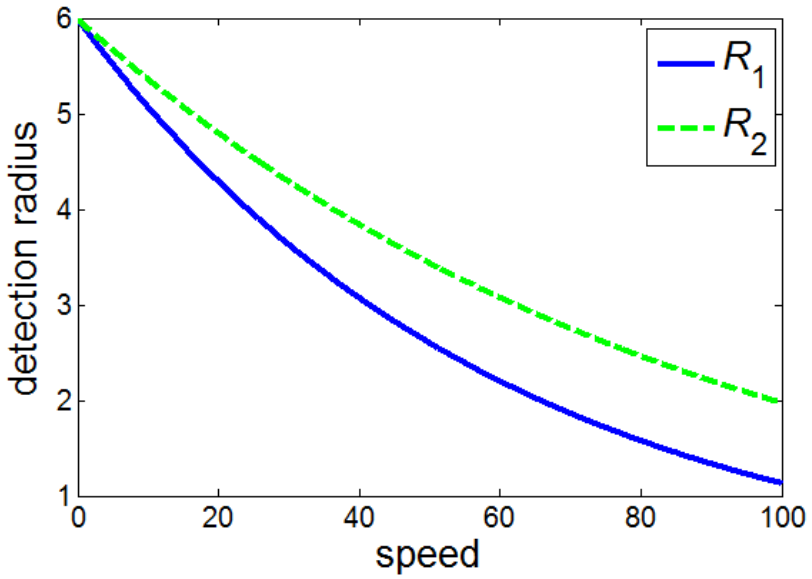


Figure 20. Detection radii of the two searchers.

We observe the optimal allocation to the first searcher with searcher speeds varying from 0 to 100 speed units in Figure 21(a). These allocations result in the probability of detection shown in Figure 21(b). The speed increment in Figure 21(a) and Figure 21(b) is 0.1 speed units.

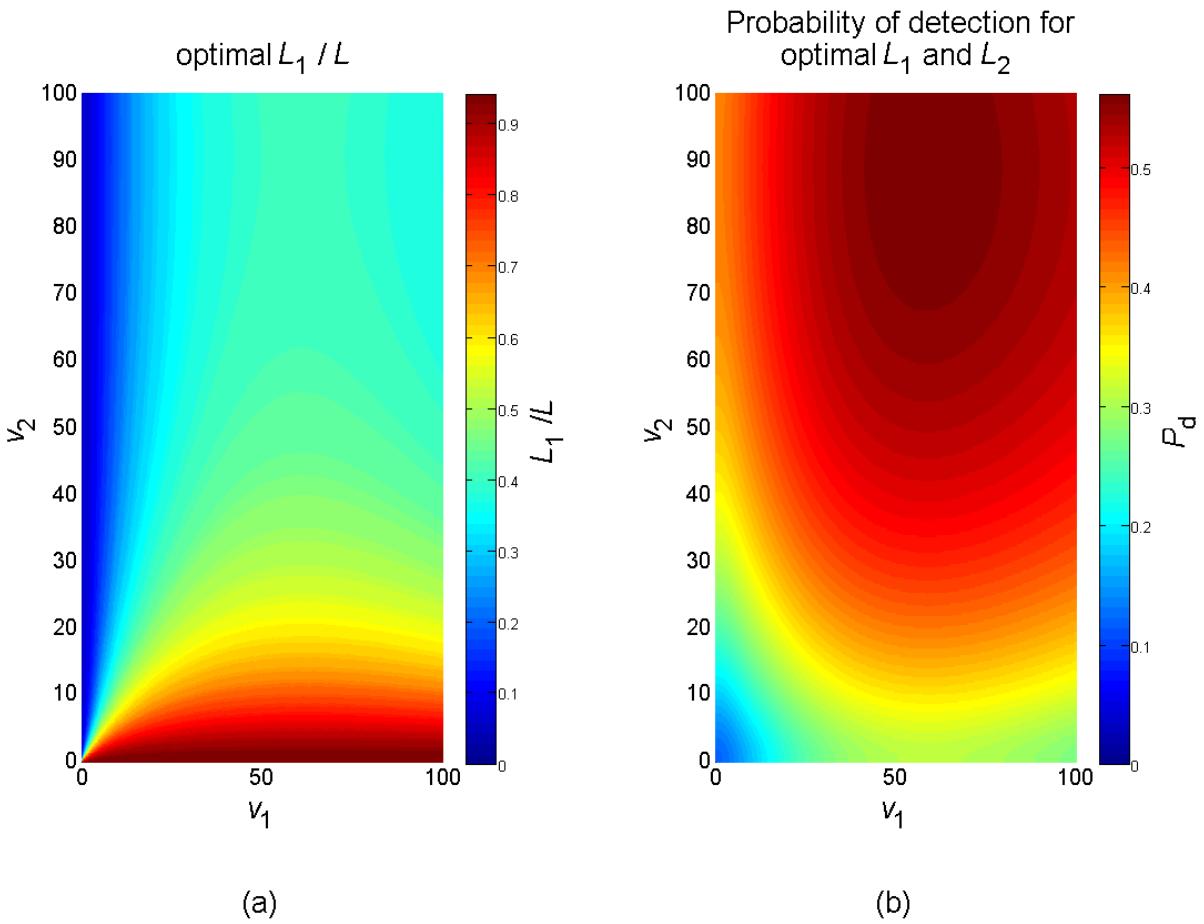


Figure 21. Varying both detection radii.

When one of the searchers is stationary, there is very little allocation to it. The allocation to a stationary searcher is twice its detection radius as in the previous cases in Figure 18 and Figure 19.

In the region where both searchers have moderate speeds, increasing v_1 up to around 60 speed units while keeping v_2 constant results in an increase in the optimal allocation to the first searcher and in the probability of detection. After that value, both the allocation to the first searcher and the detection probability start to decrease. This is not the case for v_2. When v_1 is kept constant and v_2 is increased up to around 70 speed units, the allocation to the first searcher decreases, which means that the allocation to the second searcher increases. After that value, the allocation stays the same. In the case of the probability of detection, it always increases. This can be explained by their detection

radius functions; note that the detection radius of the second searcher does not decrease as much as that of the first searcher. The positive effect of increasing speed always dominates the negative effect of the reduction in the detection radius for speeds up to 100 speed units for the second searcher.

The maximum probability of detection is observed at a value of 0.56 when v_1 is 58.3 speed units and v_2 is 88.4 speed units. At these speed values, the optimal allocation to the first searcher is 40% of the border length.

IV. GEOMETRIC CONSIDERATIONS

In this chapter we study the effect of making some changes in the geometry of the border patrol problem. We introduce two angles for studying the geometry, α which is the angle between the border and the horizontal axis, and β which is the angle between the searcher's path and the horizontal axis. The horizontal axis is perpendicular to the two barriers. The two angles α and β can be seen in Figure 22, where the horizontal axis is the black dotted line, the border is the red dashed line, and the searcher's path is the blue dashed-dotted line. The target moves perpendicular to the horizontal axis while trying to pass through the border.

In this study we made the border and searcher's path symmetric to the horizontal axis in such a way that the searcher's path intercepts the border in the middle of the border as shown in Figure 22.

In this chapter we only study the single searcher case since our goal is to get insight about the geometry.

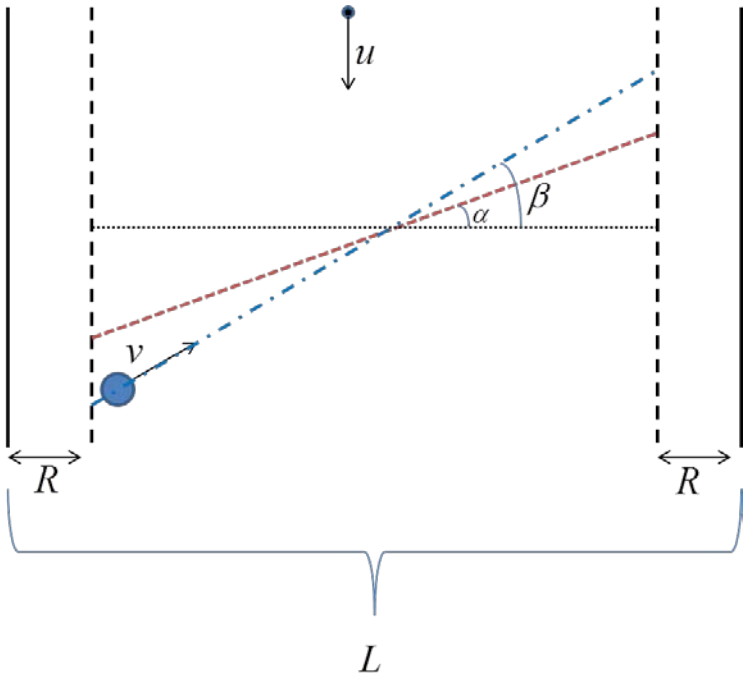


Figure 22. Geometry.

A. ANALYTICAL SOLUTION

In this subsection we only analyze the effect of changing the angle of the searcher's path (β) by keeping the angle of border (α) constant at 0. The speed vector of the searcher is shown in Figure 23(a). Figure 23(b) and Figure 23(c) show the speed of the searcher in the target stationary geometry when the searcher is moving to the right and left, respectively.

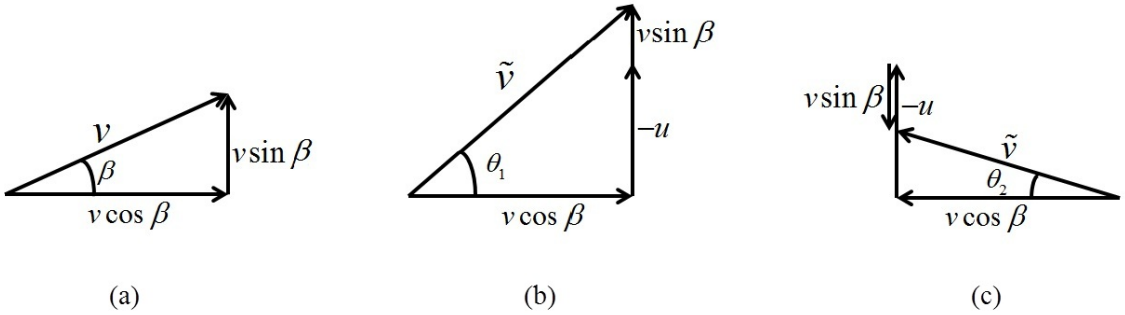


Figure 23. Speed vector of the searcher.

All the angles in Figure 23 can have negative values depending on the magnitudes and directions of the speed vectors. Figure 24 shows the detection region (coverage area) of the searcher in target stationary geometry. Although Figure 24 refers to the case when β is between 0 and 90 degrees, the area covered will be the same for other β values. Figure 24(a) refers to the case when θ_2 is greater than 0, and Figure 24(b) refers to the case when θ_2 is less than 0. Due to the geometry θ_2 will always be less than θ_1 in magnitude.

We can calculate θ_1 and θ_2 as:

$$\theta_1 = \arctan\left(\frac{|-u| + v\sin\beta}{v\cos\beta}\right) = \arctan\left(\frac{u + v\sin\beta}{v\cos\beta}\right)$$
$$\theta_2 = \arctan\left(\frac{|-u| - v\sin\beta}{v\cos\beta}\right) = \arctan\left(\frac{u - v\sin\beta}{v\cos\beta}\right). \quad (14)$$

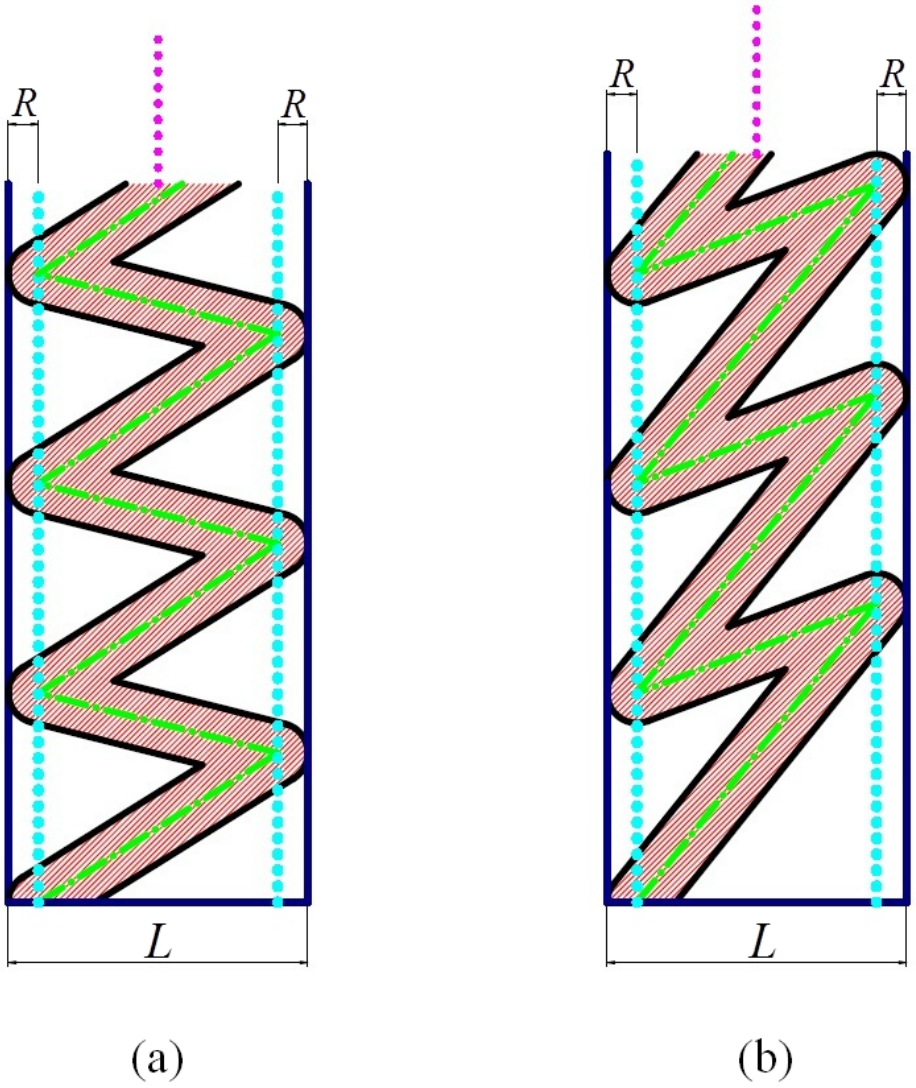


Figure 24. Geometric considerations in target stationary geometry.

In Figure 24(a) and Figure 24(b) we see that the coverage area of the searcher follows a pattern after the first cycle. Depending on the angles, it may take more cycles to follow a pattern. Since the coverage area follows a pattern, we can find the probability of detection by finding the coverage area between two points that mark the beginning and end of the pattern and dividing it to the total area (shaded and unshaded) of that pattern. Possible patterns for Figure 24(a) and Figure 24(b) are shown in Figure 25(a) and Figure 25(b). The patterns are selected between two points when the searcher is distance *R* away from the left edge of the border.

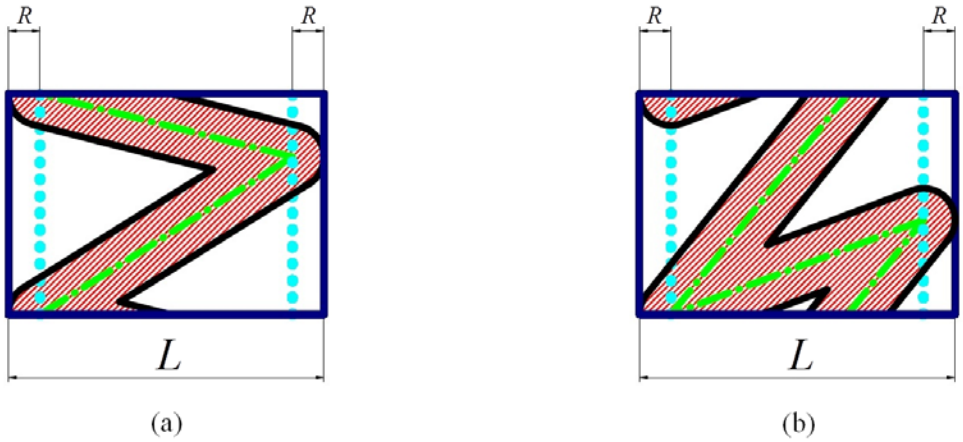


Figure 25. Patterns to be used for computations.

We see that there is a big difference in the geometry of the shaded areas in Figure 25(a) and Figure 25(b). Our goal is to find an equation for the probability of detection that will be valid for all angles β, and hence θ_1 and θ_2.

It is difficult to find the areas in Figure 25. In order to make it easy to find the areas, we generated Figure 26 in which we moved some portions of the shaded areas. The shaded areas in Figure 26 seem to be easier to compute than Figure 25, although they are exactly the same.

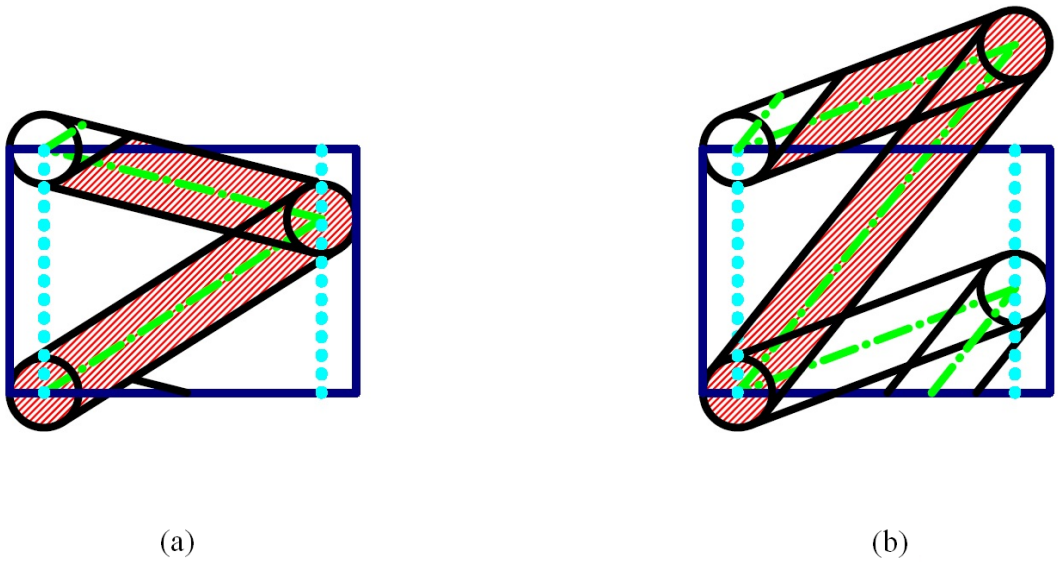


Figure 26. Modification of the areas covered.

We also see that the shaded areas in Figure 26(a) and Figure 26(b) have the same logic, which enables us to obtain a common formula for both cases. Hence, we focus on finding the shaded area in Figure 26(a). Figure 27(a) shows the shaded area to be computed in Figure 26(a). Note that it is simpler to compute the whole area in Figure 27(a) and subtract the unshaded regions than to compute only the area of the shaded region. Figure 27(b) shows the whole area.

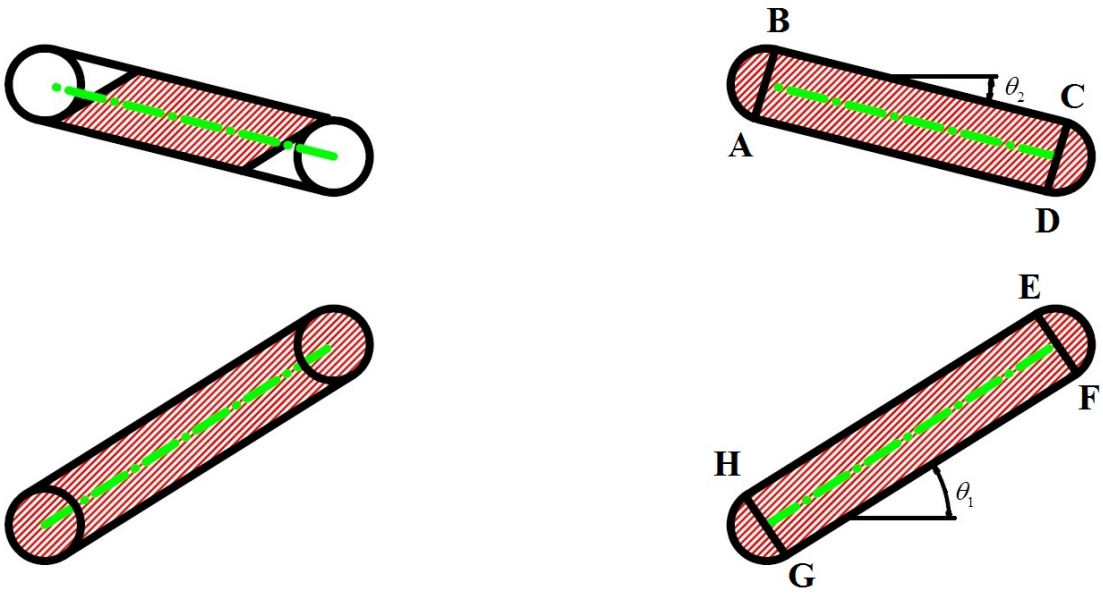


Figure 27. Simplifications in the areas.

We can find the area of the shaded region in Figure 27(b) by finding the areas ABCD, EFGH, and the four semicircular regions. The four semicircular areas are all equal, and their total area is

$$4\frac{\pi R^2}{2} = 2\pi R^2. \tag{15}$$

The area of the rectangle ABCD is

$$AB \times AC = 2R\frac{L-2R}{\cos\theta_2}. \tag{16}$$

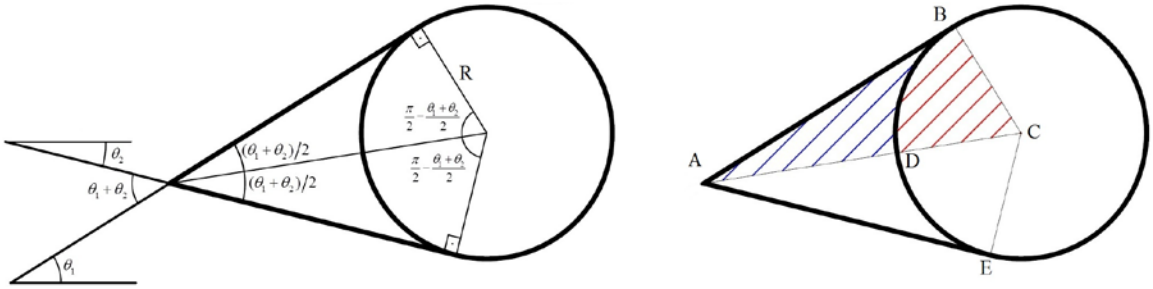
The area of rectangle EFGH is

$$EF \times FG = 2R\frac{L-2R}{\cos\theta_1}. \tag{17}$$

The total shaded area in Figure 27(b) can then be computed by using Equations (15), (16), and (17) as:

$$2\pi R^2 + 2R\frac{L-2R}{\cos\theta_2} + 2R\frac{L-2R}{\cos\theta_1}. \tag{18}$$

The areas of the unshaded regions in Figure 27(a) can be computed by using the geometry in Figure 28. Figure 28(a) shows the angles that can be used to compute the total area in Figure 28(b), which equals one of the unshaded regions in Figure 27(a). We can find the total area in Figure 28(b) by finding the area ABE and adding the circular area. We can find the area ABE by finding the triangular area ABC, subtracting the red shaded area CBD, and multiplying the result by 2.



(a) (b)

Figure 28. Unshaded areas.

The triangular area ABC is:

$$\frac{AB \times R}{2} = \frac{R\cot\left(\frac{\theta_1+\theta_2}{2}\right)R}{2} = \frac{R^2\cot\left(\frac{\theta_1+\theta_2}{2}\right)}{2} \tag{19}$$

The area CBD is:

$$\pi R^2 \frac{\frac{\pi}{2} - \frac{\theta_1 + \theta_2}{2}}{2\pi} = \frac{R^2}{2}\left(\frac{\pi}{2} - \frac{\theta_1 + \theta_2}{2}\right) \tag{20}$$

The total area in Figure 28(b) can be computed by using Equations(19), (20), and adding the circular area as:

$$\begin{aligned} &2\left(\frac{R^2 \cot\left(\frac{\theta_1 + \theta_2}{2}\right)}{2} - \frac{R^2}{2}\left(\frac{\pi}{2} - \frac{\theta_1 + \theta_2}{2}\right)\right) + \pi R^2 = R^2\left(\cot\left(\frac{\theta_1 + \theta_2}{2}\right) - \frac{\pi}{2} + \frac{\theta_1 + \theta_2}{2} + \pi\right) \\ &= R^2 \cot\left(\frac{\theta_1 + \theta_2}{2}\right) + \frac{\pi R^2}{2} + R^2 \frac{\theta_1 + \theta_2}{2}. \end{aligned} \tag{21}$$

The total area in Figure 27(a) can then be computed by using Equations (18) and (21) as:

$$\begin{aligned} &2\pi R^2 + 2R\frac{L - 2R}{\cos\theta_2} + 2R\frac{L - 2R}{\cos\theta_1} - 2\left(R^2 \cot\left(\frac{\theta_1 + \theta_2}{2}\right) + \frac{\pi R^2}{2} + R^2 \frac{\theta_1 + \theta_2}{2}\right) \\ &= 2\pi R^2 + 2R\frac{L - 2R}{\cos\theta_2} + 2R\frac{L - 2R}{\cos\theta_1} - 2R^2 \cot\left(\frac{\theta_1 + \theta_2}{2}\right) - \pi R^2 - R^2\left(\theta_1 + \theta_2\right) \\ &= \pi R^2 + 2R\left(L - 2R\right)\left(\frac{1}{\cos\theta_2} + \frac{1}{\cos\theta_1}\right) - 2R^2 \cot\left(\frac{\theta_1 + \theta_2}{2}\right) - R^2\left(\theta_1 + \theta_2\right) \end{aligned} \tag{22}$$

where θ_1 and θ_2 are as previously shown in Equation (14).

Figure 29 shows the probability of detection with respect to the angle of the searcher's path (β) while the angle of the border (α) is kept constant at 0. We vary β from -89 to 89 degrees since β =90° and β =-90° makes θ_1 and θ_2 undefined (Equation (14)). We choose border length to be 200 distance units, detection radius to be 6 distance units, target speed to be 5 speed units, and searcher speed to be 20 speed units. We observe the maximum probability of detection (P_d) when β =0°, i.e., when the searcher patrols over the border. We notice a decrease in P_d when β increases or decreases.

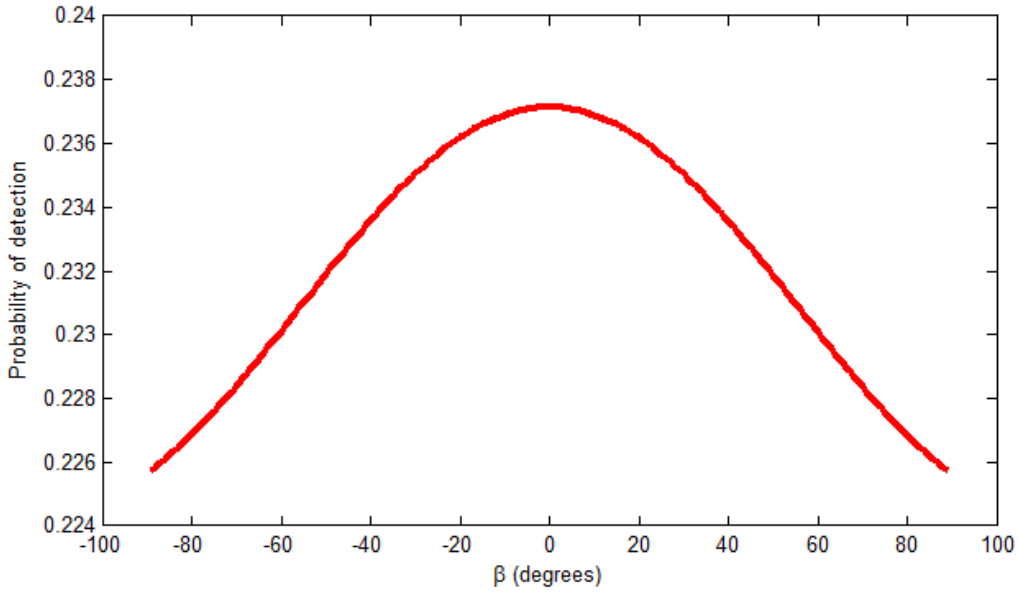


Figure 29. Analytical results for geometric considerations.

B. MONTE CARLO SIMULATION

We generate the Monte Carlo simulation model for the analysis of varying β by using the same principles of the single searcher problem, except the searcher travels on a linear path with angle β between its path and border as in Figure 22. The border is aligned with horizontal axis. We use ten million replications in the simulation and varied β from -89 to 89 degrees with 0.1 degree increments. We do not include -90 and 90 degrees, because it makes the horizontal axis component of the searcher speed to be 0, causing only vertical movement, which violates the problem assumptions. We use the same parameters that we used to analyze the analytical solution in Figure 29.

Figure 30 shows the analysis of the Monte Carlo simulation results. The red dashed line shows the analytical solution obtained in Figure 29. The blue line shows the estimated probability of detection ($\widehat{P_d}$) along with its 95% confidence interval shown with green dotted lines.

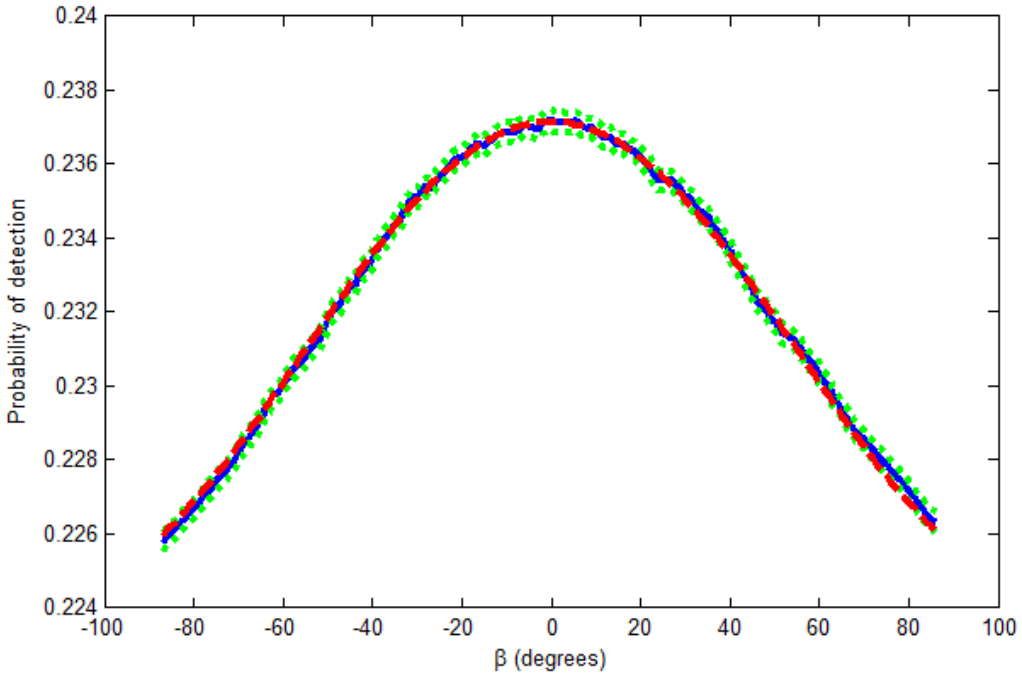


Figure 30. Simulation results of geometric considerations.

We observe that the analytical solution and Monte Carlo simulation results are very close to each other.

We also notice that the absolute difference between the maximum and minimum probability of detection in Figure 30 is only about 1%, and hence we can say that the probability of detection is nearly independent of the angle of the searcher. This may cause some problems in practice since the searcher also detects some targets which are very far away from the border for large β, which may result in very large number of false positives. For this reason we need to set a limit on the maximum distance from the border that an object detected to be classified as a target.

We introduce a new parameter d, which is the maximum distance from the border in order for an object detected to be classified as a target. In this case, if we look at Figure 31, in order for the searcher to classify an object it sees as a target, the target has to be between the two green dashed lines, both of which are distance d away from the border. If the object is outside this region, the searcher will not classify it as a target; in other words, the searcher will not detect it.

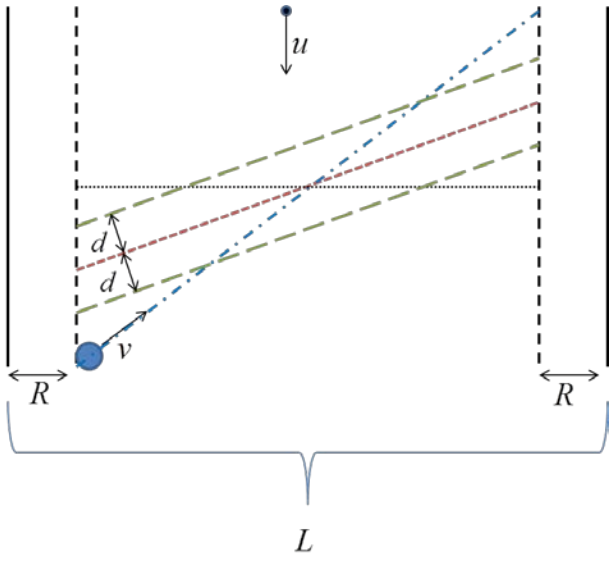


Figure 31. Introducing *d*.

Figure 32 shows the results of the same Monte Carlo simulation after the introduction of *d*. In Figure 32(a), *d* is 50 distance units, and in Figure 32(b) it is 20 distance units. In both cases we see that $\widehat{P_d}$ stays nearly constant up to some β, and then it starts to decrease. Smaller *d* causes us to observe this transition at smaller β values. Specifically, we observe the transition when the searcher starts to move out of the region in which it is capable of detecting targets.

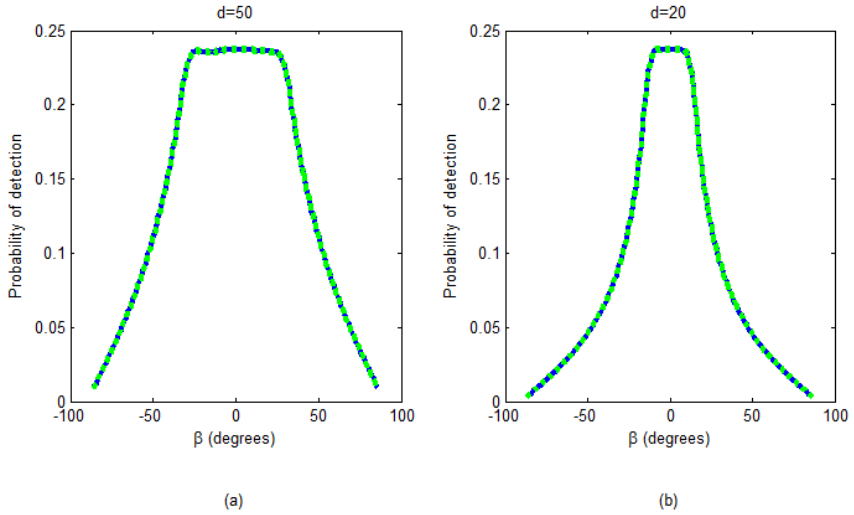


Figure 32. Varying β with the introduction of *d*.

Figure 33 shows the Monte Carlo simulation results when β is kept constant at 0 and α is varied from -89 degrees to 89 degrees with 0.1-degree increments. We observe a similar behavior as in Figure 32. The estimated probability of detection stays nearly the same for small α, then it starts to decrease.

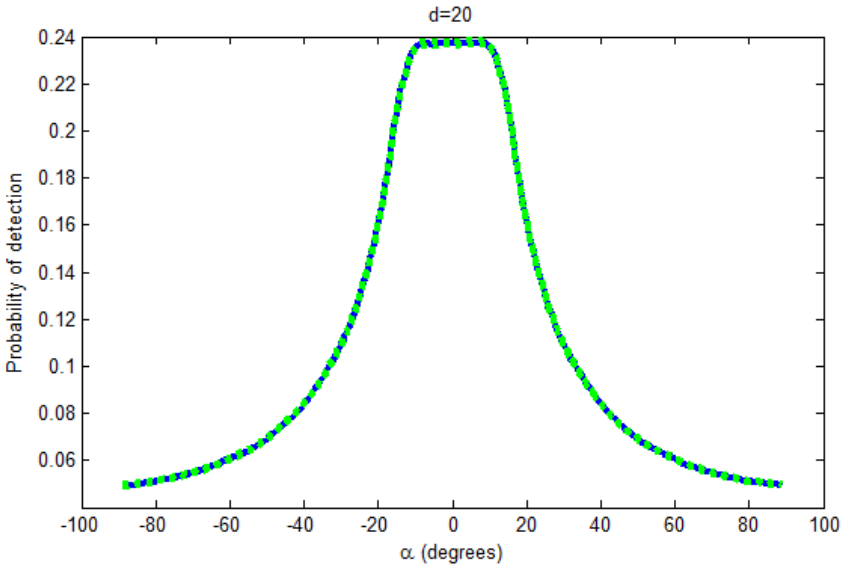
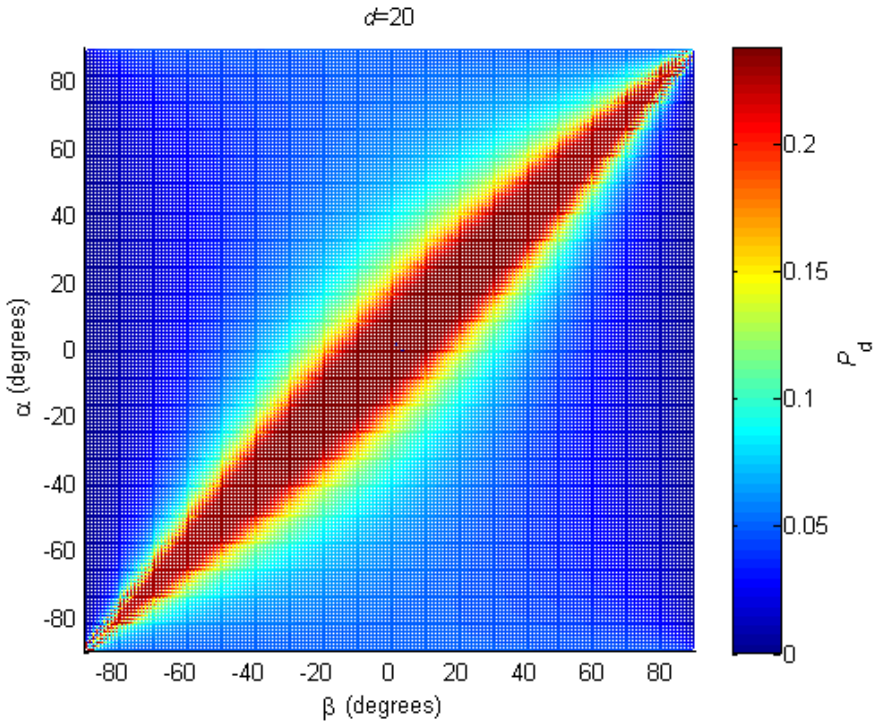


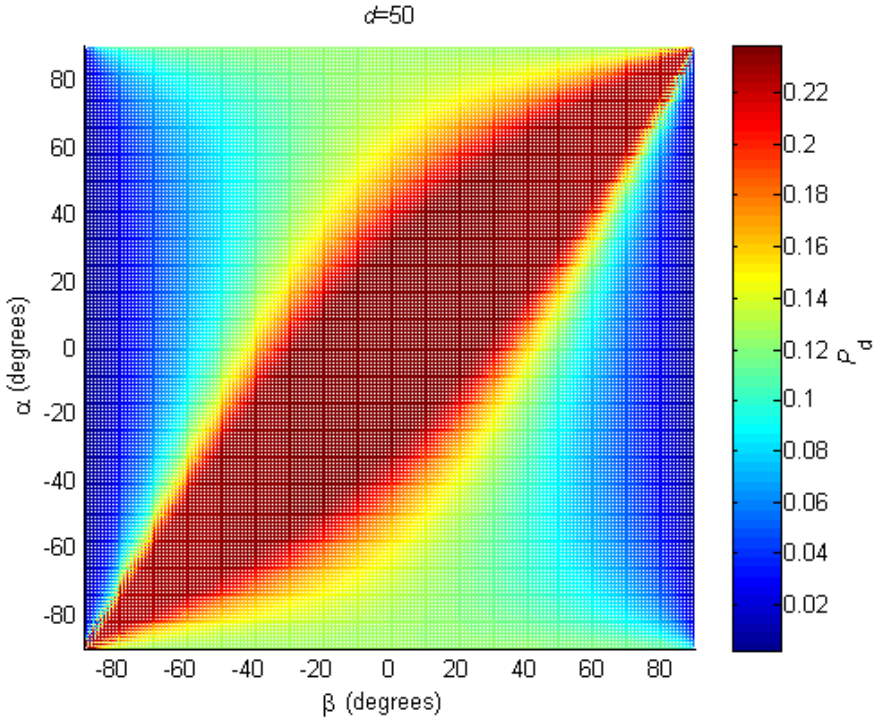
Figure 33. Varying α.

Figure 34(a) and Figure 34(b) show the Monte Carlo simulation results as a heat map when both α and β are varied from -89 to 89 degrees with 0.1 degree increments for d=20 and d=50 distance units, respectively. We observe that the estimated probability of detection is maximized when α and β have the same value, i.e., when the searcher patrols over the border. We also see that there is some flexibility in this sense; that is, the searcher's path may differ slightly from the border. For example, in Figure 34(a) when α is 0, β may be between -10 and 10 degrees and we still observe a probability of detection close to the maximum. This flexibility decreases with increasing border angles.

When we increase d to 50 units, we notice that the flexibility increases. For instance, when α = 0, β may now be between -30 and 30 degrees, and we are still close to the maximum detection probability.



(a)



(b)

Figure 34. Detection probability as a function of both α and β.

C. ANALYSIS

Up to now we worked on straight borders and straight searcher paths in order to obtain insights about the optimal deployment of UAVs. However, most borders are not straight lines. For example, assume the straight blue line in Figure 35(a) shows the border that we are trying to conduct surveillance. The dashed lines, that are separated *d* distance away from the border, mark the region that a detected object can be classified as a target.

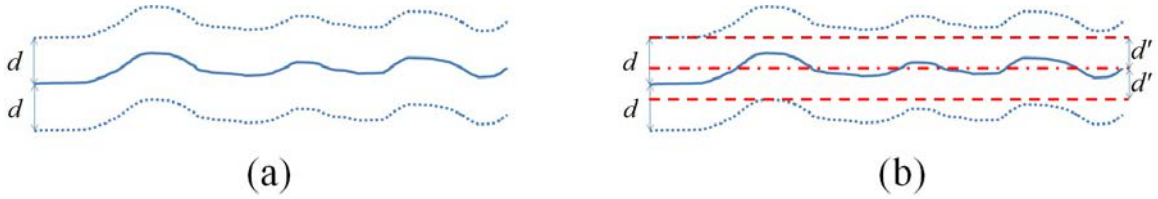


Figure 35. A sample non-straight border

By using the results just obtained, we reason that a reasonable approach might involve approximating the region as in Figure 35(b). The dashed red lines mark our new region in which we can detect the targets; note that it is a subset of the region in Figure 35(a). The dashed dotted line shows the approximate border, which is distance d' away from both dashed lines. This approximate border and detection region are simpler to analyze than the original border, and because the approximate detection region is contained within the original region, the detection probability we estimate will be a conservative estimate of the actual detection probability.

If a border cannot be approximated as a single straight line, we can divide the border into several segments and generate multiple approximations. In this way we can analyze each segment separately. A sample border that cannot be divided into a single approximation is shown in Figure 36(a), and its possible approximation is shown in Figure 36(b). The red lines show one segment and the green lines show the other.

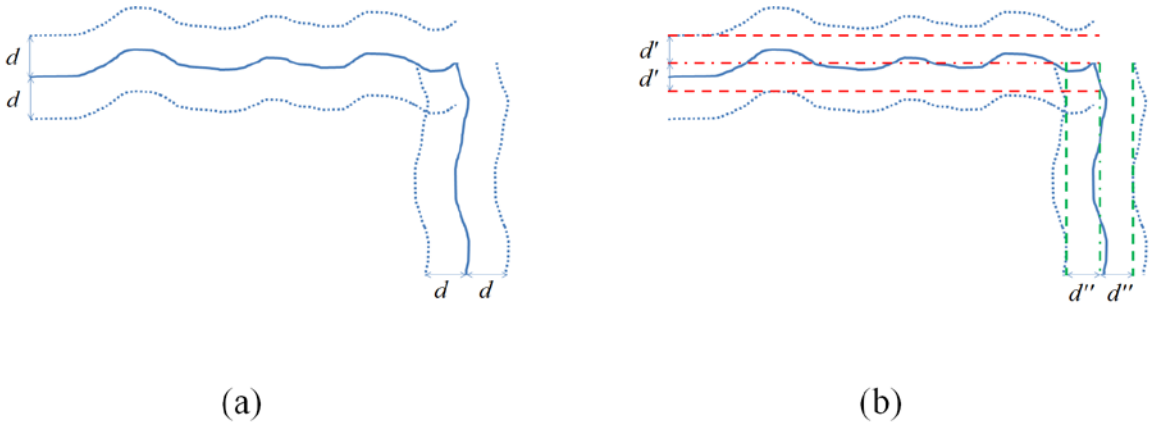


Figure 36. A sample complex border

V. IMPERFECT SENSOR

In the previous chapters, we only considered cookie-cutter sensors. Cookie-cutter sensors are perfect sensors; that is, if a target is within the detection range, it is detected with probability 1. In reality, no sensor is perfect, which means there is a probability that even if a target is within the detection range of a sensor, it may not be detected.

For example, assume the circle in Figure 37 shows the detection range of an imperfect sensor (the searcher is stationary at point A carrying an imperfect sensor) with detection radius *R*. The detection range marks its detection region, where the sensor is capable of detecting the target. If a target does not get into the detection region, it is not detected. Assuming that the target is not initially within the sensor's detection region, in order for detection to occur, the target should enter the detection region at some point, which is the point B in Figure 37. Without being detected, the target may exit the detection region at some other point, which is the point D in Figure 37. On its way from B to D, the target passes through the point C, which is its "closest point of approach". The distance between the searcher and the target at this closest point is defined as the lateral range (*x*) (Wagner et al., 1999).

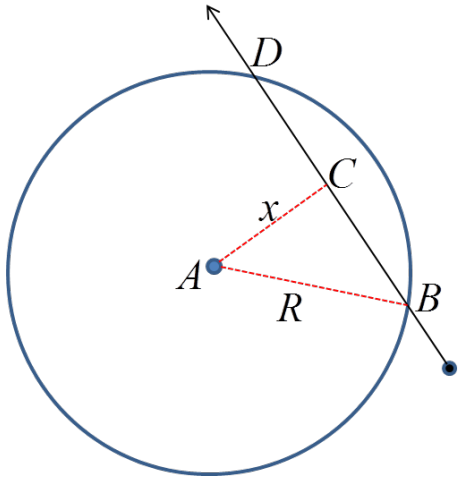


Figure 37. Lateral range (*x*).

Suppose a target is moving along a line with respect to a searcher which has an imperfect sensor. We can regard this as a searcher stationary geometry. Also suppose that the target enters and exits the detection region of the sensor at points B and D, respectively, as in Figure 37. At point B, the probability of detection is 0. After the target passes point B, the cumulative probability of detection starts to increase and keeps increasing as long as the target stays within the detection region. This results in the peak of the cumulative probability of detection to be observed at point D, when the target exits the detection region. This cumulative probability is denoted by $P(x)$, and its graphical representation for all values of x is known as a "lateral range curve", where x is the minimum distance observed between the target and the sensor while the target travels within the detection region of the sensor (Wagner et al., 1999).

A sample lateral range curve can be of the form $e^{-\left(\frac{x}{k}\right)^2}$ (Soza & Company, Ltd., 1996) like the solid line in Figure 38(a), or $1-e^{-\left|\frac{a}{x}\right|^b}$ like the dashed line in Figure 38(a). Figure 38(b) shows a family of lateral range curves of the form $e^{-\left(\frac{x}{k}\right)^2}$ for various k. Increasing k results in an increase in the width of the lateral range curve. Figure 38(c) and Figure 38(d) show a family of lateral range curves of the form $1-e^{-\left|\frac{a}{x}\right|^b}$ for various a and b, respectively. From the figures we see that a controls the witdth of the curve and b controls the steepness.

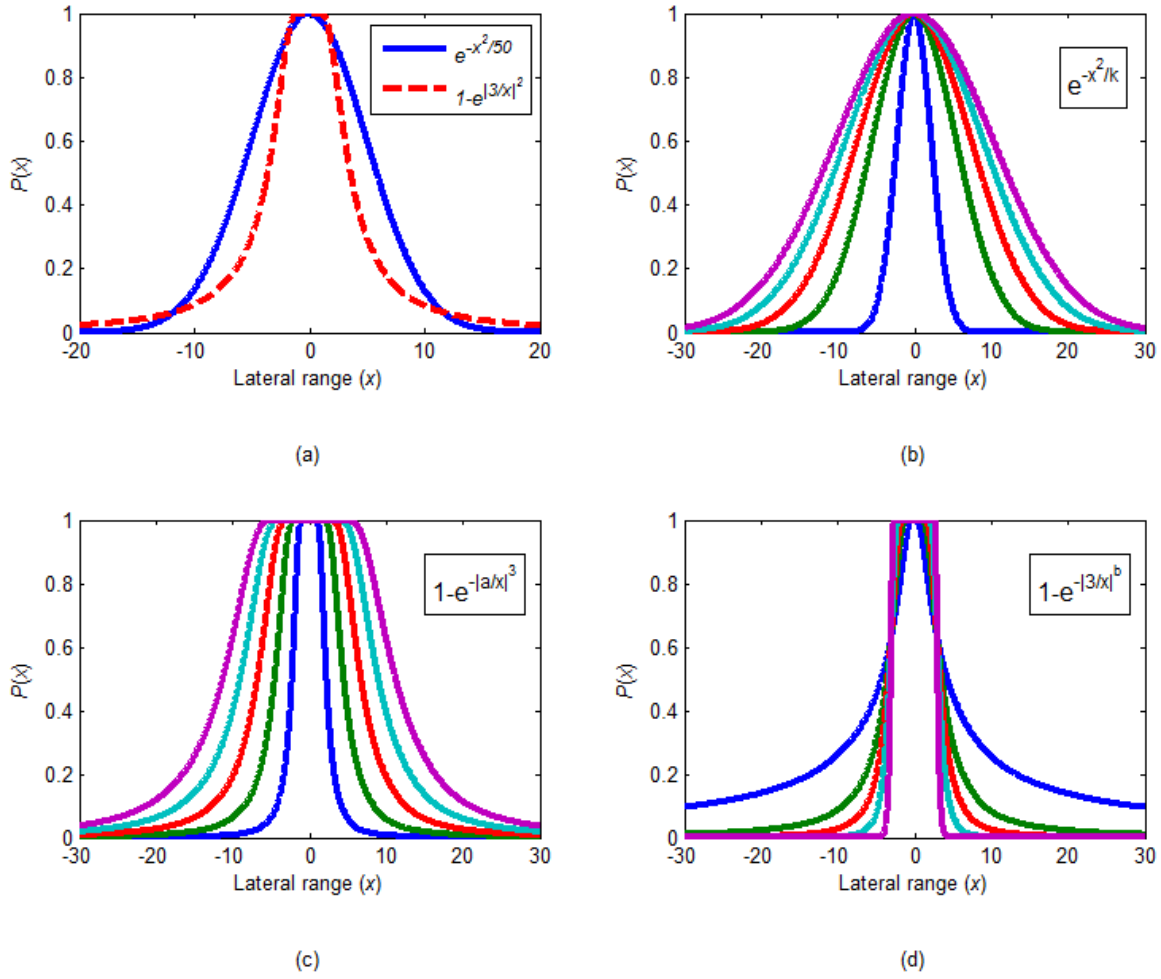


Figure 38. Lateral range curves.

Since our goal is to have an insight into imperfect sensors, we only study the simple border patrol problem with a single target and a single searcher that has an imperfect sensor. We denote this problem as the simple imperfect sensor problem. The difference between this problem and the problem in Chapter II.A, is only the sensor's imperfectness. For our studies, we choose an imperfect sensor which has the lateral range curve as

$$P(x)=\begin{cases}0.95-e^{\left|\frac{3}{x}\right|^2} & \text{if} \quad |x|\le 6\\ 0 & \text{otherwise.}\end{cases} \tag{23}$$

The lateral range curve we choose is shown in Figure 39. We decide not to choose the maximum cumulative detection probability as 1; rather, we choose 0.95, in order to always give the target a chance to escape, which is mostly the case in real life applications. We also set a limit to the maximum detection range as 6 distance units.

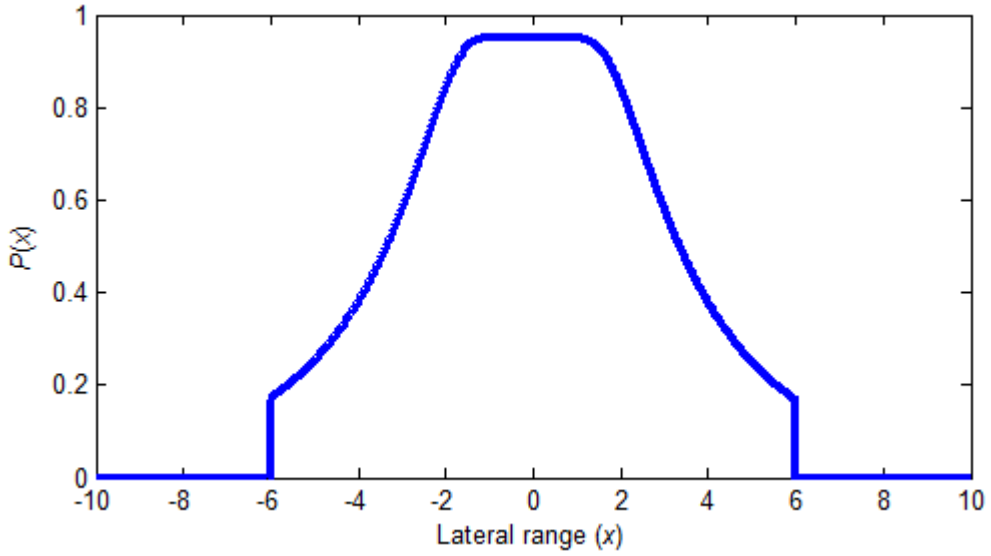


Figure 39. Chosen lateral range curve.

In Figure 39 we see that the lateral range (x) also takes negative values. The positive lateral range values refer to a target passing through one semicircle of the detection region. Likewise, negative values refer to a target passing through the other semicircle. This may be useful for some kinds of sensors (radars, etc.). Since we decided to work on a symmetric lateral range curve, we do not take into account which side the closest point of approach is observed.

In order to make an analysis on the simple imperfect sensor problem, instead of building upon our model in Chapter II.A.2, we generated an event-driven model, in which we calculate the minimum distance that is going to be observed between the target and the searcher in the simulation and decide whether a detection occurs. Calculation of this distance is provided in Appendix D. We verified our new model by comparing it to our time-step simulation in Chapter II.A.2, with the same parameters as used in Figure 9. Figure 40 shows the comparison of the time-step simulation and the event-driven

simulation. As we can see from the figure, the results are nearly the same, and hence, we have verified our event-driven model.

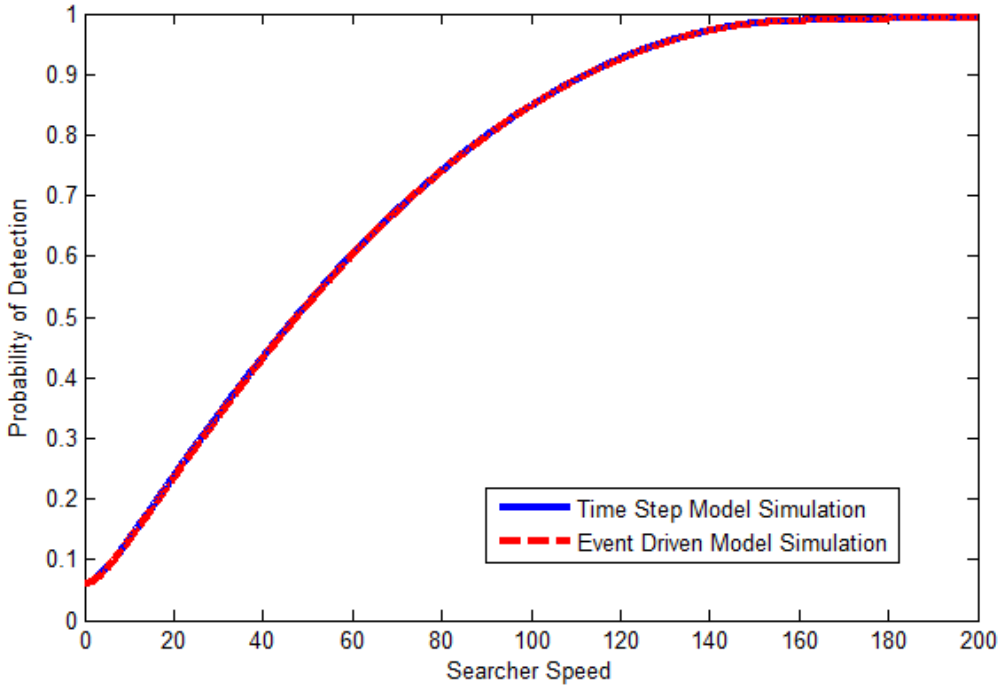


Figure 40. Comparison between the time-step and event-driven simulations.

In some complex problems (with geometric considerations, dynamic target motion, line-of-sight (LOS) obstacles, etc.), it may not be appropriate, may be very hard, or even may be impossible to use an event-driven model, which forces us to use a time-step simulation model.

In the cookie-cutter sensor cases that we studied in the previous chapters, we used time-step simulations and quit the simulation if the target gets within the detection region of the searcher; otherwise, we ran the simulation until the simulation end time, or until the target goes to a point where it cannot be detected anymore, whichever is smaller. In the case of an imperfect sensor, we cannot end the simulation, we need to log the distance between the searcher and the target, and make the necessary computations at the end of

the simulation to decide if the target is detected or not. Keeping track of this distance requires a new vector, which may have a very large size depending on the length of the simulation, leading to memory problems. Increasing the number of targets and/or searchers requires new vectors which multiplies the memory problem.

Instead of generating a vector to keep track of the distance, we may implement an algorithm that keeps track of the distance by storing it to a variable, and updates the variable if the distance is reduced further. But this approach requires new computations and new if-else statements to execute, which increases the runtime. We also cannot stop the simulation before the simulation end time, in case the target gets closer to the searcher. Alternatively, to end the simulation before the simulation end time, we can implement another algorithm which further increases computation in the simulation. The effect of the increase in computations is infinitesimal in the simple imperfect sensor problem, but the problem gets huge when there are multiple searchers, when the simulation end time is long, and when we perform a large number of replications.

For these reasons, it would be nice to decide, without any extra computations, if the target is detected or not at the first instance the target enters the detection region of the searcher. This is only possible, if the lateral range curve of the sensor is a step-like function; that is, it has a constant cumulative probability of detection for some range of x, and it has 0 cumulative detection probability for the remaining range of x. If $P(x)$ is constant within the detection region, we can decide whether or not the target is detected at the first instance the target gets into the detection region and quit the simulation.

As a matter of fact, Soza & Company, Ltd. (1996) suggest approximating lateral range curves as cookie-cutter sensors or M-Beta sensors in such a way that the areas under the lateral range curves of both the actual sensor and its approximation are equal. In the M-Beta sensor model, the cumulative detection probability is kept constant at some value M, which is between 0 and 1. The width of this approximate lateral range curve is adjusted to β in order to set the areas under the lateral range curves of the actual sensor and its approximation equal. The cookie-cutter sensor model is a specific case of M-Beta sensor model when M is chosen to be 1. Figure 41 shows the cookie-cutter approximation and M-Beta range approximation when M is set to 0.5, along with our original sensor

model. We calculate the area under the curve to be 7.2 units, so the cookie-cutter sensor approximation has $2R$=7.2, which makes its detection radius to be 3.6 units. For the M-Beta approximation, we use M=0.5, so we have $M\beta$=7.2, which results in β=14.4, and hence, the detection radius is $R = \frac{\beta}{2} = 7.2$ distance units.

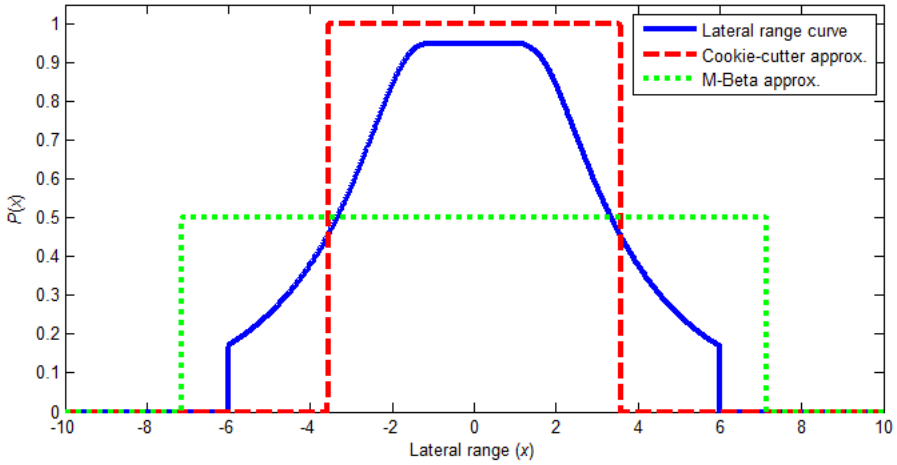


Figure 41. Sample lateral range curve approximations.

We performed three different simulations for the original sensor model and its two approximations, all of which are shown in Figure 41. For all models, the searcher patrols the border with its turning points being equal to its maximum detection range according to Figure 41. We calculate the closest distance observed between the target and the searcher, and decide whether the target is detected.

Figure 42 shows the comparison of the three models. In the three subfigures, we vary all the variables in the border search problem one by one, while keeping the others constant, and observe the probability of detection. We performed one million replications in all simulations, and only plotted the mean value of the estimated detection probability in all cases. In Figure 42(a) we vary the searcher speed from 0.1 to 300 speed units in 0.1 unit increments, while keeping the target speed and border length constant at 5 speed units and 200 distance units, respectively. In Figure 42(b) we vary the target speed from 0.1 to 15 speed units in 0.1 unit increments, while keeping the searcher speed and border length constant at 150 speed units and 200 distance units, respectively. Similarly, in

Figure 42(c), we vary the border length from 20 to 200 distance units in 10 unit increments, while keeping the target speed and searcher speed constant at 5 and 150 speed units, respectively.

When we vary the speed of the searcher and the target (Figure 42(a) and Figure 42(b)), we see that all three models are similar when the actual detection probability (probability of detection obtained from the original sensor model) is low, and they differ when the actual probability of detection increases. We get higher estimates with the cookie-cutter sensor approximation and lower estimates with the M-Beta sensor approximation.

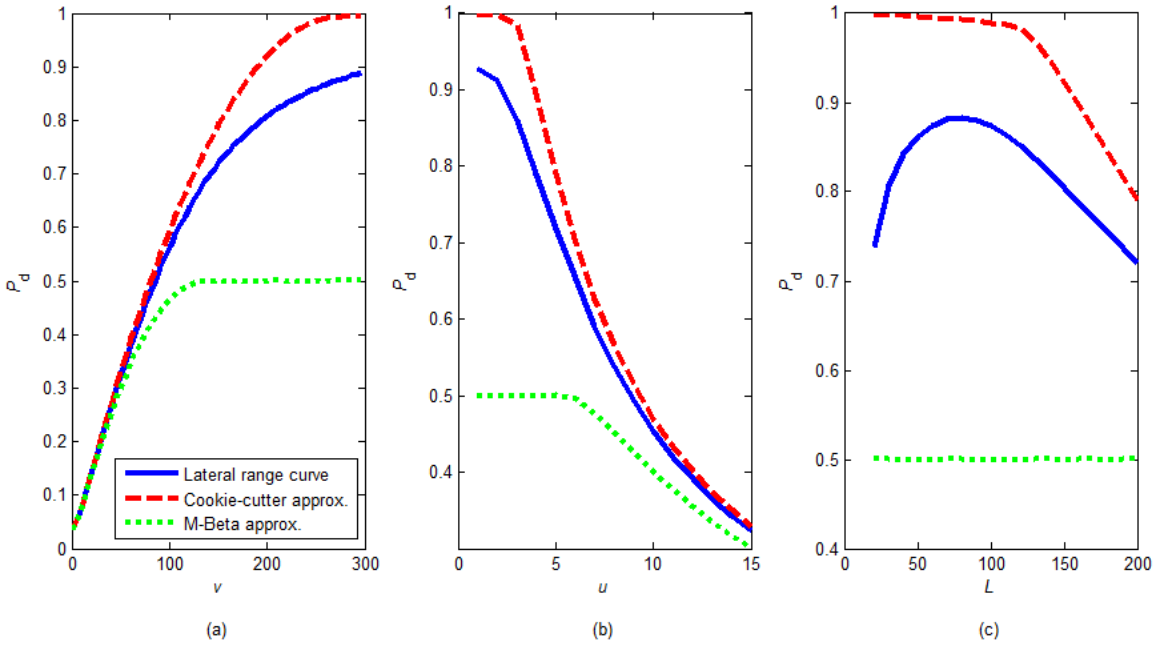


Figure 42. Comparison of the actual sensor and the approximations.

In Figure 42(c), we see that the actual probability of detection with varying border length has a concave shape, it first increases with increasing speed and then it starts to decrease. We would normally expect to see a non-increasing function since the border length negatively impacts the probability of detection. We consider the extreme case in order to see the reason for the concave shape. In the extreme case, the border length is

equal to two times the maximum detection range (2x6=12 distance units). The searcher stays stationary in the middle of the border at this border length.

When the searcher is stationary, due to the lateral range curve of its sensor which is shown in Figure 39, most of the targets that are close to the edges of the border will not be detected. This dramatically decreases the detection probability. For this reason, we conclude that the turning distance has an important impact on detection probability. We conduct an analysis on the turning distance for the actual sensor model, similar to our analysis in Chapter II.A.3, and present it in Appendix E.

With this analysis we conclude that the optimal turning distance depends on border length. We also expect a difference depending on the lateral range curve of the sensor. For this reason, the turning distance should be determined depending on the length of the border, since border length is constant for a particular problem. If a user cannot determine the turning distance, he or she can take the turning distance as 0; in other words, the searcher should go all the way to the end of the border before turning the other way, since the effect in the detection probability is negligible compared to choosing the optimal turning distance. On the contrary, choosing a bigger turning distance can yield worse results, especially when the border length is small.

Figure 43 shows the comparison of the actual sensor model with its two approximations, as in Figure 42, with the only difference being the turning distance in the simulations. We reduce the turning distance to zero in the original sensor model.

We see that changing the turning distance does not affect the detection probability considerably when we vary only the searcher and target speed. Because we set the border length to be 200 distance units, and at this border length, the difference in the detection probability was ignorable when we varied the turning distance from 0 to 6 distance units (see Appendix E).

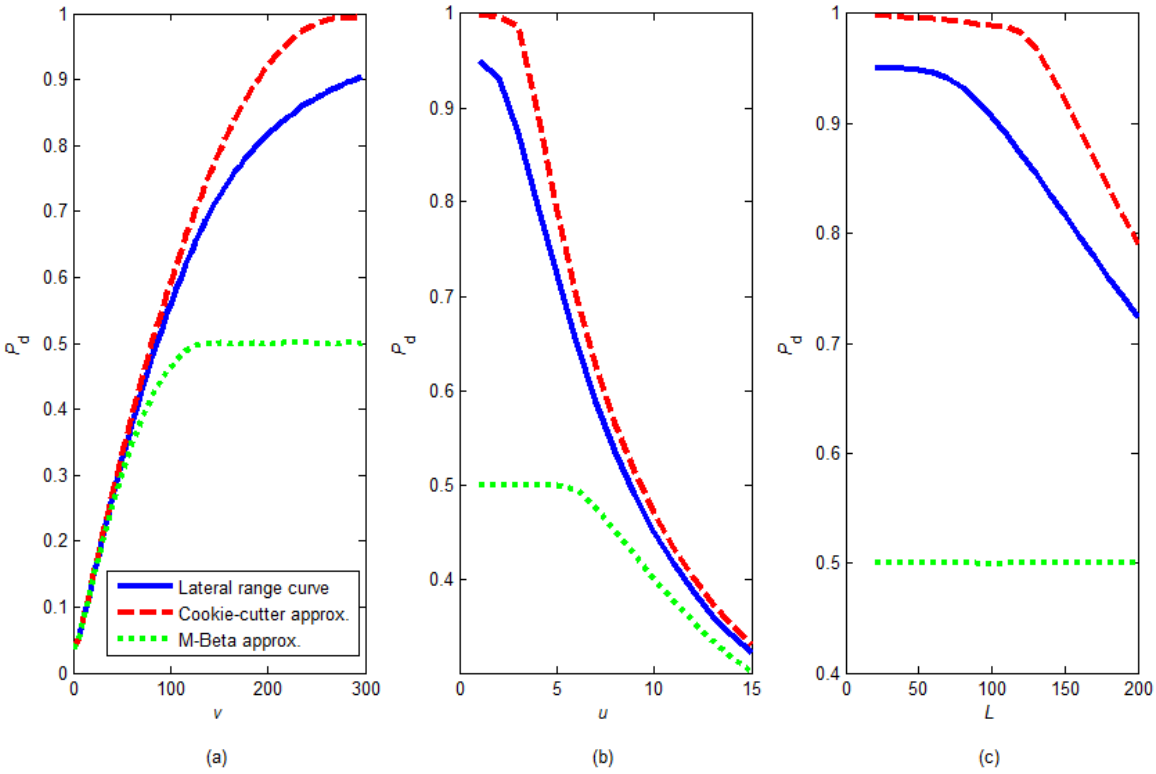


Figure 43. Comparison of the actual sensor and the approximations.

In Figure 43(c), we now see a non-increasing estimated detection probability in the actual sensor performance with increasing border length. The cookie-cutter sensor approximation has higher estimates than the actual sensor, and the M-Beta approximation has lower estimates. We also note that the estimated detection probability of the M-Beta sensor model stays nearly constant at 0.5 throughout the range of the border length we studied. This is reasonable since the maximum detection probability should be 0.5, even when the searcher covers the entire border. We would expect it to start to decrease after a high border length.

Due to the fact that in a particular problem we would know the border length, and it stays constant, we decide not to study various border lengths. Instead, we study the lateral range curve approximations only for the case when the border length is 200 distance units. In a different problem, the actual border length that is observed can be used. Since we chose the border length to be 200 distance units, we choose the turning

distance as 2.64 distance units, which results in the maximum estimated detection probability (Appendix E).

We study the cookie-cutter sensor and M-Beta sensor approximations separately in the following sections.

A. COOKIE-CUTTER SENSOR APPROXIMATION

In Figure 43(a), we study the estimated detection probability when varying the speed of the searcher and keeping other problem parameters constant. We note that the cookie-cutter sensor model approximation yields results close to the actual sensor model for searcher speeds nearly up to 100 speed units, and then it starts to differ. For high speed values, the cookie-cutter sensor approximation provides highly optimistic results.

Similarly, in Figure 43(b), we see the estimated detection probability by varying the speed of the target and keeping the other problem variables constant. In this case we note that the cookie-cutter approximation always yields higher estimates. The gap between the detection probabilities of the actual model and the cookie-cutter approximation is especially wide when the target speed is small.

In general, we note from Figure 43(a) and Figure 43(b) that the cookie-cutter sensor approximation results in highly optimistic estimates, especially in the regions where the actual sensor model also yields high detection probabilities. In this case, we would want to generate a better approximation. In the cookie-cutter sensor approximation, we first calculate the area under the lateral range curve, which is shown in Figure 39, as 7.2 area units. We calculate the detection radius of the cookie-cutter approximation by setting the area under its lateral range curve equal to that of the original sensor model, which is 7.2 area units. By this method we calculate the detection radius of the approximate model as 3.6 distance units.

Since the cookie-cutter sensor model is an optimistic model, we can make it a better approximation by somehow reducing its performance. In an approximate model, we can only change the cumulative detection probability and the detection radius from the actual sensor model. In a cookie-cutter model, the cumulative detection probability

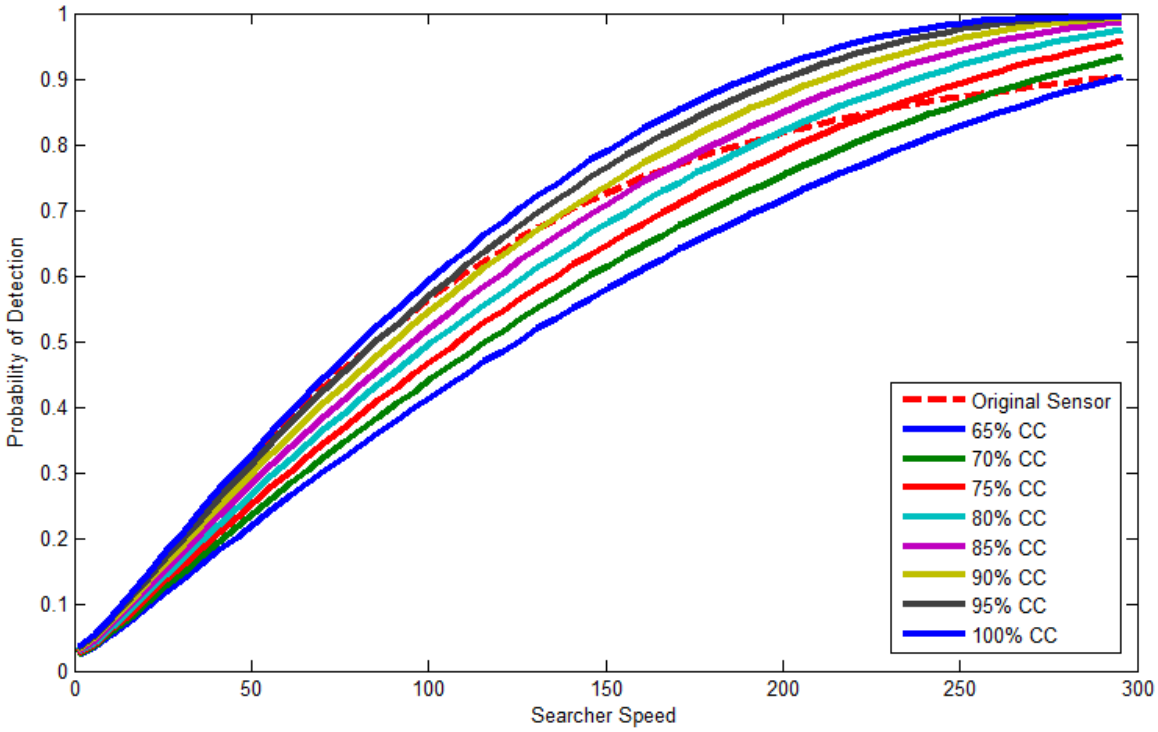
should be 1 within the detection region, which leaves us with the only variable we can adjust in the cookie-cutter approximate model, the detection radius.

We can reduce the detection radius of the approximate sensor model in order to reduce its performance. We can do this by setting the area under the lateral range curve of the approximate cookie-cutter sensor model to some proportion of the area under the curve of the actual sensor model. For example, setting this proportion to 90% reduces the detection radius of the approximate model to 3.24 distance units.

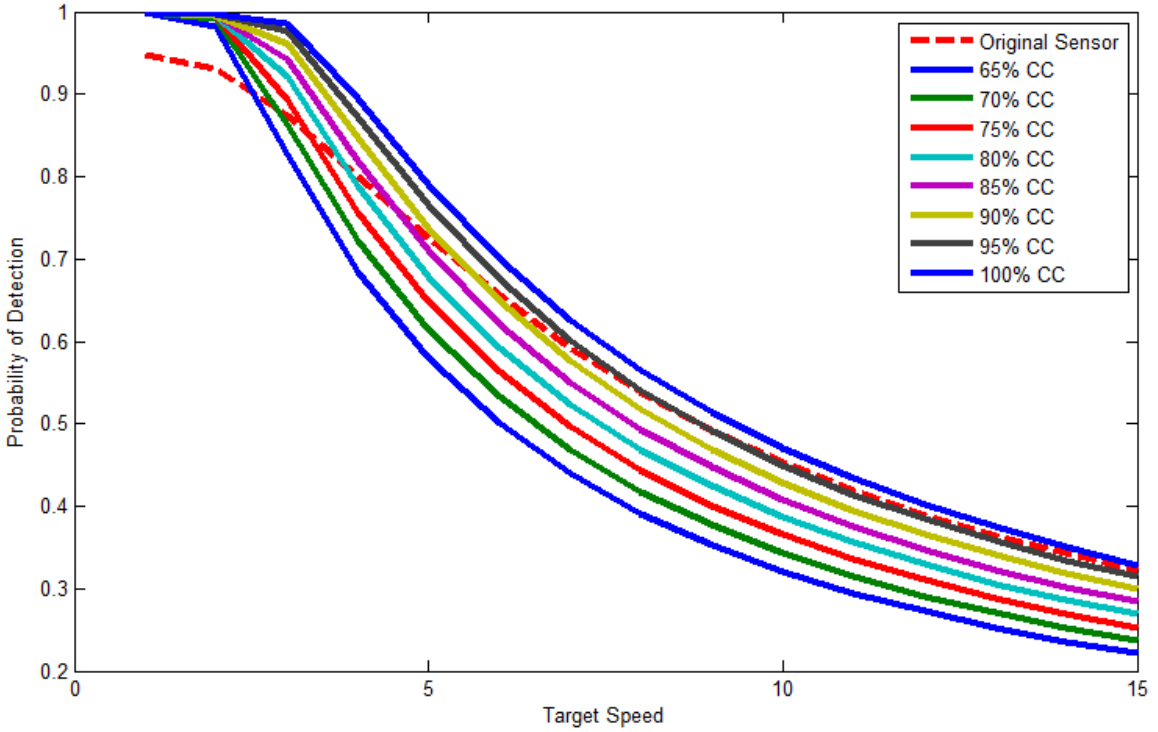
Figure 44 shows the estimated detection probability for the actual sensor model in red dashed line and for various cookie-cutter approximations in solid lines with different colors. We change the proportion of the cookie-cutter approximations as mentioned in the previous paragraph to obtain several cases. We call each approximation with their percentage values. For example, in the case when we set the area under the lateral range curve of the cookie-cutter approximation to 80% of the area under the lateral range curve of the original sensor model, we call it the 80% cookie-cutter model.

Figure 44 shows 65% to 100% cookie-cutter sensor models with 5% increments. The upper and lower curves in Figure 44(a) and Figure 44(b) show 100% and 65% cookie-cutter approximations, respectively. The curves in between them show other percentages in decreasing order from top to bottom.

Figure 44(a) shows the estimated detection probability with respect to searcher speeds between 1 and 300 speed units when we keep the target speed at 5 speed units and border length at 200 distance units. Each approximation seems to be good for some range of searcher speed. For instance, the 100% cookie-cutter sensor model is good when the searcher speed is less than 40 speed units, the 95% model is good between 40 and 110 speed units, and the 90% model is good between 110 and 150 speed units, etc. Since the searcher speed is a variable that a searcher can control, the percentage model that fits well with the speed range of the searcher can be used.



(a)



(b)

Figure 44. Various cookie-cutter approximations.

Figure 44(b) shows the estimated detection probability with respect to target speeds between 1 and 15 speed units, while keeping the border length and searcher speed constant at 200 distance units and 150 speed units, respectively. We observe similar results as we observed in Figure 44(a). Each percentage model has a target speed range for which its estimate is close to the original sensor model's estimate. For example, the 90% cookie cutter model is good when the target speed is between 5 and 7 units, and the 95% model is good when the target speed is greater than 7 speed units, etc.

We can use the results from Figure 44(a) when we are certain that the target speed is 5 speed units. We can choose the best approximation depending on the speed range of the searcher. For example, if the maximum speed of the searcher is 150 speed units, we can use the 95% cookie-cutter approximation. If we would not want to have higher estimates than the original sensor model, we can use the 90% cookie-cutter approximation. If we want to be more accurate, we can use different approximations depending on the speed of the searcher.

Similarly, we can use the results from Figure 44(b) when we are certain that the searcher speed is 150 speed units. This makes sense if we have a sensor whose performance deteriorates with increasing speed like the case we studied in Chapter III.C. In this case we may observe the best detection performance at a certain searcher speed. We can choose the best approximation depending on the speed range of the target.

Although we mostly know and can control the searcher speed, we cannot control and may not be able to know the target speed. In this case, it is more important to use the results from Figure 44(b) and use these results for a range of target speed. It makes more sense to set an upper bound on target speed and try to obtain a good approximation to the actual sensor model. If we assume that the maximum target speed is 5 speed units and try to find the best approximation, we may say that none of the approximations in Figure 44(b) seems to be good.

B. M-BETA SENSOR APPROXIMATION

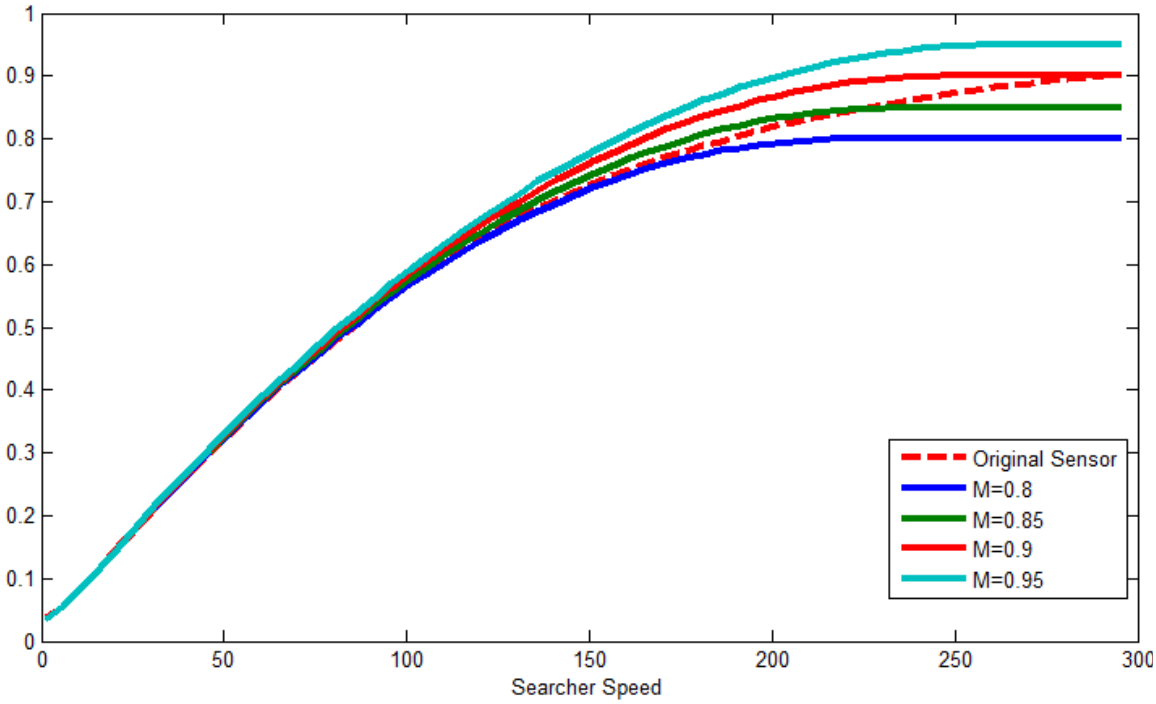
We now study M-Beta approximations to the original sensor model. In Figure 43 we study the M-Beta approximation when M=0.5, and observe that the approximation generally result in very low estimates compared to the original sensor model.

In M-Beta sensor approximations, we can adjust M, which is the cumulative detection probability, to any value between 0 and 1. After choosing M, we can compute β=2R by setting the areas under the lateral range curves of the actual sensor model and M-Beta approximation equal. We can then do simulations with the obtained detection radius.

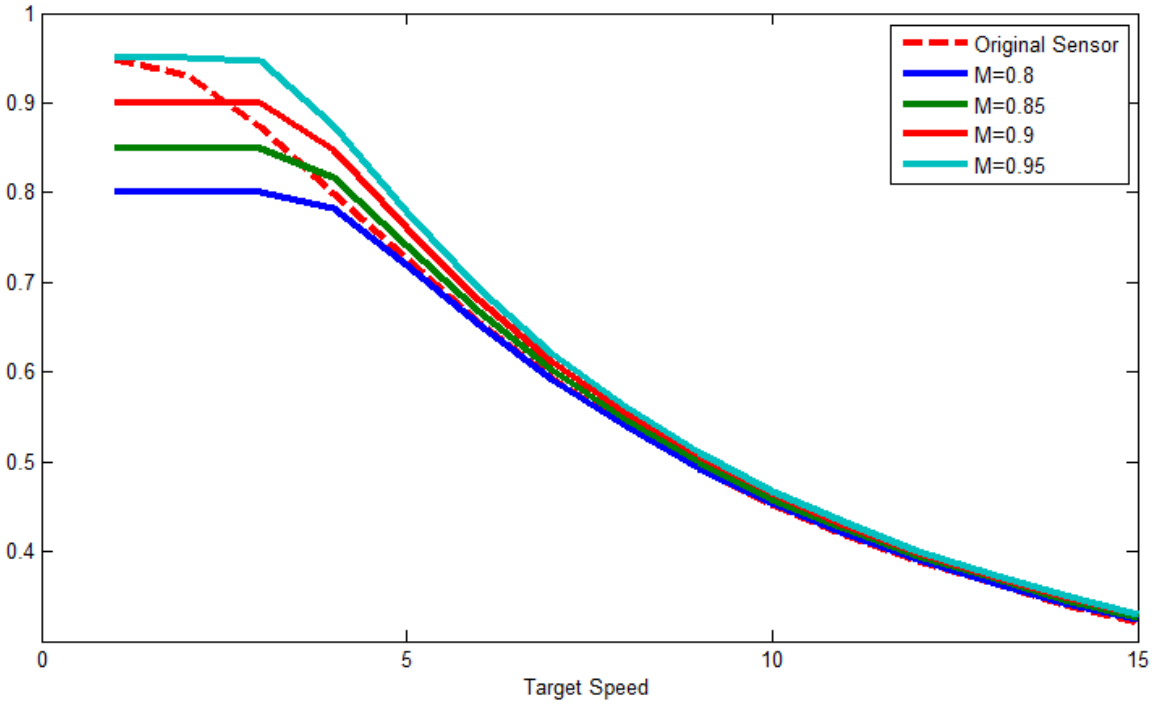
Figure 45 shows the estimated detection probability of the actual sensor model in red dashed lines and various M-Beta approximations in solid lines when we vary M from 0.8 to 0.95 in 0.5 increments. We did not simulate the M=1 case, since it is actually the cookie-cutter approximation. The upper curves in Figure 45 show the detection probability when M=0.95 and we can see the curves for other M values in decreasing order, from top to bottom.

Figure 45(a) shows the estimated detection probability for the M-Beta approximation when we vary the searcher speed up to 300 speed units while keeping the target speed and border length constant at 5 speed units and 200 distance units, respectively. In general, the results seem to be better than in Figure 44(a). This time, the M=0.8 approximation seems to be good for searcher speeds up to 170 speed units.

Similarly, Figure 45(b) shows the estimated detection probabilities for the M-Beta approximation when we vary target speed up to 15 speed units while keeping the searcher speed and border length constant at 150 speed units and 200 distance units, respectively. The results seem to be better than in Figure 44(b). For example, the M=0.8 approximation seems to be good for target speeds greater than 4 speed units. Furthermore, the M=0.85 approximation seems to be acceptable for target speeds lower than 4 speed units. Although there is a larger gap in the estimated detection probabilities for low target speeds when M=0.85, this model is conservative when target speeds are low and relatively accurate for higher target speeds.



(a)



(b)

Figure 45. Various M-Beta approximations.

From Figure 45(a) and Figure 45(b) we can say that the M-Beta approximation with M=0.85 is an acceptable approximation. But in an M-Beta approximation there are two variables, M and β. We can determine the value of M and calculate β by setting the areas under the lateral range curves of the original sensor and M-Beta approximation equal. We now analyze whether we can improve our results by setting the areas under the lateral range curve of the M-Beta approximation to some proportion of the area under the lateral range curve of the original sensor, instead of setting these areas equal. We call each approximation by their percentage and M values, such as, "80% M=0.9 approximation." In the analysis, our goal is to obtain an appropriate approximation which is not highly optimistic over the range of the target speed. We assume that the target speed is less than 5 speed units. We also do not vary the searcher speed as we assume that it can be controlled and optimal detection performance is observed at a particular speed; we assume it is 150 units in our case, and hence, the searcher tries to keep its speed constant at this value.

Figure 46 shows the estimated detection probability with respect to target speed for several percentage M-Beta scenarios. We vary the target speed from 0.1 to 5 speed units in 0.1 speed unit increments in order to study the case when we do not exactly know the target speed but know its maximum speed.

Figure 46(a) shows the percentage scenario when M=0.95. The 60% M=0.95 approximation seems to be the best approximation since it seems to be not highly optimistic over the entire target speed range. It does yield, however, much lower estimates for higher target speeds.

Figure 46(b) shows the percentage scenario when M=0.9. The 85% M=0.9 approximation seems to be the best approximation. Figure 46(c) shows the percentage scenario when M=0.85. The 95% M=0.85 approximation seems to be the best approximation.

Figure 46(d) shows comparison of the best approximations obtained from Figure 46(a) through Figure 46(c). We choose 85% M=0.9 among the three best cases,

since it is not highly optimistic and does not yield very low estimates over the range of target speed.

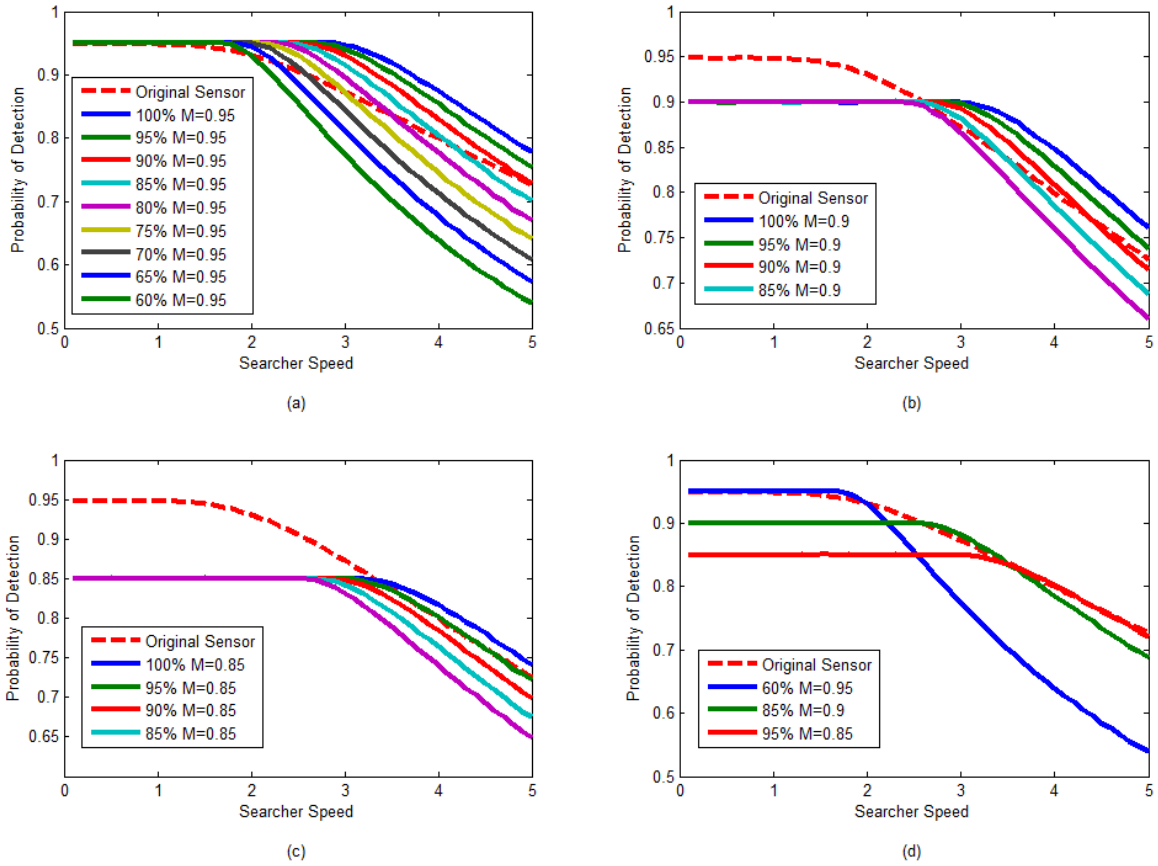


Figure 46. Percentage M-Beta approximations.

In our analysis we note that M-Beta sensor approximations provide better results since these approximations have two parameters that can be controlled. In the analysis, the resolution of the parameters can be reduced to obtain better results. Additionally, the best *M* and its percentage can be chosen by some other means instead of choosing the best by eye.

Although we perform our analysis on a particular sensor, the sensor with its lateral range curve shown in Figure 39, the same procedure may be applied to any sensor. Depending on the sensor model, different approximations may be more appropriate.

VI. CONCLUSION AND FUTURE WORK

We develop both analytical and Monte Carlo simulation models for the simple border patrol problem in which we have a single searcher having a cookie cutter sensor and patrolling over a straight border to detect a single target.

After verifying the models for the simple border patrol problem, we add some complexity to it by adding another searcher. We propose two different searcher paths for the multiple-searcher problem: the disjoint path and the common path. We develop both analytical and Monte Carlo simulation models for the disjoint path problem by building upon the single searcher case. We develop a Monte Carlo simulation model for the common path problem.

When we compare the results of the two multiple-searcher cases, we notice the importance of allocating the border to the two searchers. Therefore we study the optimal allocation, which is the allocation resulting in the maximum detection probability. We study two ways to determine the optimal allocation: by analytical methods and by Monte Carlo simulation. We conclude that we should choose the disjoint path rather than the common path and allocate the border to the searchers optimally to maximize the detection probability.

We analyze the effect of degrading detection performance with increasing searcher speed. We perform our analysis for a single searcher and observe that the maximum detection probability occurs at a certain searcher speed and hence detection radius. We extend this analysis to multiple-searcher problems. Besides analyzing the speeds that result in maximum detection probability, we determine the optimal allocation to maximize the detection probability.

We add complexity to the simple border patrol problem by studying geometric considerations, in which we vary the angles of the border and searcher's path. We develop both analytical and Monte Carlo simulation models for the geometric considerations on the searcher's path. We note that the detection probability is nearly independent of the searcher's path's angle before introducing d, the distance from the

border that a detected object can be classified as a target. After introducing *d*, we note that the detection probability decreases considerably after a certain angle, which depends on *d*.

After noting a similar behavior when we change the angle of the border without varying the searcher's angle, we vary both angles and observe the maximum detection probability when the searcher's path is aligned with the border. We also note that we still have some flexibility in this case, and we have higher flexibility when the angles are lower or when *d* is higher.

We introduce the concept of lateral range curves of imperfect sensors. For this case, we study the single searcher problem in which the searcher has an imperfect sensor. We discuss possible approximations to an imperfect sensor and study their performance on a sample imperfect sensor.

The following can be studied as an immediate future work to this study:

- Study nonlinear searcher paths, as well as searcher paths obtained by combining two separate linear border segments.
- Model line-of-sight (LOS) restrictions or, more generally, model detection probabilities that depend on the target's location.
- Control (and optimize) the speed of the searcher; for example, consider increasing its speed in the regions where it cannot detect targets.
- Instead of single independent targets, model target groups that act together.
- Explore intelligent targets that can observe and react to searcher actions.

APPENDIX A. DETERMINING THE TIME STEP

When we compute the detection probability of the simple border patrol problem in Chapter II.A.1, we calculate the area of coverage of the searcher in target stationary geometry as shown in Figure 4(a). When we perform a time-step simulation, we would like to cover the same area to obtain accurate results. However, due to the nature of time-step simulations, there may be some errors in the simulation.

For example, assume the dotted yellow lines in Figure 47(a) mark the edges of the coverage area of the searcher in target stationary geometry. Any target between these two lines is detected. Assume the position of the searcher at time $t=t_0$ is point A and its position at time $t=t_0+\Delta t$ is point B, as shown in Figure 47(a), where Δt is the time step. In this case, the target, which is shown as a red dot in the figure, will not be detected, even though it is actually within the coverage area of the searcher.

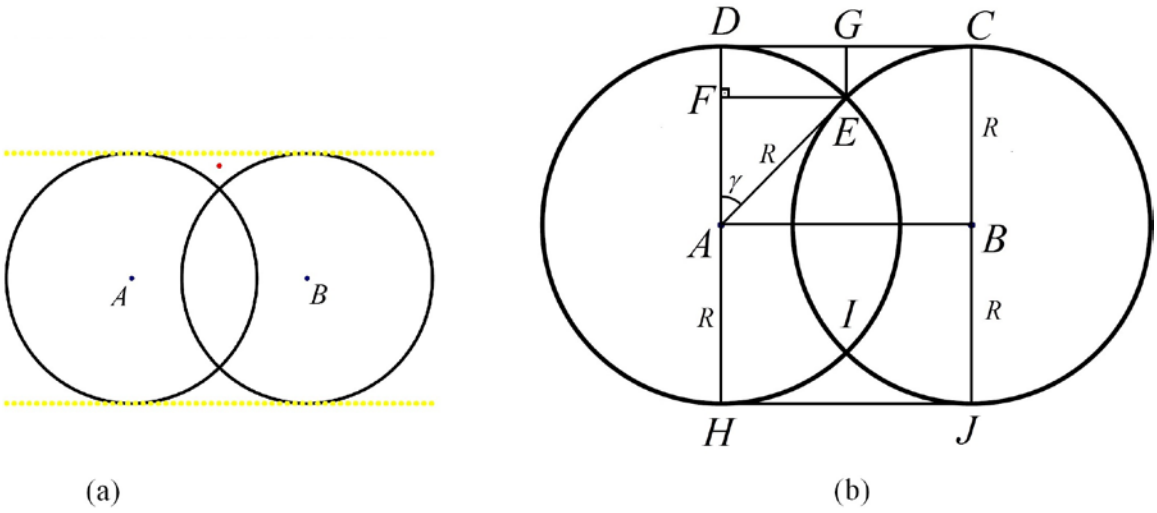


Figure 47. Error in time-step simulations.

In general, the targets that fall in the regions CDE and HIJ, which are shown in Figure 47(b), will not be detected in the time-step simulation, causing errors in the simulation results. We can compute the percentage error caused by the time-step

simulation by dividing the area of region CDE to the area of rectangle ABCD by symmetry.

In Figure 47(b) we have:

$$\begin{aligned} AF &= R\cos\gamma \\ DG &= FE = R\sin\gamma \\ DF &= GE = R - R\cos\gamma. \end{aligned} \tag{24}$$

The area of the rectangle ABCD is:

$$DC \times AD = 2 \times DG \times AD = 2 \times R\sin\gamma \times R = 2R^2 \sin\gamma \tag{25}$$

The area of the region CDE is:

$$\begin{aligned} &2 \times DEG = 2\left(DFEG - DEF\right) = 2\left(DG \times DF - \left(AED - AEF\right)\right) \\ &= 2\left(R\sin\gamma\left(R - R\cos\gamma\right) - \left(\pi R^2 \frac{\gamma}{2\pi} - \frac{AF \times FE}{2}\right)\right) \\ &= 2\left(R\sin\gamma\left(R - R\cos\gamma\right) - \left(R^2 \frac{\gamma}{2} - \frac{R\cos\gamma \times R\sin\gamma}{2}\right)\right) \\ &= 2\left(R^2\sin\gamma - R^2\sin\gamma\cos\gamma - \left(\frac{R^2\gamma}{2} - \frac{R^2\cos\gamma\sin\gamma}{2}\right)\right) \\ &= 2\left(R^2\sin\gamma - R^2\sin\gamma\cos\gamma - \frac{R^2\gamma}{2} + \frac{R^2\cos\gamma\sin\gamma}{2}\right) \\ &= 2\left(R^2\sin\gamma - \frac{R^2\cos\gamma\sin\gamma}{2} - \frac{R^2\gamma}{2}\right) = 2R^2\sin\gamma - R^2\cos\gamma\sin\gamma - R^2\gamma \\ &= R^2\left(2\sin\gamma - \cos\gamma\sin\gamma - \gamma\right) \end{aligned} \tag{26}$$

We can compute the percentage error by using Equations (25) and (26) as:

$$\frac{R^2\left(2\sin\gamma - \cos\gamma\sin\gamma - \gamma\right)}{2R^2\sin\gamma} = 1 - \frac{\cos\gamma}{2} - \frac{\gamma}{2\sin\gamma} \tag{27}$$

which is a function solely of γ.

Figure 48 shows the percentage error with respect to γ. From the figure we can say that γ smaller than 3 degrees produces satisfactory results.

In order to obtain a relationship between the percentage error and the time step, we first need to find the relationship between γ and the time step. For this case we can use the fact that in a time step, the searcher travels $\sqrt{v^2+u^2}\times\Delta t$ units in target stationary geometry, which is equal to the line segment DC in Figure 47(b). By using this we can find the relationship between γ and the time step as:

$$DC=\sqrt{v^2+u^2}\times\Delta t \Rightarrow 2\times DG=\sqrt{v^2+u^2}\times\Delta t \Rightarrow 2R\sin\gamma=\sqrt{v^2+u^2}\times\Delta t$$
$$\Rightarrow \sin\gamma=\frac{\sqrt{v^2+u^2}\times\Delta t}{2R} \Rightarrow \gamma=\arcsin\left(\frac{\sqrt{v^2+u^2}\times\Delta t}{2R}\right). \quad (28)$$

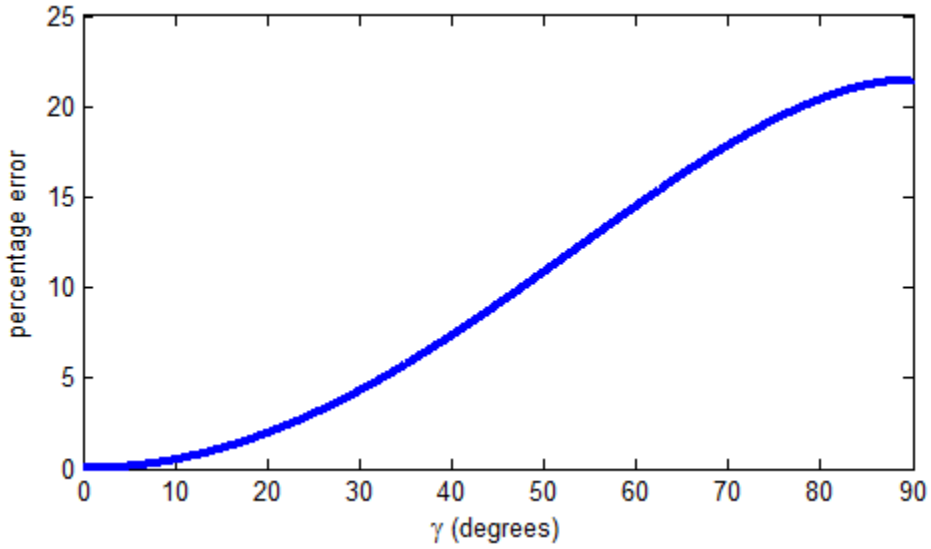


Figure 48. Percentage error with respect to γ.

By choosing the time step as a function of R, u, and v we can make the percentage error independent of them. For this reason, assume we select the time step as $\Delta t=k\frac{R}{\sqrt{v^2+u^2}}$, where k is a constant to be determined to obtain low enough percentage error. In this case we have:

$$\gamma = \arcsin\left(\frac{\sqrt{v^2+u^2}\times\Delta t}{2R}\right) = \arcsin\left(\frac{\sqrt{v^2+u^2}\times k\dfrac{R}{\sqrt{v^2+u^2}}}{2R}\right) = \arcsin\left(\frac{k}{2}\right) \qquad (29)$$

$$\Rightarrow k = 2\sin\gamma.$$

If γ=3 degrees, we have k=9.6. So if we select k>9.6 we would expect to have satisfactory results. But since we want to compare our time-step simulation results with other methods in the thesis, we selected k=25. In this way we reduced the percentage error down to $6.7\text{x}10^{-3}$ percent.

Increasing the time step reduces the runtime but increases the percentage error; likewise, decreasing it reduces the percentage error but increases the runtime. The time step should be selected in order to balance accuracy and runtime.

APPENDIX B. CALCULATING THE MULTIPLIER TO SET A LIMIT IN THE SIMULATION END TIME IN DISJOINT PATH PROBLEM

In the Monte Carlo simulation of the single searcher problem, it is easy to set a simulation end time ($t_{\max}$) since we know the time it takes for the searcher to make a complete cycle, i.e., $t_{\max} = \frac{(L-2R)}{v}$. The problem gets complicated when there are two searchers. For example, in the disjoint path problem the searchers complete their cycles mostly in different durations. We can take the lowest common multiple of these two cycle times and set it as the simulation end time; however, depending on the cycle times, this lowest common multiple may be a very large number, resulting in an unnecessary increase in the simulation end time.

Initially, we made the simulation end time the maximum of the cycle times in the disjoint path problem. Figure 49(a) shows the probability of detection with respect to the allocation to the first searcher when the number of replications is one million, the border length is 200 distance units, and the other parameters are as stated on the figure. The purple line shows the estimated value of the probability of detection ($\widehat{P_d}$) along with its 95% confidence interval. Due to the large number of replications, the confidence interval is so narrow that it cannot be seen in the figure. The dashed blue line shows the probability of detection (P_d) obtained from the analytical solution in Equation (9). We see that the Monte Carlo simulation results differ from the analytical results at some regions. Since the confidence interval is very narrow, the differences are not random; it follows a pattern caused probably by a flaw in the simulation runtime.

In order to obtain satisfactory results, we decided to introduce a simulation end time multiplier, *k*, and use it to determine the simulation end time by multiplying it with the maximum cycle time, i.e., $t_{\max} = k\max(t_{\text{cycle1}}, t_{\text{cycle2}})$. We use the largest gap in Figure 49(a), when the allocation to the first searcher is 49%, and varied k in order to obtain Figure 49(b). Figure 49(b) shows the probability of detection with respect to the simulation end time multiplier in the disjoint path problem when the allocation to the first

searcher is 49% percent and all other variables are the same as the ones used to obtain Figure 49(a). The thin green line shows the probability of detection obtained analytically, and the solid thick blue line shows the $\widehat{P_d}$ obtained from the Monte Carlo simulation, with its 95% confidence interval shown in dotted red lines. We use one million replications for the Monte Carlo simulation.

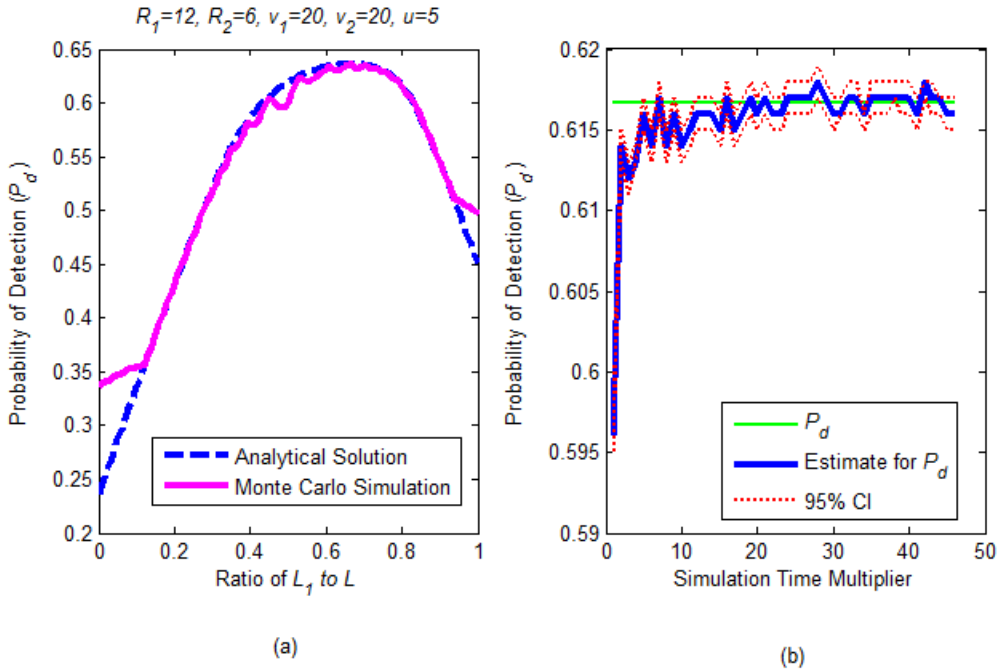


Figure 49. Study on simulation end time multiplier.

In Figure 49(b) we see that the Monte Carlo simulation results improve considerably, even when the multiplier is only 2. We also see that having a multiplier of 4 is good enough, but we decided to have a multiplier of 25 since after that multiplier the Monte Carlo simulation and analytical solutions are very close, and we may need to have higher multiplier for different problems.

Figure 12 shows the results of Figure 49(a) when we choose k to be 25. We can easily see the improvement and say that choosing k to be 25 provides satisfactory results.

APPENDIX C. CONCAVITY

The aim of this appendix is to present experimental evidence that the function to be maximized in

$$\max_{L1} \quad f(L_1) = P_d(L_1, R_1, u, v_1)\frac{L_1}{L} + P_d(\mathrm{L} - L_1, R_2, u, v_2)\frac{L - L_1}{L} \qquad (30)$$
$$st \quad 0 \le L_1 \le L$$

is concave.

A function $f(x)$ is concave if it satisfies the following condition (Boyd & Vandenberghe, 2004):

$$f(\lambda\bar{x} + (1-\lambda)x') \ge \lambda f(\bar{x}) + (1-\lambda) f(x') \qquad \forall\, \bar{x}, x' \in X,\ \forall\ \lambda \in [0,1]. \qquad (31)$$

What this condition briefly implies is shown in Figure 50. First we pick two values for $\bar{x}$ and x' that x can take, and for all $\lambda \in [0,1]$ Equation (31) should be satisfied. This condition should hold for all values $\bar{x}$ and x' can take.

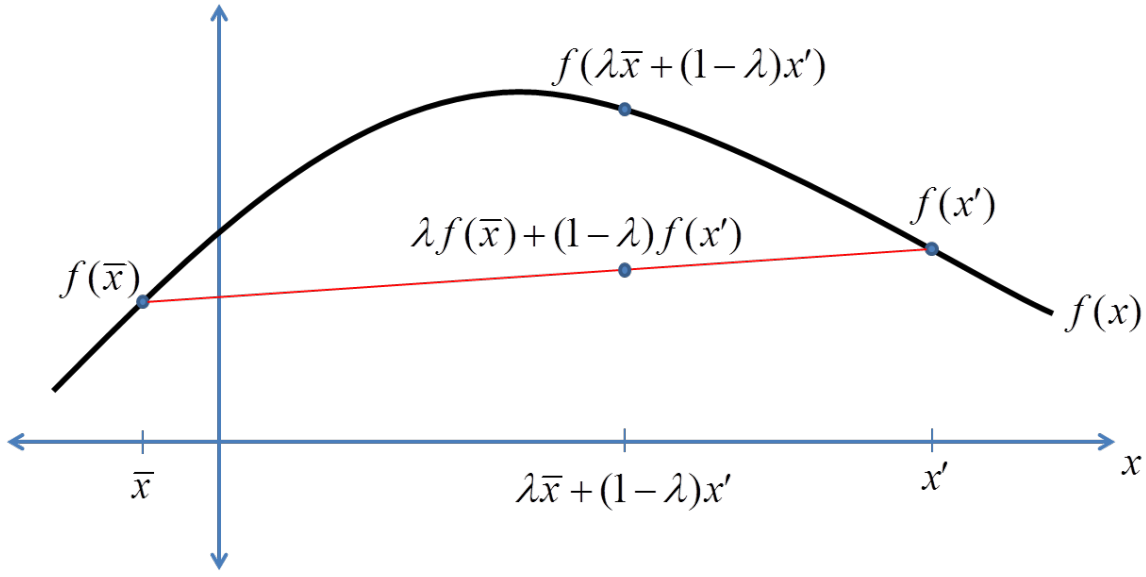


Figure 50. Concavity.

In our problem shown in Equation (30) we have $f(L_1)$ instead of $f(x)$, where $0 \leq L_1 \leq L$. We try to show that the function $f(L_1)$ is concave by showing:

$$f(\lambda \overline{L_1} + (1-\lambda)L_1') \geq \lambda f(\overline{L_1}) + (1-\lambda) f(L_1') \qquad \forall\ 0 \leq \overline{L_1}, L_1' \leq L,\ \forall\ \lambda \in [0,1] \quad (32)$$

while keeping L, R_1, R_2, u, v_1, and v_2 constant. Moreover, we want to show that the function in Equation (30) is concave over a wide range of L, R_1, R_2, u, v_1, and v_2. For this reason we vary L from 10 to 1000 distance units with 10 unit increments, R_1, and R_2, from 0.2 to 5 distance units with 0.2 unit increments, u from 0.2 to 10 speed units with 0.2 unit increments, and finally, v_1, and v_2 from 3 to 150 speed units with 3 unit increments.

For each combination of L, R_1, R_2, u, v_1, and v_2 as described, we check if Equation (32) holds by choosing $\overline{L_1}$ (from 0 to L with L/20 increments) and L_1' (from $\overline{L_1}$ + L/20 to L with L/20 increments) and varying alpha from 0.05 to 0.95 in 0.05 increments.

In our experiment, we omit the cases when R_2 is less than R_1, since we also have similar combinations when R_2 is greater than R_1. We also omit the cases when $\overline{L_1} = L_1'$, α=0, and α=1 since these result in equality of both sides in Equation (32), which does not violate the conditions. In this way, we reduce the runtime of the experiment considerably.

After all iterations, we were unable to find a counterexample showing that the function in Equation (30) is non-concave, although such a counterexample may exist.

In the experiments, we use the High Performance Computing (HPC) network at the Naval Postgraduate School. If we used a home computer, this experiment would have taken around 10 years. By using the HPC network, we manage to do it in seven days.

APPENDIX D. CALCULATING CLOSEST POINT OF APPROACH

In this appendix we briefly describe how we calculate the closest point of approach for the single searcher, single target problem. The closest point of approach is the minimum distance to be observed between the target and the searcher in a particular problem. In order to determine this quantity, we use target stationary geometry as described in Chapter II.A.1.

Figure 51 shows target stationary geometry in a particular problem. The target, the big red dot at point T, stays stationary and the searcher follows the green dashed dotted lines to detect the target. The turning points of the searcher are marked with red dotted lines. These lines divide the problem into sections with length *w*. By symmetry, we can say that the closest point of approach is observed in the section where the target stays. For this reason, we only need to study the region in which the target is observed.

In a particular problem, we can have two different sections depending on whether the searcher is moving to the right or to the left in that section. In order to determine which section the target is in, we take the mod of the target's vertical position with respect to the width of the sections, *w*. If the result is an even number, the target is in the section where the searcher is moving to the right and vice versa.

After determining which kind of section the target is in, we only work on this section. Here we only show the procedure on one type of section, but we can easily perform a similar procedure on the other section by the same logic.

Figure 51(b) shows the section in which the target in Figure 51(a) is observed. In this case we calculate the closest point of approach by finding the distance between the target's position (T) and the line passing through the searcher's path (AB). This may not be accurate if the target is close to the edges. We may obtain a lower value for the closest point of approach in this case. In order to obtain accurate results, we need to find the distance between the target and the line segment AB, not the line passing through AB.

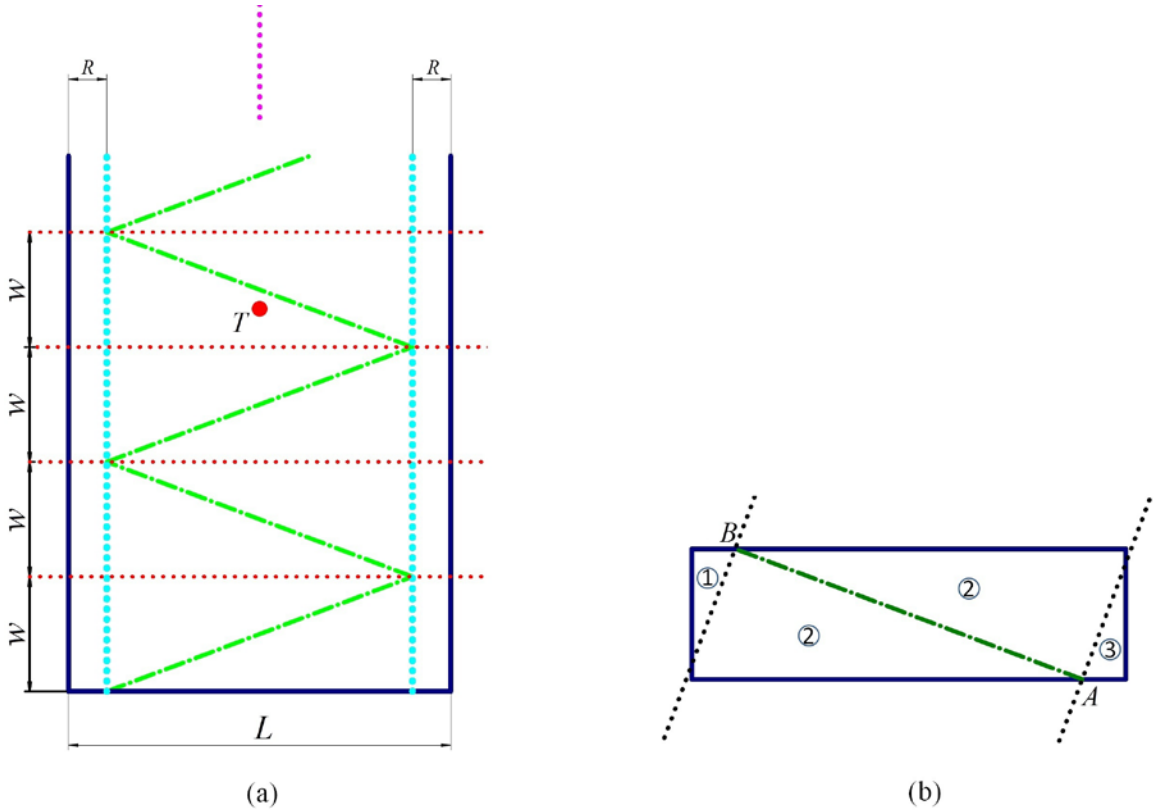


Figure 51. Target stationary geometry in calculating closest point of approach.

For this reason we divide the section into three regions as shown in Figure 51(b). The dotted lines are perpendicular to the searcher's path. These lines divide the section into three regions. After determining which region the target is in, if the target is in region 1 or 3, we calculate the distance between the target and the point B or A, respectively. Moreover, if the target is in region 2, we calculate the distance between the target and the line passing through AB. By this way we obtain the closest point of approach.

APPENDIX E. TURNING DISTANCE CALCULATION

In Chapter II.A.3, we study the turning distance for the simple border patrol problem. In this problem we have a single searcher with a cookie-cutter sensor. In this appendix, we study the turning distance when the same searcher has the lateral range curve as shown in Figure 39.

Like the analysis in Chapter II.A.3, we consider ten different scenarios defined by the border length. We begin with a border length of 100 units and incrementally increase our border length by 100 units until we reach a length of 1000 units. In doing so, we are able to see the effect of the turning distance for several values of probability of detection.

In all ten scenarios, we fix the detection radius R at 6 units, the target's speed u at 5 speed units, and the searcher's speed v at 100 speed units. We vary the turning distance from 0 to 12 units in 0.06 unit increments. We perform the analysis by running Monte Carlo simulations with one billion replications for each scenario.

Figure 52 shows the results of the turning point analysis. Each subfigure shows one of the scenarios, with the corresponding barrier length stated on each subfigure. In each subfigure, the horizontal axis shows the turning distance, and the vertical axis shows the estimated probability of detection $\widehat{P_d}$ in blue straight line, along with its 95% confidence interval in red dotted line.

We notice that when border length is low, i.e., when the max $\widehat{P_d}$ observed is high (Figure 52(a) through Figure 52(f)), $\widehat{P_d}$ stays constant up to some turning distance and then it starts to decrease considerably. For example, when the border length is 20 distance units, and if we set the turning distance to R=6 distance units, we lose around 22% (from 0.95 to 0.74) of maximum $\widehat{P_d}$ that can be observed.

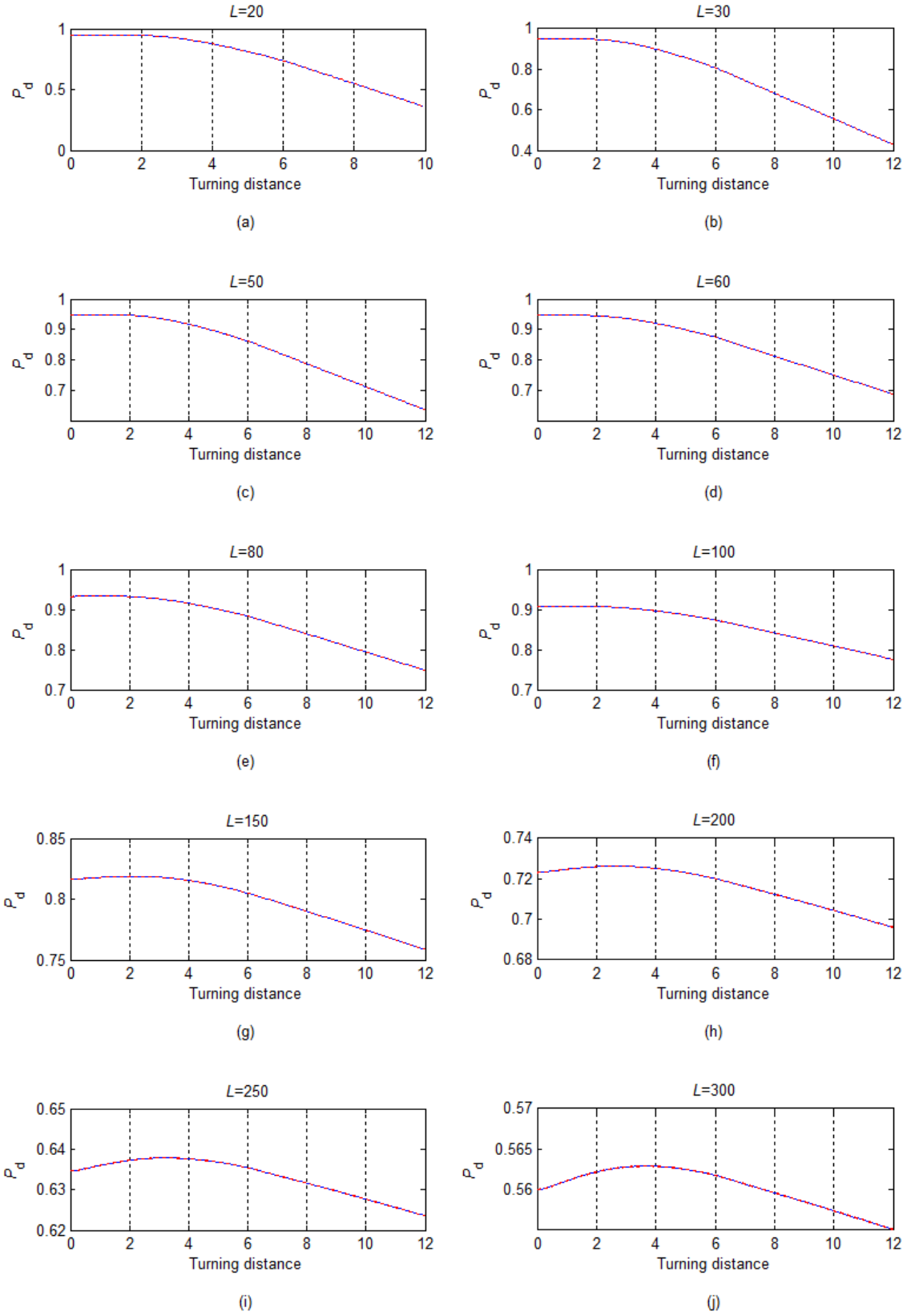


Figure 52. Turning distance analysis on imperfect sensors.

In Figure 52(g) through Figure 52(j), we note that the optimal turning distance that results in the maximum $\widehat{P_d}$ to be observed occurs between 0 and *R*. This optimal point changes depending on the problem. Furthermore, in these cases, we do not lose much of the maximum Figure 52 $\widehat{P_d}$ if we do not adjust the turning distance properly. For example in Figure 52(g), we lose about 4% of the maximum $\widehat{P_d}$ if we incorrectly set the turning distance to *R* units. In Figure 52(j), we lose nearly 5% of the maximum $\widehat{P_d}$ when we incorrectly set the turning distance to 0 or *R*.

We observe the maximum $\widehat{P_d}$ at different turning distances depending on the problem. In all the scenarios studied, we can lose much of the detection probability by incorrectly setting the turning point higher than the optimal. However, we either observe the maximum $\widehat{P_d}$ or $\widehat{P_d}$ that is close to the maximum when we set the turning distance to 0.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization.* New York: Cambridge University Press.

Eagle, J. (2013, September 27). *Search and detection theory*. Lecture notes, course OA3602, Naval Postgraduate School, Monterey, CA.

Haddah, C. C., & Gertler, J. (2010). *Homeland security: Unmanned aerial vehicles and border surveillance* (CRS Report No. RS21698). Washington, DC: Library of Congress, Congressional Research Service. Retrieved from http://www.fas.org/sgp/crs/homesec/RS21698.pdf.

Ozcan, B. Y. (2013, June). *Effectiveness of unmanned aerial vehicles in helping secure a border characterized by rough terrain and active terrorists*. Master's thesis, Naval Postgraduate School, Monterey, CA.

Soza & Company. (1996). *The theory of search: A simplified explanation*. (96-F-HNG040). Washington, D.C.: U.S. Coast Guard Office of Search and Rescue. Retrieved from http://www.uscg.gov/pdf/Theory_of_Search.pdf

Wagner, D. H., Mylander, W. C., & Sanders, T. J. (Eds.) (1999). *Naval operations analysis.* Annapolis: Naval Institute Press.

Washburn, A. R. (2002). *Search and detection.* Linthicum, MD: INFORMS.

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