

A particle filter approach to estimating target location using Brownian bridges

Jesse A. Nunez, Dashi I. Singham & Michael P. Atkinson

To cite this article: Jesse A. Nunez, Dashi I. Singham & Michael P. Atkinson (2020) A particle filter approach to estimating target location using Brownian bridges, Journal of the Operational Research Society, 71:4, 589-605, DOI: [10.1080/01605682.2019.1570806](https://doi.org/10.1080/01605682.2019.1570806)

To link to this article: <https://doi.org/10.1080/01605682.2019.1570806>



Published online: 26 Apr 2019.



Submit your article to this journal [↗](#)



Article views: 113



View related articles [↗](#)





View Crossmark data [↗](#)



Citing articles: 2 View citing articles [↗](#)

A particle filter approach to estimating target location using Brownian bridges

Jesse A. Nunez, Dashi I. Singham  and Michael P. Atkinson 

Operations Research Department, Naval Postgraduate School, Monterey, CA, USA

ABSTRACT

We study the problem of modelling the trajectory of a moving object of interest, or target, given limited locational and temporal information. Because of uncertainty in information, the location of the target can be represented using a spatial distribution, or heatmap. This paper proposes a comprehensive method for constructing and updating probability heatmaps for the location of a moving object based on uncertain information. This method uses Brownian bridges to model and construct temporal probability heatmaps of target movement, and employs a particle filter to update the heatmap as new intelligence arrives. This approach allows for more complexity than simple deterministic motion models, and is computationally easier to implement than detailed models for local target movement.

ARTICLE HISTORY

Received 4 April 2018
Accepted 11 January 2019

KEYWORDS

Simulation; military;
stochastic processes;
Brownian bridge;
particle filter

1. Introduction

Searching for a target of interest moving through an area of operations is a fundamental problem that falls within the field of search theory. The field of search theory is a prime example of decision-making under uncertainty. For an overview of search and detection, see Washburn (2002). A challenging type of search relevant to our work involves a mobile target (see Brown, 1980; Pietz & Royset, 2015; Stone, Royset, & Washburn, 2016). Search models with mobile targets must specify the behaviour of the targets and their movements and actions (Ben-Zvi & Nickerson, 2012; Eagle & Yee, 1990; Stone & Kadane, 1981; Thomas & Hulme, 1997; Vermeulen & van den Brink, 2005). Given the specified movement dynamics of a target, the searcher must decide how to best find the target in response, by deciding when and where to allocate sensors to detect the target (Ben-Zvi, 2017).

This paper does not formulate prescriptive models to allocate search resources, but rather provides a complementary analysis to traditional search methods by providing the key input to many search models: the distribution for a target's location over time. We focus on building a comprehensive model for the uncertainty in a target's continuous movement through an area of interest of arbitrary size, given initial information about its location and updates from observing portions of the area. Having a realistic stochastic model for the target's behaviour based on whatever information is available is critical to a successful search effort. Intelligence about a

target's location and movement patterns comes from many different sources, such as sonar, radar, communication and human intelligence. The intelligence is imperfect (e.g., can have false positive and false negative errors), which provides a challenge in effectively fusing together the information from different sources. In this paper, we model how the distribution of the target's location changes over time based on uncertain intelligence. We develop a new method to produce spatio-temporal heatmaps that display the uncertainty in a target's location. The heatmap updates when new intelligence arrives from sensors designed to look for the target.

There are two parts to our approach. The first part exploits the path-driven motion of targets to construct an analytical model for the spatial distribution based on prior work modelling the paths of migrating animals as Brownian bridges. A Brownian bridge is Brownian motion with fixed values at two endpoints. We model target movement between waypoints as Brownian bridges, construct heatmaps based on the Gaussian distribution, and derive values of interest using boundary-crossing results of Brownian bridges. The second part extends the analytical model to a simulation approach, which allows us to relax assumptions and add complexity to target behaviour. We apply a particle filter method to the Brownian bridge model that updates the heatmap as new intelligence arrives. The objective is to produce heatmaps for the target's location over time that could later be used to determine where to search for the target, though we do not consider the search optimisation problem here.

We now present past methods for modelling a target's distribution. Moskowitz and Simmen (1989) develop an analytical model to estimate a target's distribution using an initial probability distribution and a constant but sampled velocity. The authors calculate containment regions using level sets to determine areas that are likely to contain the target with a given probability. Miller and Moskowitz (1996) extend this work to obtain closed-form solutions in certain cases. Sklar and Ladany (1993) use angles of observation to estimate target location, and incorporate error in the observer's location and measurement instrument. More generally, the idea of modelling location data as spatial distributions has been developed in Angermann, Kammann, Robertson, Steingäß, and Strang (2001) and Turchin (1991). This paper models spatial distributions using heatmaps aggregated from individual path trajectories. Jian, Matsuka, and Nickerson (2006) use paths generated by experimental subjects drawing trajectories while attempting to avoid detection and obstacles. These trajectories are aggregated into heatmaps to estimate the frequency of a trajectory being in a certain area.

Search models also make assumptions about how targets move through the area. The simplest variants assume stationary targets who hide within one of N distinct locations, as in Kadane (1971) and Chew (1973). When the target is mobile, the models often assume the target travels in a straight line (Baumgartner, Ferrari, & Wettergren, 2009; Le Cadre & Souris, 2000; Mooshegian, 2013), possibly with uncertainty in the starting and ending points (Pietz & Royset, 2013; Stone & Kadane, 1981) or moving boxes that represent spatial uncertainty (Pietz & Royset, 2015). Lersteau, Rossi, and Sevaux (2016) assume a known target path, with uncertainty in the timing of the path traversal; we include both temporal and path uncertainty. Network models define target dynamics across a discrete set of nodes (Brown, Carlyle, Abdul-Ghaffar, & Kline, 2011; Przybyla, Taylor, & Zhou, 2010). Similarly, the target may move on a lattice according to a Markov transition function (Eagle & Yee, 1990, Gentil, Rémillard, & Del Moral, 2005; Royset & Sato, 2010) or Bayesian methods (Simonin, Le Cadre, & Dambreville, 2009). Ben-Zvi and Nickerson (2012) deliver an optimal strategy for deciding whether to interdict a target that could be hostile using a partially observable Markov decision process model over a series of discrete locations.

1.1. Proposed model

We present a method for modelling the distributional uncertainty in target movement over time

using heatmaps and a systematic way of updating those heatmaps as new intelligence arrives. Our model is motivated by the Brownian bridge movement model (BBMM), originally introduced by Bullard (1999) to represent animal movement patterns. Our two main contributions are 1) extending this method to develop spatio-temporal heatmaps, and 2) using particle filtering to update these heatmaps over time. The BBMM assumes Brownian bridge motion between a start and end point, and estimates the distribution of an animal's location based on properties of the Gaussian distribution. The BBMM has been used to estimate home ranges and migration routes (Horne, Garton, Krone, & Lewis, 2007), and has been suggested as a useful method for analysing low resolution trajectory data because it "assumes random movement between sample points" (Buchin, Sijben, Arseneau, & Willems, 2012). The idea is that knowledge of an animal's location cannot be known exactly at all times, but may be known at specified times. The Brownian bridge model allows for this known information to be incorporated as endpoints, with uncertainty represented in the trajectory between the endpoints. While Brownian motion may not be a perfect representation of the trajectory of the animal (or in our case, the target), the Brownian bridge models the relative uncertainty in location. Brownian bridges have higher uncertainty in the middle, with reduced uncertainty towards the endpoints which are fixed.

Thus, we propose using the Brownian bridge as a model for the uncertainty in target movement between two points where the location is known. A key difference between our model and the prior BBMM is that the ecology literature focuses on spatial distributions aggregated over longer time ranges, whereas we analyse the evolution of spatio-temporal distributions as the target moves through the area. Thus, while the underlying assumption on target movement is the same as in the original BBMM, our probability heatmap construction is entirely different because we model the target's distribution heatmap over time, instead of trying to normalise over all time to determine long-term areas of high activity. This modified model is referred to as the Temporal Brownian Bridge Model (TBBM). Our model development also offers new analytical metrics based on the mathematics of Brownian bridges, and develops a method of updating the heatmap estimate based on sensor information captured mid-trajectory.

The Brownian bridge model lies between two approaches for modelling target movement and generating heatmaps: the straight line methods and discretised grid/network methods mentioned in the

earlier literature review. Straight line movement assumptions that completely specify target travel are often unrealistic and do not account for complexity or uncertainty in target movement. The Brownian bridge model better captures uncertainty when location information is unavailable while allowing intelligence to be incorporated. Models that require knowledge of how a target moves over a discretised grid or network require information for calibration for each node, and detailed information may not be available.

Brownian bridges define movement only between points of known intelligence with minimal assumptions, and do not require detailed modelling of local target behaviour, which can propagate modelling errors and be computationally costly. Additionally, the Brownian bridge model allows for some analytical tractability if desired, and it is easy to simulate quickly relative to models that require parametrisation and updating over a discretised search space. The appeal of our approach is that it can incorporate intelligence information only when it is available, and does not require calibration for times when it is not available. The TBBM requires at a minimum some uncertain estimate of the starting and ending location of the target, but does not require knowledge of the trajectory. While this information is minimal, there may be cases where the target's desired endpoint is not known. For example, a straight-line model (e.g., Miller & Moskowitz, 1996; Moskowitz & Simmen, 1989) may accurately represent a target fleeing after it executes an attack, and a grid-based model (e.g., Eagle & Yee, 1990) may be more appropriate for a target loitering in an area.

Aside from developing a new method for generating spatio-temporal heatmaps for target movement, we develop a particle filter method for updating the heatmap with new intelligence as it arrives. The TBBM model proposed in Cheng (2016) and Nunez (2017) is uniquely designed to add model flexibility and incorporate intelligence updates, unlike the original BBMM model. We simulate Brownian bridge paths for target movement and treat those simulated paths as particles in a particle filter algorithm. The weights on each particle update after the arrival of new intelligence. The particles are aggregated according to their weights to form heatmaps. Numerous applications implement particle filter techniques for tracking targets, such as car and aircraft positioning and collision avoidance (Gustafsson, 2010; Gustafsson et al., 2002).

This paper is organised as follows. Section 2 describes the analytical model that is used to compute the probability the target lies in certain regions. Section 3 describes the simulation model and

particle filter implementation to update the heatmap. Section 4 presents experimental results for different implementations of the particle filter. Finally, Section 5 concludes the paper.

2. A Brownian bridge model for target movement

First, we discuss some theoretical results that can be derived by considering temporal heatmaps constructed using Brownian bridge movement between two points. A Brownian bridge is a stochastic process that is Brownian motion tied to particular spatial values at two time points (see 5.6B in Karatzas and Shreve (1998)). We consider two-dimensional Brownian bridges that consist of two-dimensional Brownian motion constrained to pass through two particular points in the xy -plane.

The target starts its journey at departure point (x_d, y_d) at time t_d , and will reach its arrival point (x_a, y_a) at time t_a . In many applications the starting and ending locations and times will not be known with certainty and we consider this case in Sections 2.1 and 3, but for now, the randomness in the model is the path taken by the target between the known start and end points. While we assume that in between these times the target moves according to a Brownian bridge, the value of the model comes from the heatmap, which is the result of aggregating over all possible Brownian bridges to represent uncertainty. Departure and arrival information represents the most basic form of intelligence that might be available, and the Brownian bridge model represents uncertainty in the target's behaviour during the times when its location is unknown between the start and endpoints. Modelling the target's movement as a Brownian bridge allows for continuous paths without assuming a particular path (e.g., a straight line path). While a Brownian bridge may not perfectly represent the target's true dynamics, it approximates the uncertainty in the target's location and produces heatmaps to inform search decisions.

If the target departs from location (x_d, y_d) at time t_d and arrives at location (x_a, y_a) at time t_a , its location at time $t \in [t_d, t_a]$ $(x(t), y(t))$ evolves according to the following dynamics:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_d + (x_a - x_d) \frac{t - t_d}{t_a - t_d} \\ y_d + (y_a - y_d) \frac{t - t_d}{t_a - t_d} \end{pmatrix} + K \begin{pmatrix} W_1(t - t_d) \\ W_2(t - t_d) \end{pmatrix} - K \frac{t - t_d}{t_a - t_d} \begin{pmatrix} W_1(t_a - t_d) \\ W_2(t_a - t_d) \end{pmatrix}, \tag{1}$$

where $W_1(\cdot)$ and $W_2(\cdot)$ are independent standard Brownian motions over $t \in [0, t_a - t_d]$ and K is a

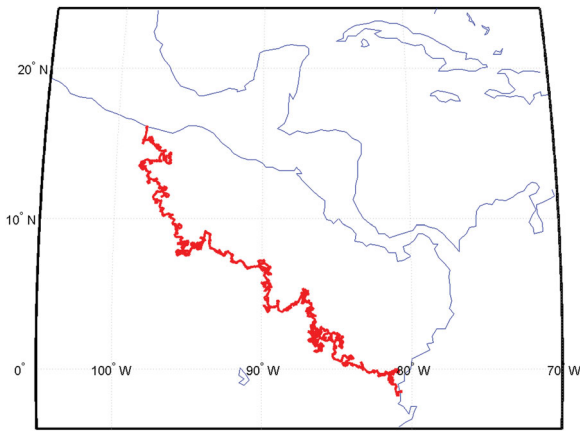


Figure 1. One realisation of a Brownian bridge path.

scalar parameter that allows us to tune the variance. This formulation assumes random fluctuations in the x and y direction are independent. Incorporating dependence requires more notation and bookkeeping without providing additional insight.

For any time $t \in [t_d, t_a]$ the expected location of the target is a fraction $\frac{t-t_d}{t_a-t_d}$ of the way along the centre line that connects (x_d, y_d) and (x_a, y_a) :

$$\mu_t = \begin{pmatrix} \mu_{x_t} \\ \mu_{y_t} \end{pmatrix} = \begin{pmatrix} x_d + (x_a - x_d) \frac{t - t_d}{t_a - t_d} \\ y_d + (y_a - y_d) \frac{t - t_d}{t_a - t_d} \end{pmatrix}. \quad (2)$$

The covariance matrix of the process $(x(t), y(t))$ is $\sigma_t^2 I$, where

$$\sigma_t^2 = K^2 \frac{(t-t_d)(t_a-t)}{t_a-t_d}. \quad (3)$$

There is little variation in the target's location near the start ($t = t_d$) or end ($t = t_a$) of the journey, and the maximum variance occurs at the halfway point and equals $K^2 \frac{(t_a-t_d)}{4}$.

Figure 1 illustrates one Brownian bridge path. While a target's movement dynamics may not literally evolve in a Brownian fashion at a micro level, at the macro level the Brownian bridge may approximate aggregate effects reasonably well. Environmental conditions may push a maritime target far off its intended course. Targets may also stop and loiter in the same location either to rest, or for tactical reasons to avoid detection. We can approximate these loiter situations in Figure 1 when the path backtracks in a small area. The variance parameter K determines the likelihood of these loiter situations and how far off-track the target may end up during its transit. A low value of K results in a near straight line path, whereas larger values of K can produce paths much more circuitous than Figure 1.

Figure 1 is useful for illustration purposes but not for planning search operations because the true path will be unknown. Rather, a heatmap that

displays the target's bivariate spatial density moving through time will provide more valuable decision support information. The target's position at any time $(x(t), y(t))$ is a bivariate Gaussian with mean given by μ_t in (2) and covariance matrix $\sigma_t^2 I$ from (3). Normality follows because $W_i(t-t_d)$ and $W_i(t_a-t_d)$ are jointly Gaussian by construction. We can plot analytic heatmaps using ellipses that contain the target with some specified probability (e.g., 0.95). Figure 2 illustrates one example for $t_d = 0$ and $t_a = 100$ h. The target departs from the star in the lower right-hand corner and arrives to the star in the upper left-hand corner. The top row fixes $K = 12$ nm and plots the circles containing 0.5, 0.75, and 0.95 of the distribution for $t = 15$ and $t = 50$ h. The bottom row presents similar figures for $K = 24$ nm. For these parameters, the standard deviation in either component in the middle of the transit is $5K$.

Figure 2 illustrates the importance of the variance parameter K . Unlike for complicated simulations where we have to specify local behaviour, or calibration of a large discrete state space, our model only has one parameter K . If the intelligence implies that the target will move quickly and directly towards its destination, then K should be small. If intelligence suggests the target may loiter or stop at various unknown waypoints along the way, then K should be large. In practice one can estimate K (or more generally the covariance structure) using maximum likelihood estimation if data is available (e.g., paths of previous targets). See Horne et al. (2007) and Pozdnyakov, Meyer, Wang, and Yan (2014) for a discussion of estimation approaches.

2.1. Gaussian uncertainty with departure and arrival locations

There may be uncertainty with the intelligence regarding the arrival and departure information. If the departure or arrival parameters are random, we can numerically integrate over those parameters via the law of total probability to compute the likelihood the target is in any region at any time. Unfortunately, we lose much of our analytic tractability when we introduce this additional source of randomness. We examine these situations in more detail in Section 3, where we present our simulation model.

In the special case where the departure location (x_d, y_d) and arrival location (x_a, y_a) follow a multivariate Gaussian distribution, we generalise our results and maintain the bivariate Gaussian nature of the target's location $(x(t), y(t))$ over time. We allow the departure and arrival locations to be correlated, so we must specify the mean and covariance

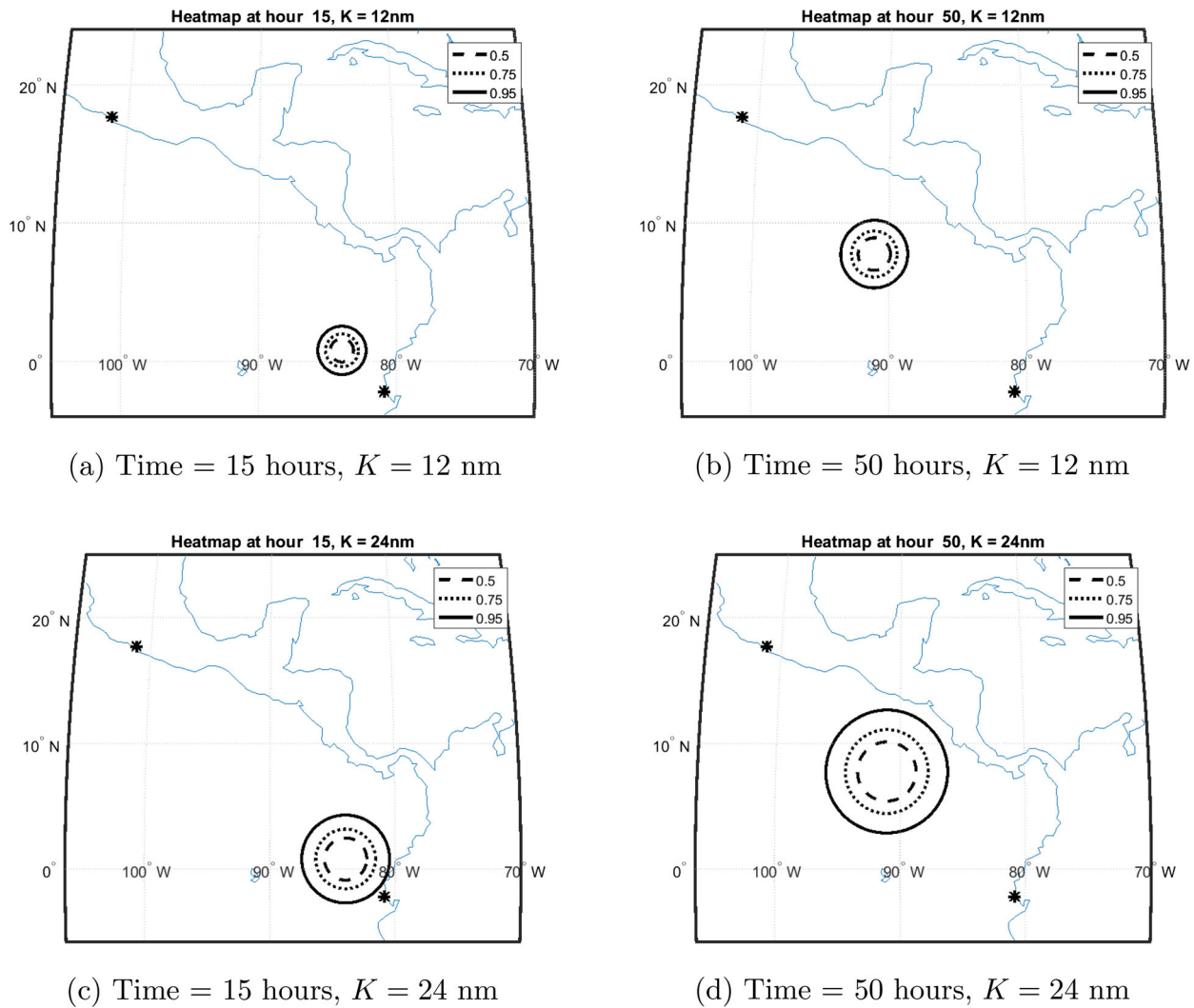


Figure 2. Evolution of heatmap for $t_d = 0$ and $t_a = 100$ h. The circles contain the target with the associated probabilities in the legend.

of a four dimensional Gaussian random vector as our input. We denote these parameters as $\hat{\mu}$ and $\hat{\Sigma}$:

$$\hat{\mu} = \begin{pmatrix} \bar{x}_d \\ \bar{y}_d \\ \bar{x}_a \\ \bar{y}_a \end{pmatrix}, \quad \hat{\Sigma} = \begin{pmatrix} \hat{\sigma}_{xd} & \hat{\sigma}_{xd,yd} & \hat{\sigma}_{xd,xa} & \hat{\sigma}_{xd,ya} \\ \hat{\sigma}_{xd,yd} & \hat{\sigma}_{yd} & \hat{\sigma}_{yd,xa} & \hat{\sigma}_{yd,ya} \\ \hat{\sigma}_{xd,xa} & \hat{\sigma}_{yd,xa} & \hat{\sigma}_{xa} & \hat{\sigma}_{xa,ya} \\ \hat{\sigma}_{xd,ya} & \hat{\sigma}_{yd,ya} & \hat{\sigma}_{xa,ya} & \hat{\sigma}_{ya} \end{pmatrix}. \quad (4)$$

The expression for the two-dimensional spatial position of the target remains the same as in (1). For the $x(t)$ term, there are now four random terms on the right-hand side of (1): x_d , $x_a - x_d$, $W_1(t - t_d)$, and $W_1(t_a - t_d)$. The spatial position of the target at time t , $(x(t), y(t))$ is bivariate Gaussian with mean μ_t and covariance matrix Σ_t . While the mean is similar to (2), the covariance is more complicated than the expression in Equation (3):

$$\mu_t = \begin{pmatrix} \mu_{x_t} \\ \mu_{y_t} \end{pmatrix} = \begin{pmatrix} \bar{x}_d + (\bar{x}_a - \bar{x}_d) \frac{t - t_d}{t_a - t_d} \\ \bar{y}_d + (\bar{y}_a - \bar{y}_d) \frac{t - t_d}{t_a - t_d} \end{pmatrix}, \quad (5)$$

and

$$\Sigma_t = \begin{pmatrix} \sigma_x^2(t) & \sigma_{xy}(t) \\ \sigma_{xy}(t) & \sigma_y^2(t) \end{pmatrix}, \quad (6)$$

where

$$\sigma_x^2(t) = \left(\frac{t_a - t}{t_a - t_d}\right)^2 \hat{\sigma}_{xd} + \left(\frac{t - t_d}{t_a - t_d}\right)^2 \hat{\sigma}_{xa} + 2 \frac{(t - t_d)(t_a - t)}{(t_a - t_d)^2} \hat{\sigma}_{xd,xa} + K^2 \frac{(t - t_d)(t_a - t)}{t_a - t_d} \quad (7)$$

$$\sigma_y^2(t) = \left(\frac{t_a - t}{t_a - t_d}\right)^2 \hat{\sigma}_{yd} + \left(\frac{t - t_d}{t_a - t_d}\right)^2 \hat{\sigma}_{ya} + 2 \frac{(t - t_d)(t_a - t)}{(t_a - t_d)^2} \hat{\sigma}_{yd,ya} + K^2 \frac{(t - t_d)(t_a - t)}{t_a - t_d} \quad (8)$$

$$\sigma_{xy}(t) = \left(\frac{t_a - t}{t_a - t_d}\right)^2 \hat{\sigma}_{xd,yd} + \left(\frac{t - t_d}{t_a - t_d}\right)^2 \hat{\sigma}_{xa,ya} + \frac{(t - t_d)(t_a - t)}{(t_a - t_d)^2} (\hat{\sigma}_{xd,ya} + \hat{\sigma}_{yd,xa}). \quad (9)$$

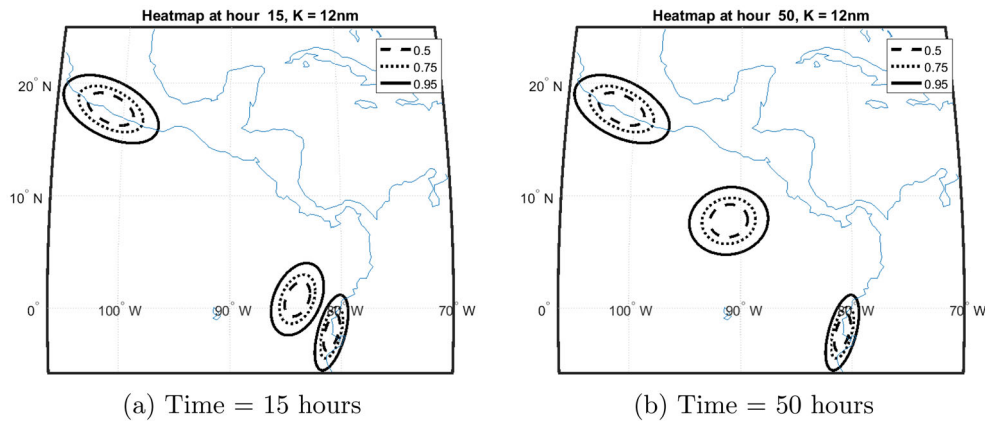


Figure 3. Evolution of heatmap at different times for $K=12\text{nm}$. The ellipse in the lower right-hand corner represents the uncertainty associated with the departure location and the ellipse in the upper left-hand corner represents the uncertainty associated with the arrival location. The ellipses in the middle capture the evolving uncertainty of the target's location as the target moves through the area.

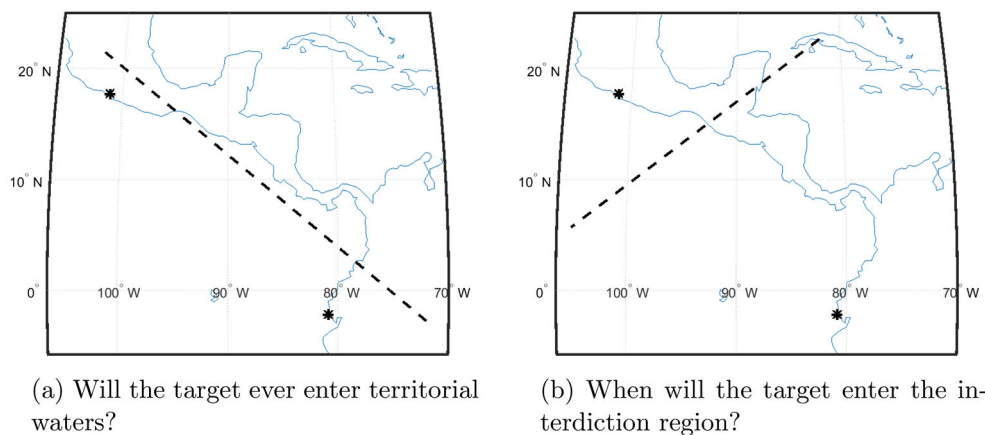


Figure 4. First passage time applications.

We assume the target's departure and arrival locations are independent of the intermediate movement fluctuations. Normality follows because (x_d, y_d, x_a, y_a) are jointly Gaussian, $W_i(t-t_d)$ and $W_i(t_a-t_d)$ are jointly Gaussian, and (x_d, y_d, x_a, y_a) is independent of $(W_i(t-t_d), W_i(t_a-t_d))$. Figure 3 presents a similar figure to the top row of Figure 2 with $K=12\text{nm}$. Comparing Figure 3 to Figure 2, we see the associated area of uncertainty (now ellipses) is much greater in Figure 3, and the elongation and orientation of the ellipse shifts over time.

2.2. First passage time results

Modelling target movement as a Brownian bridge allows us to leverage first passage time results for Brownian bridges. Figure 4 presents two examples of the relevance of these problems to our target setting. Figure 4a illustrates a situation where planners want to know whether a target will ever enter a region of interest, such as certain territorial waters. This corresponds to the probability a Brownian bridge will hit a line. The line in Figure 4b represents the boundary of an interdiction region.

Planners may only have access to detection and interdiction assets for a limited time, and they want to deploy the assets during a time window the target is likely in the area. This scenario corresponds to computing the probability distribution for the first time the Brownian bridge hits a line that separates the two endpoints.

Before presenting the results, we list several limitations. The results only hold for linear boundaries, such as those in Figure 4. The boundaries are infinite, not just line segments. Finally, the departure (x_d, y_d, t_d) and arrival information (x_a, y_a, t_a) is deterministic. If the inputs are random, we can numerically integrate the below results over the random quantities to compute the probabilities of interest; however, there will be no analytic simplifications.

We adapt the hitting time results directly from Atkinson and Singham (2015), and therefore write the linear boundary as $a_x x + a_y y = b$, where $a_x^2 + a_y^2 = 1$. The first result specifies the probability the target will hit the boundary, given that both the departure location and arrival location lie on the same side of the line (e.g., Figure 4a). The following

result corresponds to Proposition 4.6 in Atkinson and Singham (2015).

Result 1. Given that the start and end points are on the same side of the line, we present the probability that the target ever crosses the line. If $(a_x x_d + a_y y_d - b)(a_x x_a + a_y y_a - b) > 0$ then

$$\begin{aligned}
 & \mathbf{P} \left[\bigcup_{t \in [t_d, t_a]} \left((a_x x_d + a_y y_d - b)(a_x x(t) + a_y y(t) - b) \leq 0 \right) \right] \\
 &= \exp \left(-2 \frac{|a_x x_d + a_y y_d - b| |a_x x_a + a_y y_a - b|}{K^2 (t_a - t_d)} \right).
 \end{aligned}
 \tag{10}$$

The condition in Result 1 is the mathematical representation for both endpoints lying on the same side of the line. The expression in the probability is the mathematical representation for the Brownian bridge hitting the line for any $t \in [t_d, t_a]$. For the example in Figure 4a, the line representing the territorial water’s demarcation has parameters $a_x = 0.631, a_y = 0.776$ and $b = -2842$ nm. The distance from the departure location to the line is 323 nm, and the distance from the arrival location is 167 nm. If $t_d = 0$ and $t_a = 100$, then Figure 5 illustrates how the probability in (10) varies with K .

The second result specifies the probability distribution for the first time the Brownian bridge hits a line that line separates the departure location and arrival location (e.g., Figure 4b). The following corresponds to Proposition 4.7 in Atkinson and Singham (2015).

Result 2. Given that the line separates the start and end points, we present the probability that the Brownian bridge crosses the line before time t . If $(a_x x_d + a_y y_d - b)(a_x x_a + a_y y_a - b) < 0$ then

$$\begin{aligned}
 & \mathbf{P} \left[\bigcup_{s \in [t_d, t]} \left((a_x x_d + a_y y_d - b)(a_x x(s) + a_y y(s) - b) \leq 0 \right) \right] \\
 &= \exp \left(2 \frac{|a_x x_d + a_y y_d - b| |a_x x_a + a_y y_a - b|}{K^2 (t_a - t_d)} \right) \\
 & \times \Phi \left(- \frac{|a_x x_d + a_y y_d - b|(t_a - t) + (t - t_d) |a_x x_a + a_y y_a - b|}{\sqrt{K^2 (t_a - t_d)(t_a - t)(t - t_d)}} \right) \\
 & + \left(1 - \Phi \left(\frac{|a_x x_d + a_y y_d - b|(t_a - t) - (t - t_d) |a_x x_a + a_y y_a - b|}{\sqrt{K^2 (t_a - t_d)(t_a - t)(t - t_d)}} \right) \right),
 \end{aligned}
 \tag{11}$$

for $t_d \leq t \leq t_a$.

If we write the probability in (11) as $F(t)$, then $F(t)$ is the CDF for the hitting time of the boundary. The parameters of the boundary line in Figure 4b are $a_x = -0.596, a_y = 0.803$ and $b = 4008$ nm. This boundary line intersects the direct line path from the departure location to the arrival location at fraction 0.725 between departure and arrival. If $t_d = 0$

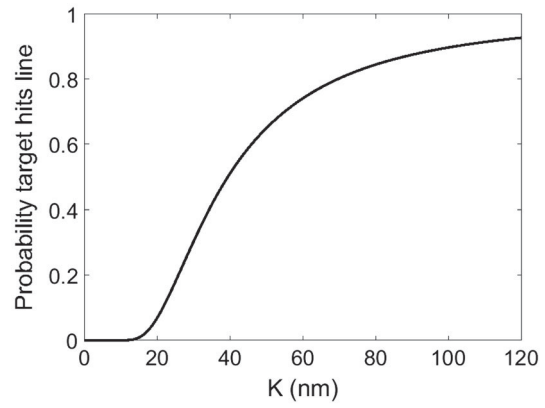


Figure 5. Probability the target hits the line for the scenario in Figure 4(a): $a_x = 0.631, a_y = 0.776, b = -2842$ nm, $t_d = 0$ and $t_a = 100$.

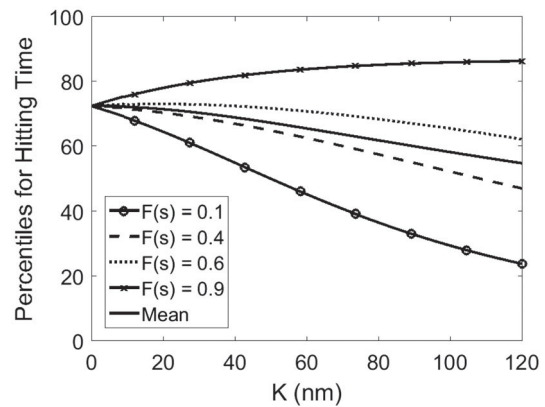


Figure 6. Percentiles for the hitting time of the boundary line in Figure 4(b): $a_x = -0.596, a_y = 0.803, b = 4008$ nm, $t_d = 0$ and $t_a = 100$.

and $t_a = 100$, then Figure 6 illustrates several percentiles of $F(t)$ as K increases. For small K , the distribution is essentially deterministic with a hitting time of 72.5. However as K increases the spread in the hitting time distribution increases significantly.

3. A particle filter algorithm for updating distributions

This section describes enhancements to the TBBM to incorporate intelligence updates into the target’s spatial distribution using a particle filter. We first develop a simulated version of the model which allows us to use more general data inputs. We next develop a particle filter algorithm that exploits unique aspects of the Brownian bridge structure and the target tracking problem to update the distribution of the target location when new intelligence arrives.

3.1. Overview

In this subsection we provide an overview of the TBBM model within a particle filtering framework.

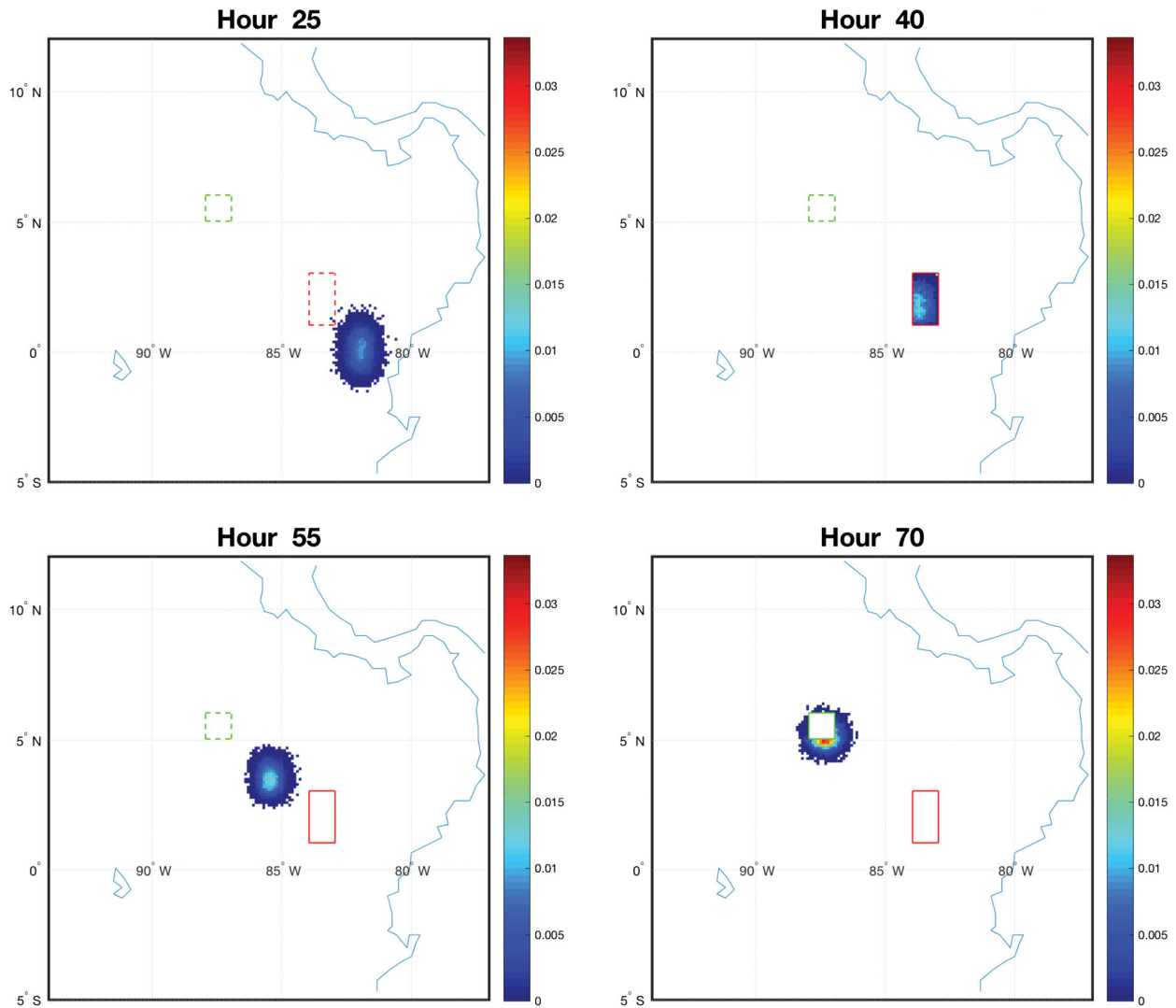


Figure 7. Intelligence updating example with rectangular sensors (red are positive and green are negative signals) and uniform sampling for start and end locations and times.

We present more formal details on the particle filter in the following subsections. The Brownian bridge model presented in Section 2 provides a way of quickly estimating characteristics of the target's distribution using an analytical model. There are two reasons to develop a simulated version of the model. The first is that simulation allows for more complex intelligence inputs. For example, the target's starting location may have a non-Gaussian spatial distribution or the arrival time may be random.

The second reason is to effectively incorporate intelligence to update the target distribution. We can include some intelligence in the model in Section 2 and maintain analytic tractability; however this limits us to a Kalman filter paradigm, which severely constrains the types of intelligence we can include. There is no limit to the types of intelligence we can incorporate into the simulated model, and the updating procedure fits naturally in the particle filter framework.

Each simulated Brownian bridge path represents one "particle." Without any additional information,

the particles have equal weight. The addition of new intelligence re-weights the particles according to how much each particle aligns with the new intelligence. Each intelligence update is a sensor signal at a specific time. The sensor returns a positive or negative signal about whether the target is in the vicinity of the sensor. The sensors may be imperfect, so a positive signal does not imply the target is necessarily inside the sensor's footprint. The updating procedure computes new particle weights based on the signal type and sensor characteristics. Before describing the components of the particle filter in more detail, we illustrate with a simple scenario using a basic updating procedure.

Suppose the sensor has perfect accuracy in determining whether or not a target is present within a given area (i.e., a cookie-cutter sensor), but does not deliver more specific information on the location of the target. If the sensor returns a positive signal, then all target paths (particles) within the sensor area of coverage are projected forward with equal weight, and all paths outside the coverage area

receive zero weight and are removed from consideration. If the sensor instead returns a negative signal, then particles within the sensor area are removed, while those outside the sensor area are projected forward. For simplicity, here we assume the sensor obtains an instantaneous glimpse that returns a positive or negative signal at one specific time. An example of such a sensor would be a satellite image or dipping sonar.

Figure 7 shows a set of plots produced by the model using this basic updating method with two perfect sensors. A positive detection is set for the lower right sensor at hour 40 and a negative detection is set for the upper left sensor at hour 70 to demonstrate the elimination of paths outside and inside of the sensors, respectively. This example models the sensor coverage areas as rectangles, which can represent images collected by satellites at a particular point in time. Other sensors could produce different footprint shapes, such as circles. The red box corresponds to a sensor that reports a positive signal, while the green box reports the absence of a target. The scale of the colour legend to the right of each plot corresponds to the range of probabilities associated with the heat map. The departure and arrival locations and times are sampled from independent uniform distributions. The solid boxes refer to intelligence that has already been observed, while dashed boxes refer to events scheduled to happen at future times.

While Figure 7 shows a simple updating method, our particle filter handles more complex weight functions using relative distance to the sensor. The particle filter also protects against degeneracy where the number of unique particles decreases with each stage. Simply removing paths that fail to meet the intelligence without employing a resampling method can result in only a handful of paths having positive weight at the end of an experiment with multiple sensor observations. This makes it difficult to accurately estimate the target’s spatial distribution. We reduce the impact of degeneracy by resampling and using a roughening technique to re-simulate new Brownian bridges with the same distributional properties as those of the sampled Brownian bridges. We summarise the particle filter algorithm below and provide additional details, including information about the weight function and roughening method, in the next several subsections.

1. Simulate N Brownian bridge paths (particles) according to initial intelligence on departure and arrival locations and times.
2. Assign each path an initial weight of $\frac{1}{N}$.
3. When intelligence arrives, update the weights of each path according to a likelihood function.

4. Resample (with replacement) a set of N paths according to the updated weights.
5. Roughen the resampled paths to create a new set of unique Brownian bridge paths.
6. Project the heatmap forward in time using the roughened resampled paths, with weights all equal to $\frac{1}{N}$.
7. At the time of the next intelligence update, go to Step 3.

3.2. Particle filter notation

For a more details on the specific mechanics of particle filters please, see Arulampalam, Maskell, Gordon, and Clapp (2002) and Doucet and Johansen (2009). The particles in our systems are Brownian bridge paths, and therefore the state of the system at time t is the path p_t the target is on at time t . We assume a fixed set of discrete times points $t \in \{1, \dots, T\}$ at which heatmaps are generated, where T is the maximum time period. This allows for a more straightforward computational implementation of the particle filter updates at particular times. Formally, a path p is a collection of two-dimensional coordinates over all time indices: $p \in \{(x_t, y_t)\}_{t=1}^T, x_t, y_t \in \mathbb{R}, \forall t\}$. Therefore our state-space can be represented as a subset of \mathbb{R}^{2T} .

While the target’s position changes across time periods, the target’s path (and hence state) remains fixed over time. This path-based state representation differs from many filter approaches that use location-based states. We could formulate a position-based variant, but prefer our approach as it allows for path inputs generated by more general movement dynamics.

A filtering algorithm allows us to update the posterior distribution for the target’s path given all intelligence received. We denote m_t as the intelligence signal received at time t . We next define our primary quantity of interest $f(p_t|m_{1:t})$: the updated posterior distribution for the target’s path given all signals received until time t . We approximate $f(p_t|m_{1:t})$ as a discrete distribution with weights assigned to particles. At time t we have N particles (i.e., possible target paths), where $p_t^{(i)}$ represents the i th particle. The particle $p_t^{(i)}$ will change with t due to resampling and roughening which will be described in the next section. Our estimate of $f(p_t|m_{1:t})$ takes the form (see Equation (49) in Arulampalam et al., 2002)

$$f(p_t|m_{1:t}) \approx \sum_{i=1}^N w_t^{(i)} I(p_t = p_t^{(i)}),$$

where $I(\cdot)$ is the standard indicator function and $w_t^{(i)}$ is the weight associated with particle i . The

particle filter re-weights successive particles $p_t^{(i)}$ using $w_t^{(i)}$.

A filter model has two main components: a movement model and a measurement model. The measurement model dictates how we update the posterior distribution in a Bayesian fashion from $f(p_t|m_{1:t-1})$ to $f(p_t|m_{1:t})$ upon receipt of signal m_t . The measurement model is represented by a function $g(m_t|p_t)$ which represents the likelihood of observing measurement m_t given that the true path is p_t . This is the key driver in updating the weights $w_t^{(i)}$. We discuss several likelihood functions relevant for our application in Section 3.3. The movement model specifies how the state transitions in the next time period and is denoted as a conditional density $h(p_t|p_{t-1})$. In our case the target remains on the same path so the movement density $h(p_t|p_{t-1})$ is deterministic at p_{t-1} . If we used a location-based state, then $h(\cdot)$ would be the conditional Gaussian density derived from the Brownian bridge.

In practice a particle filter generates the particles in the next time period via importance sampling (see Section V.A in Arulampalam et al., 2002). This importance density $q(p_t|p_{t-1}, m_t)$ is often chosen to be the movement density $h(p_t|p_{t-1})$. Because the target path is fixed, in our case $q(p_t|p_{t-1}, m_t) = h(p_t|p_{t-1})$. While we do not execute a movement update via importance sampling, the roughening procedure described in Section 3.4 that perturbs the current paths performs a similar function.

Our particle filter is a variant of the sampling importance resampling (SIR) filter. See Section V.B of Arulampalam et al. (2002) or 4.1 of Doucet and Johansen (2009) for more details. This algorithm generates the following updated unnormalised weight after every signal m_t :

$$u_t^{(i)} = \frac{g(m_t|p_t^{(i)})h(p_t^{(i)}|p_{t-1}^{(i)})}{q(p_t^{(i)}|p_{t-1}^{(i)}, m_t)},$$

For our specific values of $h(p_t|p_{t-1})$ and $q(p_t|p_{t-1}, m_t)$, the unnormalised weight simplifies greatly to $u_t^{(i)} = g(m_t|p_t^{(i)})$. Thus the updated normalised weight is proportional to the likelihood (see Equation (66) of Arulampalam et al., 2002):

$$w_t^{(i)} = \frac{u_t^{(i)}}{\sum_{j=1}^N u_t^{(j)}} = \frac{g(m_t|p_t^{(i)})}{\sum_{j=1}^N g(m_t|p_t^{(j)})}. \quad (12)$$

We use (12) to update the weight in Step 3 of the algorithm listed at the end of Section 3.1.

There are a few modifications needed for implementation in our code. A path need not be active for all time indices; for example a target might depart at index 4. In our implementation we track

the status of a path by specifying $x_t = y_t = -999$ if t is an index prior to departure and $x_t = y_t = 999$ if t is an index after arrival. Additionally, for notational simplicity above we assume a signal occurs at every time index, but in practice most signals will be vacuous reports of the form “no information” and hence will not actually trigger an update.

3.3. Measurement models

The measurement model is defined by its likelihood function, and that likelihood function fully specifies the updated weight $w_t^{(i)}$ (see Equation (12)). In this section, we update the weights $\{w_t^{(i)}\}_{i=1}^N$ from a sensor signal arriving at a specific time t . To avoid notational clutter, we suppress the dependence on t in this section. We present possible likelihood functions for updating the weights on simulated Brownian bridge particles. These schemes give more weight to particles that closely align with the sensor signal, but also account for sensor errors by giving some weight to paths that do not match the intelligence. For our case we rewrite the likelihood function $g(m_t|p_t^{(i)})$ in more specific terms to generate the updated (unnormalised) weight:

$$u^{(i)} = \mathbf{P}[\text{signal at } t | \text{target is particle } i].$$

The weights represent the relative likelihood that a specific particle triggers the sensor information. In the example given in Figure 7, the updated unnormalised weight assigned would be $u^{(i)} = 1$ if the particle aligns with the intelligence at time t and $u^{(i)} = 0$ otherwise. In practice, the likelihood is often just a function of the distance between the particle and the sensor at the measurement time t :

$$u^{(i)}(d) = \mathbf{P}[\text{signal at } t | \text{distance between particle } i \text{ and sensor is } d].$$

In this section when we refer to the location of particle i , we formally mean the particle’s spatial location $(x_t^{(i)}, y_t^{(i)})$ at the measurement time t . Similarly the distance between a sensor and particle i is the distance between the sensor and the particle’s spatial location at the sensor’s measurement time t .

Particles that are closer to the sensor at time t will have a larger weight if the sensor delivers a positive update and will have a smaller weight if the sensor delivers a negative update. The cookie-cutter example in Figure 7 falls under this category if we measure distance using a modified ℓ_∞ norm rather than the standard ℓ_2 norm (see Equation (13)). For cookie-cutter sensors, the weight function sets $u^{(i)}(d) = 1$ if the particle is within the sensor

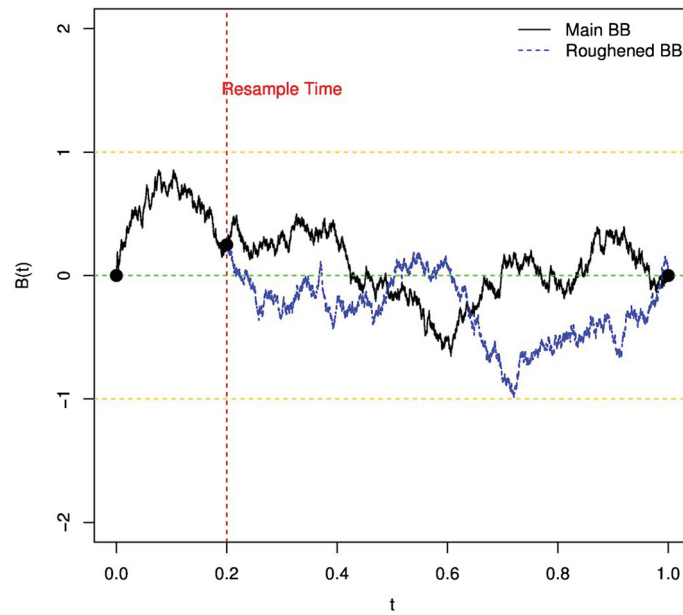


Figure 8. Illustration of roughening procedure.

bounds for a positive signal, and zero otherwise. For a negative signal, $u^{(i)}(d) = 0$ if the particle is within the sensor bounds and is 1 otherwise.

Assume the intelligence sensor box is centred at (x_0, y_0) and has width W and height H , and the particle’s location at time t is $(x^{(i)}, y^{(i)})$. We define the distance from the centre of the box to the particle at time t using a modified normalised infinity norm distance:

$$d^{(i)} = \max\left(\frac{2}{W}|x^{(i)} - x_0|, \frac{2}{H}|y^{(i)} - y_0|\right). \quad (13)$$

If the particle is on the edge of the box, then $d^{(i)} = 1$. If it is inside the box, $d^{(i)} < 1$ with $d^{(i)} = 0$ at the centre of the box. Particles outside the box have $d^{(i)} > 1$. The value of $d^{(i)}$ is the same over concentric rectangles around the centre. We can define various weight functions that depend on $d^{(i)}$, for example a linear weight function for positive and negative signals where $0 < \alpha < 1$:

$$u_p^{(i)}(d^{(i)}) = 1 - \alpha \min(d^{(i)}, 1), \quad u_N^{(i)}(d^{(i)}) = \alpha \min(d^{(i)}, 1). \quad (14)$$

For any particle i with distance greater than 1 (i.e., outside the box), the weight for a positive signal is constant at $1 - \alpha$, which can be viewed as the false positive probability produced by particles outside the sensor area. Inside the box more weight is given to particles close to the centre. A three-dimensional plot of $u_p^{(i)}(d^{(i)})$ in (14) for $d^{(i)} < 1$ produces a pyramid with the box as the base and the centre of the box as the top. For a negative signal, particles outside the box receive weight α , and those inside the box receive less weight when closer to the centre. Another weighting example is an exponential function with $\beta > 0$:

$$u_p^{(i)}(d^{(i)}) = \exp\left(-\left(d^{(i)}\right)^\beta\right),$$

$$u_N^{(i)}(d^{(i)}) = 1 - \exp\left(-\left(d^{(i)}\right)^\beta\right). \quad (15)$$

We note that when $\beta = \infty$ this function gives the cookie-cutter sensor, setting weights to 1 when particles match the intelligence, and 0 otherwise.

3.4. Roughening procedure

One major issue with particle filters is degeneracy resulting from a smaller subset of particles receiving a higher proportion of the overall weight over time. Resampling with replacement ensures that there will always be N particles, but N unique particles will not necessarily be sampled at each iteration. As more updates occur, the number of unique particles decreases.

If we use the updating procedure described above Figure 7 with perfect sensors, it is not uncommon to see a loss of over 95% of the particles with three sensor updates because only a small proportion of the originally simulated particles will meet all the intelligence. This leads to a poor approximation of the distribution because it is based on a small number of particles and increasing the number of initial particles may not be computationally feasible.

Gordon, Salmond, and Smith (1993) suggest perturbing each resampled particle with a Gaussian jitter to mitigate degeneracy. However, due to the Markovian nature of the Brownian bridge, we implement a more effective roughening procedure. Suppose \bar{t} is the time of the current intelligence update and we resample particle i . Particle i ’s current position is $(x_{\bar{t}}^{(i)}, y_{\bar{t}}^{(i)})$ and it arrives to its final

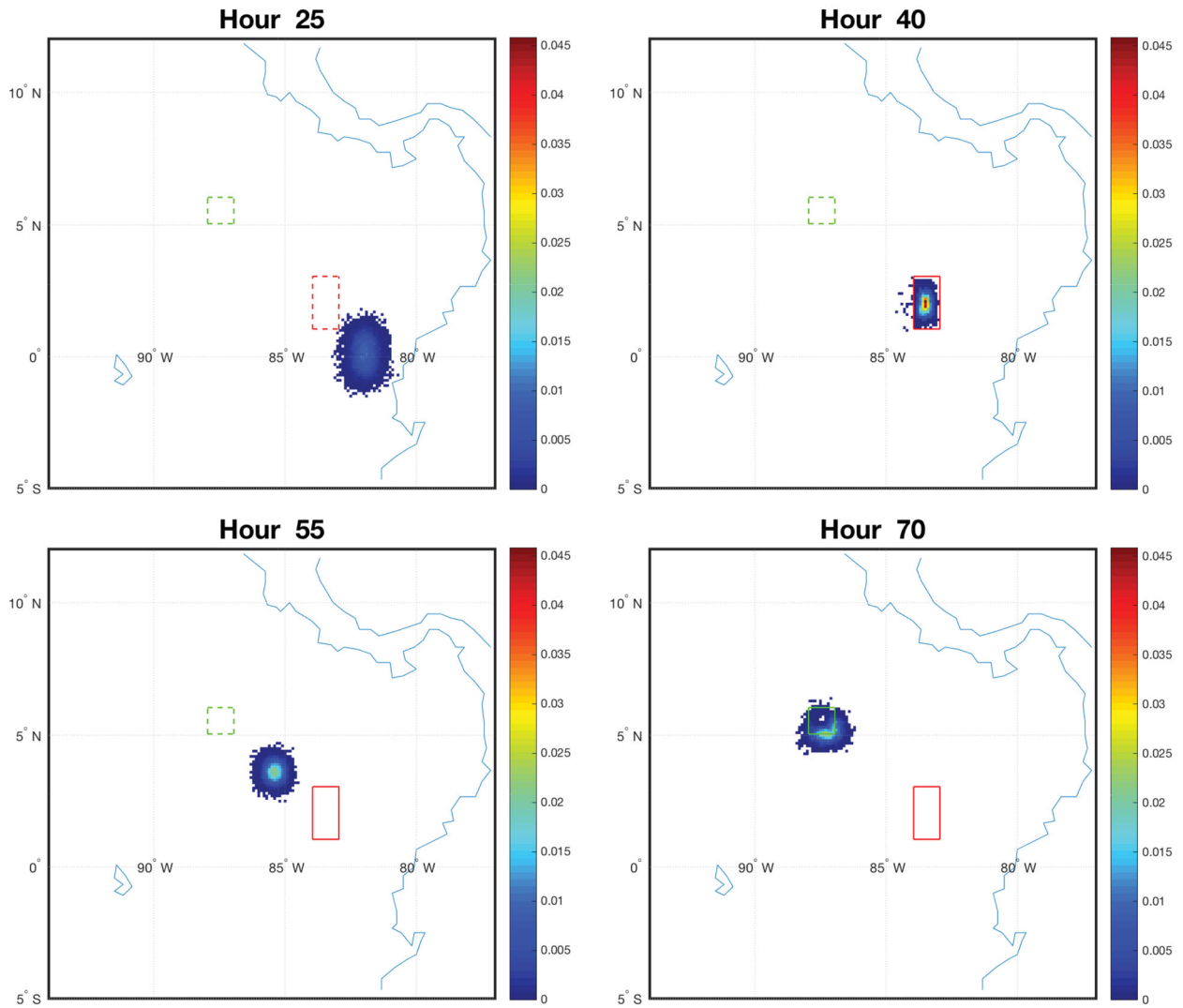


Figure 9. Heatmaps using a linear weighting function as in (14) with $\alpha = 0.95$.

location $(x_a^{(i)}, y_a^{(i)})$ at time $t_a^{(i)} \leq T$. In between its current position and final position, particle i evolves according to a Brownian bridge. Consequently, to roughen particle i we simulate a new Brownian bridge between $(x_t^{(i)}, y_t^{(i)}, \bar{t})$ and $(x_a^{(i)}, y_a^{(i)}, t_a^{(i)})$. This procedure generates a new particle with the same distributional properties as the originally sampled particle i . Figure 8 shows a one-dimensional illustration of this roughening procedure for $\bar{t} = 0.2$. As discussed in Section 3.2, this roughening procedure is similar to having a movement model in the particle filter where a state “moves” from one path to another path after one time period.

3.5. Particle filter algorithm

We now present the details of the particle filter algorithm applied to the TBBM, which employs the roughening procedure described above. Initially N particle paths are generated for all time periods $t = 1, \dots, T$ using the simulated model described in Section 3.1 and all particles are equally weighted: $w_1^{(i)} = \frac{1}{N} \forall i$. The weights associated with these

simulated particles are updated according to the likelihood function (see Equation (12)) when the first intelligence signal is received.

Suppose we have particles $i = 1, \dots, N$, and the identity of particle i at time t is $p_t^{(i)}$. Let $\tilde{p}_t^{(i)}$ be the new particle i sampled from the set of particles $P_t = \{p_t^{(1)}, p_t^{(2)}, \dots, p_t^{(N)}\}$ after the arrival of new intelligence at time t . Finally, denote $\hat{p}_t^{(i)}$ as the roughened version of $\tilde{p}_t^{(i)}$. The following summarises the particle filter algorithm which is implemented each time new intelligence arrives from the sensors.

1. Let $P_t = \{p_t^{(1)}, p_t^{(2)}, \dots, p_t^{(N)}\}$ be the particles at the current time t .
2. Update the weights of all particles according to Equation (12) to yield $W_t = \{w_t^{(1)}, w_t^{(2)}, \dots, w_t^{(N)}\}$ based on sensor intelligence at time t .
3. Resample the particles according to the weights W_t with replacement, yielding samples $\tilde{P}_t = \{\tilde{p}_t^{(1)}, \tilde{p}_t^{(2)}, \dots, \tilde{p}_t^{(N)}\}$, each with weight $1/N$.
4. Transform the resampled particle $\tilde{p}_t^{(i)}$ to the roughened particle $\hat{p}_t^{(i)}$, for $i = 1, \dots, N$, using the

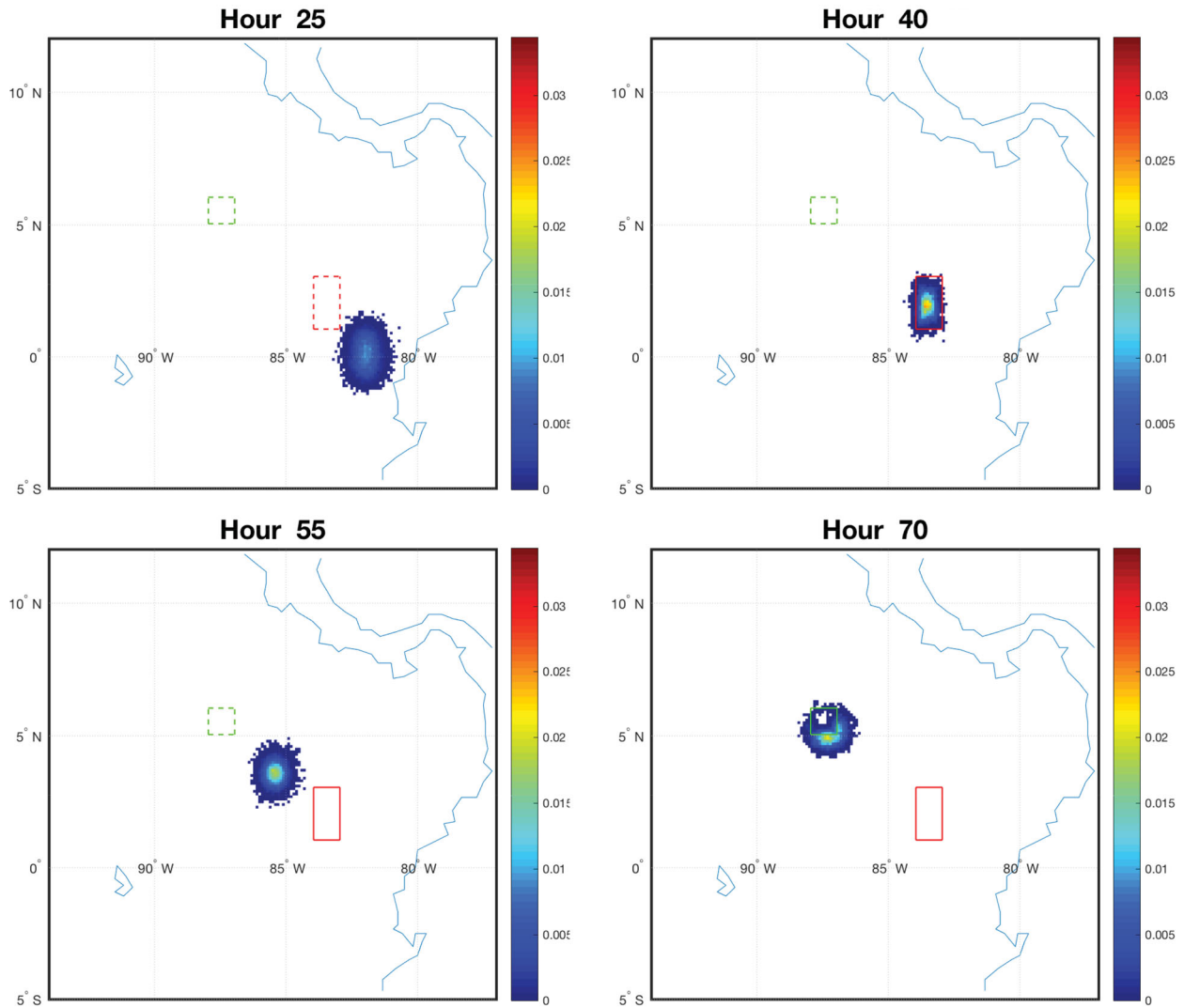


Figure 10. Heatmaps using an exponential weight function as in (15) with $\beta = 2$.

roughening method in Section 3.4 to generate new Brownian bridges over the remaining time periods.

- Proceed to the next time step setting particles $p_{t+1}^{(i)} \leftarrow \hat{p}_t^{(i)}$.

4. Experiments

There are a variety of sensor types available to collect information on moving targets, and the specific implementation details of the particle filter will depend on the nature of incoming intelligence. The properties of the individual sensor determine the area of coverage and the quality of the information within a given region. Some sensors may have a decaying quality as the distance from the sensor increases, so a linear or exponential weight function can be used. Because intelligence can come in many forms, the nature of the weight functions should be changed accordingly. Human intelligence could specify that the target is present in a given region, but may not specify the exact location, in which case a uniform weighting across a region can be used.

Finally, we note that sensor regions do not need to be rectangular, and the algorithm can be modified to be flexible to the sensor type.

We implement the TBBM with a particle filter in MATLAB (The MathWorks, 2016) to assess the performance of the results. Each experiment begins with 20,000 simulated Brownian bridges, each of which has 500 time steps simulated between the sampled start and end points. Generating a sequence of heatmaps can take anywhere from 10 seconds to a few minutes (using a single processor) depending on the number of sensor updates and time steps used. The particle filter model resamples each time a new piece of intelligence is observed, and the weight function can be easily changed to allow for modelling flexibility.

Figure 9 shows an example of the particle filter algorithm applied using the linear weight function in (14) with $\alpha = 0.95$. We only display the points contained in the top 95% of the probability mass to avoid plotting regions with a very small probability of containing the target. We see at hour 40 that the particles are concentrated around the centre of the

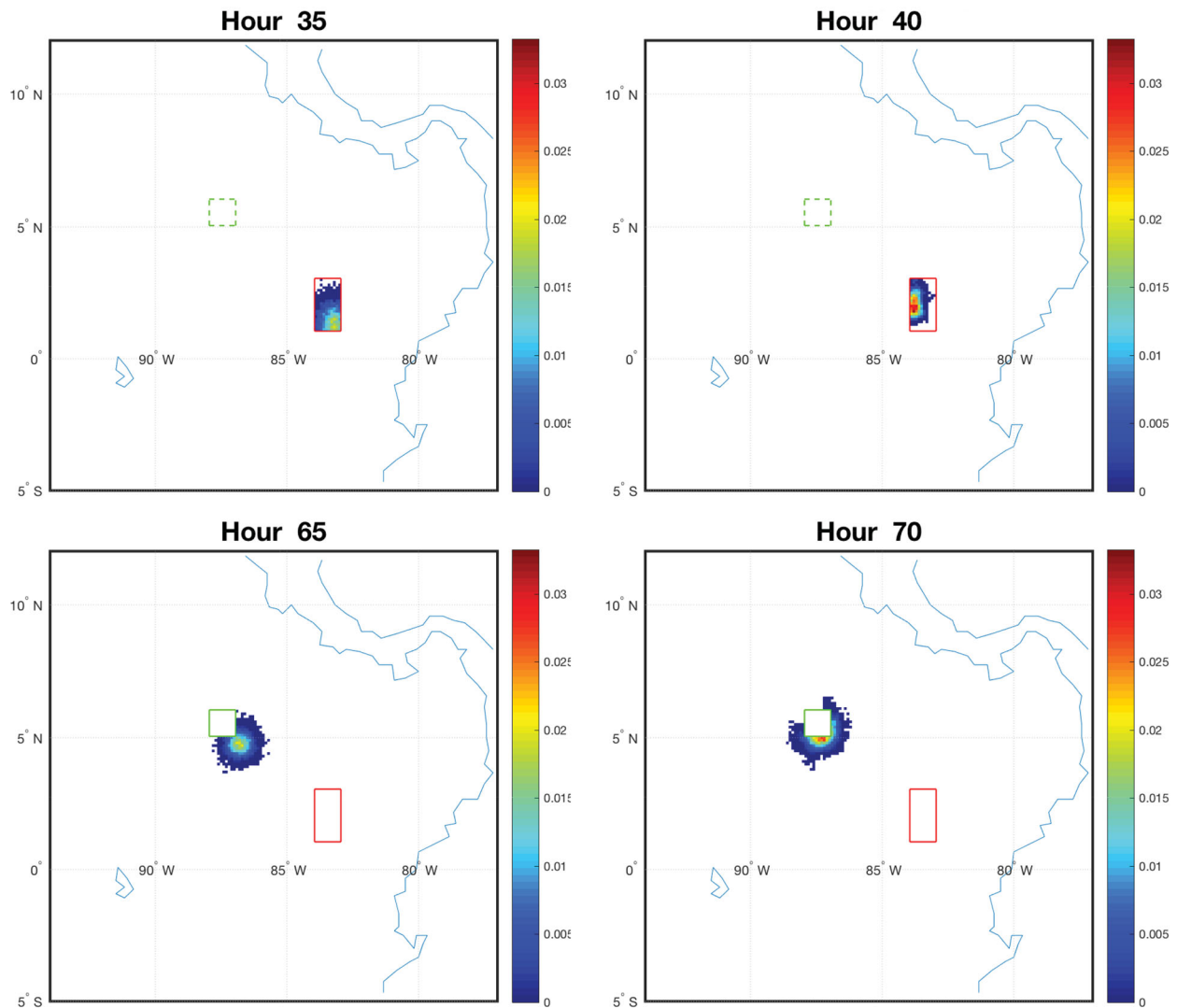


Figure 11. Heatmaps using sensor fields that deliver a positive signal over hours 35–40 and a negative signal over hours 65–70, using a cookie-cutter sensor.

sensor, and some particles outside the sensor box are resampled as well. At hour 70 with a negative signal, weight within the sensor box is reduced, while there is still significant weight around the boundary of the sensor box.

Figure 10 displays heatmaps using the exponential weight function given in (15) with $\beta = 2$. We see again that positive weight is allocated to points outside the sensor box at hour 40, with a higher concentration of particles in the centre of the box. The rate of decay in the weight is determined by the choice of β . For the negative signal at hour 70, there is a partial reduction of particles in the box, with a greater reduction close to the centre.

We can also explore the effect of a patrol, or a sensor that collects information over a time interval rather than at one specific time. For example, consider a group of many sensors, such as a sonobuoy field, which monitors for potential targets. The sensor field reports either a positive signal that the target is present in the field or a negative signal that the target is absent. If the sensor field reports a

target is present during a time interval, that would provide more information than just observing the target at one particular point in time. Additionally, the sensor field could also fail to observe the target over a range of time. Consider the cookie-cutter sensor example given in Figure 7 with two sensor fields. The southeast field is active over a 5-h time range (hours 35–40) and reports positively that the target is in the field for the entire 5 h and shows an increased concentration in location as time progresses. The northwest field is active over a different 5-h time range (hours 65–70) and fails to observe the target over the entire 5 h. Figure 11 plots the heatmap at hours 35, 40, 65, and 70.

We see at time 35 that the positive signal excludes particles outside the box. At time 40 we see a more concentrated distribution of particles, given that the target was observed in the box over times 35–40, compared to the heatmap at time 40 in Figure 7 where the positive signal was only observed at time 40. Similarly with the negative signal, at time 65 the particles inside the sensor box are removed as the first negative signal

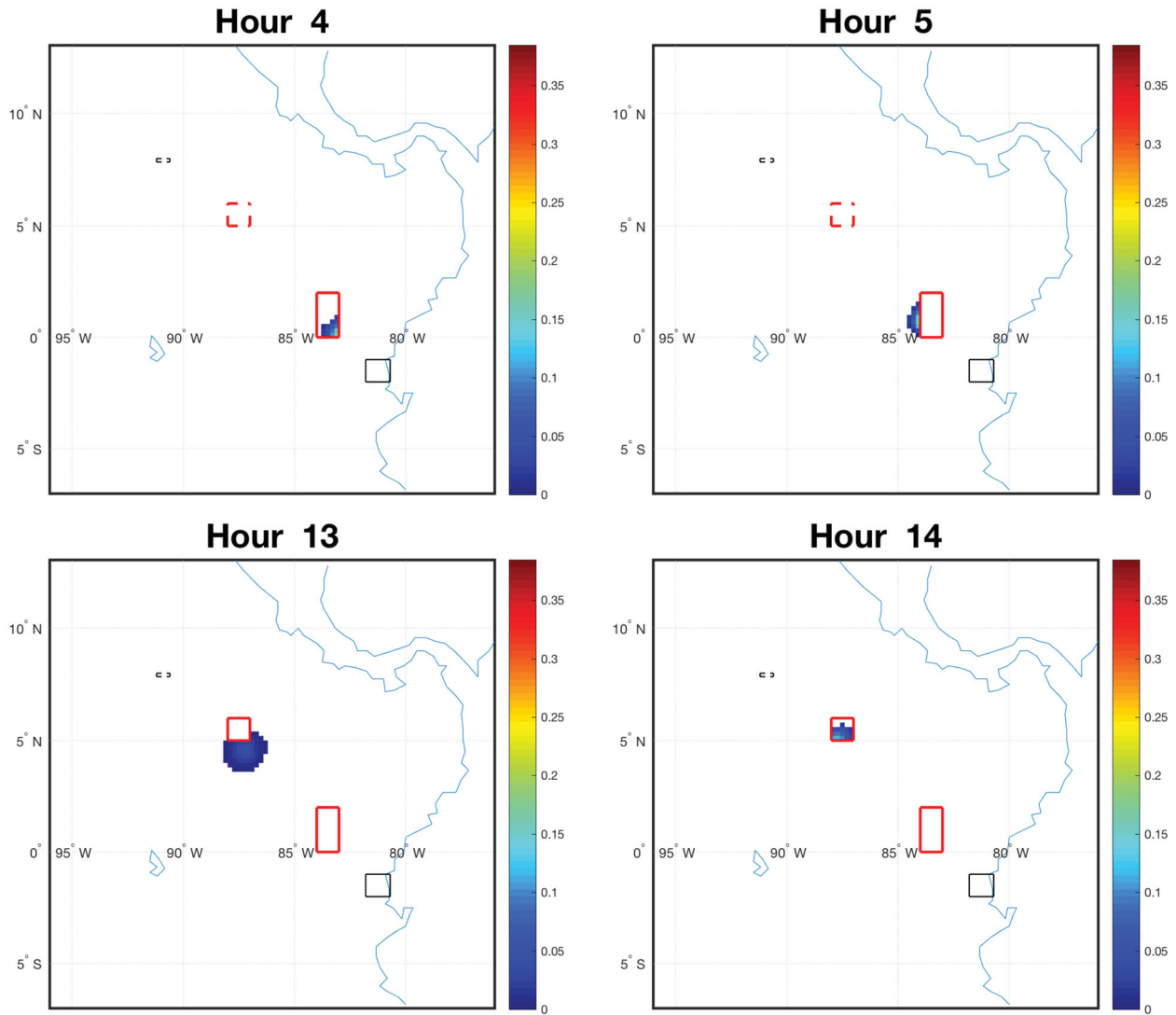


Figure 12. The first sensor switches from a positive to a negative signal in times 4 to 5, and the second sensor switches from negative to positive in times 13 to 14.

is reported. As the target fails to appear within the sensor box over times 65-70, the algorithm continues to remove particles that appear within the box. This leads to a heatmap that is concentrated around the south and east of the box at time 70 and eliminates all targets that could have been in the box just prior to time 70.

Figure 12 shows an example where a sensor’s observed signal switches in adjacent time increments. In the top plots, the first sensor initially reports a positive signal in hour 4 and then a negative signal at time 5. In the bottom plots, the second sensor fails to see the target at hour 13, but then observes the target at hour 14. Thus, the same sensor can change its observation and the heatmap can adjust to represent the updated information.

5. Conclusion

Many search methods rely on a model for a target’s location over time that incorporates uncertainty. This paper develops a robust model for target movement that produces temporal heatmaps. The model requires

only the target’s starting and ending locations (with some uncertainty), and does not assume straight-line or deterministic movement, but approximates behaviour using Brownian bridges which can be calibrated to capture variation in individual trajectories. An analytical model can be used to quickly estimate containment regions and crossing probabilities.

A simulated version of the model lends itself to enhancements and updating based on new intelligence. The distribution of the target’s locations at midpoints can be easily updated when new intelligence arrives using a particle filter, which weights simulated paths according to a likelihood function. Traditional degeneracy challenges are mitigated by using roughening and resampling techniques designed for Brownian bridges to ensure a complete heatmap can be generated as time progresses.

The Brownian bridge model for target motion has many advantages over other models. The uncertainty incorporated is more realistic than models that assume deterministic or straight paths, and the analytical methods discussed in Section 2 can

provide quick metrics without requiring complex numerical integration. The simulated model also provides an intuitive method of updating the distribution by aggregating weighted Brownian bridge paths into heatmaps, and does not require parameterisation over a discretised location space in building the model. Finally, the flexibility in the choice of weight function allows for different types of intelligence updates to be incorporated without affecting the overall computational ease of the method.

We have developed many enhanced scenarios using this model for practical analysis purposes, which are not included here for brevity. For example, the trajectory of the target can be more complex with intermediate waypoints between start and end points. Additionally, multiple possible target trajectories can be considered to create bimodal heatmap distributions. More recently, we have compared different sensor placement configurations applied to the temporal heatmaps to improve the probability of observing the target, and more formal optimisation methods are part of our future work.

Acknowledgements

We are extremely grateful to the Center for Multi-Intelligence Studies at the Naval Postgraduate School for their support of this work.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Dashi I. Singham  <http://orcid.org/0000-0003-4201-4585>

Michael P. Atkinson  <http://orcid.org/0000-0001-6606-8188>

References

- Angermann, M., Kammann, J., Robertson, P., Steingäß, A., & Strang, T. (2001). Software representation for heterogeneous location data sources using probability density functions. In *International symposium on location based services for cellular users (locellus 2001)*.
- Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 174–188. doi:10.1109/78.978374
- Atkinson, M. P., & Singham, D. I. (2015). Multidimensional hitting time results for Brownian bridges with moving hyperplanar boundaries. *Statistics & Probability Letters*, 100, 85–92. doi:10.1016/j.spl.2015.02.006
- Baumgartner, K. A., Ferrari, S., & Wettergren, T. A. (2009). Robust deployment of dynamic sensor networks for cooperative track detection. *IEEE Sensors Journal*, 9(9), 1029–1048. doi:10.1109/JSEN.2009.2025836
- Ben-Zvi, T. (2017). Learning automata decision analysis for sensor placement. *Journal of the Operational Research Society*, 69(9).
- Ben-Zvi, T., & Nickerson, J. V. (2012). Intruder detection: An optimal decision analysis strategy. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2), 249–253. doi:10.1109/TSMCC.2011.2126043
- Brown, G., Carlyle, M., Abdul-Ghaffar, A., & Kline, J. (2011). A defender-attacker optimization of port radar surveillance. *Naval Research Logistics*, 58(3), 223–235. doi:10.1002/nav.20423
- Brown, S. S. (1980). Optimal search for a moving target in discrete time and space. *Operations Research*, 28(6), 1275–1289. doi:10.1287/opre.28.6.1275
- Buchin, K., Sijben, S., Arseneau, T., & Willems, E. P. (2012). Detecting movement patterns using Brownian bridges. In *Proceedings of the 20th international conference on advances in geographic information systems* (pp. 119–128).
- Bullard, F. (1999). *Estimating the home range of an animal: A Brownian bridge approach* (Master's thesis). The University of North Carolina, Chapel Hill.
- Cheng, C. C. (2016). *A Brownian bridge movement model to track mobile targets* (Master's thesis). Naval Postgraduate School.
- Chew, M. C. Jr. (1973). Optimal stopping in a discrete search problem. *Operations Research*, 21(3), 741–747. doi:10.1287/opre.21.3.741
- Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(3), 656–704.
- Eagle, J. N., & Yee, J. R. (1990). An optimal branch-and-bound procedure for the constrained path, moving target search problem. *Operations Research*, 38(1), 110–114. doi:10.1287/opre.38.1.110
- Gentil, I., Rémillard, B., & Del Moral, P. (2005). Filtering of images for detecting multiple targets trajectories. In *Statistical modeling and analysis for complex data problems* (pp. 267–280). Springer.
- Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2), 107–113. doi:10.1049/ip-f-2.1993.0015
- Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7), 53–82. doi:10.1109/MAES.2010.5546308
- Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., & Nordlund, P.-J. (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2), 425–437. doi:10.1109/78.978396
- Horne, J. S., Garton, E. O., Krone, S. M., & Lewis, J. S. (2007). Analyzing animal movements using Brownian bridges. *Ecology*, 88(9), 2354–2363. doi:10.1890/06-0957.1
- Jian, J.-Y., Matsuka, T., & Nickerson, J. V. (2006). Recognizing deception in trajectories. In *28th annual conference of the cognitive science society* (pp. 1563–1568).

- Kadane, J. B. (1971). Optimal whereabouts search. *Operations Research*, 19(4), 894–904. doi:10.1287/opre.19.4.894
- Karatzas, I., & Shreve, S. (1998). *Brownian motion and stochastic calculus*. Springer.
- Le Cadre, J.-P., & Souris, G. (2000). Searching tracks. *IEEE Transactions on Aerospace and Electronic Systems*, 36(4), 1149–1166. doi:10.1109/7.892665
- Lersteau, C., Rossi, A., & Sevaux, M. (2016). Robust scheduling of wireless sensor networks for target tracking under uncertainty. *European Journal of Operational Research*, 252(2), 407–417. doi:10.1016/j.ejor.2016.01.018
- Miller, A., & Moskowitz, I. (1996). Generalizations of the Carlton-Kimball distribution for a target's future location. *Computers & Mathematics with Applications*, 31(8), 61–68. doi:10.1016/0898-1221(96)00031-4
- Mooshegian, M. (2013). *A probabilistic model of illegal drug trafficking operations in the eastern pacific and Caribbean sea* (Unpublished master's thesis). Naval Postgraduate School.
- Moskowitz, I., & Simmen, J. (1989). Asymptotic results in search theory. *Naval Research Logistics (NRL)*, 36(5), 577–596. doi:10.1002/1520-6750(198910)36:5<577::AID-NAV3220360504>3.0.CO;2-G
- Nunez, J. (2017). *Particle Filtering Methods for Incorporating Intelligence Updates* (Master's thesis). Naval Postgraduate School.
- Pietz, J., & Royset, J. (2013). Generalized orienteering problem with resource dependent rewards. *Naval Research Logistics (NRL)*, 60(4), 294–312. doi:10.1002/nav.21534
- Pietz, J., & Royset, J. O. (2015). Optimal search and interdiction planning. *Military Operations Research*, 20(4), 59–73.
- Pozdnyakov, V., Meyer, T., Wang, Y.-B., & Yan, J. (2014). On modeling animal movements using Brownian motion with measurement error. *Ecology*, 95(2), 247–253. doi:10.1890/13-0532.1
- Przybyla, J., Taylor, J., & Zhou, X. (2010). Locating sensors for detecting source-to-target patterns of special nuclear material smuggling: A spatial information theoretic approach. *Sensors*, 10(9), 8070–8091. doi:10.3390/s100908070
- Royset, J. O., & Sato, H. (2010). Route optimization for multiple searchers. *Naval Research Logistics (NRL)*, 57(8), 701–717. doi:10.1002/nav.20432
- Simonin, C., Le Cadre, J.-P., & Dambreville, F. (2009). A hierarchical approach for planning a multisensor multi-zone search for a moving target. *Computers & Operations Research*, 36(7), 2179–2192. doi:10.1016/j.cor.2008.08.007
- Sklar, M. G., & Ladany, S. P. (1993). Properties of a source location estimator in the plane. *Naval Research Logistics (NRL)*, 40(2), 211–228. doi:10.1002/1520-6750(199303)40:2<211::AID-NAV3220400206>3.0.CO;2-4
- Stone, L. D., & Kadane, J. B. (1981). Optimal whereabouts search for a moving target. *Operations Research*, 29(6), 1154–1166. doi:10.1287/opre.29.6.1154
- Stone, L. D., Royset, J. O., & Washburn, A. R. (2016). *Optimal Search for Moving Targets*. Springer.
- The MathWorks (2016). *Matlab 2016b*. Retrieved from <https://www.mathworks.com/products/matlab.html>
- Thomas, L. C., & Hulme, P. (1997). Searching for targets who want to be found. *Journal of the Operational Research Society*, 48(1), 44–50. doi:10.1057/palgrave.jors.2600319
- Turchin, P. (1991). Translating foraging movements in heterogeneous environments into the spatial distribution of foragers. *Ecology*, 72(4), 1253–1266. doi:10.2307/1941099
- Vermeulen, J., & van den Brink, M. (2005). The search for an alerted moving target. *Journal of the Operational Research Society*, 56(5), 514–525. doi:10.1057/palgrave.jors.2601847
- Washburn, A. R. (2002). *Search and detection* (4th ed). INFORMS.