A Multi-Robot Coordination Methodology for Wilderness Search and Rescue

by

Ashish Macwan

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy

Department of Mechanical and Industrial Engineering
University of Toronto

A Multi-Robot Coordination Methodology for Wilderness Search and Rescue

Ashish Macwan

Doctor of Philosophy

Department of Mechanical and Industrial Engineering

University of Toronto

2013

Abstract

One of the applications where the use of robots can be beneficial is Wilderness Search and Rescue (WiSAR), which involves the search for a possibly mobile but non-trackable lost person (i.e., the target) in wilderness environments. A *mobile target* implies that the search area grows continuously and potentially without bound. This fact, combined with the presence of typically rugged, varying terrain and the possibility of inclement weather, poses a considerable challenge to human Search and Rescue (SAR) personnel with respect to the time and effort required to perform the search and the danger entailed to the searchers. Mobile robots can be advantageous in WiSAR due to their ability to provide consistent performance without getting tired and their lower susceptibility to harsh weather conditions compared to humans. Thus, a coordinated team of robots that can assist human SAR personnel by autonomously performing searches in WiSAR scenarios would be of great value. However, to date, a suitable multi-robot coordination methodology for autonomous search that can satisfactorily address the issues relevant to WiSAR is lacking.

The objective of this Dissertation is, thus, to develop a methodology that can autonomously coordinate the search strategy of a multi-robot team in wilderness environments to locate a moving target that is neither continuously nor intermittently observed during the search process. Three issues in particular are addressed: (*i*) target-location prediction, (*ii*) robot deployment, and (*iii*) robot-path planning. The corresponding solution approaches devised to address these issues incorporate the influence of varying terrain that may contain *a priori* known and unknown obstacles, and deal with unique target physiology and psychology as well as found clues left behind by the target. The solution methods for these three tasks work seamlessly together resulting in a tractable MRC methodology for autonomous robotic WiSAR.

Comprehensive simulations have been performed that validate the overall proposed methodology. Moreover, the tangible benefits provided by this methodology were further revealed through its comparison with an alternative search method.

Table of Contents

List of Tables

List of Figures

Nomenclature and Acronyms

Latin Letters

$\{a\}_i$ Set of variables describing the deployment position assigned to the i^{th} search robot

$\{a_k, b_k\}$ Regression coefficients of the linear regression line that yields the estimated height values, $\hat{h}$, which are used in the computation of terrain influence when position the k^{th} control point along a given ray.

A_R Return-time attribute, yielding a measure of the time required by the robots to reach their newly-assigned deployment positions

$\tilde{A}_R$ Normalized return-time metric

A_S Success-rate attribute, yielding a measure of the probability of finding the target

$\tilde{A}_S$ Normalized success-rate metric

A_T Search-time attribute, yielding a measure of the expected total time required to find the target

$\tilde{A}_T$ Normalized search-time metric

Atr_k k^{th} attribute of the search process that is considered in formulating the objective function for robot deployment optimization

B_{dep} Objective function of the robot deployment optimization representing a measure of benefit to the ongoing search process that is entailed by a given valuation of all the deployment decision variables, based on a combination of the influences of all the attributes of the search process that are considered

c_{ip} Time that would be required by robot i to reach the p^{th} iso-probability curve via the shortest path

c_{ipj} Cost parameter representing the time that it would take for the i^{th} robot to travel to the j^{th} deployment position on the p^{th} iso-probability curve via the shortest-path, used in the inner optimization loop of the robot deployment optimization that determines the optimal assignment of robots to a selected set of iso-probability curves

$\bar{c}_{ipj}$ An element of the threshold matrix, $\bar{C}$, located at row i and column p_j

c_0^* Lower cost threshold value used in the Threshold Algorithm used to solve the LBAP that determines the optimal assignment of robots to a selected set of iso-probability curves in the inner optimization loop of the robot deployment optimization problem

c_1^* Upper cost threshold value used in the Threshold Algorithm used to solve the LBAP that determines the optimal assignment of robots to a selected set of iso-probability curves in the inner optimization loop of the robot deployment optimization problem

$\bar{C}$ The threshold matrix used in the Threshold Algorithm used to solve the LBAP that determines the optimal assignment of robots to a selected set of iso-probability curves in the inner optimization loop of the robot deployment optimization problem

Cf 'Crow's-flight' distance data-value, recorded from a past lost-person search incident corresponding to a particular target-type category

C_p Circumferential length of the p^{th} iso-probability curve

$\mathcal{C}(t)$ The particular bounded region representing the search area at time t

$d_{est,k}$ Estimated radial distance of the k^{th} control point from the LKP along a given ray, $k \in [1, N_{curves}]$; computed during iterative process to incorporate terrain influence into iso-probability curves

$d_{est\_NEW,k}$ A revised estimate of $d_{est,k}$, obtained from running another iteration of the procedure to compute terrain influence for positioning the k^{th} control point on a given ray

$d_{Rb,total}$ Conservative estimate of total distance that a search robot could travel in $T_{MaxLimit}$ time

Δd_{max} Threshold representing maximum allowable percentage difference between successive estimated radial distance values, $d_{est,k}$ and $d_{est\_NEW,k}$, above which another iteration is required in determining the position of the control point under consideration

D_c Inter-curve cumulative probability interval, equal to the difference in cumulative probability between any two adjacent iso-probability curves

e_i A measure of target-detection effectiveness associated with the i^{th} search robot

$E_{C3,HI}$ Upper bound of confidence interval for the arrival-time estimation error for the remaining search path, computed for the purposes of performing path-optimality Check #3 at a check-point on a given search path

$E_{C3,LO}$ Lower bound of confidence interval for the arrival-time estimation error for the remaining search path, computed for the purposes of performing path-optimality Check #3 at a check-point on a given search path

$E_{C3,Max}(j)$ Function yielding upper threshold on arrival-time estimation error to which $E_{C3,HI}$ is compared during path-optimality Check #3, conducted at the j^{th} check-point along a given search path

$E_{C3,Min}(j)$ Function yielding lower threshold on arrival-time estimation error to which $E_{C3,LO}$ is compared during path-optimality Check #3, conducted at the j^{th} check-point along a given search path

Eff Attribute of the search process representing a measure of effort required to implement a given assignment of robots to deployment positions

E_k Measure of curve-fitting error for the k^{th} iso-probability curve constructed from the piecewise cubic polynomial set, F_k

E_{TOTAL} Measure of total curve-fitting error for all iso-probability curves constructed from their corresponding piecewise cubic polynomial sets, F_k, $k \in [1, N_{curves}]$,

$f_{k,i}$ The i^{th} cubic polynomial segment in the set of piecewise cubic polynomials that are used to construct the k^{th} iso-probability curve

$f_{m,Rbi}$ Function yielding some measure of mobility of the i^{th} search robot over given terrain topology

$f_{phys}(\cdot)$ Function representing general relationship between terrain topology and the corresponding physiological response from the target

$f_{U\Phi}$ Function yielding some measure of uncertainty associated with a given search path (i.e., a measure of 'error' between where the path leads the robot and the region within which the path must lie)

$F_{dest,i}$ The future propagated (destination) iso-probability curve assigned to the i^{th} search robot

F_k Continuous, 2D, closed curve composed of piecewise cubic polynomials, representing the k^{th} iso-probability curve

$F_{TRUE,k}$ Continuous, 2D, closed curve composed of piecewise cubic polynomials, representing the *true* k^{th} iso-probability curve, due to the use of a relatively larger number of rays (360 rays in this Thesis)

$g_{clu}(\cdot)$ Function representing influence of found clues on probabilistic target motion prediction

$g_{psy}(\cdot)$ Function representing the influence of lost-person psychological behaviours pertinent to a given target on probabilistic target motion prediction

$g_{ter}(\cdot)$ Function representing influence of terrain on probabilistic target motion prediction

$\hat{h}$ Function yielding an estimated terrain height value (for a specified radial distance value along a given ray) based on a regression line fitted to a set of terrain height measurements

h_i — The ith terrain height measurement, used in the computation of the linear regression line representing estimated terrain heights along a particular length of straight-line travel on a ray

IP_k — kth interpolation point along a given robot search path

$I_{targ}(t)$ — Dataset representing all available information (qualitative and quantitative) about the target at time t

K — Variable value representing the 'raw' resource requirement, $N_{r,1}$, for the first iso-probability curve, used during the iterative process to determine the preferred resource requirements for each of a given set of iso-probability curves

ΔK — Increment value for variable K, used during the iterative process to determine the preferred resource requirements for each of a given set of iso-probability curves

L_k — Distance along any given ray, measured from $r = 0$ at the LKP up to the kth control point on that ray

L_{max} — Maximum distance along any given ray, corresponding to the length measured from $r = 0$ at the LKP up to the farthest control point defined on that ray

$L_{p,i}$ — Distance along the ith ray, measured from $r = 0$ at the LKP up to the position of the control point corresponding to the pth iso-probability curve

$\{m_1, b_1\}$ — Coefficients of the linear functional relationship, $q_{dec}(\gamma)$

$\{m_2, b_2\}$ — Coefficients of the linear functional relationship, $q_{inc}(\gamma)$

M — Margin-of-error, represented as a percentage of Δt_{avail}, added onto the ideal arrival-time via the alternative path for path-optimality Check #1, or onto the ideal traversal-time via the alternative path for path-optimality Check #2

$\mathcal{M}$ — Terrain topology information

M_{END} — Margin-of-error, represented as a percentage of Δt_{prop}, used to compute the upper and lower thresholds, $E_{C3,Max}$ and $E_{C3,Min}$, respectively, at the second-last check-point for any given search path

M_{START} — Margin-of-error, represented as a percentage of Δt_{prop}, used to compute the upper and lower thresholds, $E_{C3,Max}$ and $E_{C3,Min}$, respectively, at Check-point #2 for any given search path

N_{atr} — Total number of attributes of the search process that are considered in the formulation of the objective function for robot deployment optimization

$N_{catData}$ — Total number of $\{Cf, \Delta t_{TOT}\}$ data-value pairs available from past lost-person search incidents involving a lost person of a given target-type category

N_{clu_t} Total number of clues found during a search by time t

N_{curves} Total number of iso-probability curves used for target-location prediction

$N_{curves,HI}$ Number of iso-probability curves, in the range from 1 to $N_{curves,max}$, that would result in the highest possible optimal return-time value

$N_{curves,LO}$ Number of iso-probability curves, in the range from 1 to $N_{curves,max}$, that would result in the lowest possible optimal return-time value

$N_{curves,max}$ Maximum number of iso-probability curves that can be defined, given values for N_r, $N_{r,p}$, R_c, and D_c

N_h Total number of height measurements, h_i, used in computing the linear regression line, $\hat{h}$

N_{IP} Total number of interpolation points used to construct a given search path

N_r Total number of search robots available to conduct the search

$N_{r,p}$ Total number of search robots that must be assigned to the p^{th} iso-probability curve

N_{rays} Total number of rays used to construct a set of iso-probability curves

$N_{returnTrips}$ Conservative estimate of the total number of return-trips of length ($2r_{SA}$) that any given search robot can be expected to make during the available $T_{MaxLimit}$ search time

$N_{\mathcal{R}e}$ Total number of regions for search-effort allocation within the search area

N_{SD} Total number of subdivisions that a given search path has been divided into for the purposes of estimating traversal-time and for designating path-optimality check-points

$p(\theta)$ PDF yielding probability density value for target-travel direction likelihood for a specified angular heading-direction of target travel, θ

$p(r, t, \theta_i)$ 1D nominal mean target-location PDF, corresponding to the i^{th} ray with angular position, θ

$p(v)$ 1D nominal mean target-speed PDF

$p_1(\theta)$ Function for p(θ) based on a linear PDF model, and applicable for values of θ in the range: ($-180° \leq \theta \leq 0°$)

$p_2(\theta)$ Function for p(θ) based on a linear PDF model, and applicable for values of θ in the range: ($-180° \leq \theta \leq 0°$)

$P_{dest,i}$ Final (destination) position for the search path assigned to the i^{th} search robot

P_{Dj} Deployment position assigned to j^{th} search robot

$P_{est,k}$ Coordinates of the estimated position of the k^{th} control point along a given ray, $k \in [1, N_{curves}]$; computed during iterative process to incorporate terrain influence into iso-probability curves

$P_{i,k}$ Position coordinates of a point at the intersection of the curve F_k with a ray having angular position θ_i

P_{MOT} Probabilistic information about the motion of the target

P_{POS} Probabilistic information about the location of the target within the search area

P_{Rbi} Position coordinates of i^{th} search robot

$P_{TRUE,i,k}$ Position coordinates of a point at the intersection of the curve $F_{TRUE,k}$ with a ray having angular position θ_i

$q(\gamma)$ Function representing estimated target speed scale-factor for a given estimated terrain slope, γ

$q_{dec}(\gamma)$ Linear functional relationship yielding estimated target speed scale-factor for declined slopes ($\gamma < 0°$)

$q_{dirPref}(\theta_i)$ Function yielding scale factor to be applied to the nominal mean target-speed PDF associated with ray θ_i when accounting for the influence of the target psychological behavior of a preferred direction of target travel

$q_{inc}(\gamma)$ Linear functional relationship yielding estimated target speed scale-factor for inclined slopes ($\gamma \geq 0°$)

q_k Speed scale-factor computed for k^{th} control point along a given ray, using a specified estimated terrain slope value, γ_k, with the function $q(\gamma)$

$q_{max,dec}$ Speed scale-factor corresponding to $\gamma_{max,dec}$

$q_{max,inc}$ Speed scale-factor corresponding to $\gamma_{max,inc}$

q_{true} Scale-factor applied to the assumed mean target-speed PDF in order to obtain the true PDF for use in the simulations performed for the robustness tests

Q_H Ratio of high-likelihood target-travel direction probability to low-likelihood target-travel direction probability

r Random variable for straight-line target-distance from LKP

r_s Sensing range of a search robot

r_{SA} Radius of the circular search area established for the alternative, deterministic search method, used for comparison with the probabilistic method proposed in this Thesis

$r_{T,\rho\%}$ Target position point, along a ray relative to the LKP, corresponding to a cumulative probability value of ρ% based on the mean target-location PDF at time $t = T$

$\mathbb{R}$ The set of real numbers

$\boldsymbol{R}^3$ Real coordinate space of dimension 3

R_c Centre position of a set of iso-probability curves, equal to the cumulative probability measured up to the position along a ray corresponding to the highest probability density value (R_c = 50% for a normal distribution)

R_{max} Upper bound on the position of the iso-probability curves, equal to the largest cumulative probability value possible (R_{max} = 100% for a PDF with finite upper bound, or the upper truncation point for a boundless PDF)

R_{min} Lower bound on the position of iso-probability curves, equal to the smallest (non-zero) cumulative probability value allowable, as specified by the user

R_p Position of the p^{th} iso-probability curve, equal to the cumulative probability to which that curve corresponds

Rb_i The i^{th} search robot

$\mathcal{R}e_i$ Search-effort allocation region to which the i^{th} search robot is assigned

s_ε Sample standard deviation of subdivision traversal-time error data gathered by a search robot up to its current position during search-path implementation

s_k Length along a specified search path, measured from the start of the path up to the point where the path crosses the k^{th} ray

s_{TOT} Total length of a specified search path

$\mathcal{SA}$ Search area; the bounded region within which the target is presumed to be located

t Time variable

Δt_{avail} Total time available for a search robot to reach its destination curve (equal to the total time until the next iso-probability curve propagation)

$\Delta t_{C2,Max}$ Maximum path-traversal-time threshold used for comparison with $\Delta\tau_{trav,C2}$ during path-optimality Check #2 at a check-point on a given search path

Δt_{prop} Time-interval between updates of the target-location prediction information represented by the iso-probability curves

$\Delta t_{trav,j}$ Actual (observed) time required by a search robot to traverse the j^{th} subdivision of its assigned search path

Δt_{TOT} Total travel-time data-value, corresponding to a particular *Cf* data-value, representing the total time for which the target from a past lost-person search incident had travelled from the LKP before he/she was found

$T_{\varepsilon,HI}$ Upper limit of confidence interval for the subdivision traversal-time estimation error for a given search path

$T_{\varepsilon,LO}$ Lower limit of confidence interval for the subdivision traversal-time estimation error for a given search path

$T_{C1,HI}$ Upper limit of confidence interval for the arrival-time at the destination curve via the alternative path constructed for path-optimality Check #1 at a check-point on a given search path

$T_{C1,LO}$ Lower limit of confidence interval for the arrival-time at the destination curve via the alternative path constructed for path-optimality Check #1 at a check-point on a given search path

$T_{C1,max}$ Maximum arrival-time threshold used for comparison with $T_{C1,HI}$ during path-optimality Check #1 at a check-point on a given search path

T_{clu_drp} Time at which a clue was dropped by target

$\hat{T}_{clu\_drp}$ Estimated time at which a clue was dropped by target

T_{clu_fnd} Time at which a clue was found by a search robot

T_f The future point in time at which the next deployment optimality check is to be conducted

Tf_1 Transfer function yielding the probabilistic target motion information based on all relevant influences considered

Tf_2 Transfer function yielding the probabilistic target location information for a particular time based on the probabilistic target motion information obtained

T_{hs} Amount of time corresponding to the head-start of the target over the search robots from time $t = 0$

$T_{MaxLimit}$ Maximum search time-limit

T_{rem} Time difference between the time at which a given clue was found and the true or estimated time at which the clue was dropped

$\vec{u}_{P_{dest},1}$ Unit vector in the direction of the orientation of the centre robot when it reaches its destination position on the current innermost iso-probability curve (i.e., the destination position for the first part of its path)

$\{U_{terr}\}$ Dataset representing uncertainty information pertaining to the available terrain data

v Random variable for mean target-speed random variable

v_{max} Upper bound of the nominal mean target-speed PDF, representing the maximum possible nominal mean speed that the target can attain

$v_{Rb,max}$ Maximum speed that can be attained by a search robot

$v_{\rho\%}$ Mean target-speed corresponding to a cumulative probability value of ρ% based on the mean target-speed PDF

w_S Weighting applied to the success-rate term in Z_1

w_T Weighting applied to the modified-search-time term in Z_1

(x, y, z) Cartesian coordinates of a point in the search area

$(x_{clu}, y_{clu}, z_{clu})$ Position coordinates of a found clue

x_{ipj} Binary decision variable representing the yes/no decision to assign the i^{th} robot to the j^{th} deployment position on the p^{th} iso-probability curve, used in the inner optimization loop of the robot deployment optimization that determines the optimal assignment of robots to a selected set of iso-probability curves

$\mathbb{Z}$ The set of integers

Z_1 Objective function for the overall robot deployment optimization

Z_1^* Optimal value of the objective function, Z_1

Z_2 Objective function for the inner optimization loop within the overall robot deployment optimization that determines the optimal assignment of robots to a selected set of iso-probability curves

Z_2^* Optimal value of the objective function, Z_2, yielding the minimized return-time attribute, A_R

$\Delta Z_{1,max}$ Threshold representing the maximum allowable percentage difference between the Z_1 value for the current set of iso-probability curves being used and the Z_1^* value for the optimal set of iso-probability curves computed at a deployment optimality check-point; a percentage difference greater that $\Delta Z_{1,max}$ indicates that a re-deployment is required

Greek Letters

α_k	Angular distance between a ray passing through the initial point on a given search path and the ray, θ_k, along which the kth interpolation point needs to be placed
α_{TOT}	Angular distance between a ray passing through the initial point on a given search path and a ray passing through the destination point on that path
β	Angular increment for the outward trajectory from the LKP that any given search robot would travel along when implementing the alternative, deterministic search method
γ	Angular value of estimated terrain slope over a particular length of travel (e.g., over a straight line of travel along a given ray, or over a particular subdivision of a search robot's planned search path)
$\gamma_{max,dec}$	Maximum possible decline angle over which target can travel
$\gamma_{max,inc}$	Maximum possible incline angle over which target can travel
Δ_{LKP}	Straight-line distance between a newly-established LKP due to a found clue and the previous LKP
Δ_{ε}	Deviation around $\bar{\varepsilon}_{trav}$, computed to build a $\pm 3\sigma$ confidence interval for the subdivision traversal-time estimation error for a given search path
$\Delta\tau_{trav}$	Estimated time required by a search robot to traverse its entire assigned search path
$\Delta\tau_{trav,C1}$	Path-traversal-time estimate for the straight-line shortest path used for conducting path-optimality Check #1 at a check-point on a given search path
$\Delta\tau_{trav,C2}$	Path-traversal-time estimate for the alternative path constructed for path-optimality Check #2 at a check-point on a given search path
$\Delta\tau_{trav,j}$	Estimated time required by a search robot to traverse the jth subdivision of its assigned search path
$\bar{\varepsilon}_{trav}$	Estimate of the average error in the estimated traversal-time per path subdivision of a given search path due to terrain variations
$\varepsilon_{trav,j}$	Error in the estimated traversal-time for the jth subdivision of a search path assigned to a search robot
η_k	Path-segmentation ratio, used in positioning the kth interpolation point along a given search path
$\hat{\eta}_k$	Angular estimate of path-segmentation ratio, η_k, corresponding to the kth interpolation point along a given search path

θ Angular position of a ray, which represents a possible heading-direction of target travel

$\theta_i^{[k]}$ The k^{th} outward trajectory from the LKP that the i^{th} search robot would travel along when implementing the alternative, deterministic search method

θ_{hi} High-likelihood direction of target travel when the psychological behaviour of a preferred direction of travel is applicable

θ_{lo} Low-likelihood direction of target travel when the psychological behaviour of a preferred direction of travel is applicable

$\mu_{r,t}$ Mean of the nominal mean target-speed PDF

μ_v Mean of the nominal mean target-speed PDF

$\rho\%$ Cumulative probability value computed for a given mean target-speed PDF or target-location PDF

$\rho_{p,i}$ Probability area under the target-location PDF, established on the i^{th} ray, that is swept by a search robot travelling along the p^{th} iso-probability curve with a target-detection radius of r_s extend outwards in all directions from the centre of the robot (i.e., the probability area under this PDF between positions $(L_{p,i} - r_s)$ and $(L_{p,i} + r_s)$)

$\sigma_{r,t}^2$ Variance of the nominal mean target-speed PDF

σ_v^2 Variance of the nominal mean target-speed PDF

$\{\Phi\}_i$ Set of variables describing the search path assigned to the i^{th} search robot

Acronyms

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
CASP	Computer-Assisted Search Planning
EGLD	Extended Generalized Lambda Distribution
GB	Generalized Bootstrap
GLD	Generalized Lambda Distribution
GSO	Generalized Search Optimization
KS	Knowledge Source

LBAP	Linear Bottleneck Assignment Problem
LKP	Last Known Position
MRC	Multi-Robot Coordination
NP-Hard	Reference to the computational complexity of an optimization problem, indicating that the time required to find the optimal solution is a Non-Polynomial function of the number of variables involved in the problem.
PDF	Probability Density Function
PRM	Probabilistic Roadmap
RFID	Radio-frequency Identification
RRT	Rapidly-Exploring Random Tree
SAR	Search and Rescue
UAV	Unmanned Aerial Vehicle
USAR	Urban Search and Rescue
WiSAR	Wilderness Search and Rescue

Chapter 1

1 Introduction

The use of autonomous robot teams has been shown to be beneficial in a number of applications. Some examples include area-exploration (coverage), surveillance, mapping, foraging, and demining [1-6]. The main advantage that robots provide in such applications is their ability to perform tasks in an expeditious and consistent manner. An additional application area for the use of multiple robots that has been receiving increasing attention in recent years is Search and Rescue (SAR). SAR missions have been further classified into two types: Urban Search and Rescue (USAR) and Wilderness Search and Rescue (WiSAR). In USAR, the objective is to search for survivors trapped underneath the rubble of collapsed structures, typically occurring after some type of natural or man-made disaster [7, 8]. Thus, the survivors that need to be found are stationary. In addition, the search area is bounded, extending up to the limits of the site of the collapsed structure(s).

In contrast to USAR, the WiSAR problem involves the search for a potentially moving lost-person (i.e., the target) in wilderness environments [9-11]. Throughout the search, the target is never directly observed by the searchers, either continuously or intermittently, since, naturally, once the target has been sighted, he/she would be considered to be found and the search would be complete. Hence, one does not have any sensor readings to use during the search in order to attempt to track and pursue the target. Instead, one must rely on predictions of the target's motion and possible location at any given time in order to conduct the search. Moreover, a moving target implies a search area that grows with time, and potentially without bound. These characteristics present rescuers with a significantly different problem than that for USAR. The main challenge in WiSAR, therefore, is to locate a possibly moving, unobservable (i.e., non-trackable) target, whose state at any given time is unknown, but can, nevertheless, be predicted through the use of available probabilistic information.

Robots can be particularly useful in SAR applications due to their ability to: effectively move over rugged terrains, access highly confined areas, and provide consistent operation (i.e., not get tired or adversely affected by harsh climatic conditions the way humans do). These and other benefits have been commonly cited in the literature [9, 12-15], with a general consensus that the efficiency and success-rate of SAR operations can be significantly improved through the increased automation of the search process. It is envisioned that in a not-too-distant future, human SAR personnel will be augmented with teams of autonomous robots, at least for the search aspect of the operation. Thus far, research on the use of robots for SAR has focussed on a number of issues related to the use of single or multiple robots for the USAR domain, including robots that are tele-operated as well as those with varying levels of autonomy. The issues addressed include robot locomotion system design, formation of effective mixed human-robot teams, robot exploration and navigation, and real-time 3D mapping and landmark identification [16-36].

The use of rescue robots for WiSAR has not yet been explored as widely as in USAR [12, 37-39]. Robots can, however, be advantageous for the WiSAR domain as well. Since robots can be designed to endure various harsh weather conditions, their use can help reduce the risk to human lives by decreasing the number of human SAR personnel that need to be used and/or decreasing the amount of time that human searchers need to spend exposed to the elements outside. Furthermore, certain robots, by virtue of their design, may also be able to access and navigate over difficult outdoor terrains that humans cannot. For example, Unmanned Aerial Vehicles (UAVs) would be able to search over rugged, steep, mountainous regions that a human searcher either cannot negotiate as quickly and easily, or cannot access at all. As well, a ground-based robot of smaller size relative to a human would be able to access a cave (which a small child may have crawled into for shelter) if the entrance is too small for a human searcher to pass, or if one is uncertain about any potential dangers inside. Moreover, while individual robots can provide advantages such as these, a team of robots, carrying out a search autonomously, could further increase the efficacy of the search, and would, thus, provide even greater benefit. This opens up an interesting yet challenging research area, where it would be a worthwhile endeavour to develop a methodology that can optimally coordinate a team of multiple robots, on-line, to autonomously perform an effective search in WiSAR scenarios.

To date, existing works aimed at WiSAR robotics have mainly focussed on issues related to their implementation as tele-operated, assistive robots for human SAR personnel [12, 37-41]. Moreover, although other areas of research also exist that address multi-agent search methods, none are able to provide a solution that satisfactorily addresses the unique challenges of WiSAR, while also lending itself to automation by a team of multiple robots. In particular, for one, Search

Theory literature contains research dealing with the use of analytical techniques to solve an optimization formulation of a generalized search problem. The methods devised are not necessarily for use with robots, as the optimal search strategy that is determined is expressed in terms of an abstract quantity called "search effort", coupled with specification of the amounts of and the locations where this search effort should be applied [42-44]. A key assumption that is made in these methods is that a 2D probability density function for the location of the target over the search area, as well as a corresponding probabilistic motion model, are known *a priori*. The literature on Multi-Robot Coordination (MRC) research also includes methods for coordinating a multi-robot search [45-49]. However, most of the methods presented are intended to be general coordination mechanisms and are not aimed at the WiSAR application in particular. As such, they make certain simplifying assumptions such as stationary targets, *a priori* known probability information, and/or bounded environments, which limit their usefulness for WiSAR.

1.1 Thesis Statement

It is evident from the literature, then, that autonomous, coordinated, multi-robot WiSAR, especially for ground-based search operations, is an expanding research field. There is a clear need to develop a methodology to autonomously coordinate the search strategy of a team of robots for WiSAR searches. Such a methodology, however, must be able to address the unique challenges of this application. In particular, these are:

i. a non-trackable, possibly mobile target,

ii. difficulty in predicting the exact motion of the particular target sought,

iii. varying terrain that may contain *a priori* known and unknown obstacles,

iv. inexact terrain topology information (low resolution of available terrain information),

v. a time-varying and potentially boundless search area,

vi. unique target physiology (impacting the target's motion over varying terrain),

vii. found clues that are left behind by the target.

Existing methods are unable to address these challenges concurrently, and on-line, with a single solution method. In addition, one of the main difficulties in this problem is that the target is non-trackable. Thus, unless found, he/she is neither continuously nor intermittently within the sensing range of any of the robots. As such, in order to conduct effective coordination of the search resources (i.e., the robots), it would be necessary to derive and utilize some prediction of the location of the target within the search area at any given time. Due to the dynamic nature of the WiSAR scenario, and to account for possible target motion, this prediction would also have to be updated as time passes. Once such a prediction is derived, it can, then, be used to guide how the search effort is distributed throughout the search. Therefore the objective of this Thesis is to *develop a novel on-line-feasible multi-robot coordination methodology for the autonomous, optimal navigation of multiple robots to conduct a search for a potentially moving target in WiSAR scenarios, using time-varying predictions of the target's location*.

Although the use of the term WiSAR implies that the overall problem involves both *searching* for the lost-person and *rescuing* that person when found, this Thesis will only focus on the search problem. Thus, the rescue component of WiSAR is not within the scope of this research. Moreover, it is clear that there are significant technological and design challenges that must be addressed in order to have robots that are capable of traversing some of the more difficult wilderness terrains and performing the types of sensing tasks that are required for implementing the search methodology proposed in this Thesis. These challenges point to yet another possible area of research and development in robotics that would be worthwhile. Nevertheless, the design of physical rescue robots is also beyond the scope of this Thesis. The research conducted for this Thesis focuses only on developing the MRC methodology required to perform an autonomous search in WiSAR scenarios. Before describing the MRC problem for WiSAR in greater detail and presenting the proposed solution approach, a review of the pertinent literature is provided in the following section.

1.2 Literature Review

In the following, a review is first provided of research that has been conducted pertaining to search methods within the areas of search theory and MRC. In addition, a review of the literature on target-location prediction, robot deployment, and robot path-planning is also provided, as these topics are relevant to the methodology proposed in this Thesis.

1.2.1 Search Theory

Search theory encapsulates the body of work dealing with analytical methods for determining the optimal allocation of search effort for locating a stationary or moving target and is, in fact, a discipline within the field of Operations Research [44, 50]. The basic tenets and general approach of search theory methods are the same, the ground-work for which was established in the seminal work by B. O. Koopman during World War II in an attempt to establish theory to help guide search strategies for locating enemy submarines in marine warfare applications [42]. Over the years, continued research has built on this theory, extending its applicability to general SAR operations.

Search theory methods apply probability theory to develop analytical solutions to the search problem. In these methods, the term ‘search effort’ refers to an abstract quantity that measures the level of search-resource usage [42, 44]. Typically, this measure takes the form of the total length of distance to be travelled to search within a specified region, or the amount of time that should be spent searching in that region, and is assumed to be ‘infinitely divisible’ (i.e., divisible into as fine increments as desired). It is also assumed that this search effort can be moved from one location to another within the search area instantly, so that there is no cost associated with ‘switching’ the application of search effort among different locations; an assumption that stands as a glaring incompatibility with the constraints of multi-robot, autonomous WiSAR.

For the case of moving targets, target motion is modelled probabilistically via an *a priori* known stochastic process, allowing for the computation of a 2D Probability Density Function (PDF) for the location of the target within the search area at any given time. In addition, it is also assumed that a PDF for the conditional probability of detecting the target in a region, for a specified amount of search effort applied in that region, given that the target is, indeed, located there, is available. This PDF corresponds to the sensor being used (be it an electronic sensor, or a person), and expresses the effectiveness (i.e., the detection ability) of that sensor probabilistically. As such, it is sometimes referred to as the ‘detection function’ in search theory literature [42, 44].

Given the abovementioned two PDFs, analytical techniques can, then, be used to combine this probabilistic information and determine a search plan dictating the optimal allocation of search effort over a specified time period to maximize the probability of finding the target, subject, of course, to constraints on the search effort available [44, 50]. For example, in [51], the target’s motion is modelled as a Markov diffusion process, which includes an initial target-location probability distribution as well as a transition probability distribution describing how the initial target-location PDF changes with time. Both these PDFs are assumed to be *a priori* known. Theory is educed that describes how the target-location PDF at any given time can be modified based on the search that has already been performed by the searcher and the detection function corresponding to that searcher. The detection function is assumed to be an exponential PDF. The optimal search to be performed is, then, modelled as an optimal control problem. However, the purpose in [51] was to show the theory and the mathematical formulations involved. Thus, no efficient and effective solution algorithm was proposed, and neither was the theory implemented on a practical search example.

Using the above-described overall formulation of the search problem, necessary and sufficient mathematical conditions for optimal solutions have been determined [52-55], as well as algorithms that find solutions to meet these conditions [52, 56-58]. Together, these mathematical conditions and corresponding optimization algorithms devised are termed Generalized Search Optimization (GSO) techniques [43, 50]. They characterize the basic search theory approach to the search problem, and form the basis of search methods so inspired.

There have been search methods proposed and subsequently implemented that are based on GSO techniques, albeit for applications different from WiSAR. In [59], for example, the Computer-Assisted Search Planning (CASP) system, used by the U. S. Coast Guard, is presented, which uses search theory principles to help determine how to best allocate search effort in marine search and rescue situations. A probability distribution for the location of the target (a vessel or a person adrift on the water) at the start of the search is first derived using oceanographic models to predict ocean currents and wind velocities, with estimation of associated uncertainties, and Monte Carlo simulation. As well, this target-location PDF is propagated forward in time and space (assuming a Markov process for target motion) to produce a sequence of probability maps for the location of the target at different points in time over the time-period of interest. Using search theory, the optimal allocation of search effort is computed and used to help guide the search manager in assigning search resources. Moreover, the impact of the ongoing search that is being conducted is also incorporated into the target-location predictions. At the end of each day, the target-location PDFs are modified through Bayesian inference techniques based on the areas that have been searched, the amount of search effort applied to those areas, and the target-detection ability of the sensors used. A new optimal search effort allocation is computed for the following day and, subsequently, implemented.

In [60], search theory concepts are used in developing a robotic SAR method for a single autonomous robot. The objective is to determine an optimal control solution for the robot using two candidate utility functions: mean time to detection and cumulative probability of detection. Mathematical functions for both of these quantities are derived using basic probability theory. The initial target-location PDF is assumed to be given, and is propagated over time and space using a known non-linear probabilistic (Markov) target-motion model. Moreover, Bayesian analysis is used to modify each propagated target-location PDF based on the regions already searched by the robot as time progresses, as well as the target-detection PDF corresponding to the sensor (i.e., uncertainty in the sensor's ability to detect the target, which is also assumed to be known). An optimization formulation is set-up where either of the two aforementioned utility functions can be used as the objective. The decision variables are the control inputs to the robot at discrete points in time over a finite time horizon that can be specified by the user. The objective is to find the optimal action sequence of control inputs over the specified time horizon that either minimizes the mean time to detection or maximizes the cumulative probability of detection. However, finding optimal control inputs on-line using this approach is computationally prohibitive. As such, suggestions are made on how to lower computation time at the cost of sub-optimal solutions. In particular, a one-step time-horizon is used, and fewer control inputs spaced farther apart (and maintained constant in between) are specified. Nevertheless, the method depends on the assumed availability of the initial target-location PDF, the probabilistic target-motion model, as well as the sensor's target-detection PDF. Moreover, the computation costs involved in propagating and updating all probabilistic information to account for target motion makes it difficult to use this approach as an effective on-line planning method.

The work in [61] also focuses on a marine SAR scenario, where multiple autonomous UAVs search for a floating target being carried by winds and waves. Theoretically, the same approach as that in [60] is used, where recursive Bayesian estimation updates and propagates an *a priori* known initial target-location PDF with the use of a given probabilistic target-motion model. Also, a target-detection PDF corresponding to each search sensor is available, and is used to modify the propagated target-location PDF based on the areas searched. However, this modification must be done by combining the sensor data of all the multiple sensor platforms based on the search that they have performed, which adds to the computational effort required. It is assumed that all sensor platforms can communicate with all others, so that each robot maintains its own copy of the target-location PDF over the search area and updates it based on the computed influence of all the sensor platforms involved. The proposed method also uses dynamic space reconfiguration, where the search area boundaries are reconfigured to account for target motion. This involves incrementally expanding the search area to include new regions that the target could occupy, when required, while also reducing the search space, if possible, by removing cells that do not contain any significant information. This helps to address the practical issue of limited computational resources to some degree when attempting to perform all the required computations during implementation.

In [62], the basic Bayesian probabilistic framework of [60] is used for the coordinated search of a target using multiple robots. The probability distribution of target location is updated based on a probabilistic target motion model, in the same way as in [60], but is subsequently modified to incorporate the effect of information obtained from noisy, non-line-of-sight sensors. Although multiple robots are considered, each robot plans its path independently by determining the next sequence of cells of the discretized search area to visit that minimizes a cost function. This cost function represents the entropy of the posterior target-location PDF over the search area. It is assumed, however, that robots can share their path-plans with each other, and each robot, thus, plans its path assuming that the others will continue to follow their most recently planned paths without change. This method still requires the assumption that all the necessary probabilistic information needed to compute the robot paths is available.

Thus, search theory approaches, although rigorous in their theoretical development, nevertheless pose difficulties for adoption into an autonomous multi-robot search framework, especially for the WiSAR application. A technical report, [63], published by the U. S. Coast Guard Operations, made some interesting observations regarding the issue of the compatibility of search theory principles with land-based SAR. The report noted that search planning techniques that have been developed based on search theory are most applicable to marine SAR operations, and do not readily transfer over to land-based searches. This is understandable as the original research work was borne out of the need for a theory to search for submarines in marine warfare during WWII. The report called for more research to consider how search theory principles could be applied to devise search-planning methods for inland searches, and that a standard methodology for land-based search-planning specific to inland SAR should be developed. The report also suggested that a computer-based decision-support tool for land-based SAR would be helpful in order to assist in guiding resource allocation. Finally, it was noted that better procedures are required for obtaining the type of probabilistic information that search theory methods need, when dealing with searches on land. Although the report was concerned with inland SAR using human search-personnel, these observations speak to the scarcity of formal theory for guiding WiSAR search operations that could perhaps help in developing methods for robotic WiSAR.

The last observation from the report in [63] noted above would especially pose a challenge if one attempts to apply search theory methods to address WiSAR. Indeed, quantities such as the PDFs for the location of the target and for the probability of target-detection for a given amount of search effort applied, as well as the probabilistic target-motion model, are difficult to obtain in practice, and search theory does not provide clear guidelines on how these probability distributions can be derived. Moreover, although search optimization can be formulated mathematically, and the necessary and sufficient conditions for optimality can be expressed in mathematical terms, solving this optimization in practice, especially in an on-line manner, poses a challenge. Once the problem is complicated further by including restrictions on searcher motion in order to obtain an optimal search path under the search-theory formulation of the search problem, the optimization becomes NP-hard and is, thus, not very amenable to an on-line feasible methodology. This is especially true when multiple paths for multiple searchers must be determined [64-66].

Furthermore, a WiSAR search presents even more challenges, including the unavailability of specific prior probabilistic information about the *particular* target being sought; influences of varying terrain on target motion (dependent upon the target's physiology) and, thus, on target-location probability; a growing and potentially boundless search area; robot deployment and path-planning in order to search different regions of the search area (i.e., non-negligible switching costs); and, the handling of clues found during the search. Not only would it be non-trivial to model these complexities mathematically and incorporate them into the optimization formulation of the search problem in the search theory approach, but repeatedly solving this optimization on-line within a growing search area in order to constantly update the optimal search assignments of all the robots would be computationally prohibitive. In addition, the solutions obtained from search theory approaches only indicate the location and amount of search effort to be placed. In multi-robot WiSAR, one is faced with the additional challenge of determining an optimal assignment of robots to perform those search tasks, so that simply specifying the location and amount of search effort would be insufficient. Consideration of robot assignment issues would have to be incorporated into the optimization formulations that determine optimal search effort allocation.

1.2.2 Multi-Robot Coordination

Multi-robot coordination has been defined as the planning, direction, and control of a team of multiple autonomous robots in a common environment to carry out a single task or multiple tasks in order to achieve a common global-level goal [67-72]. The problem of coordinating a team of robots to perform a complex set of tasks remains an open topic of much interest, and has been addressed in a variety of ways in the literature. Nevertheless, at a high level, approaches that are developed can be classified into two categories: centralized and decentralized. These are reviewed below.

1.2.2.1 Centralized Multi-Robot Coordination

In centralized approaches, a single control unit allocates specific tasks to each member of the multi-robot team [73-75]. As such, it maintains (intermittent or continuous) communication with each member and has access to all the information pertaining to the entire workspace obtained by all team members. A significant advantage of centralized coordination is that it allows the central controller to allocate all tasks in an optimal manner due to its access to global-level information.

Centralized approaches have been proposed for the coordination of multiple robots for a variety of tasks, such as motion planning [48, 49], formation control [74], coverage/exploration [1, 75], transportation [76], deployment [73], and scheduling servicing [77]. Research in [49], for example, deals with the development of a central-planning system responsible for assigning time-optimal paths to each member of a team of robots in order to guide them toward pre-specified destinations. The method is designed to function in a dynamic environment by having the ability to re-plan and re-assign new motion tasks to robots in real-time based on changing world knowledge. Hence, it is able to cope with surprises such as the emergence of unforeseen obstacles or unexpected terrain. A specific grammar for stating mission expressions (motion assignments to particular robots in a certain order) is developed so as to allow for addressing any type of motion-planning problem scenario regardless of what the global objectives are. The proposed system design consists of three asynchronous processes operating together. Each robot is equipped with a Local Navigator that chooses a steering direction based on what the corresponding robot perceives. A Dynamic Planner exists for each mission goal that calculates updated paths and costs for each robot to the goal as new map information arrives from the navigators. A central Mission Planner continually updates the assignment of robots to goals such that the overall mission cost is minimized. Although the method is general in the sense that it abstracts the data passed between processes, it is still limited to this particular class of MRC problems (i.e., a multiple travelling-salesperson problem), and in this sense is specialized with limited scope.

A centralized strategy for on-line coordination of multiple unmanned vehicles is also presented in [48]. An optimization model is developed to determine feasible robot control inputs on-line for multiple autonomous mobile robots to move toward specific, stationary target locations. There are as many target locations as robots, and the exact target locations are known at all times. Static and moving obstacles exist in the environment, and the robots are assumed to be able to detect them and estimate their velocities. The optimization problem is formulated by integrating model predictive control (which allows for the incorporation of motion constraints and changes in mission objectives) with mixed integer-linear programming (which can model logical constraints pertaining to the interaction behaviour of multiple robots). However, all target locations are *a priori* known and fixed, so that one does not need to be concerned with planning within an environment with expanding boundaries. Moreover, the added complexity of terrain topology is not taken into account.

A variety of other examples of centralized control methods for multiple autonomous vehicles exist, albeit for specialized tasks. The research in [74] focuses only on path-planning of multiple robot agents to maintain a certain platoon formation. In [75], a robot population is modelled using a stochastic hybrid automaton model [78] and a method is proposed to control the evolution of this population so as to maximize the probability of robotic presence in a specified region of the environment, given uncertainty in each individual robot’s reactions to inputs. The work in [77] proposes an optimal scheduling method for the autonomous refuelling of multiple UAVs by a single tanker (the central controller). These approaches to MRC point out a key advantage to the centralized approach. Since a central controller has knowledge of all the information pertaining to the system, it can obtain, or at least work towards, a globally optimal solution for the system [79]. This is of great benefit to applications that are time-critical, or have limited quantities of resources required to accomplish the global task, because in such instances, one would need to make the best use of all available information to allocate tasks to the robots so that time and resources are not wasted.

1.2.2.2 Decentralized Multi-Robot Coordination

In decentralized approaches, no central controller exists. Instead, each team member functions mostly independently and acts based solely on the local information that it receives from its on-board sensors [80-82]. A number of decentralized approaches have been proposed in the literature for multi-robot teams that allow the robots to function independently, but also exhibit cooperative behaviour. In [80], for example, a set of software tools for the development of controllers and estimators for MRC are discussed. These tools include a set of decentralized control, planning, and sensing algorithms, through which the overall task of controlling the robots is broken down into a set of modes or behaviours for each robot, including path planning and obstacle avoidance. The focus is on formalizing the infrastructure for local robot interactions with each other and with their environment. Through these interactions, the robots are able to complete the global tasks without the need for a central controller.

Similarly, in [82], a distributed control approach for a cooperative multi-robot hunting task is described. The robots have limited sensing and communication, and must search for and pursue an evading target. Conditions are defined to indicate which state the robots can be in: search, pursuit, predict, or catch. These, combined with state transition rules, allow each robot to function independently. It is assumed that each robot is able to determine the position of other robots within its sensing range, and modifies its actions accordingly. In this way, cooperation may emerge among the robots in the team. Searching, however, is conducted by making random turns within the search area, and prediction of the target's location is only initiated for a finite period of time after occasions when a detected target manages to escapes from within a robot's sensing range. After this period, the robot returns to the search state, where no further target prediction is used until the next target-sighting and subsequent target-escape occurs.

In [83], a Swarm Robotics approach is used to coordinate a group of homogeneous robots for an abstracted problem scenario involving the distribution of robots to multiple locations. The robots use biologically-inspired behaviour to dynamically re-distribute themselves to ensure task completion at each location. In [5], the impact on system performance for three tasks, namely, foraging, consuming, and grazing, is studied for different levels of inter-robot communication incorporated into the basic swarm robotics approach. For the foraging and consuming tasks, at least, communication is found to provide significant performance improvement over communication-free swarming behaviour.

A decentralized algorithm with a focus on searching is presented in [84]. The algorithm coordinates a robot team to navigate within a known search space to find a target. Each robot maintains its own map of the search area and navigates by selecting regions in which to take measurements in decreasing order of the probability of containing "good" information. An initial underlying density function for the probability of detecting the target over the search area is, nevertheless, assumed to be known, and must be updated based on the regions searched over time. As well, the target is assumed to be stationary, and the search area, bounded, so that the density function does not propagate outward with time.

Some decentralized approaches have also proposed the formation of small sub-groups, where robots within a sub-group share information, but little or no communication exists across sub-groups [72, 85]. For example, research reported in [85] addresses the area-exploration task, and presents a negotiation-based approach to coordinate the movements of the robots. A distributed bidding model allows each robot to submit a bid to other robots within its communication range in a bidding session for the next position that it should move to that will (locally) maximize information gain and maintain communication with nearby robots. The winner of each session is allowed to move according to its bid calculation. In this manner, the robots cooperatively explore the given area, resulting in a totally distributed exploration algorithm.

Although decentralized approaches have the advantage of reduced communication and computation requirements, they are mostly applicable to situations where the global task to be accomplished can be divided into smaller independent sub-tasks that can be carried out in parallel [72]. For such problem types, the impact on global system behaviour can typically be measured by simply aggregating the impacts of the individual robot behaviours. In these cases, it can be expected that optimal local behaviour of each robot individually will have an aggregated effect to increase the optimality of the global system behaviour. Hence, applications such as foraging or carpet cleaning are suitable. However, in more complicated application domains, such as automated WiSAR, where effective task allocation depends on global-level information, which would be unavailable to independently-functioning robots with local scope, the performance of distributed MRC methods may be questionable.

In addition, since the robots in decentralized MRC approaches act independently based on local information, it can be difficult to program behaviours that guarantee any intended or required type of cooperation between robots in all circumstances. It is stated in [80], for example, that the emergence of cooperative behaviour itself is uncertain. In [82], as well, it is acknowledged that cooperation may arise through the behaviours programmed into each robot, but is not guaranteed. This type of uncertainty could further diminish global system performance in applications where cooperation is important [49, 86, 87].

1.2.3 Multi-Robot Search and Rescue

To date, research on autonomous robotic WiSAR has mainly focused on robot implementation issues rather than on the development of an overall coordination methodology for searching. Moreover, since WiSAR often requires the rapid coverage of rugged terrains, research focus has typically been placed on the use of tele-operated UAVs to assist the searches. For example, research done in [12] considers human-robot interaction issues when using UAVs to provide aerial imagery of the search area in order to assist WiSAR personnel. The UAVs apply a coverage search pattern, with no form of autonomous coordination among multiple UAVs or utilization of autonomously generated and updated probabilistic target-location information to guide the search. This work is extended further in [37] to study the roles of the UAV aviation and navigation operator and the UAV sensor operator, who typically form a team of tele-operators that control UAVs for WiSAR missions. Information gathered through field-studies is analyzed to help determine the ideal level and type of UAV autonomy, and the information display requirements, that would be needed to minimize the size of the tele-operation team required. The eventual goal is to design a UAV system that allows a single operator to simultaneously fill both the navigation and sensor operator roles.

Research focusing on tele-operation of robots and formation of effective mixed human-robot teams for USAR was also reported in [18-20]. For example, research in [20] addresses the issue of situational awareness in tele-operated USAR robots by proposing a means to enable robots to autonomously and dynamically find the ideal balance of autonomy between the robot and teleoperator during a USAR mission. The proposed approach utilizes hierarchical reinforcement learning to devise a semi-autonomous controller that allows a robot team to autonomously determine the appropriate rescue tasks to perform during a USAR operation, and to request assistance from a human operator when necessary.

The study in [38] proposes the design of a robust, real-time, trajectory planning system for fixed-wing UAVs intended to serve in uncertain environments such as those characteristic of WiSAR. Given a set of waypoints, the proposed algorithm plans a trajectory on-line that satisfies UAV kinematic constraints and that accounts for dynamic obstacles. A tracking algorithm is also employed to ensure that the trajectory is followed satisfactorily. Custom-designed UAVs are used to implement and test this approach. In [88], a method is proposed for constructing an observation model to quantify the efficacy of any given target detection algorithm applied to video-camera information obtained from UAVs that assist in WiSAR.

Recent research interest has also been shown in the physical design of robots for SAR. However, such works pertain to the USAR domain, where specialized robots are designed to address mobility and sensing challenges particular to this application. For example, [16] and [17] focus on the design of robot locomotion systems suitable for USAR terrains, while [89] and [90] address the design of sensing and control systems for real-time 3D mapping and landmark identification in USAR environments.

With respect to autonomous multi-robot searching, research in [91] compares various coordination mechanisms and provides insight into how they can be applied to the problem of automated SAR. However, their focus is on USAR, and, as such, the problem considered is one of searching for stationary targets in a bounded search environment (e.g., a collapsed building with targets buried under rubble), where target motion prediction is not required. Robot search strategies are not addressed in any detail either, and a simple random-walk algorithm, or a shifting grid-search pattern, is assumed. Similarly in [92], a method is proposed to allow for cooperative exploration by a team of autonomous robots in a post-disaster USAR environment. They address the ineffectiveness of vision-based sensing for localization and mapping caused by smoke and fire by employing an indirect form of communication based on the use of RFIDs.

A mixed team of mobile robots and distributed static and mobile sensors are considered in [93] for use in assisting firefighters during SAR missions inside buildings that are on fire. Thus, the search area is bounded and the targets are stationary (i.e., trapped, immobile humans). The static sensors are carried by the mobile sensors as well as the human firefighters and are deployed within the search area. As the search progresses, the static and mobile sensors create ad-hoc networks as necessary for localization and mapping, and also to share and update their information on located targets and on sensor readings of temperature and toxin-concentration gradients in the building. The firefighters use this information to guide their search for survivors. Although the static and mobile sensors work together in an automated fashion, the system still relies on the human firefighters to conduct the actual search. The other mobile robots that are used employ formation control protocols based on information obtained from their local sensors and from the ad-hoc sensor network that is set-up in order to gradually find and move towards the source of heat in the building. Moreover, no form of target location prediction is used either.

In [94], the proposed search method for SAR combines a behaviour-based approach with the concept of potential fields, though still for the search of stationary targets within a 2D, bounded, albeit unknown, environment. Coordination is an emergent property that results from the behaviours assigned to the robots. Potential fields are shared locally among robots within a predefined “catchment” radius, allowing robots to react to obstacles they may not directly detect, but which others do. The approach does not address many of the issues involved in real-life WiSAR, such as moving targets with unique psychologies, target location prediction, the utility of clues left behind by the lost-person, and the influence of varying terrain.

In [61], a marine SAR scenario is addressed involving a floating target being carried by winds and waves. The Recursive Bayesian Estimation technique maintains and propagates the target-location PDF for use in target state prediction. This paper is one of the very few to incorporate search theory into automated, multi-robot SAR. However, one of the drawbacks is that each robot is assumed to be able to independently calculate the target position update equations after each time interval and, then, independently decide where to go using a five-step look-ahead with receding horizon method. Hence, this approach lacks proper global-level coordination and optimal task assignment. Furthermore, additional issues that are important in WiSAR, such as varying terrain and its impact on target motion, as well as the utilization of clues found during the search, are not considered.

Other methods proposed for automated search are similarly incompatible with robotic WiSAR due to focus on a different application and/or the use of simplifying assumptions [14, 95-98]. The search method in [96], for example, addresses a visibility-based version of the pursuit-evasion problem, assuming an unlimited searcher-vision range in a bounded environment with a moderate distribution of obstacles. It was modified in [97] to accommodate multiple searchers and a searcher field-of-view model that more closely resembles that of a physical robot, but, nevertheless, still maintained the other limiting assumptions of [96]. A human-immune-system-inspired coordination methodology for multi-robot search and rescue is proposed in [98], where the robots are required to work cooperatively in order to obtain debris from a disaster area and transport them to another site. The debris is subsequently assembled into a structure to be used for evacuating the trapped survivors. The method focuses on optimal task allocation in which the most suitable robots are selected to transport each piece of debris using a multi-criteria cost function that is optimized, off-line, through a genetic algorithm implementation. A method is also devised to prioritize transportation of different pieces of debris based on their order dependencies. Thus, the application domain is USAR and the tasks to be performed by the robots are quite specialized to the nature of the particular problem addressed.

It is evident, then, that autonomous, coordinated multi-robot WiSAR is still an ongoing research field. From the abovementioned works and others on robotic SAR, it is clear that there exists interest and effort toward the use, as well as the increased autonomy, of robots for this application domain. Nevertheless, existing methods use simplifying assumptions such as stationary targets and bounded environments, and/or do not utilize target-prediction based on probabilistic information, making them incompatible as comprehensive search methods for autonomous robotic WiSAR. However, they do indicate recognition of the advantages that an autonomous multi-robot solution, with target-behaviour prediction, could bring to an application such as WiSAR.

1.2.4 Target-Location Prediction

Target-location prediction is an integral part of search theory, as well as the proposed methods for autonomous search in the literature that make use of this theory. For a moving target, this prediction refers to a continuous or discretized probability distribution for the location of the target defined over the search area, as well as some type of stochastic process for propagating this probabilistic target location information over time and space to account for target motion (sometimes referred to as a ‘target motion model’). However, what is common among search methods in the literature that use target-location prediction is the assumption that both the initial probability distribution for the target’s location and the probabilistic description of how the target moves are given inputs that are known *a priori* (e.g., [42-44, 59-62]).

As discussed above, search theory works have mainly utilized theoretical GSO techniques for the optimal allocation of the search effort [43, 44]. Target prediction is addressed via a stochastic process, assumed to be a given input to the adopted GSO technique. However, no guidance is provided as to which stochastic process is applicable for a given search scenario. In [60], for example, Bayesian analysis is used to propagate a 2D initial target-location PDF over the search area. A probabilistic Markov motion model is used in order to propagate the target location uncertainty over discrete time-steps. However, this motion model is assumed to be given, without explanation of how it can be derived. A probabilistic observation model is also assumed to be known, which gives the conditional probability of detecting the target at any position within the search area given that the target resides at that position. This PDF is combined with the propagated target-location PDF to compute the posterior PDF for each successive time-step. The research in [61] extends the above work to include a means for reconfiguring the search boundaries, but continues to use the same approach to predicting the location of the target. In [62], the Bayesian probabilistic framework in [60] is used, and two alternate means of analytically incorporating the effect of imperfect sensors (i.e., the probabilistic observation model) are presented.

It is evident from the literature, however, that in works like the above that use analytical approaches to compute and propagate a 2D target-location PDF over the search space, target prediction primarily involves only three main considerations. In particular, these are: the probability of where the target is at any given time (the prior target-location PDF), how the target moves (the target motion model), and the search unit’s detection ability in the given environment (the detection function). Other factors that complicate target prediction in WiSAR scenarios, such as the impact of terrain on target motion, or the influence of clues on the target location probability, are not explicitly considered. This limits their usability and versatility in real-world environments.

Other approaches to target location prediction are also evident in the literature. As part of the research on human-robot team formation in WiSAR scenarios in [12], the use of UAVs to assist human search personnel is investigated. Human search commanders use background information about the target (gathered from interviewing friends and family of the lost person) and their own experience and expert knowledge to delimit regions of high probability of target presence within the search area. This results in a type of coarse probability distribution for target location over the search area, which is then used to assign search tasks to the UAVs. However, the UAVs simply execute search patterns over their assigned regions, without autonomously evaluating and updating the probabilistic information for target location within this region on-line and using it to modify their search flight-paths. In [99], a method is proposed for constructing the prior target-location PDF as well as the transition matrix (assuming a first-order Markov process). The method requires that expert opinion be initially available to help in generating the PDF, but allows for incorporating the influence of terrain features.

Examination of missing-person incidents in wilderness areas of Alberta, Canada was carried out in [100]. The Wakeby Distribution, [101], was used to characterize the data for different categories of wilderness user. This distribution provides a function for computing percentiles of ‘crow’s-flight’ distances in planning the boundaries of a search. The term ‘crow’s-flight’ distance refers to the straight-line distance between the last known position of the target at the start of the search and the position at which the target was found. Such data from past lost-person search incidents are gathered and stored by SAR organizations, and are further organized into relevant categories of lost-persons, such as hikers, mountain-bikers, cross-country skiers, etc., since lost-persons within such categories tend to behave in a similar manner. This information was used and analyzed in [100] to derive a unique Wakeby distribution for each of several lost-person categories that were considered in that work.

A search planning system used by the U.S. Coast Guard is discussed in [59]. Given estimates on the location and time of the distress incident, the system uses Monte Carlo simulation, combined with oceanographic models for target drift computation, to build an initial target-location PDF, and to propagate it. Bayesian updating is also used to reflect ‘negative information’ obtained from unsuccessful searches. The output is a sequence of time-phased PDFs giving target position likelihoods over the time period of interest.

For WiSAR, target-location prediction not only requires a means for constructing and representing an initial prediction of the target's location at the start of the search, but also a means for updating this probabilistic information efficiently over time and space. This must be performed on-line, autonomously, so that the probabilistic target location information can be used to autonomously guide the search paths of the robots as this information evolves with time. Moreover, to be practically useful, the probabilistic information on the target's location must be obtained and updated autonomously using readily available information, rather than information that is simply assumed to be known *a priori*. In addition, a means for incorporating other influences important in WiSAR must be clearly delineated, such as the influence of terrain on target motion and the effect of clues found. None of the existing target location prediction methods in the literature provide a means to address all these requirements simultaneously.

1.2.5 Robot Deployment

In the MRC literature, deployment is generally viewed as either one of attaining a specified formation of multiple agents [102-104], or one of appropriately dispersing multiple agents so as to cover a bounded area [45, 105-107]. In [102], for example, a directed potential field approach is used to initially bring a group of robots into a particular formation pattern and to maintain this pattern as the group moves. Similarly, in [103], a general class of attraction and repulsion functions is presented that can be used to aggregate a swarm of robots together. In [104], the user specifies the deployment pattern in terms of the deployment locations, and a general deployment strategy that dictates how the robots are allowed to move in order to reach these locations. An integrated control system then determines a lowest-cost assignment (in terms of route-length and time) of robots to those locations based on hierarchical task network planning and constraint-reasoning techniques. Deployment is considered to be complete once all robots have attained their assigned positions.

Deployment in terms of general coverage typically requires robots to be distributed in a given region so as to maximize the amount of area that is within the field of view of the robot sensors. In [45], this involves determining how many robot-groups should be unloaded from a carrier, the number of robots per group, and the initial robot locations, in order to cover a given area. Pre-determined motion strategies for deployment are utilized, which dictate patterns of travel-paths along which the robots are restricted to move. A deployment solution that can cover the search area within the maximum deployment time allowed is, then, determined iteratively by varying the number and sizes of groups based on heuristics.

In [105], a probability density distribution over the search space is used to model uncertainty about the environment. As the robots move, the uncertainty along their paths is reduced. Multi-robot deploy-and-search strategies are used to compute and regularly update deployment positions based on a centroidal Voronoi partition of the search space, so as to minimize the time required to decrease the uncertainty density. However, the method assumes a static, bounded environment where uncertainty does not propagate over time and space. This same centroidal Voronoi partitioning approach is used in [107] for the deployment of sensors for optimal coverage, but the solution is found using decentralized coordination algorithms.

The above methods mostly consider deployment as a singular problem – one that is solved only once, and is considered to be complete when the intended deployment positions have been attained. Methods that re-compute deployment do so within the context of a simplified scenario where the search area is bounded and, in some cases, where no time-varying target-location prediction modelling exists. A WiSAR scenario is more complex due to its dynamic nature, and it may be necessary to re-distribute the search resources within the search area during the ongoing WiSAR mission in order to maintain search effectiveness. This would require regular re-computation of the optimal deployment, and a subsequent re-deployment of the robots, if necessary, throughout the search. Furthermore, in WiSAR, optimal deployment must be based on available probabilistic target-location information, which is regularly propagated over time and space within an unbounded environment with varying terrain. Moreover, the event of finding a clue can significantly affect the prediction of the target's location, and would impact the optimal deployment as well. However, the literature indicates that current methods for deployment are not designed to address such complexities in a single solution approach.

1.2.6 Robot-Path Planning

The majority of research in the multi-robot path-planning literature has so far focused on probabilistic path-planners, including the Probabilistic Roadmap (PRM) and Rapidly-Exploring Random Trees (RRT) approaches, and augmentations thereof [108-112]. In [109], for example, to address the 'curse of dimensionality' that still plagues these approaches when path-planning for large multi-robot teams, the sub-dimensional expansion technique is proposed. This technique dynamically varies the dimensionality of the path-planning problem by planning within the joint configuration spaces of only those robots that happen to be in close proximity of one another, rather than always planning within the joint configuration spaces of *all* the available robots combined. This approach has the effect of generally keeping the overall dimensionality of any given path-planning-computation instance relatively small. Simulations are conducted where PRMs are enhanced with this technique, and it is shown that paths for a 32-robot team can successfully be planned in less than 10 minutes.

Similarly, research in [110] proposes ways to reduce the path-planning search space by decoupling path-planning. One method involves exploiting the multi-robot path-planning problem structure in order to intelligently alternate between planning the optimal robot paths independently (decoupled) and constructing the roadmaps considering a subset of robots simultaneously (coupled). This variable-coupling technique helps to expedite the basic PRM/RRT approaches, while still maintaining completeness (i.e., the guarantee that if an optimal solution exists, it will be found).

Strategies also exist that determine optimal paths for each robot in a team sequentially based on robot priority, (e.g., [113] and [110]) and address robot interactions subsequently as they arise. Moreover, some employ additional techniques to help ensure feasible solutions (e.g., through robot velocity re-tuning along planned paths, [114], or on-line path re-planning, [111]). However, although the resulting paths are sub-optimal, and may possibly not be found at all, these approaches do reduce computation time, making them more applicable for on-line path planning.

Nevertheless, since probabilistic path planning methods involve randomly sampling the configuration space (of a single robot individually or multiple robots jointly) to construct roadmaps and subsequently find paths, their effectiveness depends on how densely the roadmap fills the environment. This quickly becomes intractable for multi-robot path planning in applications such as WiSAR, where the search area can be only sparsely populated with obstacles and grows within a potentially boundless environment. Moreover, for WiSAR, path planning must also take into account the probabilistic information about the location of the target within the search area, as well as the impact of varying terrain and clues found, in order to plan meaningful and effective paths that will optimize the chances of finding the target being sought.

Other approaches to path planning also exist in the literature, but are designed to address certain particular challenges. The research in [115], for example, presents a path planning method intended specifically for WiSAR scenarios, but focuses on the use of a single fixed-wing UAV that acts as an assistive tool to live search personnel. The method requires a 2D, discretized target-location probability map, constructed using the procedure in [99] mentioned in Section 1.2.4 above, to be overlaid onto the terrain. In this approach, the UAV 'sweeps up' probability as it moves along its planned path. The volume of probability removed as a result of the UAV moving along a particular path is based on the UAV's sensing footprint. The search-path is found through an optimization that seeks to maximize the probability volume removed in this manner within the available search time. This results in a 'priority-search' strategy, where the highest probability areas are searched first. However, with this approach, the complete path over the entire allotted search time is planned in a single computation run rather than incrementally, on-line, making it difficult to adapt path planning to unforeseen situations autonomously. The planned path is also constructed in segments that are restricted to straight-lines that can only emanate in either of the four cardinal directions from the robot's position, resulting in paths that are non-smooth.

Another example can be found in [116], where a path planning method is devised to address a multi-UAV formation control problem, with applications to search and coverage as well as SAR. This work points out a key difficulty that one is faced with in the path planning problem for multiple robots, namely, that a means must be provided to reconcile the discrete nature of the decision-making involved with the continuous nature of the multi-robot system as the assigned paths are executed. The work in [116] proposes a hybrid supervisory control approach to address this challenge. The problem that is considered, in particular, is a 2D leader-follower formation scenario, where the UAVs must: (*i*) reach a formation relative to a designated leader UAV; (*ii*) maintain that formation as the leader UAV moves to follow a particular assigned path; and (*iii*) avoid colliding with other UAVs in the team throughout the whole process. A modular, decentralized supervisor is designed at the path-planner level based on a finite discrete event system model. Through this model, the multi-UAV formation control system is reduced to a finite state machine so that it can be controlled via existing discrete event system supervisory control theory. Moreover, a polar partitioning of the state space is proposed for the path planning. However, for searching in WiSAR scenarios, path planning must also consider target-location prediction in devising and allocating the search tasks, and must dynamically respond to changes in this prediction on-line. This introduces additional challenges that need to be addressed in a cohesive and seamless manner. Thus, a path planning method that can feasibly construct search paths for multiple robots, on-line, in WiSAR scenarios, based on time-varying probabilistic target-location information, is still a challenge that must be met. Path planning may also be further complicated by the necessity of maintaining a specified robot distribution over the search area as the search progresses. These issues, when combined, make path planning a complicated problem for autonomous multi-robot WiSAR.

1.3 Thesis Contributions

The literature reveals that there is a lack of a suitable approach for the autonomous coordination of multiple robots for the WiSAR application. Search theory does offer a mathematical optimization-based method for the allocation of search-effort, which could possibly be automated. However, approaches to searching for a moving target based on search theory are not amenable to robotic WiSAR, mainly due to the use of an analytical approach to modelling and optimizing the allocation of search effort. This approach not only makes it difficult to incorporate the complex issues that must be addressed in realistic WiSAR scenarios (since everything must be modelled mathematically and incorporated into a single 2D posterior target-location PDF), but it is also not feasible as an on-line approach due to the computational complexity involved in having to solve the resulting mathematical optimization problem. In addition, the 2D initial target-location PDF and the probabilistic target motion model form the critical elements of search theory approaches, and it is unclear how these can be obtained in practice for any given WiSAR lost-person. Similarly, multi-robot coordination methods found in the literature address generalized applications, such as foraging, searching, or coverage, and often include simplifying assumptions that are incompatible with WiSAR. As such, there is no single approach proposed that can address all the issues relevant to robotic WiSAR, and it is not evident how existing methods can be modified to do so either.

As will be discussed in Chapter 2, the purpose of this Thesis is to address this need by proposing a MRC methodology that can deal with the issues relevant to WiSAR in an on-line manner. In particular, the method must be able to coordinate a team of robots to perform a search in which the target is possibly mobile, resulting in a growing and potentially boundless search area.

Probabilistic information on the location of the target must be derived using readily-available data, represented in an appropriate manner, propagated to account for target motion, and utilized to guide the search. Moreover, this method must account for the influence of varying terrain topology and non-traversable obstacles (both *a priori* known and unknown) on the motion of the target, based on some knowledge of the target's physiology. As clues are found during the search, the method must be capable of utilizing the relevant information from those clues and modifying the search effort accordingly. Finally the coordination methodology must be able to address these issues concurrently, yet still remain feasible as an on-line approach.

The overall solution approach that has been developed in this Thesis to address this MRC problem for autonomous robotic WiSAR has resulted in a number of contributions as follows:

i. A procedure for constructing a prediction for the location of a mobile target within a search area at any given time based on the use of empirical data relevant to WiSAR and conservative estimates of target motion.

ii. A method for modifying the target-location prediction in order to incorporate the influence of target physiology, target psychology, varying terrain, obstacles, and found clues.

iii. A procedure for propagating the target-location prediction over time and space to account for target motion.

iv. A mechanism for representing target-location prediction that can be computed rapidly, and that facilitates the tasks of both deploying robots and planning their paths.

v. A method for determining an optimal deployment of robots at the beginning of a WiSAR search operation, based on the impact on a measure of the time required to find the target (search-time) and on a measure of the probability of success in finding the target (success-rate).

vi. A method for determining a new (revised) optimal deployment solution during the ongoing search process based on measures of search-time, success-rate, and effort that would be required by the robots to implement the new deployment solution.

vii. An evaluation method to regularly gauge the optimality of the current deployment solution, on-line, throughout the search process, in order to determine the necessity for implementing a new optimal deployment.

viii. An on-line path-planning method to determine optimal search-paths for each robot, individually, that recursively constructs each path in conjunction with the propagating target-location prediction.

ix. A rapid checking method for evaluating the optimality of each robot's path during path-implementation.

x. A decision-making procedure to determine the necessity for re-planning a robot's path based on the results of the path-optimality evaluations.

xi. A contingency procedure for robot-path planning when *a priori* unknown obstacles are encountered.

At a high level, a solution to the MRC problem for robotic WiSAR would have to address three main tasks: target-location prediction, robot deployment, and robot-path planning. The different contributions listed above correspond to methods and procedures that have been devised that, together, address these three tasks.

Firstly, in order for a search in a WiSAR scenario to be effective, it must be an informed search. Namely, there has to be some prediction of the location of the target within the search area that has to take place if one wishes to deploy and utilize the search resources in an intelligent manner that maximizes the effectiveness of the search. As mentioned in [63], [117], and [118], search managers in actual WiSAR missions involving live personnel must commence a search by first estimating the likelihood of target-presence within different regions of the search area, even if this implies using their intuition and experience to come up with crude initial probabilities. For a robotic WiSAR methodology, this requires a technique to autonomously generate a prediction for the initial location of the target at the start of the search. To be practically useful, such a prediction must be generated based on readily available information about the target, which poses a significant challenge since probabilistic information about the location and motion specific to the particular target being sought is difficult, if not impossible, to obtain.

Another challenge in target-location prediction is that the resulting probabilistic information, once obtained, must also be represented in a useful way. As reviewed in Section 1.2.4, search methods that use target state prediction represent this information through the use of a 2D target-location PDF that spans the search area. This form of representation is either a continuous distribution represented by a closed-form mathematical function, or a matrix of probability values resulting from discretizing the search area with a grid array. Not only is the information required to construct such a representation difficult to obtain, but this approach is also computationally intensive to maintain on-line. Furthermore, a means must be provided to propagate the probabilistic target-location information to account for target motion as the search progresses, and the representation method that is used must facilitate this propagation on-line.

This Thesis will, thus, present a method for target-location prediction in Chapter 3 that addresses the above challenges. It will propose a source of information, and means to process it, in order to derive a suitable initial target-location prediction. Additionally, a means of propagating this information over time and space to account for possible target motion throughout the search will be proposed. Furthermore, a novel means of representing this probabilistic target-location information will also be presented that not only provides an effective means of portraying this probabilistic information and its variation over time, but also facilitates the process of identifying key regions for search effort allocation, distributing the search resources, and planning the robot search paths. In addition, this solution will be shown to be feasible as an on-line approach through simulations.

Secondly, given probabilistic information about the target's location within the search area, and an effective means of representing it, the next major task involves the deployment of the robots within the search area. The optimal deployment task requires solving two main problems. First, regions must be identified within the search area where it would be most beneficial to place search effort. The challenge here is that this benefit must be expressed as a function of important characteristics of the pending search that will help distribute search resources in an optimal way. In search theory, one typically tries to allocate search resources in a manner that either minimizes the time required to find the target, or maximizes the chances of finding the target, since these form the overriding concerns during any WiSAR search endeavour. Using the novel mechanism devised for representing probabilistic target-location information, this Thesis will propose a means of quantifying both these search attributes in a way that is relevant to robotic WiSAR, and a method by which these quantified attributes can be used to guide the delimitation of where search effort should be placed. Since search resources are limited, and the likelihood of target presence within the search area varies according to the underlying probability distribution, a corresponding optimization problem will have to be formulated and solved to determine the best regions for search effort allocation.

Upon identifying the optimal regions for search effort allocation, the second problem to be addressed for the optimal deployment task is the determination of how the search resources should be distributed among these regions. For robotic WiSAR, this entails determining an effective assignment of the robots to the delimited regions. Since robots must traverse terrain, and will, therefore, take some finite time to arrive at their assigned positions, this assignment of robots to regions must be done as efficiently as possible, and would, therefore, also require an optimization. What further complicates this second main task of the overall MRC methodology is that both the delimitation of the optimal regions in which to apply search effort, as well as the assignment of the robots to these regions, is dependent on the search resources available. Therefore, these two problems cannot be addressed independently, but rather, will require a unified solution approach. In Chapter 4, this Thesis will propose such a unified method that can address these two problems optimally and concurrently.

Thirdly, once robots have been optimally deployed, the next main task deals with how the robots would move about the search area in order to find the target. Namely, this is a problem of robot-path planning. However, the search paths that are planned must be constrained by the need to apply effort within the optimal search regions that were determined, while maintaining the optimal distribution of the search resources that was dictated. In general, though, there will be uncertainty about how the terrain varies, and this impacts how successfully any given robot is able to implement a given path-plan. Therefore, an appropriate path-planning method will also be proposed in this Thesis, in Chapter 5, that satisfies the abovementioned constraints, but that is also able to address path-implementation uncertainties in a dynamic, on-line manner to ensure that all robot paths remain effective throughout the search.

A final challenge that must be addressed is that each of these three main tasks for MRC in WiSAR must be conducted within a dynamic environment. In WiSAR, a potentially moving target results in a growing search area that progressively encompasses new and varying terrain which may contain new clues as to the target's likely whereabouts. Moreover, this target movement also implies time-varying probabilistic target-location information which impacts robot deployment, and, consequently, robot-path planning. Hence, the methods devised to address the three main tasks of target-location prediction, optimal deployment, and robot-path planning must also be correspondingly designed to contend with this dynamic nature of the WiSAR search problem. This Thesis will, thus, present an MRC methodology for WiSAR that dynamically and concurrently addresses these three main tasks to conduct an effective search for the target, while still remaining feasible as an on-line method.

The remainder of this dissertation is, thus, organized as follows. The detailed problem formulations for each of the aforementioned three main tasks, as well as an overview of the solution approaches proposed for each, are given in Chapter 2. Following this, Chapters 3, 4, and 5 will present the detailed solution methods for these three tasks, respectively. Chapter 6 will present the results of simulations that were conducted to test the overall proposed MRC methodology for WiSAR, including an example search simulation, a comparison test with an alternative search method, and a test of robustness. Finally, Chapter 7 will conclude the Thesis and will discuss possible directions for future research.

Chapter 2

2 Problem Formulation and Proposed Methodology

The SAR problem comprises two main constituent tasks: search and rescue. The problem addressed in this Thesis focuses only on the former task, namely, the search process. Search refers to moving through an environment and obtaining and analyzing relevant information in order to determine the location of a specified target. The specific branch of SAR for which a global strategy for autonomous MRC is proposed, herein, is search in wilderness environments (i.e., WiSAR). The main characteristic of WiSAR that differentiates it from other branches of SAR is that the target is, in general, mobile within an in-land, yet potentially boundless, environment, and which cannot be tracked during the search (i.e., the target is not continuously or intermittently observed by any of the searchers while the search is in progress). The target is considered to be found only when it enters within the sensing range of a searcher and is positively identified as being the target.

As delineated in Chapter 1, at a high level, a solution approach to the MRC problem at hand must address three main tasks. First, a means to address target-location prediction within an ever-expanding search area must be devised. In addition to predicting the location of a possibly moving target at any given time, this prediction must also address other complicating issues, such as target physiology and psychology, the influence of terrain on target motion, the impact of clues, etc. Since one of the aims of this Thesis is to develop a robotic search methodology, target-location prediction must be performed autonomously and on-line.

The second main task of the solution approach to the MRC problem is robot deployment. Since search resources are limited and the search area grows with time, it is imperative that the available search robots are always distributed within the search area in the most effective way possible. This distribution refers to the specification of position coordinates for each search robot, which, once attained, will correspond to an optimal distribution of the search resources within the search area. This optimality must be judged based on the impact of a specified robot distribution on some measure of the quality of the ongoing search. The specification of this optimal distribution corresponds to the robot deployment. However, robot deployment must also address the dynamic nature of WiSAR. Namely, as the search area grows, and new relevant information is obtained (e.g., varying terrain or found clues), the distribution of the robots within the search area may need to be modified in order to remain optimal. Thus, robot deployment must also be dynamic to properly address MRC in WiSAR.

The third main task to be addressed by the overall MRC methodology is robot-path planning. Given an optimal deployment of the search resources, the robots must subsequently move about the search area in order to find the target. However, the need to maintain a specified optimal deployment constrains this motion. Hence, appropriate paths must be planned for each robot that allows the robots to perform the search while adhering to the constraints imposed by the optimal deployment. Moreover, since deployment is dynamic, path planning must also be correspondingly dynamic, with path re-planning conducted as needed. An added difficulty is that, although the environment is dynamic and multiple robots are involved, path planning must be conducted in an on-line manner.

Each of these three main tasks will be formulated in more detail in the following sections, and an overview of the proposed solution method for each will subsequently be provided as well. Throughout these formulations, as well as in the solution methods presented, it is important to note that the research work in this Thesis deals only with the methodology for coordination rather than the low-level issues of the physical design of robots and their payload for performing the search tasks. Furthermore, it is assumed that each robot is capable of performing low-level implementation tasks, such as trajectory-planning to follow a specified path, and obstacle-avoidance. Although robot platforms and control-methods do currently exist that can make the proposed MRC methodology practically feasible, research still continues on the practical design aspects of robots for WiSAR environments. As technology advances, we can expect the quality of robot locomotion, sensing, and other required tasks to improve as well, making an autonomous multi-robot solution to WiSAR increasingly practical, logistically and otherwise.

2.1 Problem Formulation

A robotic WiSAR scenario is assumed to proceed as follows: at a certain point in time, a notification arrives of a missing person, hereby referred to as the ‘target’. The Last Known Position (LKP) of the target at time $t = 0$ is given. Next, since the target is assumed to be non-trackable and non-communicative, probabilistic information about the location of the target within the search area needs to be derived. This information would be used to compute an optimal initial deployment of robots within the search area. All available robots are, then, transported to the search region, and, upon arriving there, are deployed according to an optimal initial deployment at time $t = T_{hs}$ (i.e., the target's 'head-start' over the robots from time $t = 0$). Optimal search-paths are, then, planned for each robot, and the robots commence the search by executing these paths.

It is assumed that basic topographic information of the region is available, and represented in the form of a height-map matrix that is created by discretizing the search area into a rectangular grid. Moreover, much of the information that is derived and used in conducting the search comes from knowledge of the target. In a WiSAR scenario, notification of a missing person typically arrives from those who know the lost-person (i.e., friends and family), and who have reason for concern as to the lost-person's whereabouts because the person is 'overdue' in some sense. Namely, the person presumed to be lost has not returned home (or to some other expected location) after having indicated their intention to pursue a certain activity (e.g., going camping, skiing, hiking, hunting, etc.). As such, the target's identity and other personal information can be determined through interviews with those who report the person missing. Therefore, in a WiSAR scenario, elementary knowledge of the personal characteristics of the target, such as age, clothing, provisions, purpose of visit, health, experience, and familiarity with the area, is also assumed to be available.

Figure 2.1 shows the three main tasks, outlined earlier, that must be conducted in an MRC methodology for robotic WiSAR. These tasks represent the three major sub-problems that comprise the overall robotic WISAR problem addressed herein.

The proposed 3-task cyclical structure shown in Figure 2.1 is necessitated by the logical order of the information flow that is required for the overall problem: target-location prediction, followed by interpretation of this probabilistic information for search-effort distribution (i.e., robot deployment), and finally, planning and implementation of the optimal search-paths (i.e., robot-path planning). The system needs to continually update the probabilistic model of the target's location, and re-compute the optimal deployment as well as the optimal search-paths for all the robots, to maintain the effectiveness of the overall search, given the current system state. This would keep the probabilistic information on the target's location up-to-date, and ensure that optimal allocation of search effort is maintained. The optimal deployment, although regularly re-evaluated, would only be modified when deemed necessary. Similarly, the optimality of the search-paths also need to be regularly evaluated (*during* path-implementation), and re-planned if necessary. As the search progresses and the robots implement their planned paths, new information will be obtained. This may include new terrain information (e.g., previously unknown obstacles encountered, or new terrain topology experienced that is different from that indicated on the available terrain map) and clues found. This information must be used to update target-location prediction, which, in turn, would impact robot deployment and robot-path planning.

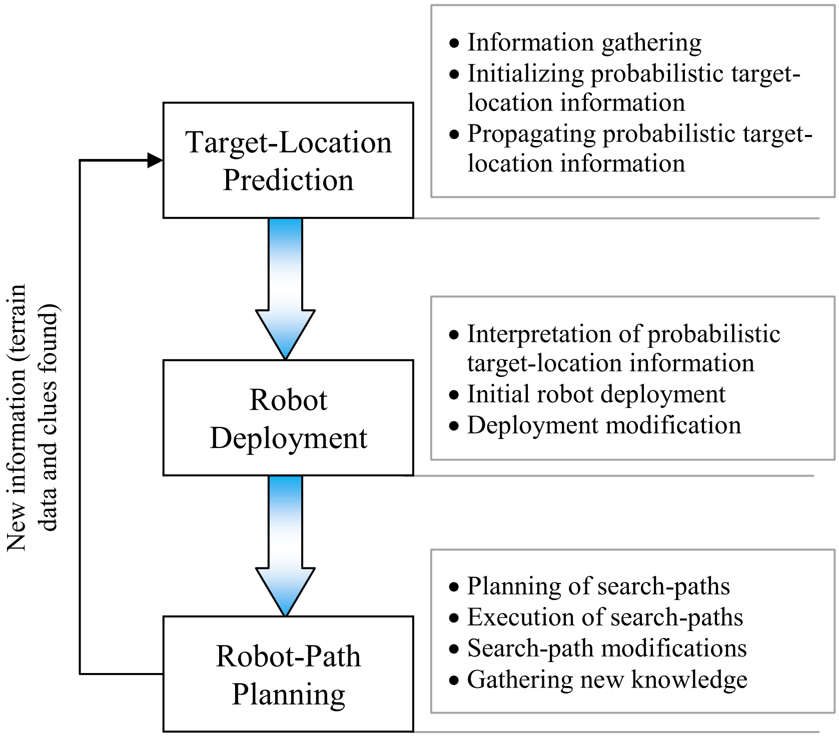

Target-Location Prediction
Robot Deployment
Robot-Path Planning
New information (terrain data and clues found)
• Information gathering
• Initializing probabilistic target-location information
• Propagating probabilistic target-location information
• Interpretation of probabilistic target-location information
• Initial robot deployment
• Deployment modification
• Planning of search-paths
• Execution of search-paths
• Search-path modifications
• Gathering new knowledge

Figure 2.1. The 3-task structure of the overall robotic WiSAR problem.

2.1.1 Target-Location Prediction

The foremost sub-problem in WiSAR is obtaining accurate probabilistic information about the target for target-location prediction. This task primarily involves generating probabilistic information about the location of the target within the search area and updating this information over time, *t*. As mentioned above, elementary knowledge of the target can be obtained through interviews with friends and family. In addition, once the lost-person can be classified according to a given target category, any relevant historical data associated with that category would also be considered as a part of the overall information about the target. Thus, through initial information-gathering, a collection of relevant target information would be available at the start of the search. However, it is entirely possible that as the search progresses, new information about the target can be learned and/or existing information revised. Therefore, in general, the relevant target information can be abstracted as some time-varying set, $\{I_{targ}(t)\}$, of mixed qualitative and quantitative pieces of data. This data must be used to help in constructing the probabilistic information about the location of the target within the search area.

The search area, $\mathcal{SA}$, refers to the bounded region within which the target could possibly reside at any given time. In general, $\mathcal{SA}$ can be described by three-dimensional Cartesian coordinates: $\mathcal{SA} = \{x, y, z\}_{SA}$. However, there may still be some natural limits to this area, such as a large body of water over which it is known that the target cannot travel. Therefore, in general, the search area may not include all of $\boldsymbol{R}^3$ space. Moreover, the search area is a dynamic entity, with boundaries that grow over time due to target motion. Hence, more accurately, the search area would be a time-varying space, $\mathcal{C}(t)$, that is a subset of $\boldsymbol{R}^3$ space, giving:

$$\mathcal{SA} = \{(x, y, z)\}_{SA} \in \mathcal{C}(t) \subseteq \boldsymbol{R}^3 \tag{2.1}$$

Thus, the objective in target-location prediction is to derive a probability distribution, P_{POS}, for the location of the target within $\mathcal{SA}$ at any given point in time, t. However, obtaining such a distribution that accurately describes the target-location likelihood requires the consideration of several influences specific to WiSAR scenarios. This significantly complicates this first main sub-problem of robotic WiSAR. Outlined below are four important influences that will be considered in this Thesis.

2.1.1.1 Influence of Target-Motion Model

Firstly, any probabilistic information on the location of the target within the search area must come from some probabilistic information describing generally how the target is likely to move. As reviewed in Section 1.2.4, search theory approaches typically refer to this target-motion information as the target-motion model. For instance, this model could represent a probability distribution for the speed of the particular target being sought, which, through an appropriate transformation, can be converted into a probability distribution for target position within the search area for a specified point in time. In any case, such a model must be obtained from available information about the target, $\{I_{targ}(t)\}$, and for any given time, t, it would be used to compute the target-location probability distribution, P_{POS}. Let P_{MOT}, then, represent the available probabilistic information about the motion of the target from which P_{POS} must be obtained.

2.1.1.2 Influence of Terrain Topology and Target Physiology

However, in addition to elementary target information, the probable motion of the target will be influenced by a number of other factors. For one, the terrain over which the target moves would impact the target's motion. The impact of the terrain contained within the search area at any given time, though, would, in turn, depend to a significant extent on the target's physiology. Target physiology covers that set of knowledge dealing with the physical characteristics of the target, and its implications on the target's mobility over different terrain topologies. Data relevant to determining target physiology would be contained in $\{I_{targ}(t)\}$, and a particular function, $f_{phys}\left(\{I_{targ}\,(t)\}\right)$, would have to be defined that represents the general relationship between terrain topology and the corresponding physiological response from the target. For example, this function could represent a search within an empirical database listing average walking-speed changes over different inclines relative to flat ground for males and females of different age groups; or, perhaps, a mathematical function that has been fit to such data. Once such a function is available, it can be used along with information on the terrain topology within $\mathcal{SA}$ at time t to determine the overall terrain influence on P_{MOT}. Let $\mathcal{M}$ represent the terrain topology information (typically, in the form of a height-map matrix corresponding to a rectangular grid-array overlaid onto the search area). Then, the influence of terrain within the search area on P_{MOT} can be represented by some function defined as:
$g_{ter}\left(f_{phys}\left(\{I_{targ}\,(t)\}\right), \mathcal{M}, x, y, z, t\right)$, where $\{(x, y, z)\} \in \mathcal{SA}$.

2.1.1.3 Influence of Found Clues

Yet another influence to be considered is that of clues found during the search. As the robots follow their planned paths and move through the search area, information about the possible location of the target in the form of clues may be found. Clues can be of different types, and it would be necessary to abstract this concept into some type of information that can be autonomously used to compute the relevant influence on P_{MOT}. Typically, each clue that is found will have position information, (x_{clu}, y_{clu}, z_{clu}) (i.e., *where* the clue was found) and time information (i.e., *when* the clue was found, T_{clu_fnd}, and, possibly, *when* it was 'dropped' by the target, T_{clu_drp}). As time progresses, multiple clues may be found, resulting in the information set,

$\{(x_{clu}, y_{clu}, z_{clu}, T_{clu\_fnd}, T_{clu\_drp})_c\}$, containing information on each clue $c \in [1, N_{clu,t}]$, where $N_{clu,t}$ is the total number of clues found during the search by time t. Since clues contain information corresponding to past positions of the target, it is evident that this information would influence one's prediction of where the target is likely to be. A method must be devised, therefore, to compute what the influence of clues that have been found would be on P_{MOT}. Let this influence of clues, then, be represented by the following quantity:

$g_{clu}\left(\left\{\left(x_{clu}, y_{clu}, z_{clu}, T_{clu\_fnd}, T_{clu\_drp}\right)_c\right\}, N_{clu,t}, t\right)$, where $c \in [1, N_{clu,t}]$.

2.1.1.4 Influence of Target Psychology

A fourth influence on the derivation of P_{MOT} would also be target psychology. Accounting for the influence of psychology presents a particularly difficult challenge due to the fact that information on the psychological behaviour of lost-persons tends to be in the form of qualitative descriptions of behavioural tendencies, given the type of target and the type of environment within which the target is lost. For instance, information gathered from the past experiences of search commanders in SAR missions, and contained in publications such as [119], indicate that lost children between the ages of 1 to 3 years tend to display a shelter-seeking behaviour where, upon moving randomly for a period of time, will seek-out the nearest shelter location (e.g., inside a cave, under thick brush, or under an overhanging rock) and remain there. Children 7 to 12 years of age, on the other hand, resort to 'trail-running' behaviour, where they may simply follow dirt roads, train tracks, or pipelines that they may happen to find, in the hopes that it will lead them out of the wilderness. It would, therefore, be necessary to devise a method to quantify the influence of such behaviours on the probabilistic information on target motion, given the target behavioural information in $\{I_{targ}(t)\}$ as well as the terrain topology $\mathcal{M}$. Let $g_{psy}\left(\{I_{targ}(t), \mathcal{M}\}\right)$ represent a function that computes such an influence.

2.1.1.5 Amalgamation of Influences

Given all these influences, a method would have to be devised that can combine their impacts, compute associated uncertainties, and derive the probabilistic information on the target's motion. Letting Tf_1 represent a transfer function that performs such an operation, the following is obtained:

$$Tf_1 \begin{bmatrix} g_{ter}\left(f_{phys}\left(\{I_{targ}(t)\}\right), \mathcal{M}, x, y, z, t\right), \\ g_{clu}\left(\left\{\left(x_{clu}, y_{clu}, z_{clu}, T_{clu_{fnd}}, T_{clu_{drp}}\right)_c\right\}, N_{clu,t}, t\right), \\ g_{psy}\left(\{I_{targ}(t), \mathcal{M}\}\right) \end{bmatrix} \longrightarrow P_{MOT} \tag{2.2}$$

Finally, once the probabilistic target-motion information is obtained, a second process must take the result and derive the corresponding target-location probability information, P_{POS}, which is the main objective of the overall target-location prediction task. Using Tf_2 to represent the transfer function that carries out this process, the final result can be expressed as:

$$Tf_2\left[P_{MOT}, \{I_{targ}(t)\}, \mathcal{SA}, t\right] \longrightarrow P_{POS} \tag{2.3}$$

Thus, it is evident that simply obtaining probabilistic information about the location of the target within the search area is a significantly complicated task. Not only is it non-trivial to appropriately model and represent the different influences that need to be considered, but it is also a difficult matter to incorporate the impact of those influences into P_{MOT} and subsequently derive P_{POS}. Moreover, it is clear that some of the components involved in these influences are time-dependent. Thus, as the corresponding information changes with time, these influences, and therefore, P_{MOT} and P_{POS}, would have to be modified accordingly. Indeed, the probabilistic information on the position of the target changes continuously with time due to target motion. This entails having to re-compute P_{POS} at regular time-intervals to ensure that the most up-to-date probabilistic target-location information is available for the search.

Given such complexity, it is apparent that a search theory approach where all these influences must be modelled as mathematical functions and P_{MOT} and P_{POS} must be computed analytically would not be feasible for robotic WiSAR. This is especially true when one considers the additional constraint that the computation of P_{POS} must be done on-line, multiple times throughout the search.

2.1.2 Robot Deployment

The second sub-problem in robotic WiSAR is the deployment of the multi-robot team. As mentioned earlier, this sub-problem requires distributing the search resources (i.e., the robots) within the search area in the most effective way possible. Thus, a unique position within the search area must be assigned to each robot. The determination of what positions within the search area require robots, and the assignment of specific robots to specific positions, constitutes the robot deployment sub-problem.

Moreover, deployment also entails ensuring that, at any given time, the distribution of robots within the search area continues to remain the most effective one. Hence, at the start of a search (at time $t = 0$), an initial distribution of the search resources is determined and implemented. Thereafter, the effectiveness of the distribution of these resources must be continually monitored (as the robots move about to perform the search), and a re-distribution must be computed and implemented, when required.

However, attention must be drawn to a key distinction regarding the terminology used. The term ‘deployment’ used here refers only to the act of allocating robots to particular position coordinates within the search area. The deployment task does not involve the planning or execution of the paths that the robots must take to perform the search after having reached these positions.

Given the above objectives of the deployment problem, it is evident that robot deployment consists of two main sub-tasks: (i) determination of where robots need to be placed, and (ii) assignment of individual robots to the locations determined.

2.1.2.1 Deployment Sub-Task #1: Selecting Locations for Search-Effort Allocation

In the first sub-task, the objective is to determine where, within the search area, would it be most beneficial to allocate the search effort. To make this determination, the benefit of any given search-effort allocation within the search area has to be quantified. The term ‘benefit’ refers to how useful a given search-effort allocation would be to the ongoing search. Therefore, to quantify this benefit, specific attributes of the search process must be selected and expressed quantitatively. Each such attribute would be a measure quantifying the impact of a given distribution of search resources to that attribute. These individual measures would, then, have to be combined in some fashion to produce a single, final measure representing the overall benefit of that given distribution of search resources to the ongoing search process. For instance, one of the main concerns in any SAR situation is to find the lost-person as soon as possible. Thus, one attribute of interest could be the time that would be taken to find the target, given a particular search-effort allocation. Quantifying such an attribute, however, can be a very difficult task. The time required to find the target, for example, would be impossible to compute exactly in a WiSAR scenario given its complexity, and the fact that the target is never observed during the search. Nevertheless, an estimate or indication of such an attribute can still be made using the available information.

Generally, how well a given distribution of search resources will perform in addressing any given attribute (i.e., the impact of a given distribution of search resources on an attribute-measure) would depend on:

i. the quantity of search resources available,

ii. the target-detection effectiveness of those resources (e.g., sensing radius of a search robot),

iii. the mobility of those resources (e.g., a robot's speed-response over different terrains),

iv. probabilistic information about the target's location, and

v. probabilistic information about how the target moves.

Although one may have deterministic knowledge of the search resources at hand, the pertinent information on the target is probabilistic. Clearly, though, robot deployment must be conducted in accordance with the probabilistic target-location information at any given time so that search effort is proportioned according to the relative likelihoods of target presence in the search area at that time. Therefore, quantification of the important attributes of the search, as well as the overall benefit entailed by any given distribution of search resources, would depend on the available probabilistic information on the target, namely, P_{MOT} and P_{POS}. Given quantified measures of these attributes, they would, then, be combined through an overall benefit function.

However, in a WiSAR scenario, in general, the search area is large relative to the area that can be covered within the combined sensing ranges of all the available search resources in any reasonable amount of time. Certainly, as time passes and the search area grows, it is assured that such would eventually be the case. Therefore, there would be an infinite number of possible search-effort allocations that can be selected (i.e., there would be an infinite number of options as to where robots can be placed within the search area). Each choice of search-effort allocation would have a different impact on the different attributes of the search, and therefore, on the benefit to the overall search. It is clear, then, that the selection of the most beneficial distribution of search resources would require an optimization. The decision variables would be the deployment positions to select, and the objective of this optimization would be to select the optimal deployment positions that maximize the overall benefit function. An additional point to note is that, in general, although the main purpose of robot deployment is to find deployment positions, the process of determining such positions would require an intermediate step to identify regions within the search area where search-effort must be allocated. Identification of regions for search-effort allocation would be an internal task performed in the computation of one or more of the attributes. Delimiting such regions helps to narrow down the areas within which deployment positions must be selected. For example, the search area may contain a particular region within which there is a very high probability of target presence, as indicated by P_{POS}, and optimal deployment positions would be positions within such a region. The computation of certain attribute-measure values would involve identifying those regions and computing some characteristic of them. Thus, although the final output in which one is interested during robot deployment are optimal deployment position coordinates, through the process of finding these positions, specific regions for search-effort allocation will also have been determined. Such regions are a by-product of optimizing robot deployment, and they play an important role in the last main task of MRC for robotic WiSAR (i.e., robot path planning).

2.1.2.2 Deployment Sub-Task #2: Assigning Robots to Deployment Positions

With respect to the second sub-task, the objective is to determine which robot to assign to each of a given set of optimal deployment positions. Any given assignment of this type would be associated with a corresponding level of effort required to accomplish the entailed deployment. Since this Thesis deals with the problem of robotic WiSAR, one cannot ignore the costs involved in trying to attain a specified deployment. In fact, this issue has a counterpart in search theory approaches. As mentioned in Section 1.2.1, early search theory methods simplified the search-effort allocation problem by assuming that ‘switching’ costs were negligible. In other words, although multiple regions at different locations in the search area can be specified within which search-effort must be allocated, one assumes that the cost required in travelling from one region to the next is negligible. When robots are involved, such an assumption cannot be made. Given a particular deployment position to attain, a robot must expend a finite amount of effort to reach that position, where that effort would be some measurable quantity, such as distance to travel, travel time required, etc. In the interest of overall efficiency, it is logical that one would wish to minimize such effort. Thus, the decision as to which robot should be assigned to which deployment position becomes yet another optimization problem, where one wishes to find the optimal assignment of robots to a specified set of optimal deployment positions such that the effort entailed by that assignment is minimized.

However, since an ‘assignment’ in the context of robot deployment refers to a particular deployment position that each robot must reach, a further complication arises. Namely, the selection of deployment positions affects the effort that any given assignment scheme entails. Therefore, the optimal amount of effort that can be achieved is limited by the optimal deployment positions that have been selected. Although a given selection of deployment positions may provide great benefit to the overall search as expressed by the overall benefit function, even the best assignment of robots to those deployment positions may entail a significant amount of effort required by the robots to simply attain those positions. This dependency between the two objectives of determining optimal deployment positions and determining the optimal assignment of robots to those deployment positions means that these two objectives must, in general, be considered together. In fact, the effort-measure quantified for any given assignment scheme would represent yet another attribute of the overall search, since minimization of this effort would be beneficial to the search that is in progress. Thus, this effort measure would be represented as such, and must be combined with all other attributes specified above when composing the overall benefit function. The result would be a single optimization that accounts for both the objectives listed above for robot deployment. The choice of the assignments of robots to deployment positions would, then, constitute additional decision variables, over and above the deployment positions to be selected, that must be specified when optimizing the overall benefit function.

2.1.2.3 Formulation of Deployment Optimization

To express the foregoing more formally, and to summarize the overall optimization involved in the robot deployment sub-problem, let Rb_i refer to the i^{th} search robot, where $i \in [1, N_r]$ and N_r robots are available for the search. Let us further assume that each robot, Rb_i, has a particular associated measure of target-detection effectiveness, e_i, as well as a function describing its mobility, $f_{m,Rbi}$, over given terrain topology, $\mathcal{M}$.

One of the objectives of the deployment problem is to determine particular (optimal) deployment positions for the robots, which form one set of decision variables. Let the set $\{P_{Dj}\} = \{(x_D, y_D, z_D)_j\}, j \in [1, N_r]$, represent the N_r deployment position coordinates that need to be selected. Recall that robot deployment involves assigning *each* robot to a *unique* deployment position. Hence, there must be as many deployment position coordinates as there are robots. Next, as discussed above, in order to compute a measure of the overall benefit to the search entailed by a given set of search-effort allocations (i.e., a given selection of deployment positions), particular attributes of importance of the search must be defined and modelled, depending on what the user considers to be relevant and what they wish to account for (e.g., estimate of time required to find the target).

Let Atr_k, $k \in [1, N_{atr}]$, represent the k^{th} attribute among the N_{atr} attributes that have been defined. It is important to note that computation of an attribute could involve any of a number of different processes (e.g., evaluating a closed-form mathematical function, running a simulation, performing a black-box process involving several different types of computations, etc.), depending on what any given attribute is meant to model, but with the end-result being a single attribute-measure value. Each of the attributes that are modelled would be a function of the deployment position coordinates $\{P_{Dj}\} = \{(x_D, y_D, z_D)_j\}$, as well as the parameters related to the search resources, which are the number of robots available, N_r, and the target-detection effectiveness e_i, of each robot, Rb_i, $i \in [1, N_r]$. Also, since a significant source of information used to guide the search will come from the prediction of the target's location at any given time, the attributes would also be functions of the available probabilistic information on the target's motion and location: P_{MOT} and P_{POS}.

The second objective, corresponding to the second main sub-task of robot deployment, is the optimal assignment of robots to a given set of deployment positions. This optimal assignment depends on minimizing some measure of the effort required to implement a given robot assignment, and this measure becomes another attribute of the ongoing search that influences the overall benefit function. Let *Eff* represent this particular attribute, which would be a function of the positions of all the robots, P_{Rbi}, $i \in [1, N_r]$, the deployment position coordinates, $\{P_{Dj}\}, j \in [1, N_r]$, and the mobility functions, $f_{m,Rbi}$, $i \in [1, N_r]$, of the robots. The additional decision variables that must be considered in order to address the optimal assignment sub-task are variables specifying which deployment position a given robot is assigned to. Since an 'assignment' of a robot to a position can be represented in different ways (e.g., the position coordinates of the deployment position assigned to a robot, or a binary variable representing a yes/no decision on the assignment of each robot to each deployment position, etc.) let some set of variables $\{a\}_i$, $i \in [1, N_r]$, represent variables describing which deployment position the i^{th} robot is assigned to. Since the effort attribute, corresponding to a given assignment of robots to a specified set of deployment position coordinates, influences the overall benefit of a robot deployment to the search, this attribute must be considered together with all the others that have been defined. As such, the assignment variables, combined with the deployment position coordinates, constitute the entire set of decision variables for the overall robot deployment optimization problem. Finally, let B_{dep} represent the objective function of this optimization problem. This function quantifies the overall benefit to the search for a given selection of all the decision variables, by combining the influences of all the defined attributes of the search.

The overall optimization problem for robot deployment can, therefore, be represented by the following:

$$Maximize\ B_{dep} \\ = f\big(\{Atr_k\}, Eff, \{(x_D, y_D, z_D)_j\}, \{a\}_i, P_{MOT}, P_{POS}, N_r, \{e_i\}, \{f_{m,Ri}\}, \mathcal{SA}, \mathcal{M}, t\big), \tag{2.4}$$

$$\forall i \in [1, N_r], \forall j \in [1, N_r], \forall k \in [1, N_{atr}],$$

subject to:

$$(x_D, y_D, z_D)_j \in \mathcal{SA} = \mathcal{C}(t), \forall j \in [1, N_r], \tag{2.5}$$

$$\{a\}_i \neq \{\emptyset\}, \forall i \in [1, N_r], \tag{2.6}$$

$$\{a\}_m \neq \{a\}_n, \forall m, n \in [1, N_r], m \neq n. \tag{2.7}$$

The first constraint given by Eq. 2.5 indicates that the deployment positions that are specified must lie within the search area corresponding to the given time *t* for which the optional deployment is computed. The second constraint given by Eq. 2.6 represents the requirement that every robot *must* be assigned to a deployment position, while the constraint given by Eq. 2.7

indicates that each robot must be given a *unique* deployment position (i.e., no two robots can have the same deployment position assigned to them).

It is evident that the resulting robot deployment optimization problem is quite complex, since it involves the consideration of many different issues, each of which could itself be difficult to model and compute. As a result, the computational complexity in solving this optimization problem would be significant. Thus, making the computation of optimal robot deployment an on-line feasible task would be a challenge, and concessions in terms of the optimality of the solutions or the level of detail in which different parts of the above optimization are modelled and implemented, may have to be made to ensure on-line computational feasibility. In addition, the different attributes involved in Eq. 4, and the overall benefit function itself, would, obviously, be functions of time. This implies that at a later time, a given optimal deployment solution may no longer be optimal, and a re-optimization and corresponding re-deployment may be required. A given optimal deployment must, therefore, be regularly re-evaluated during the search, and a new optimal deployment must be implemented, whenever necessary, to maintain the optimality of the search. This necessitates the additional requirement that the checking of an implemented deployment for optimality throughout the ongoing search be done in a time-efficient and on-line-feasible manner as well.

2.1.3 Robot-Path Planning

The third sub-problem of robotic WiSAR to be addressed is robot-path planning, which involves determining how the robots should move throughout the search in order to find the target. Once again, since the target is non-trackable, the determination of the best way in which each robot should move would have to be based on the available probabilistic information about the motion of the target, P_{MOT}, and the location of the target at any given time, P_{POS}. Just as with robot deployment, robot paths must be planned in such a way that their motion provides the most benefit to the ongoing search.

However, the solution to the optimal robot deployment sub-problem already involves a quantification and evaluation of a measure of benefit to the search based on important attributes that have been defined. Moreover, this solution dictates the optimal regions where search-effort must be placed, as well as the optimal deployment positions within these regions, which, together, correspond to a particular distribution of the search resources within the search area. As such, the optimal deployment solution constrains the robot search paths. In particular, robots must move such that they remain within the regions to which they have been allocated. This region information would be contained within the attributes that were defined for the robot deployment optimization. Namely, by virtue of optimizing the individual attribute values (which, in turn, maximize the overall benefit function), the optimal regions for search effort allocation will have been delimited, and the optimal deployment positions within them will have been selected. For example, if one of the attributes were to represent an estimate of the time required to find the target, the optimization (i.e., minimization) of this attribute's value would have involved delimiting the regions where search effort should be allocated to minimize this search time. Subsequently, the optimal deployment positions would have been selected that lie within these regions. Therefore, when robot paths are planned, the robots would be constrained to remain within the regions corresponding to their assigned optimal deployment positions.

Even with such search-effort allocation regions pre-specified, robot-path planning is not a trivial task since there would be uncertainty involved in how a given robot moves and whether or not it remains within its respective region. A major source of this uncertainty would be related to the terrain information available. The terrain topology information in $\mathcal{M}$ may not necessarily be accurate, either because of errors in the terrain data, low resolution of the terrain information, or both. Moreover, *a priori* unknown obstacles may be present in the terrain, which, when encountered by a robot, would disrupt that robot's trajectory and its ability to follow its assigned path. Hence, in planning robot paths that adhere to the constraints placed by the corresponding search-effort allocation regions, it is necessary to compute paths that minimize the uncertainty involved in doing so. Thus, once again, it is evident that an optimization must be involved in performing this third sub-task of the WiSAR problem. Namely, robot path planning is an optimization problem.

For any given robot, then, planning its path will require specifying a starting position, a destination position, and a path-definition connecting the starting position to the destination. Naturally, the starting position of the path would be the robot's position at the time at which the robot must commence implementation of its path. Given uncertainty information about the terrain data, a function must be defined which takes this uncertainty information and computes the corresponding uncertainty along any defined robot path. In other words, this function must compute the amount of error that can be expected between the actual region within which a given path would lead the robot, and the assigned region within which the robot is constrained to move. The objective of the optimization would be to minimize this path uncertainty. The decision variables would be the position coordinates of the destination of the path, as well as all parameters corresponding to the chosen method of path-definition. For example, if a path is to be represented by a single cubic polynomial, the coefficients of the polynomial, combined with the coordinates of the path's destination, would constitute the decision variables. This optimization would be constrained by the need to ensure that the defined path and the destination position coordinates, lie within the search-effort allocation region to which the robot in question has been assigned through the optimal robot deployment solution.

Let $P_{Rbi} = (x_{Rb}, y_{Rb}, z_{Rb})_i$ represent the position of robot Rb_i at the time for which a path is to be planned. Let the path destination position be represented by $P_{dest,i} = (x_{dest}, y_{dest}, z_{dest})_i$, and let the variable set $\{\Phi\}_i$ be the set of parameters that define any given path for robot Rb_i, based on the chosen method of path-modelling. Therefore, $P_{dest,i}$ and $\{\Phi\}_i$ would be the decision variables in the optimization of robot Rb_i's path. If $\mathcal{R}e_i \in \mathcal{SA}$ delimits the search-effort allocation region to which robot Rb_i has been assigned through optimal robot deployment, then this region, along with the terrain topology data, $\mathcal{M}$, the uncertainty information about the terrain data, $\{U_{terr}\}$, and the robot's path-definition, $\{\Phi\}_i$, would form the inputs to a function, $f_{U\Phi}$, that computes a measure of uncertainty associated with this defined path (i.e., a measure of 'error', as explained above, between where the path leads the robot and the region, $\mathcal{R}e_i$, within which the path must lie). The function, $f_{U\Phi}$, then, is the objective function that must be minimized. Formally, the optimization to determine the path for any given robot Rb_i, for time t, can be represented as follows:

$$Minimize\ f_{U\Phi} = f\big(P_{dest,i}, \{\Phi\}_i, P_{Rbi}, \mathcal{R}e_i, \{U_{terr}\}, \mathcal{M}, t\big), \tag{2.8}$$

subject to:

$$P_{dest,i} \in \mathcal{R}e_i, \tag{2.9}$$

$$\{\Phi\}_i \in \mathcal{R}e_i. \tag{2.10}$$

This optimization problem must be solved for each robot Rb_i, $i \in [1, N_r]$, in the multi-robot team, every time a path needs to be planned. Path-planning would be required at the start of a search, but also must be re-computed any time the optimal deployment is changed, since deployment affects the search-effort allocation regions to which the robots are assigned. Further still, path re-planning may be required in-between re-deployments. Among the many challenges involved in solving the path optimization problem is the determination of the uncertainty in the terrain information. Available information about this uncertainty could be very limited, and, furthermore, could change as the search progresses and new information is obtained. Thus, due to this dynamic nature of WiSAR, a single robot-path optimization computed at the start of the search or after any given re-deployment may not be enough to ensure that the path uncertainty remains minimized. This implies that the optimality of each robot's path must be checked regularly, on-line, during path-implementation, and a path re-plan must be initiated, when required. Thus, path planning must be a recurring and dynamic process, and a method must be developed that efficiently checks path-optimality for each robot during the search. This method must, again, be on-line feasible in order to allow a robotic WiSAR solution to be viable.

2.1.4 Integration

Although the three sub-problems formulated above that comprise the overall multi-robot WiSAR problem represent three main sub-tasks that must be performed, it is important to realize that these sub-problems must be addressed concurrently, in an on-line manner, throughout the ongoing search. Moreover, due to the dynamic nature of WiSAR, each sub-problem involves components that vary with time. As this information varies, so too will the applicability of the current solution to each of those sub-problems. Thus, in general, each sub-problem would have to be solved multiple times during any given search mission. This requires an effective means to coordinate the computations and optimizations involved, as well as the transfer and update of all information involved.

Primarily, an effective coordination architecture must be selected. As discussed in Section 1.2.2, multi-robot coordination methods in the literature can be either centralized or decentralized approaches. Some methods also attempt to find a compromise between these two extremes by proposing hybrid architectures that can be partially centralized under a certain set of circumstances. The robotic WiSAR problem, however, is a time-critical application where the target must be found as soon as possible. As more time passes, the chances of target-survival drop. Thus, one must conduct a search in an optimal manner, which requires access to, and processing of, all relevant available information. This can most directly be accomplished by a centralized approach.

Given the fact that a central process must carry out all the necessary computations corresponding to the three WiSAR sub-problems, and must do so in an on-line manner, the way in which this centralized architecture is implemented is critical, and poses a challenge. In other words, a centralized architecture must be designed that always allows for the successful on-line computation of target-location prediction, robot deployment, and robot-path planning, whenever necessary during the search.

Target-location prediction requires not only the computation of the initial target-location probability information based on the information about the target, $\{I_{targ}(t)\}$, the probabilistic target motion, P_{MOT}, and the available terrain map, $\mathcal{M}$, and the other influences as described in Section 2.1.1, but also the regular update of this information to account for a growing search area, $\mathcal{SA} = \mathcal{C}(t)$. As the robots implement their paths and carry out the search, new information may be obtained, such as corrected terrain topology information from sensors, the location of *a priori* unknown obstacles, and clues found, which would also necessitate updates of the target-location prediction.

Thus, the central controller must maintain appropriate contact with all search robots and determine when to trigger an update of the target-location prediction information. Although the target-location prediction is continuously changing, there would not be sufficient computational resources available to conduct continuous updates. A decision needs to be made on when to update the target-location prediction to account for target motion so that this information is kept sufficiently up-to-date to effectively perform all other tasks that depend on it. Hence, this requires the specification of a finite time-interval for updating target-location prediction information. Additionally, however, it must also be decided when, exactly, the discovery of new information should trigger additional updates of target-location prediction, over and above the scheduled ones, and what types of information changes should serve as the triggers.

With respect to robot deployment, the optimal deployment solution must be computed at the start of the search, but the optimality of this deployment must be monitored. Thus, it must be decided how often this optimality check must be done, given knowledge of the computational complexity of the checks involved. Moreover, any given checking-instance may trigger a re-deployment, and an approach must be devised to effectively manage this randomly occurring necessity for re-deployment, given the computational resources available and the other tasks being computed at the time.

Lastly, robot-path planning, as formulated in Section 2.1.3, requires the specification of an optimal destination and path-definition for each robot. However, it is obvious that a single path for the entire search cannot be planned in one path-planning instance. Thus, robot-path planning must be incorporated into a particular schedule that dictates time periods for which a suitable path must be planned for each robot that allows the robots to search within its respective assigned region, $\mathcal{R}e_i \in \mathcal{SA} = \mathcal{C}(t)$, $i \in [1, N_r]$. Establishing such a schedule naturally breaks-up robot-path planning into a set of regular planning events that can be implemented over the course of a search for an indefinite amount of time. This would add further constraints to the robot-path planning problem that, nevertheless, helps to remove ambiguity in the path planning process as to the appropriate set of possible destination positions to consider. Still, the problem becomes complicated when one is, once again, forced to contend with the uncertain, dynamic nature of the robotic WiSAR problem. Each robot's path must be monitored for optimality and, when required, re-planned. This requires an on-line-feasible method for checking path-optimality, a schedule for optimality checks, as well as a means to deal with the fact that the need for re-planning any given robot's path may occur at any time.

The abovementioned challenges involved in sub-task integration further emphasize the importance and necessity for ensuring that the method used to solve each sub-problem is computationally feasible on-line. The faster and simpler each operation is to compute, the easier it will be to integrate all the processes involved and develop a viable overall MRC methodology for robotic WiSAR.

2.2 Overview of Proposed Methodology

An effective method must be developed to address each of the sub-problems of robotic WiSAR that were formulated above. The research conducted for this Thesis has resulted in the successful development of effective and on-line-feasible approaches to solve the three main sub-problems, namely, target-location prediction, robot deployment, and robot-path planning. Each of the following three sub-sections provides an overview of the proposed approaches for these three sub-problems, respectively. The fourth sub-section will describe the proposed approach to the successful integration of the three main sub-tasks.

2.2.1 Target-Location Prediction Method Overview

In order to conduct effective target-location prediction, a source of probabilistic information about the motion of the target is required. In the approach proposed herein for this first sub-problem, the probabilistic information is obtained from lost-person data on past search-incidents that are kept by SAR organizations. Key pieces of information about such incidents are recorded and kept on file to serve as reference information for search managers in future WiSAR search missions [119]. The stored data is sorted according to different categories of lost-persons. These categories are selected based on observed commonalities in lost-person behaviours. For instance, it has been observed that children 7 to 12 years of age, upon becoming disoriented, will tend to follow trails, drainages, stream/river banks, or the like, in order to return to familiar ground or to find help [11, 119]. Hunters, on the other hand, typically become disoriented while chasing wounded game into thick areas of trees or bush. They are usually experienced enough to build shelters for the night, and then continue their attempt to regain their bearings at daybreak. As a result, identifying the lost-person category of the target being sought can help in predicting how the target will behave based on readily available information from past experiences.

More specifically, in addition to qualitative descriptions of target behaviour, data regarding ‘crow’s-flight’ distances are also recorded for all successful searches (i.e., searches where the target was found). The term ‘crow’s-flight’ distance refers to the straight-line distance measured from the LKP to the position where the target was found. In this Thesis, a technique will be proposed whereby this distance data can be analyzed and used to derive a probability distribution for average target speed along a straight line of travel. A method is, then, proposed that can use this probability distribution to subsequently derive probabilistic information about the location of the target at any specified point in time, which serves the role of P_{POS}, as discussed in the problem-formulation presented in Section 2.1.1.

Thus, it is evident that the overall approach being used here is to derive probabilistic information about the motion and location of the *category* to which the target belongs, rather than attempting to derive such information about the particular target. Although this would only give a rough idea of how the target is likely to move throughout the search, it still serves as a structured technique by which to obtain relevant probabilistic information and avoids having to simply assume that such information is given at the beginning of the search. Moreover, this Thesis will propose an overall MRC methodology for robotic WiSAR that can use this ‘rough’ information to, nevertheless, conduct an effective search.

Typically, in the literature, as reviewed in Section 1.2.4, methods that use target-location prediction will assume the availability of a 2D target-location PDF that is described either by a continuous function, or discretized into a probability matrix overlaid onto the search area. However, in WiSAR, with the need to incorporate a significant amount of information in terms of other influences into such a probability distribution, as well as the need to update and propagate this information over time and space, using such a means for defining and representing target-location prediction is not viable if the proposed methodology is to be an on-line method. Thus, target-location prediction in the method proposed in this Thesis is addressed using a novel approach that involves delimiting boundaries of regions of target presence, similar in concept to iso-elevation contours used in topographic maps. These boundaries are defined based on computed target-*motion* probability information, derived using readily-available data pertaining to the target’s category as mentioned above. This probabilistic target-motion information (serving the role of P_{MOT}), in turn, can be used to derive P_{POS} for any time-value specified.

In the approach proposed in this Thesis, multiple such boundaries are established, where each boundary bounds a region associated with a particular probability for target-presence. For example, four boundaries could be established, corresponding to 100%, 70%, 50%, and 30% probabilities for target-presence, respectively, as illustrated in Fig. 2.2. The location of the boundaries in the search area as well as the boundary densities in different regions of the search area, allow one to say something about where search-effort should be focussed in order to have the best chance of finding the target. This approach of using such probabilistic boundaries provides a novel means of representing the relevant probabilistic target-location information that is also useful in guiding robot deployment and robot-path planning. One of the advantages of this approach is that the task of storing probability information on target-location reduces to storing the static raw data corresponding to the target category identified at the start of the search, and the established boundaries. Moreover, this approach also facilitates the task of updating probability information to account for target motion and of incorporating other relevant influences.

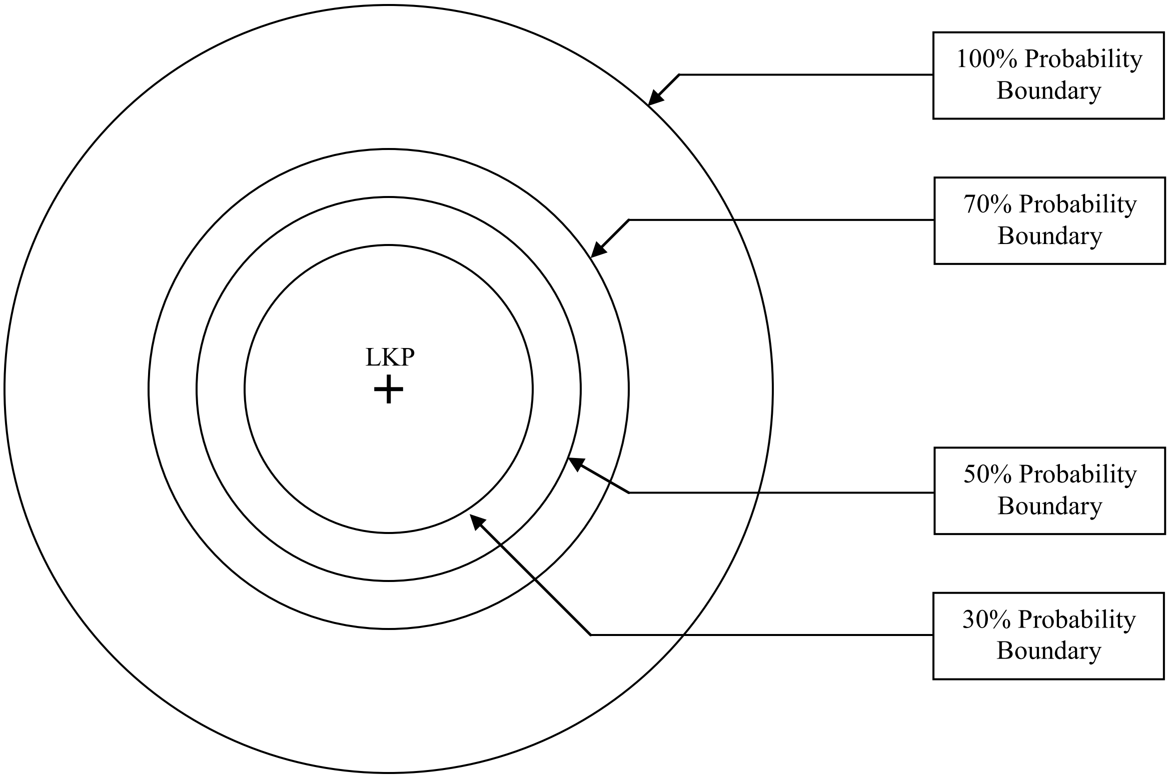


Figure 2.2. Illustration of the concept of probabilistic boundaries for target-location prediction.

In fact, the use of this novel representation mechanism for target-location prediction allows for a modular approach to incorporating the pertinent influences that were discussed in the problem formulation in Section 2.1.1. As illustrated conceptually in Fig. 2.3, the proposed method for target-location prediction involves first computing a set of 'nominal' boundaries describing the probabilistic information about the target's location at any given time. This target-location information is 'nominal' in the sense that it only portrays target location probability based on simple target travel outward from the LKP that does not account for any other influences, such as terrain, target physiology, obstacles, clues found, etc. Once this 'nominal' information is established, the impact of each relevant influence can be incorporated individually and sequentially to gradually modify the boundary definitions.

This incorporation of each influence would be associated with its own method. For instance, to account for the influence of terrain on the target's probable location, a mapping function is first established that analyzes the relevant portions of the terrain to determine a measure of difficulty that the corresponding topology would impose on the target. This difficulty measure allows for the modification of the probabilistic information used to construct the boundaries to reflect this difficulty. This, in turn, results in modified boundaries through which target-location prediction is represented. In this way, the impact of terrain on the motion, and thereby, the location, of the target at any given point in time is accounted for. Similarly, methods will be proposed to incorporate the effect of other relevant influences, including *a priori* known and unknown obstacles, clues found, and particular target psychologies, and it will be shown that these methods allow for fast computation, thereby facilitating on-line employment of this approach.

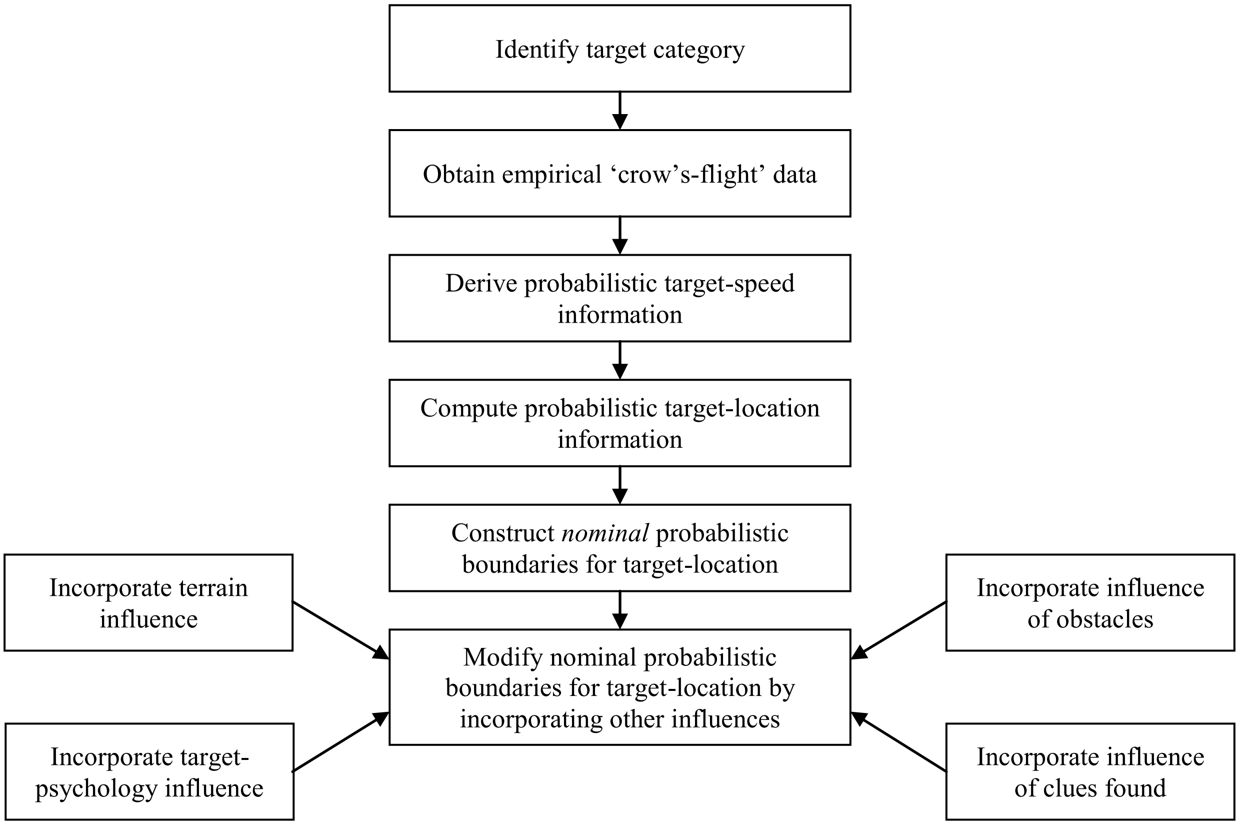

Identify target category
Obtain empirical 'crow's-flight' data
Derive probabilistic target-speed information
Compute probabilistic target-location information
Construct *nominal* probabilistic boundaries for target-location
Modify nominal probabilistic boundaries for target-location by incorporating other influences
Incorporate terrain influence
Incorporate target-psychology influence
Incorporate influence of obstacles
Incorporate influence of clues found

Figure 2.3. Modular approach to target-location prediction.

The other major requirement for target-location prediction is the update and propagation of this prediction over time to account for target motion. With the novel approach to deriving and representing this prediction, update and propagation is a straight-forward matter of re-computing the nominal set of boundaries from the probabilistic target-motion information for the new point in time to be considered. Each of the other influences is, then, re-applied, and an updated set of boundaries, representing the updated target-location prediction, is obtained. The computationally efficient methods devised to perform these operations allow for this approach to be conducted in an on-line manner.

2.2.2 Robot Deployment Method Overview

In order to determine an optimal deployment of the search resources, two tasks need to be addressed, as discussed in the robot deployment problem formulation in Section 2.1.2. First, the optimal deployment positions within the search area must be determined. Second, the robots must be optimally assigned to these deployment positions. However, as mentioned in Section 2.1.2, an intermediate step in determining the optimal deployment positions is the establishment of corresponding optimal regions for search-effort allocation. Since all the pertinent information available about the target at any given time is the probabilistic information on its location within the search area, it is evident that optimal deployment must be guided by this information. This information is concisely contained in, and represented by, the probability boundaries established for the target-location prediction sub-problem. The use of this probability-boundary-based representation mechanism for target-location prediction allows for a natural and convenient means for specifying the regions to be considered for optimal deployment. Thus, the proposed method uses these boundaries to help in delimiting the regions where search-effort should be allocated, which, in turn, are used to determine the optimal deployment positions.

2.2.2.1 Deployment Sub-Task #1: Optimization for Selecting Locations for Search-Effort Allocation

The first task in robot deployment, then, begins with the selection of the optimal regions for search-effort allocation, which is transformed into a task of establishing an optimal set of boundaries. These boundaries bound the regions in which search-effort should be placed, and in which the optimal deployment positions must lay. As described in the problem formulation for robot deployment, this optimization would depend on the benefit that the choice would have on the search in progress. This benefit must be quantified by establishing specific attributes of the search that are of importance and subsequently determining the impact that any given selection of search-effort allocation regions (i.e., selection of a set of boundaries), and the deployment positions within those boundaries, would have on those attributes. From the literature on search theory, as well as studies of how real WiSAR operations are conducted, it is evident that there are typically two main concerns when determining how to coordinate a search. The first is ‘search-time,’ which refers to the expected total time that would be required to find the target, and the second is ‘success-rate,’ which refers to the probability of finding the target. Thus, in the method proposed in this Thesis, determination of the optimal regions for search-effort allocation, and the optimal deployment positions, involves an optimization in which the objective function includes a weighted-sum of metrics that quantify the effect that any specified selection of these deployment variables would have on search-time and success-rate. These two quantities represent two of the attributes of importance when optimizing robot deployment.

In devising metrics for search-time and success-rate, the proposed approach first involves application of a technique to determine a particular balance in distribution of search resources among a given set of search-effort allocation regions. This balance depends on the relative size of the regions as well as the total number of search robots available. This technique allows for the specification of the *number* of robots that must be allocated to each region, without having to specify which robots, in particular, should be assigned to each region, or the exact deployment positions for each robot. In this way, an indication of the amount of search-effort that each region would get is made available. From this distribution information, as well as information on the capabilities of the search robots (including their speed and their target-sensing radius) and the size of each of the search-effort allocation regions, quantitative measures that are commensurate with search-time and success-rate of the ongoing search can be computed. It is important to note, however, that such measures would only give an indication of search-time and success-rate, since it would be impossible to compute ahead of time how long it would take to find the target and how likely it is that the target will be found. Furthermore, the values of the search-time and success-rate metrics would guide the selection of search-effort allocation regions, as well as the number of robots required per region. However, deployment positions within these regions are not selected directly at this stage. Instead, the establishment of the deployment positions is intertwined with the second task in robot deployment, namely, robot assignment.

2.2.2.2 Deployment Sub-Task #2: Optimization for Assigning Robots to Deployment Positions

In the second task in optimal robot deployment, robots must be optimally assigned to specific deployment positions within the search-effort allocation regions. However, with multiple robots to assign, and with each robot having, in general, an infinite number of possibilities for a possible deployment position within the search-effort allocation regions delimited, this could very quickly become an intractable problem. Thus, to address this issue, and to help ensure on-line feasibility of the method, an alternate, simplified approach is proposed. In particular, rather than finding optimal deployment positions and optimal assignment of robots to those positions, the strategy proposed herein is to optimally assign robots to the finite set of search-effort allocation regions, and, once accomplished, to let the corresponding shortest distances from the robots to their assigned regions dictate the deployment positions. The advantage of this approach is two-fold. Firstly, it significantly decreases the computational complexity involved in having to search for specific deployment positions for each robot from among an infinite number of possibilities. Secondly, the use of shortest-paths minimizes the effort required by the robots to attain their assigned deployment positions, thereby allowing more time and effort to be spent on actual searching through search-path implementation, which benefits the overall ongoing search.

Indeed, as discussed in the formulation for robot deployment in Section 2.1.2, the optimal assignment problem *requires* the minimization of the corresponding effort entailed, which, in turn, is a function of the regions selected for search-effort allocation. It is this dependency that necessitates modelling the effort entailed by a given assignment scheme as an attribute of the search that must be considered concurrently with all the other attributes in the overall deployment optimization. In general, an assignment scheme would refer to a specified paring of each robot with a unique deployment position within a specified search-effort allocation region. Moreover, any given assignment scheme can be characterized by a value representing the amount of effort required by the robots to attain their deployment positions. Since one wishes to conduct a search in as efficient a manner as possible, it is clear that this effort value must be minimized. In the method proposed in this Thesis, this effort is a function of the estimated time that would be required by the robots to reach specific assigned search-effort allocation regions via the shortest paths. The point at which a shortest path, corresponding to a particular robot, intersects the robot's assigned region, becomes the deployment position assigned to that robot.

2.2.2.3 Overall Optimization Structure for Robot Deployment

With the approach described above for the two main sub-tasks, robot deployment becomes a simpler problem of searching for and delimiting optimal search-effort allocation regions within the search area, and an optimal assignment of robots to those regions, where each region can be assigned more than one robot. This requires a nested optimization structure, where a second optimization, *within* the overall deployment optimization, needs to be conducted to optimally assign robots to each set of regions that are considered by the optimization search engine, and to determine the corresponding value for the time-based assignment effort attribute. Through this 'inner' optimization of robots to regions, the use of shortest-paths allows for the determination of specific deployment positions for the robots as well. As mentioned above, a technique has been developed that allows for the computation of the *number* of robots required in each search-effort allocation region that results in a balanced distribution of search resources among regions. These robot-number requirements per region form the constraints for the 'inner' optimization.

Figure 2.4 illustrates the dependency between the two main sub-tasks of the robot deployment sub-problem and the two optimizations involved. In this figure, the circular area represents the search area $\mathcal{SA} = \mathcal{C}(t)$ at some time *t* for which an optimal deployment needs to be determined. In order to determine the optimal deployment positions, the optimization search-engine would perform a search in which it would have to consider different options for the search-effort allocation regions. Figure 2.4 illustrates an example instance of this search in which a particular option of regions is being considered. Two regions, labelled $\mathcal{R}e_k$, $k \in [1, 2]$, are being considered, and are represented as annular areas of different sizes within the search area. The robot-requirements for the two regions have been computed to be three and two, respectively, as indicated in the figure. Moreover, five robots, Rb_i, $i \in [1, 5]$, indicated by the blue dots, must be assigned to these two search-effort allocation regions, in accordance with these regional robot requirements.

During any such instance of robot deployment optimization, as shown in Fig. 2.4, there is enough information to compute the search-time and success-rate attributes in order to evaluate their contribution to the deployment optimization objective function, B_{dep}. However, the attribute corresponding to the effort, *Eff*, entailed by the optimal assignment of robots to deployment positions within the two regions requires its own optimization. In other words, every time the optimization search engine considers a particular set of search-effort allocation regions during the overall deployment optimization, the corresponding value of the assignment effort attribute, *Eff*, is determined by finding an optimal assignment of robots to those regions based on shortest-paths, and taking the corresponding endpoints of those paths to be the corresponding optimal deployment positions. In this way, once an optimal set of regions are found, the corresponding set of optimal assignments of robots to those regions, as well as the corresponding set of optimal deployment positions, will automatically have been found as well. In Fig. 2.4, the dashed red lines emanating from the robots represent the shortest paths used to determine the optimal assignment of the robots to the two regions, as well as the corresponding optimal deployment positions indicated by the five red '×'s, labelled P_{Dj}, $j \in [1, 5]$. Each of the shortest paths is associated with a time-value, $\Delta\tau_i^*$, $i \in [1, 5]$, which are aggregated in some fashion to compute the corresponding optimal effort attribute value, Eff^*.

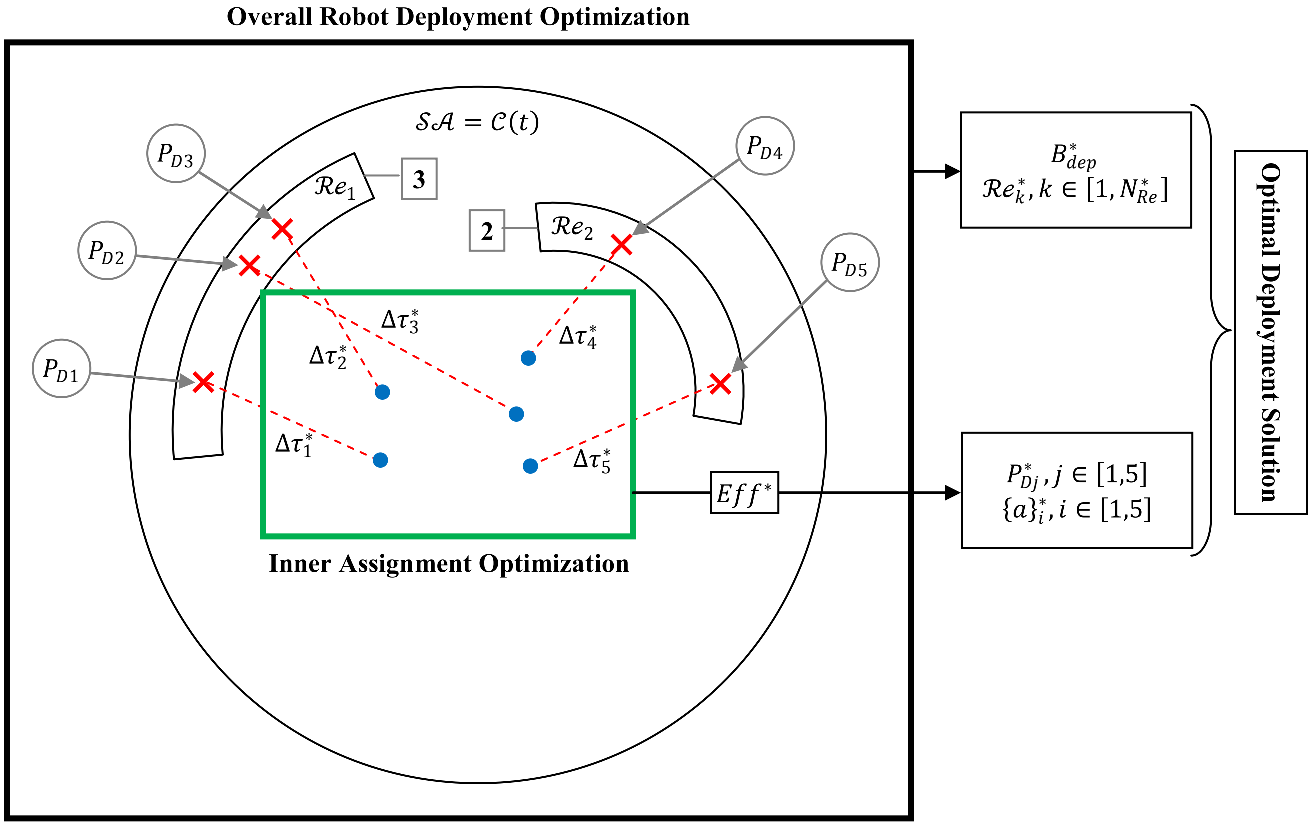


Figure 2.4. Conceptual illustration of the optimizations involved in the robot deployment sub-problem.

The thick black rectangular border in Fig. 2.4 represents the overall deployment optimization that is underway, which, upon completion, will output an optimal overall solution in terms of an optimal set of deployment positions, P_{Dj}^*, $j \in [1, 5]$, within an optimal set of regions, $\mathcal{R}e_k^*$, $k \in [1, N_{\mathcal{R}e}^*]$ for search-effort allocation, as well as an optimal set of assignments, $\{a\}_i^*$, $i \in [1, 5]$, of robots to those optimal deployment positions. This overall optimization solution maximizes the overall benefit that the selected deployment has to the search. This optimal benefit is quantified by the deployment benefit function, B_{dep}^* (Eq. 4 from the formulation in Section 2.1.2). The thick green rectangular border represents the inner optimization that is used to determine the optimal assignments of robots to search-effort allocation regions and the corresponding deployment positions. Through this inner optimization, the optimal deployment positions, P_{Dj}^*, $j \in [1, 5]$, are determined, as well as the optimal assignments, $\{a\}_i^*$, $i \in [1, 5]$, of robots to those optimal deployment positions.

For the overall robot deployment optimization, then, the proposed approach uses search-time, success-rate, and the optimal assignment effort value, Eff^*, as the attributes of importance of the search, when evaluating the overall deployment benefit objective function (Eq. 4). In particular, a weighted-sum of these attributes forms this objective function. The computation of these attributes involves specification of the regions for search-effort allocation, deployment positions within these regions, and optimal assignments of robots to these deployment positions. However, one can recall that specification of search-effort allocation regions requires the use of the probabilistic boundary-based representation mechanism used to represent the target-location prediction information at any given time. This constitutes the link between the target-location prediction information and the optimal robot deployment, thereby ensuring that the available probabilistic target-location information is used to optimize the robot deployment. The final result of robot deployment optimization, then, is the best way of distributing the search resources within the search area at any given time, using all the relevant information available, so that the overall search will be performed optimally.

A major consequence of using the probability boundary-based representation mechanism to delimit the search-effort allocation regions is that these regions, then, become inextricably linked to the probability boundaries defined. Therefore, the search-effort allocation regions, as defined and used in the proposed method for MRC in WiSAR, are portions of the continuously changing target-location prediction information. As this probability information propagates with time due to target motion, so too do these search-effort allocation regions. Thus, the search-effort allocation regions are not stationary, and they propagate *with* the target-location prediction.

This raises the last issue of relevance in robot deployment, namely, the necessity for re-deployment. At the start of the search, an initial deployment of robots is computed based on the target-location prediction derived at that time and the positions of the robots. Through this initial optimization, optimal regions for search-effort allocation are delimited, optimal deployment positions within these regions are found, and these positions are optimally assigned to the robots. Upon attaining their respective assigned deployment positions, the robots start to move within their regions using paths that are individually planned for each robot using the robot-path planning method to be outlined in the next section and detailed later. Nevertheless, it is clear that the optimality of a given deployment depends to a great extent on the target-location prediction, since this prediction is used to decide where search-effort should be allocated, which, in turn, constraints the optimal deployment positions. However, this prediction changes with time. As time progresses, the search area enlarges, and the target-location prediction propagates correspondingly to account for potential target motion. This will naturally affect the optimality of the initial deployment that was implemented. In particular, at some point in time, it is entirely possible that the distribution of search resources dictated by the initial optimal deployment solution is no longer optimal. In such a situation, the deployment must be revised. This revision entails a re-optimization to find the new optimal deployment solution. A new optimal deployment solution could change the optimal regions where search-effort should be allocated, the number of robots required in each of these new regions, the optimal deployment positions within those regions, and the optimal assignment of robots to those optimal deployment positions. When this happens, the robots must cease implementation of their current paths and plan new paths in order to move to and attain their newly assigned deployment positions. Upon attaining these new deployment positions, search-paths are planned and implemented as usual to continue the search. This process of revising the optimal deployment and having the robots stop their current search-paths to attain their new deployment positions is termed ‘re-deployment.’ Furthermore, since the target-location prediction changes continuously throughout the search, it is also clear that other re-deployments at future points in time throughout the search may be necessary. Therefore, throughout any given search, re-deployment may occur multiple times.

As mentioned in Section 2.2.1, the target-location prediction is updated and propagated at regular discrete time-intervals in order to account for target motion. In the proposed approach to robot deployment, the optimality of a given implemented deployment is checked at these same time-intervals. This is a logical approach since any change to the target-location prediction could require a change to the optimal deployment, and in a time-critical application such as WiSAR, one would be interested in maintaining the optimality of the search as best as possible. However, at the same time, all robots would still require a finite amount of time in order to travel to and reach any newly-specified deployment positions when a re-deployment is implemented. Time and effort spent in trying to attain deployment positions is time and effort taken away from actual searching, and this is undesirable. Hence, although a new optimal deployment solution may exist that is different from the current one after a target-location prediction update occurs, the benefit in increased optimality from this new deployment may not be sufficiently large to outweigh the negative effects of the undesirable time and effort that must be spent in attaining the new deployment positions. A method has, therefore, been devised that gauges the relative increase in optimality provided by a new optimal deployment solution, computed at any given target-location prediction update instance, relative to the current deployment solution that is being used at that time. If this increase in benefit is found to be sufficiently large, re-deployment is conducted. Otherwise, the current deployment solution is maintained. Figure 2.5, then, summarizes the overall proposed robot deployment approach that has been outlined here.

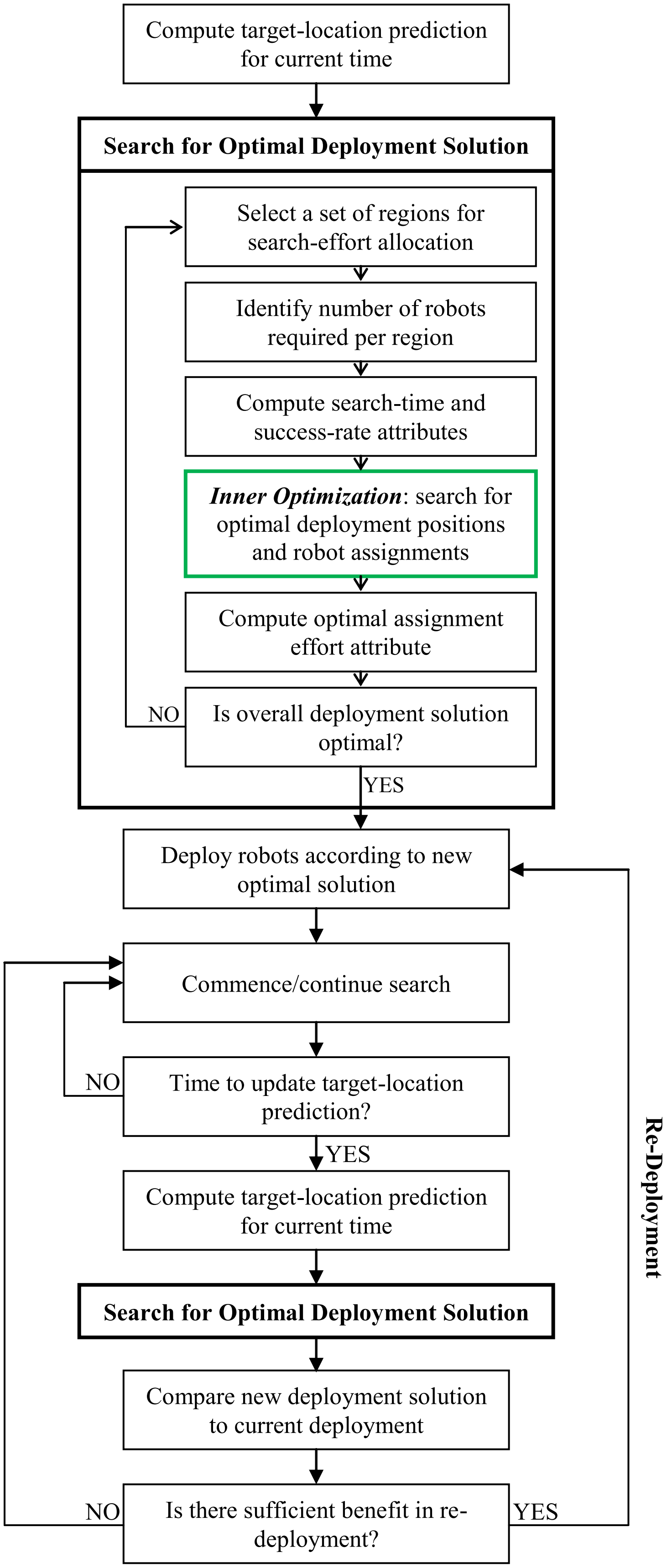

Compute target-location prediction for current time
Search for Optimal Deployment Solution
Select a set of regions for search-effort allocation
Identify number of robots required per region
Compute search-time and success-rate attributes
Inner Optimization: search for optimal deployment positions and robot assignments
Compute optimal assignment effort attribute
Is overall deployment solution optimal?
NO
YES
Deploy robots according to new optimal solution
Commence/continue search
Time to update target-location prediction?
NO
YES
Compute target-location prediction for current time
Search for Optimal Deployment Solution
Compare new deployment solution to current deployment
Is there sufficient benefit in re-deployment?
NO
YES
Re-Deployment

Figure 2.5. Flowchart summary of the overall robot deployment approach.

2.2.3 Robot-Path Planning Method Overview

As stated above in Section 2.2.2, the proposed MRC methodology for robotic WiSAR separates the concepts of deployment and searching. Deployment, as outlined above, begins when an optimal deployment solution has been found and has been decided to be implemented, and ends once all robots have reached their assigned optimal deployment positions. As soon as these positions are reached, searching begins, where robots move about their assigned regions by following specific paths that have been planned for them. Since this Thesis focuses on the methodology for coordinating robots autonomously for WiSAR, it is assumed that the robots have the ability to navigate on their own and follow path-plans assigned to them, and that they are capable of autonomously implementing simple procedures when required, such as boundary-following when an unknown obstacle is encountered. In this way, each robot requires relatively less computational ability, and, aside from each being supplied with its own copy of the latest terrain topology map, does not need to share any other information.

Robot-path planning involves the determination of the best way in which each robot should move (after having attained its optimal deployment position) in order to provide the most benefit to the ongoing search. However, the paths that are planned are constrained by the output of optimal deployment. Namely, in addition to the deployment positions to be attained, the optimal deployment solution also dictates and delimits the optimal regions for search-effort allocation. These regions will have been selected in such a way as to maximize the overall benefit to the ongoing search. As such, it is imperative that all search-paths are planned in such a way that they keep the robots within their respective search-effort allocation region. This will ensure that the search progresses in an optimal fashion, since the search-effort will continue to be expended within the optimal regions. This imperative forms the fundamental principle that governs robot-path planning, namely, that the paths must be planned such that each robot remains within its assigned search-effort allocation region.

The degree to which this imperative can be achieved, however, is affected by the continuously changing target-location prediction information as well as the uncertainty in the available terrain topology information. As noted previously, an important point to realize is that the search-effort allocation regions, although optimally selected during robot deployment, are not stationary. They are, in fact, continuously changing by virtue of the fact that the probabilistic-boundary based approach to target-location prediction representation is used to delimit the search-effort allocation regions. As this prediction changes and propagates with time, these regions will propagate as well. This produces a challenge for robot-path planning. Specifically, if a robot is to search within its assigned search-effort allocation region, then a path must be planned that allows the robot to move *with* this changing region. In other words, it must, somehow, be ensured that each robot remains within its assigned region at all times as it moves to perform the search.

To address this challenge, the proposed method for robot-path planning uses estimates of the future location of its assigned search-effort allocation region in order to plan a valid path that adheres to the aforementioned fundamental path-planning principle. In particular, since the target-location prediction is updated at fixed discrete time-intervals, it is possible to compute, ahead of time, the future location of each region corresponding to the future state of the target-location prediction, one time-interval into the future. Thus, if the target-location prediction is updated at Δt_{prop} time-intervals, then for a robot Rb_i situated within its assigned search-effort allocation region $\mathcal{R}e_i \in \mathcal{SA} = \mathcal{C}(t)$ at some time t, the future location of its assigned region, $\mathcal{R}e_i \in \mathcal{SA} = \mathcal{C}\left(t + \Delta t_{prop}\right)$ at time $(t + \Delta t_{prop})$ can be computed. This would necessarily require the computation of the future state of the target-location prediction at time $(t + \Delta t_{prop})$ ahead of time as well. However, the computational efficiency in the proposed method for computing and propagating the target-location prediction allows this approach to feasible on-line.

Having computed the future location of each search-effort allocation region, a path can, then, be planned for each robot from its current position within the current location of its assigned region to a destination position within the computed future location of this region. In order to help ensure that any given robot remains within its region at all times, then, one simply has to ensure that the path is planned such that it takes the robot precisely Δt_{prop} time to traverse, thereby ensuring that at time $(t + \Delta t_{prop})$, the robot will still be within its assigned region. If one continues this approach of successively computing the next future location of each robot's region Δt_{prop} forward in time, planning an appropriate path with traversal-time of Δt_{prop}, and having the robot implement this path, then it can be ensured that the robot always remains within its region throughout the search.

However, in general, there would be an infinite number of possible paths that can be planned leading from a given robot's current position to a destination position within the future location of its assigned region. Each path would have a unique traversal-time associated with it. Thus, a search must be conducted to find the path that has a traversal-time as close as possible to Δt_{prop}. This implies that an optimization is required. The proposed method employs a path-construction technique whereby, for a specified destination position, a single unique smooth path can be planned. Moreover, a method is also devised to estimate the time required to traverse any given path. For each robot, then, the optimization to find a suitable path involves a search through the possible destination positions within that robot's destination region (i.e., the future location of the robot's assigned region), to find a destination position for which the corresponding path can be traversed in as close to Δt_{prop} time as possible. This optimization is illustrated conceptually in Fig. 2.6 for a single robot that must move from its current position within the current location of its assigned search region (represented by a solid annular area) to the computed future position of this region (represented by a dashed annular area). Some possible destination positions are shown as red '×'s and the unique paths corresponding to each of those destination positions are shown as dashed red lines, each of which has a unique traversal-time value, $\Delta\tau_{trav}$, associated with it. This optimization must be carried out for each robot individually. The output would be an optimal destination position and a corresponding path.

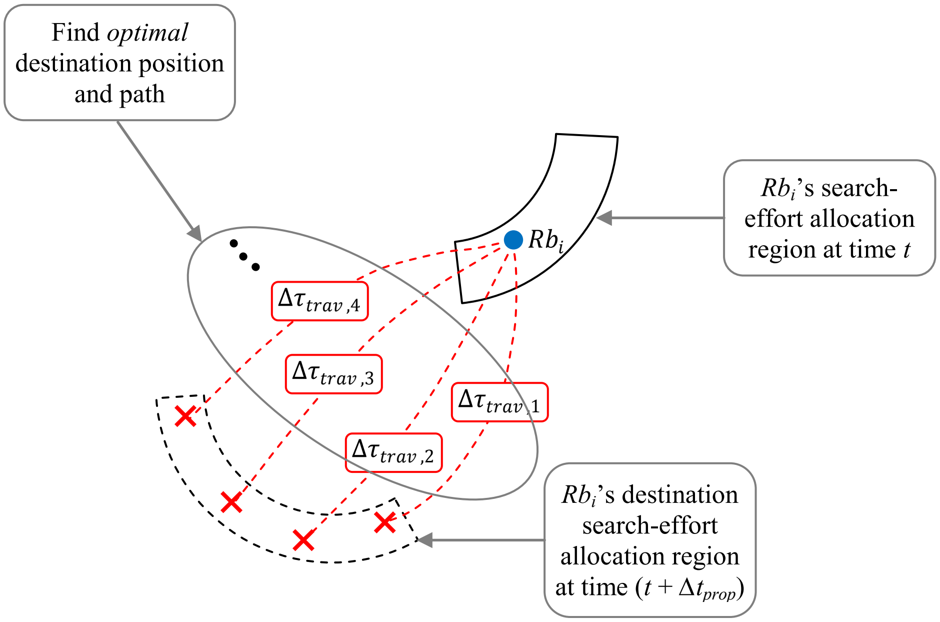

Find *optimal* destination position and path
Rb_i
Rb_i's search-effort allocation region at time t
$\Delta\tau_{trav,4}$
$\Delta\tau_{trav,3}$
$\Delta\tau_{trav,1}$
$\Delta\tau_{trav,2}$
Rb_i's destination search-effort allocation region at time $(t + \Delta t_{prop})$

Figure 2.6. Conceptual visualization of robot-path planning optimization for any given robot.

The challenge in computing this optimization, of course, is that for any given robot and its specified path, this path-traversal-time cannot be computed exactly. This is due to the inherent uncertainty in the terrain information available. The time taken by a robot to traverse a specified path will naturally depend on the terrain over which the path takes the robot. Areas of steep slope would tend to slow the robot down, whereas flat terrain or a downhill slope would allow the robot to travel relatively faster. If the terrain were known exactly (i.e., without error compared to the actual terrain the robots encounter), then traversal-time for any specified path can be computed ahead of time, exactly, based on knowledge of the robot's dynamics. However, if available terrain information is uncertain, then any pre-computed path traversal-time based on this information would have uncertainty as well. Thus, in general, it is not sufficient to simply minimize the difference between the computed path traversal-time and the desired path traversal-time of Δt_{prop}. Instead, as was formulated earlier in Section 2.1.3, one must also account for this uncertainty in the minimization. Doing so, however, would be a complex problem that would be difficult to solve, and which would not be very feasible as an on-line method.

In order to deal with this difficulty, the proposed method for robot-path planning separates the selection of an optimal path based on traversal-time, from consideration of the impact of uncertainty. This strategy involves planning an initial path for each robot from the robot's current position to a destination position within the estimated future location of its corresponding search-effort allocation region. This optimization, as illustrated conceptually in Fig. 2.6, computes path traversal-times based on the available terrain data by assuming that this terrain information is known with certainty. Once an optimal destination position and corresponding path is found, the robot begins to implement this path. As the robot is implementing its path, it gathers terrain uncertainty data, on-line, based on the terrain it experiences as it travels along its path. Then, at pre-designated check-points along the path, an evaluation of path-feasibility is conducted where the uncertainty data gathered is used to give an indication of path sub-optimality – the more sub-optimal the path is found to be, the more likely it is that the robot will not arrive at the destination position at the desired arrival time of $(t + \Delta t_{prop})$. Since several such check-points are designated along each robot's path, and the determination of path sub-optimality must be conducted rapidly on-line, this evaluation process takes the form of three time-efficient checks that must be performed, which give an indication of the degree of sub-optimality of the path. If significant path sub-optimality is detected, then the remainder of the path is re-planned by finding a new optimal destination position in the future propagated assigned region of the robot in question.

Using a decoupled strategy such as this for optimal robot-path planning, thus, provides for an on-line feasible solution. Consequently, the proposed method for robot-path planning is dynamic in two respects: firstly, the search paths must be reconstructed every time the target-location prediction is updated and propagated; and secondly, the optimality of the search-paths must be regularly re-evaluated in-between these updates (i.e., while the paths are being executed by the robots) and re-planned, when necessary, to maintain optimality. Figure 2.7 provides a flowchart summarizing the proposed overall approach to robot-path planning.

Finally, it should be noted that the proposed robot-path planning method also has contingencies in place that describe how robots must behave upon encountering *a priori* known and unknown obstacles. The underlying strategy in these cases is to guide the robots to circumvent the obstacle in the most efficient way so that the original planned path can be rejoined and resumed by the robot as soon as possible. Naturally, *a priori* known obstacles can be accounted for ahead of time, whereas *a priori* unknown obstacles must be dealt with on-line as soon as they are encountered.

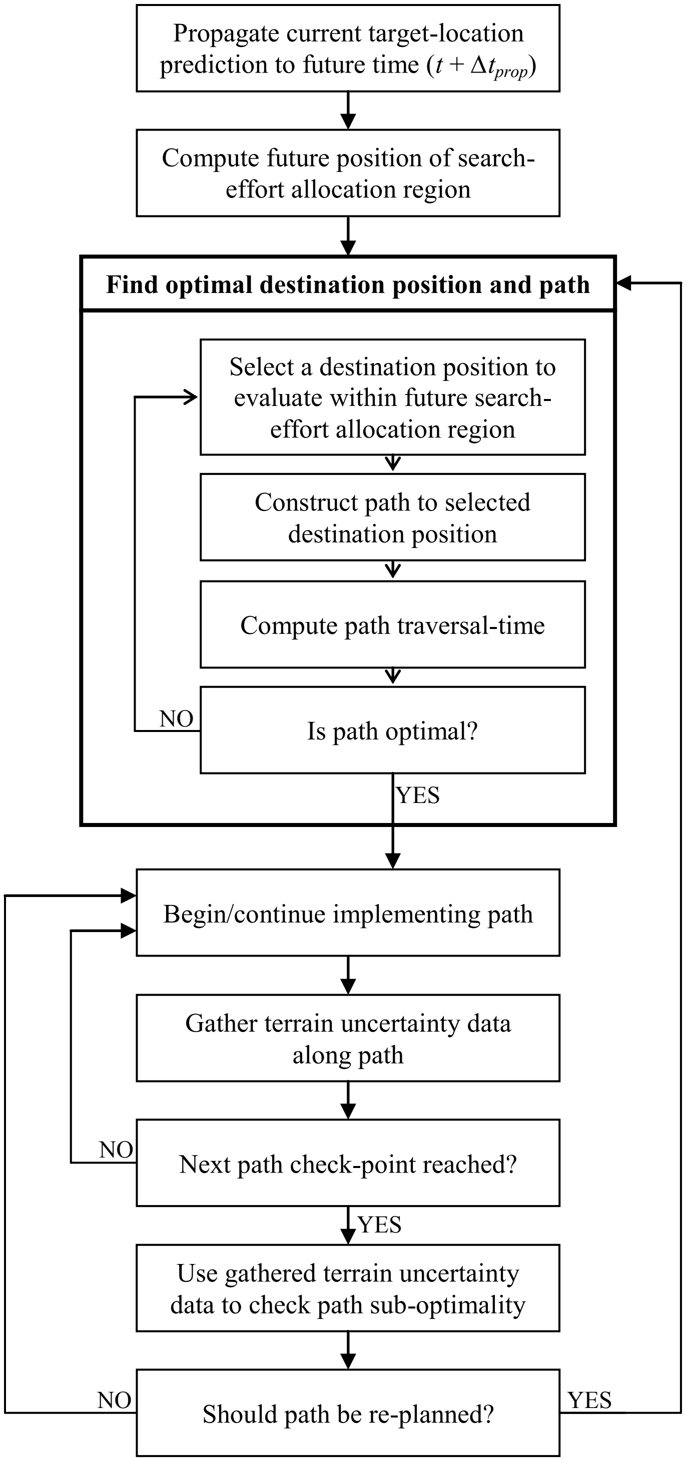

Propagate current target-location prediction to future time ($t + \Delta t_{prop}$)
Compute future position of search-effort allocation region
Find optimal destination position and path
Select a destination position to evaluate within future search-effort allocation region
Construct path to selected destination position
Compute path traversal-time
Is path optimal?
NO
YES
Begin/continue implementing path
Gather terrain uncertainty data along path
Next path check-point reached?
NO
YES
Use gathered terrain uncertainty data to check path sub-optimality
Should path be re-planned?
NO
YES

Figure 2.7. Summary of overall approach for robot-path planning.

2.2.4 Integration

As discussed in Section 2.1.4, integration of the three main sub-problems of multi-robot WiSAR requires some means to manage the computation tasks and information exchanges that take place on-line when coordinating the robots to perform the search. A centralized coordination structure is chosen, since it allows access to up-to-date global-level information, thereby making it possible to work toward an optimal solution by considering all relevant influences. To implement centralized coordination, a centralized Blackboard architecture is proposed [120].

A Blackboard architecture is comprised of three main components:

i. The Blackboard

ii. The Knowledge Sources

iii. The Control Shell

The Blackboard serves as a central, dynamic database of information. All information pertaining to the problem is placed here and updated on-line. A Knowledge Source (KS) represents an intelligent agent that has expertise in a particular aspect of the problem. When the status of the information on the Blackboard is such that a given KS can make a contribution (based on its expertise), then that KS contributes to solving the problem by adding or modifying information on the Blackboard. Multiple such KSs exist, and each will access and contribute to the information on the Blackboard whenever they are able to do so. Finally, the Control Shell serves as a 'scheduler', controlling when information is added to or updated on the Blackboard, and by which KS.

For the proposed MRC methodology for robotic WiSAR with centralized control, the Blackboard would be a central database of information to which all robots have access. Each of the important aspects of the methodology is represented by a KS. At regular time-intervals, each KS posts its corresponding output, updating the Blackboard accordingly with the most recent relevant information from the search. In this way, a separate, decision-making module then has continuous access to all the pertinent (global-level) information, and is, thus, able to determine the optimal task allocations for the robots (i.e., their deployment destination positions, when necessary, and their search-paths). In the proposed method, a central controller with real-time access to the Blackboard assumes the role of this decision-making module. The KSs necessary for the overall MRC methodology are:

i. Background Information – this KS provides all the necessary information to initialize the search process, including the initial terrain topology height-map, $\mathcal{M}$, all the information relevant to the target (i.e., elementary data, target category, etc.), $\{I_{targ}(t)\}$, and information about the capabilities of all the available search robots.

ii. Map Update – this KS incorporates any new terrain data obtained by the robots into the current terrain topology height-map, $\mathcal{M}$.

iii. Target-Location Prediction – this KS initializes, updates, and propagates the target-location prediction information, P_{POS}, using the available probabilistic information on target motion, P_{MOT}.

iv. Robot Deployment – this KS uses all the information on the Blackboard and determines the optimal regions for search-effort allocation, the optimal deployment positions for the robots, and the optimal assignment of robots to those positions at the beginning of the search. During the search, this KS also evaluates the system state, using the latest information available on the Blackboard, to decide if a re-deployment is required. When a re-deployment is found to be necessary, this KS computes the new deployment solution as well, and posts it on the Blackboard.

v. Robot-Path Planning – this KS uses relevant information from the Blackboard to plan initial optimal search-paths for the robots, and to evaluate path-optimality during their implementation by the robots at the designated check-points on the paths. Whenever it determines that a robot's path needs to be re-planned, this KS also computes the new optimal path for that robot and posts it on the Blackboard.

vi. Robot Data – this KS regularly updates the Blackboard with any new information obtained by, and pertaining to, the robots. This includes current position, current orientation, terrain data (such as terrain topology that has been experienced during search and any *a priori* unknown obstacles encountered), clues found, and a target-sighting. This information is needed by the Map Update KS to modify the terrain topology height-map, if required, and by the Robot Deployment and Robot-Path Planning KSs to perform their checks and optimization tasks.

vii. Central Controller – this KS uses the latest information posted by the Robot Deployment and Robot-Path Planning KSs to assign the appropriate tasks to all the robots. The initial solutions output by these two KSs is used to initially deploy the robots and, subsequently, to get them moving to perform the search. Updates posted by these two KSs during the search are used to submit modified task allocations to the robots that need it (e.g., a re-planned path for a robot when it reaches a check-point, or a re-deployment of all robots after a target-location prediction update). In addition, the central controller indicates when a search should be terminated, either because a specified time-limit has been reached, or because a robot reports a target-sighting.

Control within this Blackboard architecture is maintained based on prescribed time-intervals for key tasks and specific triggers during the search. From the method overviews in Sections 2.2.1 to 2.2.3, it is evident that the time-interval by which the target-location prediction is updated sets the cadence for the execution of the other sub-tasks. Each time the probabilistic information on the location of the target is updated, a new deployment optimization has to be conducted in order to determine whether a re-deployment is required. If the answer to this inquiry is affirmative, then the robots must first move to their new deployment positions before continuing with the search. Otherwise, the robots continue with their search right away.

In addition, robot-path planning is also dependent upon the frequency with which the target-location prediction is updated. Namely, the proposed method for path-planning, as outlined in Section 2.2.3 above, involves planning a path from each robot's current position to a position within the computed future location of the search-effort allocation region. This future location is computed based on the same time-interval as that used for updating the target-location prediction. As such, at the time of each such update, the robots would (ideally) have completed their assigned path and would be within their assigned search-effort-allocation region corresponding to that point in time. This triggers another path-planning instance where the next search-paths are planned for the robots.

Any intermediate instances of re-deployment or path re-planning would be triggered whenever deemed necessary based on relevant evaluations of the optimality of the deployment and the robot paths. Thus, whenever a re-deployment is found to be necessary, the Robot Deployment KS computes the new optimal deployment and the Robot-Path Planning KS determines appropriate paths to attain these deployment positions. Subsequently, the Robot-Path Planning KS also computes the optimal search-path that each robot must implement upon arriving at its assigned deployment position. Furthermore, as will be detailed in the following section, clues that are found inflict a significant change to the target-location prediction, which, in turn, necessitates re-deployment as well as path re-planning. When a robot reports a clue-find, the Target-Location Prediction KS modifies P_{POS} as necessary, and the Robot Deployment and Robot-Path Planning KSs, in turn, perform their computations to determine deployment positions and subsequent search-paths for the robots. In this manner, all these random events trigger random instances of re-deployment and path re-planning computations as well.

The central controller must monitor the Blackboard to keep a look-out for these triggers, whether they are pre-scheduled or occur randomly. Upon subsequently querying the Blackboard for the latest information provided by the other KSs, the central controller can, then, gather all necessary information to allocate the appropriate tasks to the robots. Sub-task integration, then, is handled by the architecture described above, with stipulated time-intervals controlling the periodicity of the important computations. Specific triggers are used to indicate when additional computation instances and queries for information from the Blackboard are required. This strategy ensures controlled and organized execution of the search and all of the processes that take place within it. The next three Chapters will detail the proposed methods to address the three main sub-problems that were outlined above, respectively.

2.3 Chapter Summary

In this chapter, the overall MRC coordination problem for WiSAR was formulated. Three main tasks were identified, resulting in a three-task cyclical structure to the methodology. These three tasks, namely, Target-Location Prediction, Robot Deployment, and Robot-Path Planning, form three sub-problems that need to be addressed in order to devise the overall methodology. Detailed formulations for each of these three sub-problems were provided, and an overview of the approach proposed to address each sub-problem was given. Firstly, target-location prediction involves obtaining and representing probabilistic information about the location of the target at any given time within the search area in order to help guide the search. This probabilistic information must be based on an appropriate target-motion model, and must incorporate influences relevant to WiSAR, including the influence of terrain topology, target physiology, found clues, and obstacles. The notion of circular, probabilistic boundaries was introduced in order to help address the target-location prediction problem. Multiple such boundaries can be established, each corresponding to a particular probability of target presence. The relative density of these boundaries within the search area is indicative of the distribution of target-presence likelihood over the areas.

The second task, robot deployment, refers to the distribution of the search resources within the search area in accordance with the target-presence likelihood as indicated by the target-location prediction information. This is accomplished by optimally determining the regions within the search area where search effort should be placed and optimally assigning specific robots to specific positions within those optimal regions. The resulting distribution of robots would optimize some measure of benefit to the ongoing search. To maintain the effectiveness of an implemented robot deployment, an initial deployment at the start of the search may also have to be revised at different times throughout the search.

Finally, in robot-path planning, the objective is to construct effective search paths for each robot. These paths are constrained by the need to maintain the optimal robot distribution within the search area as dictated by the most recent deployment solution that was implemented. Thus, each robot's path is planned individually such that it allows the robot to search within the region to which it has been assigned. However, uncertainty in the available terrain information results in uncertainty about whether a constructed path would keep the robot within its assigned region. This complicates path planning and necessitates regular checks of path-optimality, and, when necessary, path re-planning. This sub-problem is addressed by estimating the future position of each established search-effort allocation region from the optimal deployment solution and planning a time-optimal path for each robot to a position within the future location of its assigned region.

Integration of the three main tasks is accomplished through the use of a Blackboard architecture to form a contiguous, overall MRC coordination methodology for WiSAR. Such an architecture allows for a central database of information that is populated with the latest available information on-line from all knowledge sources involved, and caters well to the needs of an MRC system with centralized control.

Chapter 3

3 Target-Location Prediction

In a multi-robot WiSAR scenario, it is assumed that, during the search, the target is non-trackable (not observed continuously or intermittently) by any of the robots, and also non-communicative (cannot be communicated with remotely), as this is typically the case in real-life scenarios. This necessitates the sub-problem of target-location prediction – if one cannot sense or communicate with the target, one must attempt to predict his/her likely whereabouts based on known information and search accordingly. Thus, the purpose of target-location prediction is to derive probabilistic information about the target's probable location and represent it in some useful form to facilitate search-planning and execution. Existing probabilistic search methods in the literature, as noted earlier, typically assume that a 2D probability distribution for the location of the target within the search area, as well as a corresponding probabilistic target motion-model, is *a priori* available. Realistically, however, since a lost person's exact movements in a search scenario are unpredictable, such an assumption is not justifiable. In this Thesis, an alternative approach is proposed.

As described in the formulation of the target-location prediction sub-problem in Section 2.1.1, derivation of the probabilistic information on the location of the target, P_{POS}, begins with obtaining and analyzing probabilistic information about how the target moves, P_{MOT}. In the proposed approach, as outlined in Section 2.2.1, P_{MOT} is obtained using relevant data pertaining to past lost-person search-incidents recorded by and available from SAR organizations. This data is further organized into particular categories of target types (e.g., children 7 to 12 years of age, hikers, skiers, etc.), since experience indicates that lost-persons falling into each such category tend to behave in a similar manner. This similarity in behaviour not only allows for the identification of certain pertinent psychological behaviours that can be exploited to help in devising the search strategy, but relevant quantitative data corresponding to each category can help in deriving probabilistic information that can be used to construct P_{MOT} and P_{POS}.

The probabilistic information that results from this approach would describe how any target within the relevant lost-person *category* would be likely to move, and where he/she would likely be located at any given time. It does not, however, describe such information about the *particular* target being sought. Nevertheless, the robotic MRC method for WiSAR proposed herein is still able to use this target-category-based information to conduct an effective search. This approach, then, precludes the need to predict exactly how the particular target being sought would move and where he/she would be located (or to assume that such information is available) and is able to derive probabilistic target-location information using readily available data.

Moreover, this Thesis also presents a novel method of representing the derived target-location prediction information. As briefly described in Section 2.2.1, this representation involves the establishment of probabilistic boundaries, similar in concept to elevation contours in topological maps. Each such boundary bounds a region containing a certain probability for target presence, and is referred to as an 'iso-probability curve.' Multiple such curves can be constructed around the LKP, each representing the boundary of a region corresponding to a different probability. In the proposed approach to target-location prediction, nominal probabilistic information on the motion and location of the target are first established, resulting in an initial set of iso-probability curves that do not account for any other influences. Then, the influence of terrain, target physiology, *a priori* known obstacles, target psychology, and clues found, are incorporated to produce a modified set of iso-probability curves that more accurately reflects the likely location of the target at a given point in time. The methodology presented in this Thesis will show how an appropriate selection, as well as regular update, modification, and propagation of these iso-probability curves, can help to effectively predict the location of a potentially mobile target, and to intelligently guide the distribution and application of search-effort over time to find the target. Section 3.1 provides the details of how the nominal set of iso-probability curves are constructed. The sections that follow will describe how the other influences are incorporated into these curves.

3.1 Basic Construction of Iso-Probability Curves

It is assumed that at the start of a search, the LKP, (x_0, y_0, z_0), of the target is known. This position provides a starting point from which to begin constructing the target-location prediction. For the construction of the nominal set of iso-probability curves, no terrain influence is considered. The terrain information is contained in the terrain topology height-map, $\mathcal{M}$, which gives the terrain height-values (i.e., z-values) for a specified pair of planar coordinates (x, y). This topology information will be used when the influence of terrain is incorporated. For the purposes of constructing the nominal set of iso-probability curves, then, it is sufficient to work in the planar coordinate system where only the x and y coordinates are considered. Thus, the LKP can be taken to be (x_0, y_0) without loss of generality.

In general, a lost person can move in any direction, $\theta \in [0°, 360°]$, from his/her LKP, achieving a trajectory that is difficult to predict. Instead of trying to predict exact target trajectories, the approach proposed here is to establish bounds on the target's location using conservative estimates of his/her motion. In particular, as the worst-case motion scenario, it is assumed that the target may travel in a straight line radially outward from the LKP at his/her average speed, yielding a maximum distance from the LKP. Although the exact mean speed of a particular target cannot be known for certain, it is possible to construct a probability distribution of the mean speeds of all targets of a general category, into which the target being sought belongs, using pertinent data gathered by SAR organizations from past lost-person search incidents [119]. Categories of lost persons could include: 1 to 6 year old children, hikers, hunters, skiers, despondent persons, etc. [11, 119]. One particular type of data gathered that is relevant for target-location prediction is 'crow's-flight' distances for successful searches [100, 119]. For any given past search incident that was successful, the crow's-flight distance, Cf, refers to the straight-line distance from the LKP that was used at the start of the search to the position where the target was found. Also recorded in incident reports for such cases is the total time, Δt_{TOT}, for which the target had been travelling from the position of the LKP until he/she was found. Thus, for each search incident, u, one can obtain the pair of data, $\{Cf_u, \Delta t_{TOT,u}\}$. Furthermore, since this data is categorized by target-type, each target category would have a set of some $N_{catData}$ data pairs, one pair for each search incident that falls into that category.

For each such data pair, u, the crow's-flight distance, Cf_u, gives a measure of the average distance that the lost-person managed to travel outward from the LKP in a total time of $\Delta t_{TOT,u}$. Thus, a measure of the average target speed, v_u, for that search-incident can be obtained by dividing the total crow's-flight distance with the total time. The result, then, would be a set of average speed data, $\{v_u\}$, $u \in [1, N_{catData}]$, for each category of target. This raw speed data can, then, be analyzed and fitted to a continuous function by matching relevant data statistics to the parameters of an existing, well-defined probability distribution (e.g., a Normal Distribution), yielding a 1D distribution of mean target speeds along a straight-line of travel. Parametric methods that perform such matching can be found in the literature, and include Moment

Estimation, L-Moment Estimation, and Maximum Likelihood Estimation methods [121], as well as methods that fit data to a generic, multi-parameter distribution providing greater flexibility, which include the Generalized Lambda Distribution (GLD), Extended Generalized Lambda Distribution (EGLD), and Generalized Bootstrap (GB) methods [122, 123].

Alternatively, one could also use the raw speed data to compute a discrete, relative frequency distribution. Subsequently, a cumulative frequency distribution can be computed to obtain data-points corresponding to a cumulative probability function for the mean speed data [124]. These cumulative probability data-points can, in turn, be fitted to a continuous function through a regression method [125]. The resulting function would provide a mapping between cumulative probability values and mean target speed. This continuous function for cumulative probability could be integrated to provide a probability density function for mean target speed, or could be used as-is to obtain cumulative probability values as required (which would still be sufficient for target-location prediction when using the proposed approach of iso-probability curves, as will soon become evident in the following).

One of the advantages of deriving target-location prediction based on a (1D) distribution of mean target speeds along a straight-line of travel is that one is not constrained to using data from SAR organizations categorized by target-type. In the literature in fields such as Biomechanics and Human Biology, a variety of studies exist that have involved the collection and organization of empirical physiological data within more general categories, such as average walking speeds for males and females of different age groups [126-128]. Such sources of information can also provide raw speed data pertaining to the target being sought, which, in turn, can be analyzed to obtain a corresponding probability or cumulative probability distribution using the aforementioned methods.

Regardless of the data sources and analysis methods used, the end result of the above process will be a 1D probability distribution. In particular, this distribution would be a Probability Density Function (PDF) for average target speed for a target travelling along a straight line, outward from the LKP. Thus, if $\{v_u\}$, $u \in [1, N_{catData}]$, represents the average speed data for a given target category (be it a SAR target-type category or a more general category), then one can obtain from it a 1D PDF, $p(v)$, for the mean-speed random variable, v, of targets falling into that category, for travel along a single straight-line path. This distribution is referred to, herein, as the 'nominal mean target-speed PDF,' and serves the role of the nominal P_{MOT} (i.e., a P_{MOT} where terrain is taken to be a planar surface, and no other influences have been included) for the target-location prediction method proposed in this Thesis. For brevity, this PDF will, hereafter, be referred to more simply as the 'mean target-speed PDF,' where the distinction between the nominal mean target-speed PDF and an otherwise modified mean target-speed PDF will be evident from the context of the discussion, or will be made explicitly clear when required. Although any probability distribution that models the available data may be used by the approach describe here, for convenience, in this Thesis, a normal distribution will be used for the nominal mean target-speed PDF, $p(v)$, with mean, μ_v, and variance, σ_v^2, for all the mean speeds, v, that individuals within a particular target category can have.

Having obtained the nominal mean target-speed PDF pertaining to the target-category of interest, it can be converted into another corresponding PDF, $p(r, t)$, for the probable distance $r \in \mathbb{R}$, $r \geq 0$, of the target along any straight-line of travel relative to the most current LKP for any time, t. This can be done through a linear transformation of variable from v to r, where:

$$r = vt. \tag{3.1}$$

The PDF, $p(r, t)$, that results from carrying out this transformation is referred to as the 'nominal target-location PDF,' further shortened to 'target-location PDF' when the qualifier 'nominal' is self-evident. It serves the role of the nominal P_{POS}.

Therefore, if a normal distribution is assumed for $p(v)$, then, due to the use of a linear transformation (Eq. 3.1), $p(r, t)$ would also be normal [124], which gives:

$$\mu_{r,t} = E(r) = E(vt) = E(v)t = \mu_v t, \tag{3.2}$$

$$\sigma_{r,t}^2 = \sigma_{vt}^2 = t^2\sigma_v^2 \text{ (thus: } \sigma_{r,t} = t\sigma_v\text{)}, \tag{3.3}$$

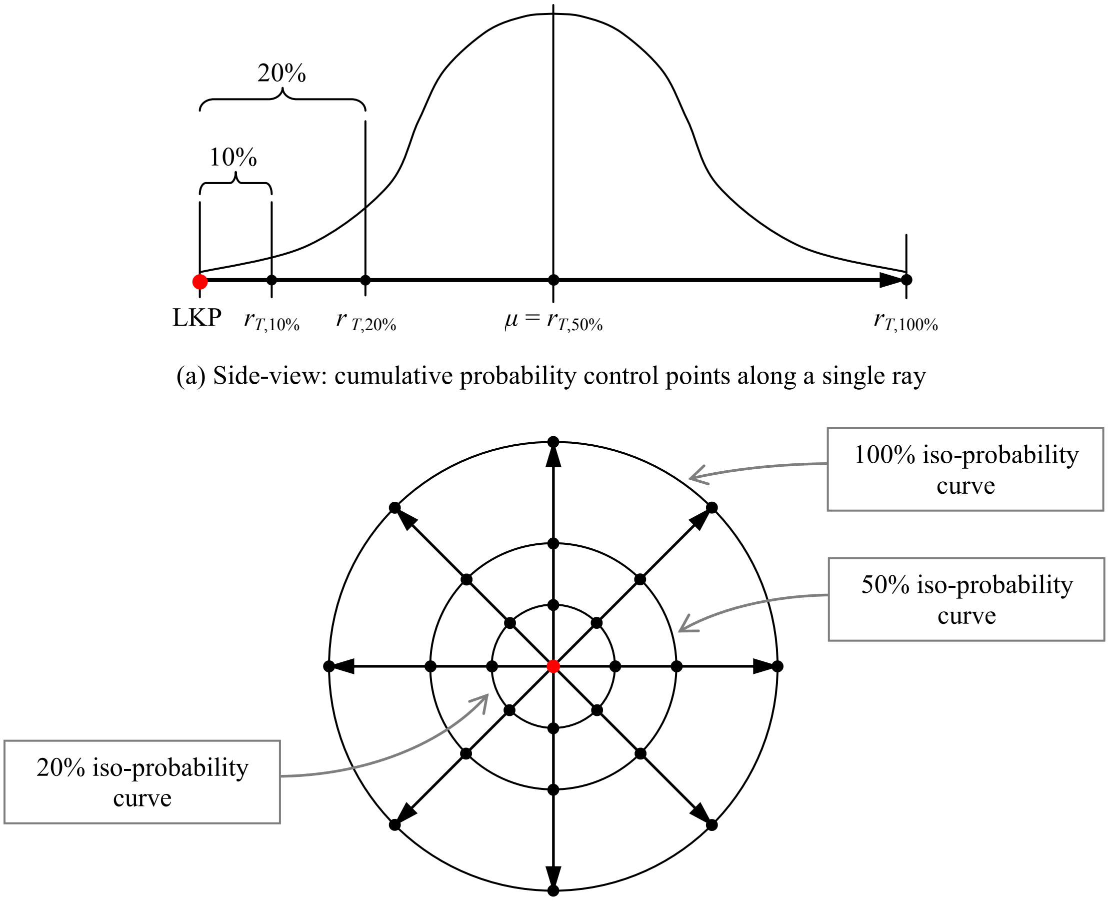
$$p(r,t) = \frac{1}{\sqrt{2\pi}t\sigma_v} exp\left[-\frac{(r-\mu_v t)^2}{2t^2\sigma_v^2}\right]. \tag{3.4}$$

Given nominal distributions for P_{MOT} and P_{POS}, the nominal set of iso-probability curves can now be constructed. Recall that the target-location PDF, $p(r, t)$, derived in the above manner is based on the assumption that the target moves at a constant mean speed, v, in a straight line outward from his/her LKP. This choice represents a worst-case-scenario estimate, since in reality the target's path would weave even as he/she tried to travel in a fixed direction. It was further assumed that the target may travel in any direction, $\theta \in [0°, 360°]$, from the LKP. Since the target can move along any direction, one can consider a subset of N_{rays} directions, θ_i, $i \in [1, N_{rays}]$, (i.e., potential lines of travel). A potential straight line of travel will hereafter be referred to as a 'ray.' Given a set of rays, the same target-location PDF, $p(r, t)$, derived above is associated with each ray, θ_i. Thus, continuing with the normal distribution assumption for $p(r, t)$, the foregoing yields:

$$p(r,t,\theta_i) = \frac{1}{\sqrt{2\pi}t\sigma_v} exp\left[-\frac{(r-\mu_v t)^2}{2t^2\sigma_v^2}\right], \forall i \in \left[1, N_{rays}\right]. \tag{3.5}$$

Using $p(r, t, \theta_i)$, derived for any particular time $t = T$, for any ray, θ_i, it is possible to identify a target-position point, $r_{T,P\%}$, along the ray, corresponding to any desired cumulative probability value, P%, Fig. 3.1a. Figure 3.1a shows the side-view of one ray, θ_i (bold arrow), where the position along the ray is given by the variable, $r \in \mathbb{R}$, $r \geq 0$. The derived (normal) target-location PDF for some time, $t = T$, is overlaid onto this ray. The ray and, thus, the target-location PDF, originates from the LKP ($r = 0$ in Fig. 3.1a) and extends up to the upper limit of this PDF ($r_{T,100\%} = v_{100\%}T$ in Fig. 3.1a). Here, $v_{100\%}$ is the maximum speed for the given target-category, and is the speed corresponding to a cumulative probability of 100% on the mean target-speed PDF, $p(v)$. Thus, the $r_{T,100\%}$ point represents the maximum distance that the target can travel in the given time, T, as dictated by the target-location PDF for this time. Each $r_{T,P\%}$ would correspond to a unique point along the ray, referred to herein as a 'control point,' and represents the average distance that a target with the P% cumulative-probability mean speed would have travelled by time, $t = T$, if he/she were moving at a constant speed and in a straight line along a ray.

However, Fig. 3.1a only shows one ray. Since multiple rays are to be considered, the same set of cumulative probability points (control points) must be considered on each ray individually. These control points established on each individual ray, θ_i, $i \in [1, N_{rays}]$, incorporate probabilistic target-location information for each separate direction, θ_i, of target motion, independently. Since only a finite set of rays are explicitly considered here, to estimate this probabilistic information for all the intermediate directions, the loci of all common control points among the established rays are taken through interpolation, Fig. 3.1b. This produces a set of continuous, 2D, closed contours over the search area, one for each cumulative percentage value, *P*%, considered. The resulting closed, 2D contours are referred to as the 'iso-probability curves.' Each such curve forms a continuous representation of the farthest (i.e., the limit) that the target could travel by time $t = T$, along any straight-line direction outward from the LKP, if the target were to have the $v_{P\%}$ mean speed. As such, each iso-probability curve corresponds to the particular cumulative probability value, *P*%, for which the control points on which that curve is based were established. Figure 3.2 illustrates another example set of iso-probability curves, where five curves, corresponding to 10%, 30%, 50%, 70%, and 100% (going from inner-most curve, closest to LKP, up to outer-most curve) cumulative probabilities for target-presence, respectively, are shown.


20%
10%
LKP
$r_{T,10\%}$
$r_{T,20\%}$
$\mu = r_{T,50\%}$
$r_{T,100\%}$
100% iso-probability curve
50% iso-probability curve
20% iso-probability curve

(a) Side-view: cumulative probability control points along a single ray

(b) Top-view: loci of iso-probability control points among 8 rays

Figure 3.1. Control points and iso-probability curves.

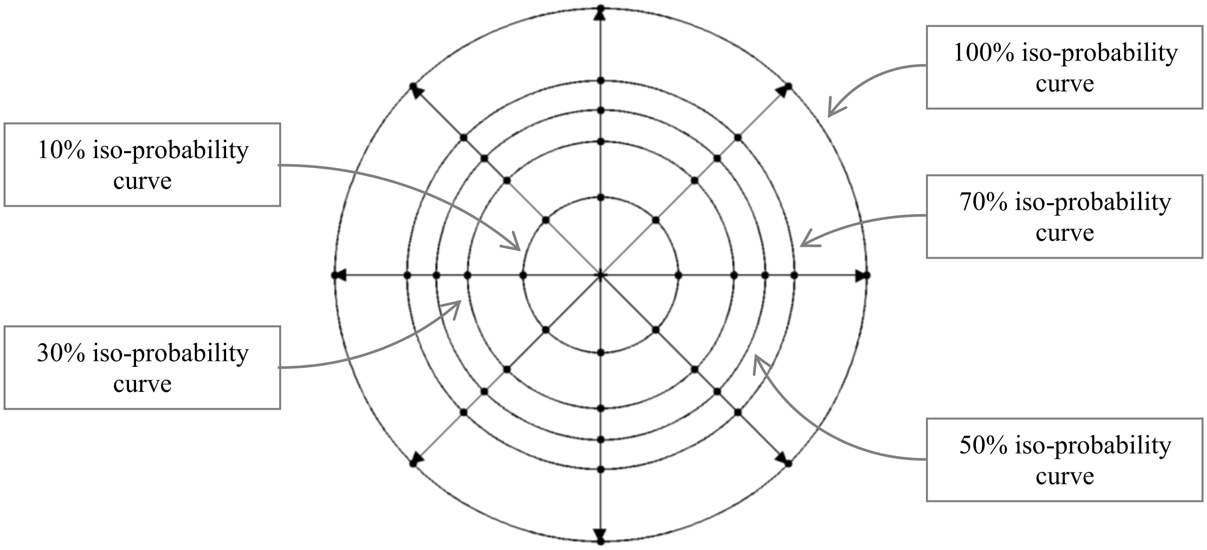

100% iso-probability curve
10% iso-probability curve
70% iso-probability curve
30% iso-probability curve
50% iso-probability curve

Figure 3.2. A set of iso-probability curves.

In the approach used in this Thesis, the interpolation that is conducted to construct each iso-probability curve (i.e., the process of taking the 'loci' of control points of the same cumulative probability among rays) is done through the use of piecewise cubic polynomials [129]. The term 'piecewise,' here, means that any given iso-probability curve is constructed of multiple 'pieces' (i.e., cubic polynomial segments). Thus, if N_{rays} rays are to be used to construct N_{curves} iso-probability curves, then for any given curve, $k \in [1, N_{curves}]$, a single cubic polynomial segment, $f_{k,i}$, $i \in [1, N_{rays}]$, would connect the control point on ray θ_i with the adjacent control point on ray $\theta_{(i+1)}$. Of course, since each iso-probability curve is to be a closed curve, for the last such segment, $i = N_{rays}$, the corresponding cubic polynomial, $f_{k,N_{rays}}$, would connect the control point on ray $\theta_{N_{rays}}$ to the adjacent control point back onto ray θ_1. In addition, it is evident that every such cubic polynomial, $f_{k,i}$, $i \in [1, N_{rays}]$, would be connected to two adjacent, neighbouring cubic polynomials, $f_{k,(i+1)}$ and $f_{k,(i-1)}$, at its end-points (i.e., at control points). The first derivative at each of those connection points should be the same for both the cubic polynomials that meet at that point. This is illustrated in Fig. 3.3. With these position and first-derivative continuity conditions specified at the end-points of each cubic polynomial segment, the resulting system of equations can be solved to obtain a continuous, 2D, closed curve, F_k, to represent the k^{th} iso-probability curve, $k \in [1, N_{curves}]$. Each F_k, then, is a piecewise cubic polynomial function composed of a set of single cubic polynomials:

$$F_k = \{f_{k,i}\}, i \in [1, N_{rays}], \forall k \in [1, N_{curves}]. \tag{3.6}$$

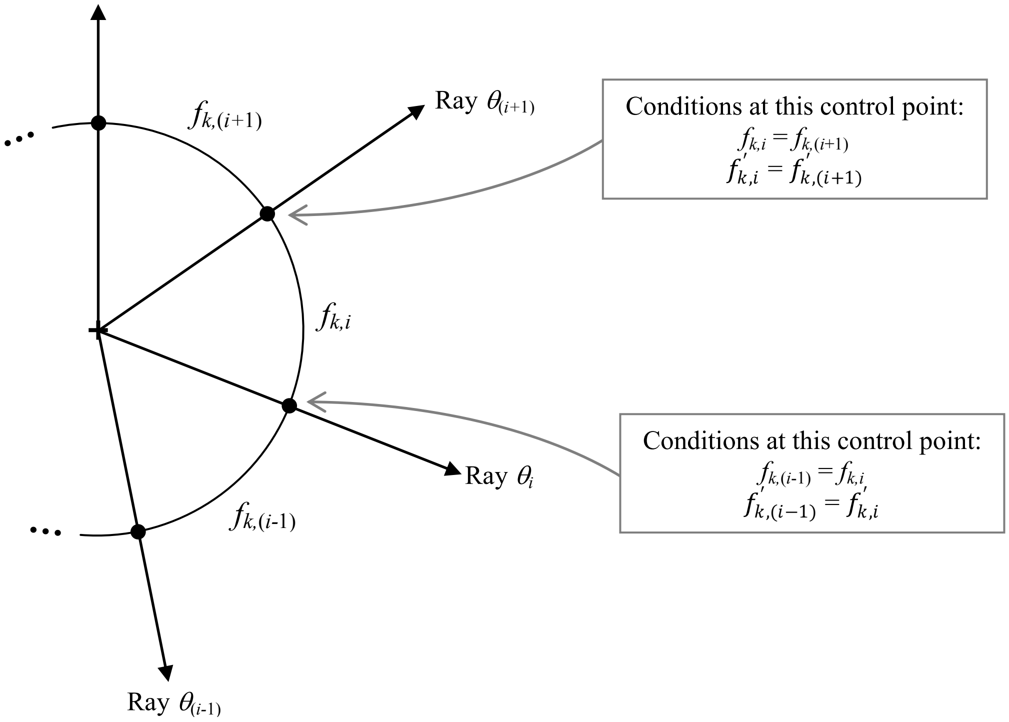


Figure 3.3. Piecewise cubic polynomials used to construct an iso-probability curve.

It is important to note, however, that the resulting set of iso-probability curves does not represent a 2D probability distribution for target presence over the search area. Therefore, the curves are not a 2D representation of the topology of a continuous target-location PDF over the search area, where the volume under that topology should be '1'. Instead, the curves are specifying probabilistic boundaries, where any given boundary, constructed through the loci of the P% cumulative-probability control points, bounds the region within which a target having the P% cumulative-probability mean speed can be expected to exist at time $t = T$. This way, there is a direct correspondence between the physical position along each ray, θ_i, and the probability density function value of $p(r, t, \theta_i)$ at that position, so that a point on a given ray corresponding to any cumulative probability specified can be directly computed whenever desired. As a result, the iso-probability curves maintain a direct physical correspondence with the boundaries of target presence, yet they also contain probabilistic target-location information by virtue of their positions relative to one another and their densities within different parts of the search area. In addition, since the conservative estimate of straight-line target-travel is used to establish the control points along any given ray, the resulting iso-probability curves delineate *conservative* boundaries on the presence of the target at time t, for any set of target-presence cumulative probability values desired. Using such a worst-case scenario estimate along each ray provides for reliable boundaries since the estimated outward travel of the target along any direction is maximized. Since one may not know *a priori* how the target may move, constructing probabilistic boundaries such as these is a reasonable strategy. By using this approach to target-location prediction, then, one does not need to be concerned with guessing the exact trajectory that the target is moving on at any given time, yet an effective search can be performed.

In Figures 3.1b and 3.2, eight uniformly distributed rays were used to construct the curves. Since the same target-location PDF is used for each ray, the common control points among the rays are all located the same distance away from the LKP on their respective rays. This produces circular nominal iso-probability curves. Interpolation between control points to construct the curves can, in fact, be conducted through any 2D curve-interpolation technique. However, in all the iso-probability curves that will be presented in this Thesis, the curves will be constructed using piecewise cubic polynomials as described above [129].

It should be noted that since this Thesis assumes a normal target-location PDF for illustrative purposes, the unbounded nature of the normal distribution requires that this PDF be truncated to have a finite 100% cumulative probability point, as well as a finite 0% cumulative probability point. As the normal distribution is being used here simply for illustration, the truncation points can be selected arbitrarily. In probability theory, the $\pm 3\sigma$ range is typically considered to cover effectively all the area under a normal distribution. Thus, in this Thesis, the 100% control point position is defined as the one corresponding to $r = (\mu_{r,t} + 3\sigma_{r,t})$. Similarly, at the lower end, the target-location PDF is truncated at $r = 0$, which corresponds to $r = (\mu_{r,t} - 3\sigma_{r,t})$ due to symmetry.

In the next three sections, methods are developed for incorporating other relevant influences into P_{MOT}, including the influences of terrain (which involves target physiology considerations), found clues, and target psychology, respectively. Following this, the issue regarding the accuracy in the construction of iso-probability curves is addressed, and, subsequently, the process of updating and propagating iso-probability curves is described.

3.2 Incorporating the Effects of Terrain

One of the important influences on the mean target-speed PDF that must be accounted for, as indicated in Eq. 2.2, is that of terrain. The nominal iso-probability curves shown in Fig. 3.1b and in Fig. 3.2 assume uniform terrain, yielding circular curves. Terrains in WiSAR scenarios may vary from rugged mountainous regions, to dense forests, and to smoother, rolling hills. The search terrain could also include non-traversable obstacles, such as large rocks or boulders, deep bodies of water, swamps, ravines, etc., which may be *a priori* known or unknown. The terrain topography would influence the target's motion, and this effect must be reflected in the prediction of the target's location. In the proposed approach, this requires modifying the iso-probability curves by incorporating terrain influence into the mean target-speed PDF (i.e., into P_{MOT}). Two factors are considered here: (*i*) difficulty of terrain, and (*ii*) *a priori* known obstacles. The former considers the impact of the difficulty in traversing large-scale land formations. The latter considers relatively smaller-scale obstructions, such as large rocks or boulders, which the target may navigate around.

3.2.1 Terrain Difficulty

In order to consider terrain difficulty in the determination of iso-probability curves, the approach proposed here is to scale the nominal mean target-speed PDF along each ray according to the instantaneous terrain surface slope and its potential impact on target motion. This impact, as discussed in Section 2.1.1, is dependent on the target's physiology. As with the construction of the nominal mean target-speed PDF, data on average human walking speeds for different inclines, categorized by age-groups, can be obtained from the literature in Biomechanics and other related fields of study [130]. Such data can be used to construct an empirical relationship between different inclines and their impact on average walking speed for different age-groups, thereby allowing for the determination of appropriate speed scale-factors for the purposes of the approach proposed here. This relationship will be referred to in this Thesis as the 'target physiological model.' Thus, the target physiological model and the data on which it is based are assumed to be part of the available target information, $\{I_{targ}(t)\}$. In Sections 3.2.1.1 and 3.2.1.2 that follow, the theoretical details as well as an illustrative example will be presented, respectively, for the proposed method of incorporating the effect of terrain difficulty into P_{MOT}.

3.2.1.1 Theoretical Details

The calculations for determining terrain difficulty begin with three basic assumptions: A topographic map of the area is available, the LKP of the lost person is known, and an appropriate target physiological model has also been derived. Given these inputs, the terrain is first discretized by overlaying a grid of cells onto the search area. Each cell is assigned its representative average terrain height value from the topographic map, yielding a height-map, $\mathcal{M}$. For the purposes of simulations conducted for this Thesis, different terrains were generated using the Terragen$^{\text{TM}}$ Classic scenery-generation software [131].

Next, in order to apply the abovementioned proposed approach of scaling the nominal mean target-speed PDF to account for terrain difficulty, it is necessary to scale the position of each control-point on each ray independently. Thus, the impact of terrain difficulty is incorporated into each nominal mean-target speed PDF along each ray separately, and, furthermore, each control-point on each ray is scaled individually. This is necessary because, clearly, each ray would be laid out over a different portion of terrain, and any potential target travel over those different portions would affect the target's motion differently. Therefore, the more control points that are defined (i.e., the more rays that are established and the more cumulative probability points selected along each ray), the more portions of the terrain will be 'sampled' in order to account for the impact of terrain difficulty when constructing the iso-probability curves.

On the iso-probability curves, each control point, $r_{T,P\%}$, on each ray corresponds to the position that would be reached by the target if he/she were to travel along that ray at a mean speed of $v_{P\%}$. However, the portion of terrain that the target would travel over influences the speed at which the target would move, and, thus, also the position where the target would end-up (i.e., the position of $r_{T,P\%}$ along the ray). Therefore, to account for the influence of terrain difficulty, it is the average *slope* of the terrain along each ray, from the LKP up to the position of the control point being addressed, that is of interest here. If this slope is known, then the target physiological model can be used to determine what the corresponding mean target speed would be in order to account for the terrain-difficulty for that control point. To compute this slope, a linear least-squares fit to ground-elevation measurements, over the portion of terrain from the LKP up to the position of the control point in question, is used.

Of course, the final position of this control point is not yet known, since this quantity is precisely what needs to be determined based on the impact of terrain difficulty. Therefore, an iterative process is needed to determine the exact location of each control point on each ray. Namely, although at a given time, one could calculate the position of a control point along a ray using the corresponding nominal mean target speed, the terrain traversed due to this distance travelled from the LKP to this position must be considered in order to scale the nominal mean speed to incorporate the terrain's influence on target motion. Scaling the mean speed would, in turn, change the distance travelled, and the terrain influence would have to be re-computed for this new travel-distance. This process must be iterated until the difference between two successive computed travel-distances is less than or equal to a user-specified maximum difference threshold value. Thus, for any given ray on which N_{curves} control points have been established, the following steps must be used to iteratively determine the impact of terrain difficulty on each control point along that ray, and to thereby establish the final position of each control point on that ray (i.e., the radial distance of each control point from the LKP along that ray).

Step 1: Estimate the radial distance, $d_{est,k}$, of the k^{th} control point, $k \in [1, N_{curves}]$, computed along the ray under consideration using the corresponding nominal mean target speed, and determine the coordinates of the control point, $P_{est,k} = (x_{est,k}, y_{est,k})$.

Step 2: Determine the height measurements (from the height-map) at regular distances along a straight-line representing the distance travelled by the target from the LKP up to the current estimated position of the control point along the ray. This should result in a set of data $\{h_i, r_i\}$, $i \in [1, N_h]$, of height measurements, h_i, and the corresponding radial distances, r_i, along the ray at which those height measurements are taken.

Step 3: Calculate the average slope of the portion of the terrain lying along the vector from the LKP up to the current estimated control point using the sample terrain height measurements obtained in Step 2.

This average terrain slope is obtained through a linear regression-line fit to the height data-points, using the slope, b_k, of the resultant line to compute an angular measure of the average terrain slope sought after:

$$\hat{h} = a_k + b_k r, \tag{3.7}$$

where $\hat{h}$ is the estimated terrain height value given by the fitted regression line; r is the radial distance at which the terrain height value estimate is evaluated; and, $\{a_k, b_k\}$ are the estimated values for the two regression coefficients, respectively.

The data used to compute the regression coefficient estimates are the given terrain height measurements, h_i, $i \in [1, N_h]$, taken at the radial distances, r_i, $i \in [1, N_h]$.

Once found, the slope value, b_k, is converted into an angular value, $\gamma_k = \tan^{-1}(b_k)$, which represents the estimate of the average angular slope over which the target could have travelled in the direction of the given ray up to the estimated position of the control point.

Step 4: Use the computed average terrain slope to determine the speed scale-factor corresponding to the control point being determined.

As noted earlier, determination of how much the mean speed of the target scales with different terrain slopes may be made by referring to relevant empirical data. Herein, for illustrative purposes, a linear relationship between average ground surface slope angle, γ, and speed scale-factor, q, is assumed. This relationship is obtained by stipulating a maximum incline angle, $\gamma_{max,inc}$, and a maximum decline angle, $\gamma_{max,dec}$, that the target can handle, as well as corresponding speed scale-factors, $q_{max,inc}$ and $q_{max,dec}$, respectively. In addition, an incline of $\gamma = 0°$ is given a scale-factor of $q = 1$. The linear relationship consists of two portions – one corresponding to negative slope angles (i.e., declines), and another corresponding to positive slope angles (i.e., inclines):

$$q(\gamma) = \begin{cases} q_{dec}(\gamma) = m_1\gamma + b_1 & \gamma_{max,dec} \leq \gamma < 0° \\ q_{inc}(\gamma) = m_2\gamma + b_2 & 0° \leq \gamma \leq \gamma_{max,inc} \end{cases}, \tag{3.8}$$

where m_1 and m_2 are parameter constants of the two portions of the linear relationship, respectively. The overall relationship given by Eq. 3.8, then, represents the target physiological model.

If the computed slope exceeds the stipulated bounds, the corresponding terrain is considered to be non-traversable by the target. The iterative process, thus, ceases and the control point being computed remains at its current position, $P_{est,k}$.

Step 5: Scale the corresponding nominal speed using the computed speed scale-factor, q_k, obtained from Eq. 3.8 for the angular slope value, γ_k. Compute a new estimate of the radial distance, $d_{est\_NEW,k}$, of the k^{th} control point, using this scaled speed. With speed having changed, the corresponding radial distance estimate of the control point must be changed as well. This relocates the control point along the ray and represents an improved estimate.

Step 6: Compare the updated control point radial distance estimate to the prior one used at the start of the current iteration in the following manner to determine whether another iteration is necessary:

$$100\% \times \left(\frac{\left| d_{est\_NEW,k} - d_{est,k} \right|}{d_{est,k}} \right) \leq \Delta d_{max}\,. \tag{3.9}$$

If the above percentage difference is greater than the threshold, Δd_{max}, another iteration is deemed necessary. The new radial distance estimate just obtained is set as the current radial distance estimate (i.e., set $d_{est,k} = d_{est\_NEW,k}$), and the process returns to Step 2. Otherwise, the iterative process is stopped, and the current estimate for the radial distance, $d_{est\_NEW,k}$, is accepted. The threshold value, Δd_{max}, is selected by the user and specified at the beginning of the search. Its setting depends on how accurately the user wishes to account for the terrain as well as on the available computational resources; a smaller threshold value means greater accuracy but more computational time, while a larger threshold value would compromise accuracy but decrease computational time. The value of Δd_{max} could also be modified on-line if lesser or greater accuracy is later found to be desired.

Using the above procedure, then, every control point on the target-location PDF on each ray can be shifted to the proper position that appropriately represents the impact of terrain difficulty. It should be noted that through this procedure, every control point on the target-location PDF is addressed individually, with the scaling of the mean target-speed PDF happening ‘internally’ through intermediate steps (namely, steps 4 and 5). Thus, in this approach, the target-location PDF is addressed directly, without having to go back to the nominal mean target-speed PDF, modify this entire distribution, and then re-derive the corresponding target-location PDF through an appropriate transformation of variable. Also, since each control-point position is modified individually on each ray, the resulting iso-probability curves will, in general, be irregular in shape. This will be seen in the example that follows.

3.2.1.2 An Illustrative Example

The approach described above for incorporating the effect of terrain difficulty into the target-location PDF is illustrated through the following example: A hiker is lost in a rugged region that does not allow for aerial surveying. An isometric view of the entire terrain is shown in Fig. 3.4. The initial search region considered encompasses a 3000 × 3000 m area. The LKP of the lost person is located at the centre of this area. A coordinate system is also established with its origin at the LKP, so that LKP = (0 m, 0 m). The terrain is discretized by overlaying a 513 × 513 grid array. Each cell is assigned the average height of the terrain enclosed within the cell, yielding a height-map, $\mathcal{M}$. Overlaying such a grid on a 3000 × 3000 m terrain means that the width of each cell is approximately 3000 m / 513 cells = 5.85 m. Given these settings, the LKP, thus, lies in row-cell #256 and column-cell #256. The mean target-speed PDF is a bounded normal distribution. Ten nominal iso-probability curves are constructed for the elapsed search time of t = 300 s using 36 uniformly-distributed rays (i.e., the inter-ray spacing is 10°).

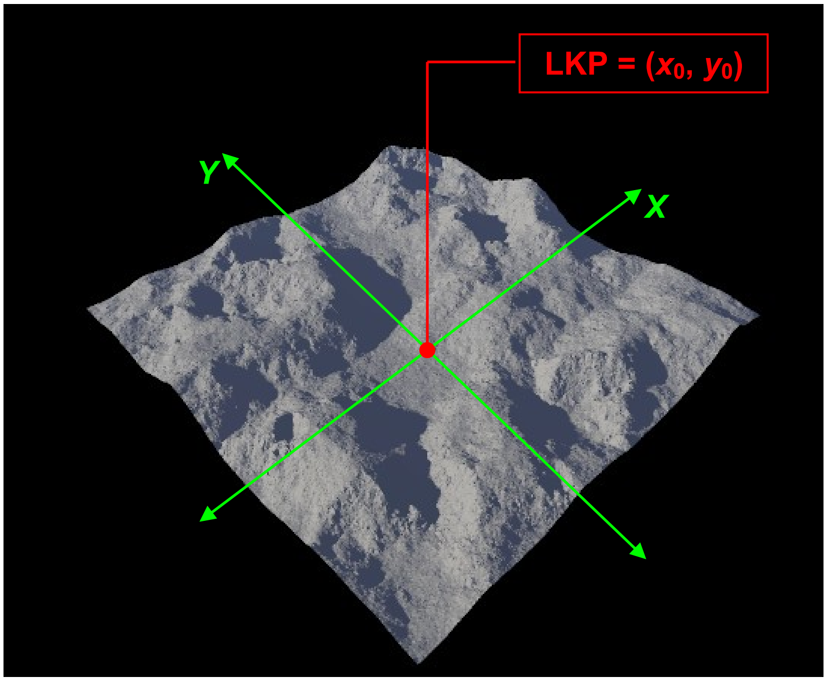


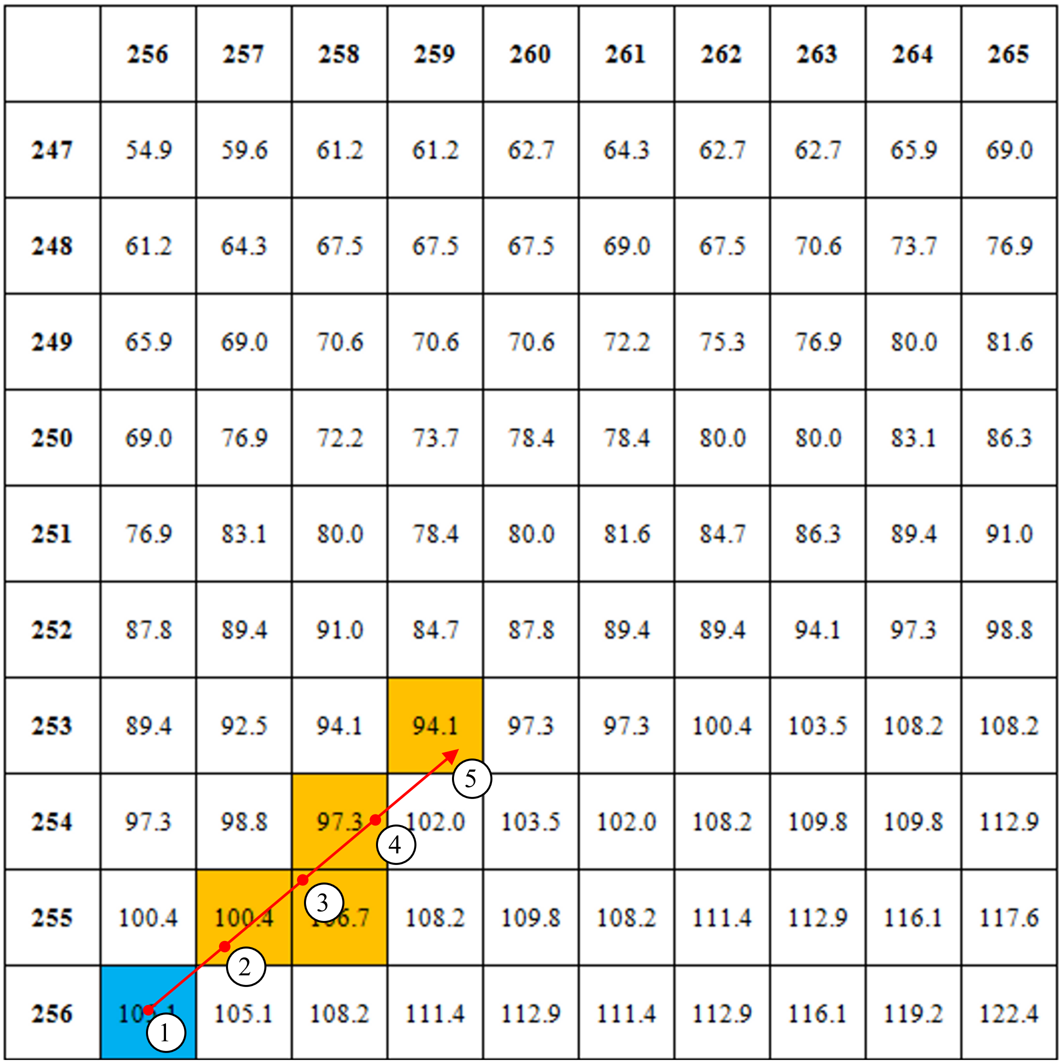
Figure 3.4. An example terrain topography.

To illustrate the iterative process used to obtain the correct position of a given control point, a sample set of calculations will be shown for the computation of the control point, $r_{300\mathrm{s},10\%}$, corresponding to the 10% iso-probability curve, for the $\theta_i = 40°$ ray. The nominal mean target-speed associated with the 10% control point, given the bounded normal distribution used for the mean target-speed PDF, is taken to be $v_{10\%,nom} = 0.080$ m/s. Figure 3.5 shows a 10×10 portion of the height-map that will be relevant to the sample computations. The first column and first row of the matrix shown in this figure give, respectively, the row-cell and column-cell numbers for this portion of the height-map grid array. All other entries are the average terrain height values in metres. The cell corresponding to the LKP, and also to the origin of the coordinate system used, is highlighted in blue.

Figure 3.6 shows a vector (red arrow) originating from the LKP up to the initial position estimate for the control point, overlaid onto the height-map. It is assumed that height measurements are taken at increments equal to the cell width along this straight line of travel (i.e., every 5.85 m). Rounding down to the next lowest integer value, this produces $N_h = 5$ height values to be obtained from the terrain height-map along this direction for the given initial length of this vector. Figure 3.6 shows the positions of these sample height points (red dots, including the arrow head) on the direction vector in question. The height value assigned to each point is the terrain height value of the cell within which the point lies. The height values and the radial distances relative to the LKP of each point are summarized in Table 3.1. The variable, r_i, represents the radial distance value of the i^{th} sample point, while h_i represents the terrain height value assigned to it. Table 3.1 also gives the cell coordinates corresponding to each sample point, and these cells are highlighted in orange in Fig. 3.6. Table 3.2 summarizes the computations for the first iteration of the proposed procedure to determine the position of control point $r_{300\mathrm{s},10\%}$ corresponding to the 10% iso-probability curve for the 40° ray.

	256	**257**	**258**	**259**	**260**	**261**	**262**	**263**	**264**	**265**
247	54.9	59.6	61.2	61.2	62.7	64.3	62.7	62.7	65.9	69.0
248	61.2	64.3	67.5	67.5	67.5	69.0	67.5	70.6	73.7	76.9
249	65.9	69.0	70.6	70.6	70.6	72.2	75.3	76.9	80.0	81.6
250	69.0	76.9	72.2	73.7	78.4	78.4	80.0	80.0	83.1	86.3
251	76.9	83.1	80.0	78.4	80.0	81.6	84.7	86.3	89.4	91.0
252	87.8	89.4	91.0	84.7	87.8	89.4	89.4	94.1	97.3	98.8
253	89.4	92.5	94.1	94.1	97.3	97.3	100.4	103.5	108.2	108.2
254	97.3	98.8	97.3	102.0	103.5	102.0	108.2	109.8	109.8	112.9
255	100.4	100.4	106.7	108.2	109.8	108.2	111.4	112.9	116.1	117.6
256	105.1	105.1	108.2	111.4	112.9	111.4	112.9	116.1	119.2	122.4

Figure 3.5. Portion of height-map for sample calculations.



	256	257	258	259	260	261	262	263	264	265
247	54.9	59.6	61.2	61.2	62.7	64.3	62.7	62.7	65.9	69.0
248	61.2	64.3	67.5	67.5	67.5	69.0	67.5	70.6	73.7	76.9
249	65.9	69.0	70.6	70.6	70.6	72.2	75.3	76.9	80.0	81.6
250	69.0	76.9	72.2	73.7	78.4	78.4	80.0	80.0	83.1	86.3
251	76.9	83.1	80.0	78.4	80.0	81.6	84.7	86.3	89.4	91.0
252	87.8	89.4	91.0	84.7	87.8	89.4	89.4	94.1	97.3	98.8
253	89.4	92.5	94.1	94.1	97.3	97.3	100.4	103.5	108.2	108.2
254	97.3	98.8	97.3	102.0	103.5	102.0	108.2	109.8	109.8	112.9
255	100.4	100.4	106.7	108.2	109.8	108.2	111.4	112.9	116.1	117.6
256	105.1	105.1	108.2	111.4	112.9	111.4	112.9	116.1	119.2	122.4

Figure 3.6. Vector to the control point for the 10% mean target-speed on the 40° ray, and corresponding terrain height data (t = 300 s).

Table 3.1. Terrain height sample data for initial control point position estimate.

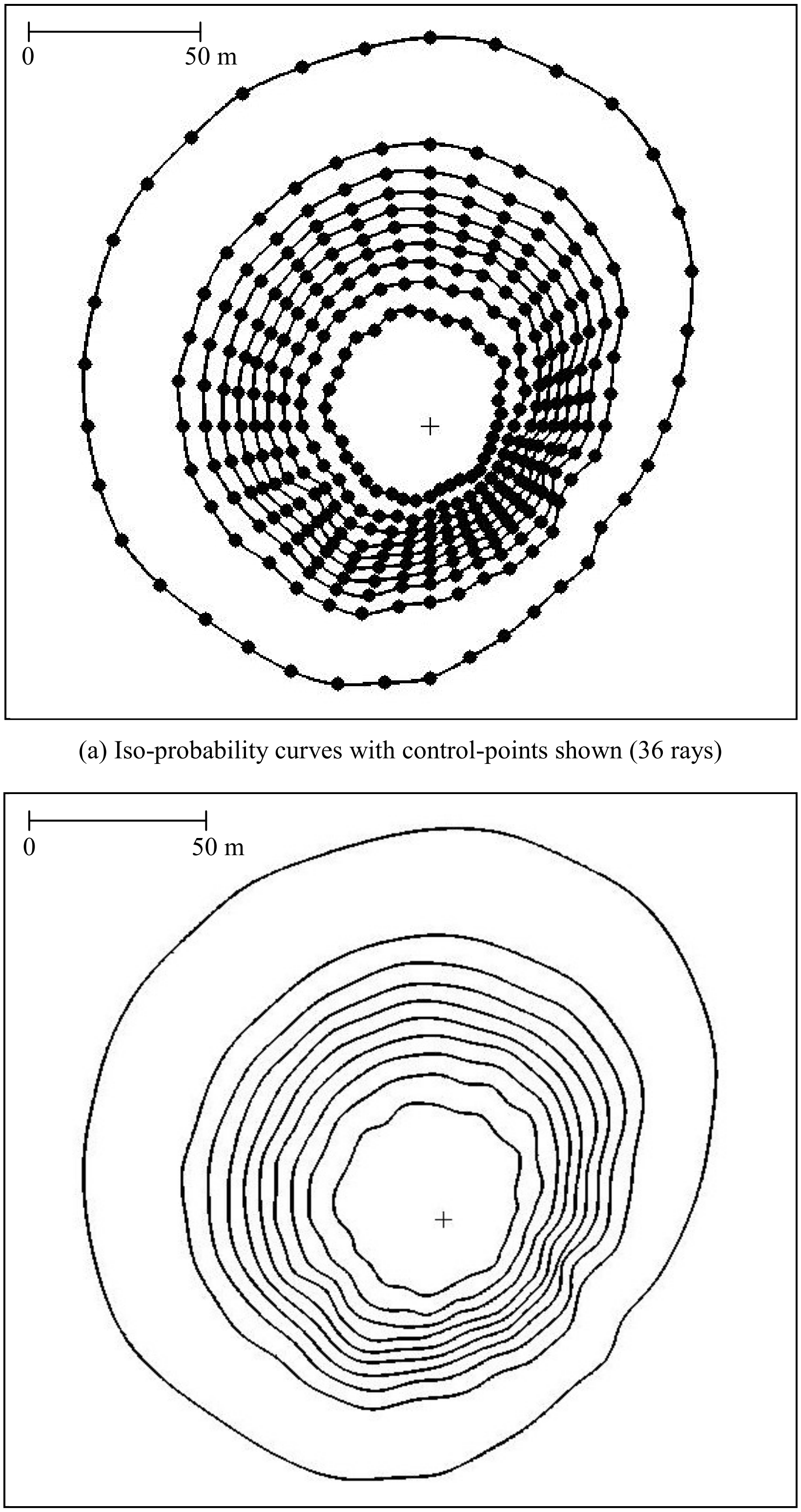
i	r_i [m]	h_i [m]	Row-cell #	Column-cell #
1	0.00	105.10	256	256
2	5.85	100.39	255	257
3	11.70	106.67	255	258
4	17.54	97.25	254	258
5	23.39	94.12	253	259

Table 3.2. Computations for Iteration 1 of the illustrative example.

Step #	Results
1	$d_{est,k} = (v_{10\%,nom}) \times (t) = (0.080 \text{ m/s}) \times (300 \text{ s}) = 24.00 \text{ m}$ $x_{est,k} = (24.00 \text{ m}) \times (\cos 40°) = 18.39 \text{ m}$ $y_{est,k} = (24.00 \text{ m}) \times (\sin 40°) = 15.43 \text{ m}$. The corresponding cell coordinates are [253 259].
2	Sample height data points from height-map (see Figure 3.6): $h_i = \{105.1 \text{ m}, 100.4 \text{ m}, 106.7 \text{ m}, 97.3 \text{ m}, 94.1 \text{ m}\}$
3	From linear least-squares regression-line fit to sample points: $b_k = -0.43$ $\gamma_k = \tan^{-1}(b_k) = -23.27°$
4	Equation 3.8 parameters used: $\gamma_{max,inc} = 60°$ $\gamma_{max,dec} = -60°$ $q_{max,inc} = 0.5$ $q_{max,dec} = 1.5$ $\therefore\ q_k = 1.19$
5	$v_{10\%,scaled} = v_{10\%,nom} \times q_k = (0.080 \text{ m/s}) \times (1.19) = 0.095 \text{ m/s}$ $d_{est\ NEW,k} = v_{10\%,scaled} \times t = (0.095 \text{ m/s}) \times (300 \text{ s}) = 28.50 \text{ m}$
6	Using $\Delta d_{max} = 1\%$: $\Delta d_k = 100\% \times (\mid d_{est\_NEW,k} - d_{est,k} \mid / d_{est,k})$ $= 100\% \times (\mid 28.50 - 24.00 \mid / 24.00)$ $= 18.75\% > \Delta d_{max}$ $\therefore$ since $\Delta d_k > \Delta d_{max}$, another iteration is necessary.

In Step 6, a maximum percentage difference threshold value of $\Delta d_{max} = 1\%$ was employed. Since the percentage difference between $d_{est,k} = 24.00$ m and $d_{est\_NEW,k} = 28.50$ m exceeded the 1% threshold, another iteration must be conducted. For the second iteration: $d_{est,k} = d_{est\_NEW,k} = 28.50$ m; $P_{est,k} = (21.8 \text{ m}, 18.3 \text{ m})$; b_k and γ_k remain –0.43 and –23.27°, respectively, so that q_k and $v_{10\%,scaled}$ also remain 1.19 and 0.095 m/s, respectively; thus, $d_{est\_NEW,k} = d_{est,k} = 28.50$ m. Since the percentage difference between $d_{est\_NEW,k}$ and $d_{est,k}$ is less than the 1% threshold value, the stopping condition in Step 6 is satisfied and no further iterations are required after iteration 2. The final position of control point $r_{300\text{s},10\%}$ on the 40° ray for time $t = 300$s is taken to be 28.50 m from the LKP along this ray.

Carrying out the above iterative procedure individually for all ten control points on each of the 36 rays produces the final set of modified control points. Connecting similar control points with piecewise cubic polynomials, then, forms the iso-probability curves. Figure 3.7 illustrates the resulting set of iso-probability curves for the $t = 300$ s time point. In subsequent control-point computations, when the iso-probability curves need to be updated, only the additional distance that would be traversed by each control point, between its previous and new positions, needs to be considered when computing the terrain slopes and scale-factors in Steps 1 to 4 above. Thus, the LKP would be replaced with the 'current control point position' in these steps during subsequent iso-probability curve propagations.



(a) Iso-probability curves with control-points shown (36 rays)

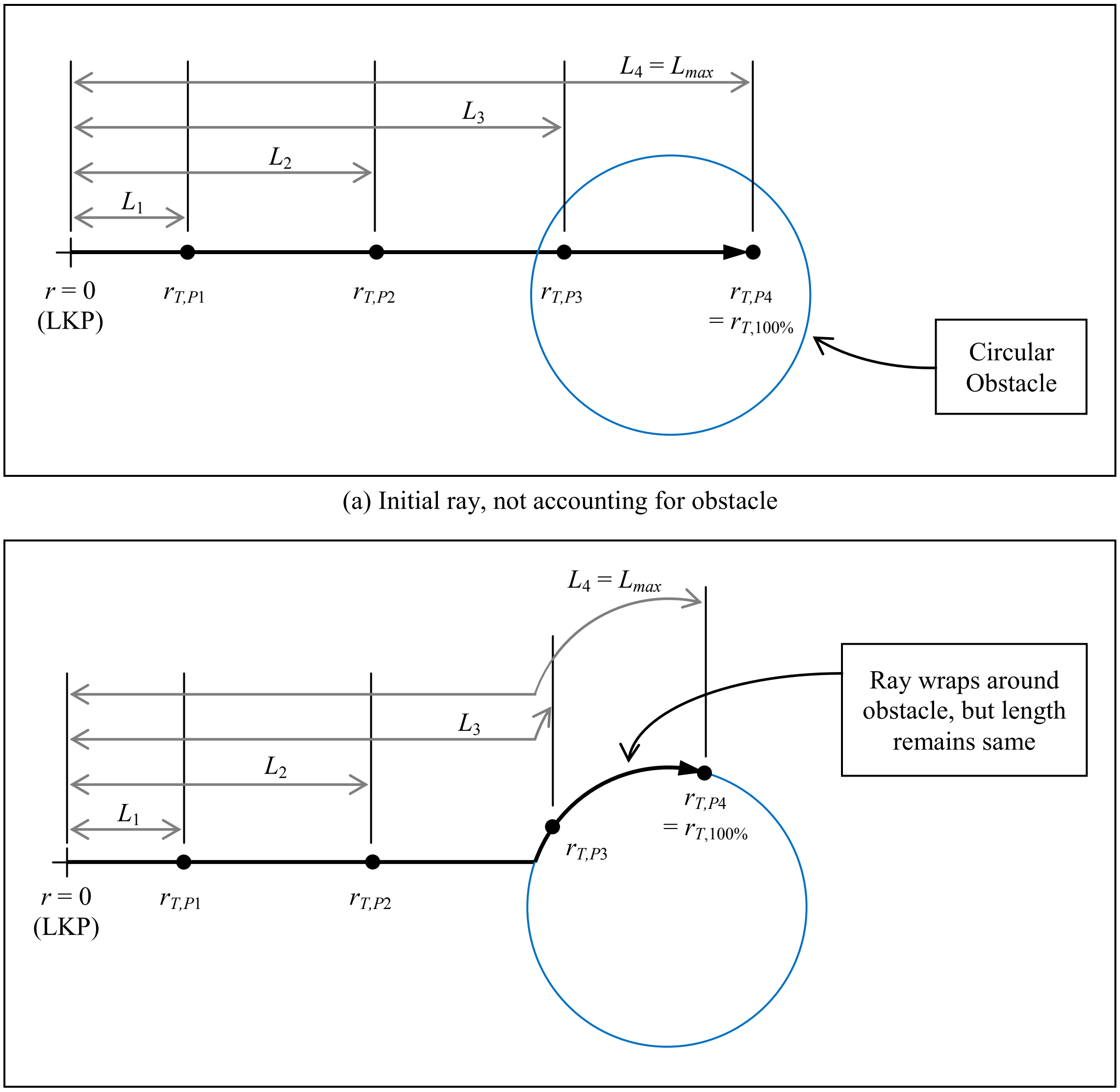
(b) Iso-probability curves without control-points

Figure 3.7. Iso-probability curves for illustrative example, shown for *t* = 300 s.

3.2.2 Known Obstacles

The second factor to consider with respect to determining the influence of terrain on the mean target-speed PDF (and, thus, on the target-location PDF) along any given ray, is *a priori* known obstacles. The effect of *a priori* known obstacles is reflected through changes in the shapes of the rays. The proposed approach to account for the influence of obstacles is to 'wrap' any ray that intersects an obstacle around that obstacle's boundary, favouring the side leading to the shortest path around the obstacle as a conservative estimate. Neither the original path-length of the ray nor the positions of the control points along the length of the ray are altered (i.e., the ray is not stretched). This approach is illustrated in Fig. 3.8, where a single ray emanating from the LKP, with four control points defined along it, is shown intersecting a circular obstacle.

In Fig. 3.8a, the initial ray is shown, which does not yet take the obstacle into account. In Fig. 3.8b, the ray is shown 'wrapped' around the boundary of the obstacle as per the proposed approach, with some of the control points now lying on the obstacle boundary as well. Let the variable, $L_k \in [0, L_{max}]$, $k \in [1, 4]$, represent the length (i.e., travel-distance) along the ray, starting from $r = 0$ at the LKP and going up to the k^{th} control point at $r = r_{T,Pk\%}$ ($L_4 = L_{max}$, which corresponds to the $r_{T,100\%}$ control point). Then, the total length, L_{max}, of the original, straight ray, and the length-values, L_k, corresponding to the positions of each of the control points, all remain the same before and after wrapping the ray. As can be seen in Fig. 3.8b, even though control points $r_{T,P3}$ and $r_{T,P4}$ have been moved up to the boundary of the obstacle, they have been placed on the obstacle in such a way that their corresponding distances, L_3 and L_4, respectively, as measured along the length of the ray (thick black line), are still the same as those in Fig. 3.8a.



(a) Initial ray, not accounting for obstacle

(b) Ray ‘wrapped’ around obstacle boundary

Figure 3.8. Wrapping rays around obstacle boundaries.

By wrapping the rays and having them trace the obstacle boundaries, the outward progress of the corresponding target-location PDF is slowed down, mimicking the slowing down of the outward progress of the target if he/she were to encounter and attempt to bypass the obstacle. Figure 3.9 shows a set of ten iso-probability curves, illustrating the effect that this approach to addressing *a priori* known obstacles has on the curves for circular obstacles.

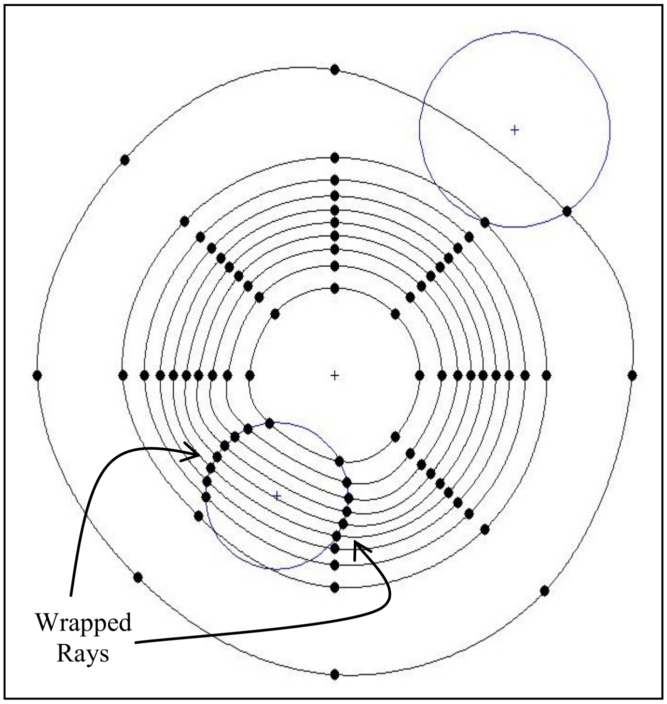


Figure 3.9. Iso-probability curves modified to reflect effect of obstacles.

It is important to note that only the positions of the control points defined are changed to accommodate the *a priori* known obstacles. The interpolated curve segments between the control points are not influenced by the obstacles directly. As a result, these curve segments may pass through the obstacles, as can be seen in the example in Figure 3.9. However, such occurrences are not to be construed as a representation problem, since it is the rays, and the control points defined along them, that explicitly model the impact of the different influences on the target-location PDF. As will be explained further in Chapter 5, the iso-probability curve segments provide guidance for robot search-movements, but only do so in those regions unobstructed by obstacles. When iso-probability curves intersect obstacles, which, in turn, obstruct the planned path of the robots, the robots in those situations would inevitably need to engage in some type of obstacle-avoidance manoeuvre until the obstacle is circumvented.

It should also be noted that the influence of obstacles has, yet again, been applied directly to the control points, thereby affecting the target-location PDFs along those rays that intersect obstacles. There was no need to return to the P_{MOT}, modify it first, and to then apply a transformation to obtain the modified P_{POS}. This approach, therefore, saves computation time and helps to make the process of constructing iso-probability curves on-line-feasible.

3.3 Incorporating the Effects of Clues Found

An evolving source of information critical to the search process is clues left behind by the target. Examples include articles of clothing belonging to the target, the time and location of a target-sighting by an eyewitness, etc. Clues can be abstracted into two general types: those that only give positional information about the target, and those that give both a confirmed position and the time at which the target was at that position since the start of the search.

The proposed approach for addressing the effect of clues on target-location prediction is to take the coordinates of the newly-found clue to be the new LKP. The iso-probability curves are, then, reconstructed relative to this new LKP, based on the elapsed time since the target had dropped the clue. For clues that only indicate position, a conservative speed and path estimate is made to estimate the time at which the clue was dropped, and to thereby estimate the additional time for which the target would have been moving since dropping the clue.

In particular, if at some time $t = T_{clu\_fnd}$, a clue is found without any information on its drop-time, $t = T_{clu\_drp}$, an estimate, $\hat{T}_{clu\_drp}$, is calculated, as illustrated in Figure 3.10. To compute this estimate, first, it is assumed that the target has moved on a straight-line path from the first LKP position, (x_0, y_0), to the newly-found clue position, (x_{clu}, y_{clu}). This constitutes a conservative path-estimate, yielding a travel-distance of:

$$d_{est} = |(x_{clu}, y_{clu}) - (x_0, y_0)|. \tag{3.10}$$

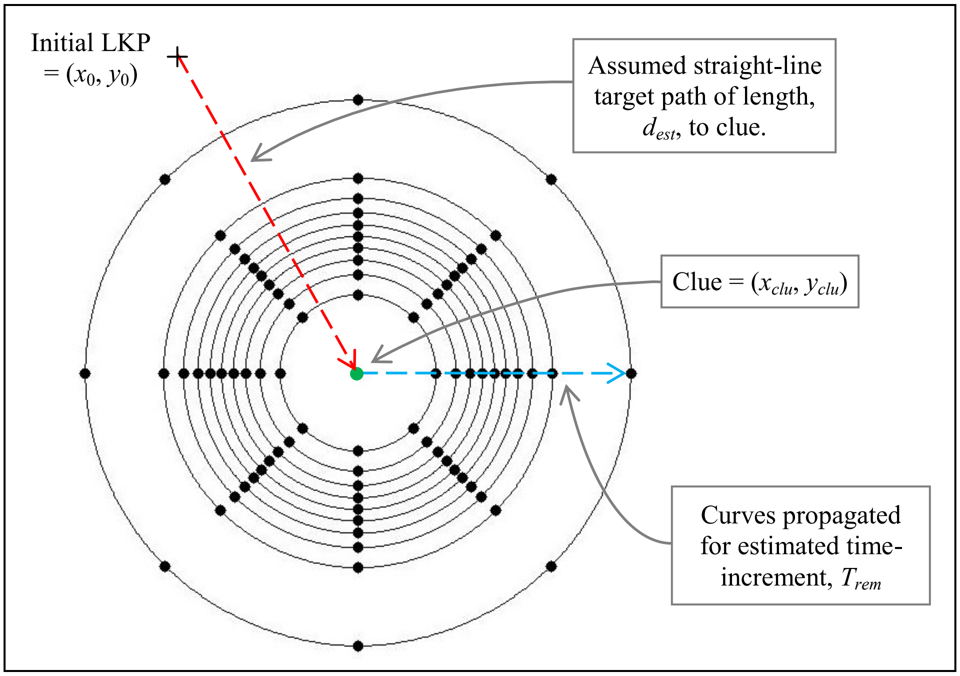

Initial LKP
= (x_0, y_0)
Assumed straight-line target path of length, d_{est}, to clue.
Clue = (x_{clu}, y_{clu})
Curves propagated for estimated time-increment, T_{rem}

Figure 3.10. Relocation and reconstruction of iso-probability curves when a clue is found.

Next, a conservative target-speed estimate is made by assuming that he/she moved along this path at the maximum possible mean speed, v_{max}, as defined by the upper bound of the nominal mean target-speed PDF (i.e., $v_{max} = v_{100\%}$). Consequently, the iso-probability curves are reconstructed for the current time, $t = T_{clu\_fnd}$, where the position of the new LKP becomes (x_{clu}, y_{clu}). The iso-probability curves propagate out from this point a distance corresponding to a time increment of $t = T_{rem}$, where:

$$T_{rem} = T_{clu\_fnd} - \hat{T}_{clu\_drp}. \tag{3.11}$$

Of course, if the time, T_{clu_drp}, at which the clue was 'dropped' is known, then estimating this time is not required and $\hat{T}_{clu\_drp}$ in Eq. 3.11 would simply be replaced by the known T_{clu_drp}. More details on how iso-probability curves are propagated out, in general, from any given LKP, will be given in Section 3.6.

Using this approach to address clues, it is evident that the iso-probability curves stretch out further for clues found closer to the LKP (since, for such clues, $\hat{T}_{clu\_drp}$ would be smaller, making T_{rem} larger), as opposed to clues found further out. This corresponds to what one would expect intuitively, since a clue found closer to the LKP implies that the target left it early on in the search, giving him/her more time to travel since the clue-drop, and thereby resulting in greater uncertainty as to his/her current location. Conversely, a clue found further from the LKP would imply that the target left it more recently, affording less time for him/her to travel since the clue-drop, and thereby resulting in relatively lesser uncertainty in his/her current location. Moreover, the use of the conservative estimate that the target travelled to the position of the clue in a straight-line path at his/her maximum nominal speed minimizes the estimated time taken by the target to reach the clue position, and, thus, maximizes the estimated travel-time of the target since having dropped the clue. This maximization of the latter travel-time maximizes the outward propagation of the iso-probability curves from the new LKP (i.e., from the clue position) when the curves are reconstructed, which ensures that the target is still contained within the bounds of the 100% iso-probability curve. This accounts for the inability to know the actual order in which clues without time-information are dropped.

3.4 Incorporating the Effects of a Psychology

Lost-person psychology information, based on experiences from past lost-person WiSAR search incidents and organized by relevant target-type categories, can be found in the literature, usually in publications from SAR organizations [11, 119, 132]. The difficulty with addressing target psychology during an autonomous robotic WiSAR search is that the pertinent information is highly qualitative in nature, typically describing behavioural tendencies. Examples include such tendencies as shelter-seeking behaviour (where the target will passively or actively look for any structure that they feel would serve as a sufficient shelter) or route-travelling behaviour (sometimes called trail-running behaviour, where the target will tend to follow things like hiking trails, pipelines, railroad tracks, etc. in hopes that it will lead him/her out of the wilderness). Such qualitative descriptions make it difficult to incorporate target psychology influences into the iso-probability curves through modification of the mean target-speed PDF or the target-location PDF.

Nevertheless, there are certain types of psychological behaviours whose influences can be incorporated into the proposed approach for target-location prediction. One important class of behaviour is that of moving in a possible preferred direction of travel. Depending on knowledge of the target and his/her purpose in a given wilderness environment, it may be possible to identify such a preferred direction of target travel based on an assumed probable destination. For example, it can be assumed that a lost tourist may be aware of the existence and general direction of a popular destination (such as a well-known village or chalet) where help/refuge could be obtained. Hikers and walkers, even those who are experienced outdoorsmen, will often display view enhancing behaviour, where they tend to travel in the direction of higher elevation so that they can attain a position of height that allows them to get a better view of the area in order to identify familiar landmarks in the distance. Thus, if this type of psychological behaviour can be identified as being applicable to the target being sought, then it may allow for the determination of a possible intended destination for the lost-person, which, in turn would make a certain direction of target motion more likely than others.

In the proposed approach to target-location prediction, the influence of such a travel direction can be addressed via the application of an appropriate scale-factor to the mean target-speed PDF of a ray that is specifically chosen to align with the probable destination point. Namely, the mean target-speed PDF along the ray in this direction would receive a higher scaling factor relative to other directions. Search commanders in real-life WiSAR scenarios often use their past experience and knowledge of the target at hand to assign relative likelihoods to different regions of the search area [63, 117, 118]. Thus, a consensus among multiple such experienced personnel may be used to establish the relative likelihoods of different directions at the start of the search, and, in turn, to obtain the scale-factors.

As an example, let us consider a linear relationship between directional likelihood and scale-factor. In particular, two contrasting directions can be identified and assigned a relative likelihood: a 'low' and a 'high'. The high-likelihood direction, θ_{hi}, can be designated to be Q_H times more likely than the low-likelihood direction, θ_{lo}. Thus, a probability distribution, having a linear probability model, can be established for the possible angular directions of travel by the target. This linear function would give the probability density value, $p(\theta)$, for any specified ray direction angle, θ, and would range over all the possible directions, from −180° to +180°.

Let us suppose, then, that the 0° direction was considered to be the high-probability direction, while the opposite direction, ±180°, was taken to be the low-probability direction (i.e., $\theta_{hi} = 0°$ and $\theta_{lo} = \pm 180°$). Furthermore, let us assume that the relative likelihood is such that $Q_H = 3$ (i.e., the high-likelihood direction is three times more likely than the low-likelihood direction for target travel). A corresponding linear probability model can be obtained as shown in Fig. 3.11. Solving for the value of the parameter, a, in this model would specify the distribution completely. This can be done as follows. Given that this is a probability density function, the area, A_{total}, under the curve must equal '1'. This gives:

$$A_{total} = 2(A_1 + A_2) = 1, \tag{3.12}$$

where A_1 and A_2 are as shown in Fig. 3.11.

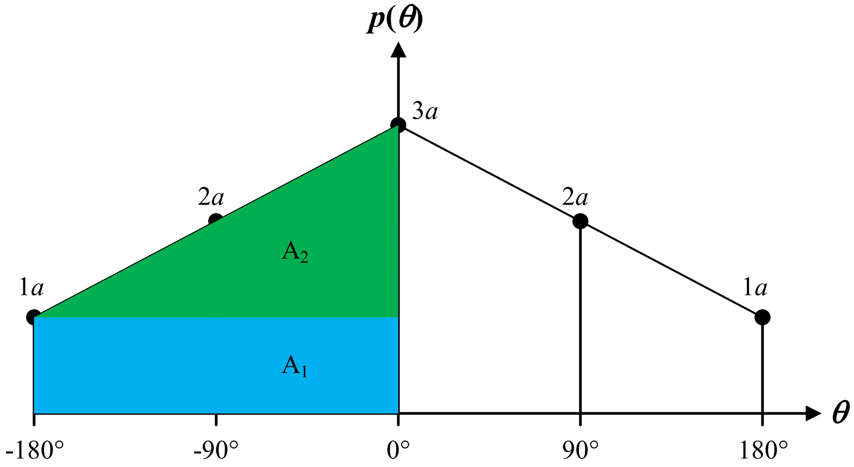


Figure 3.11. A linear probability distribution for target travel-direction likelihood.

Also, it is known that:

$$A_1 = (1a)(180° - 0°) = 180a. \tag{3.13}$$

$$A_2 = \left(\frac{1}{2}\right)(3a - 1a)(180° - 0°) = 180a. \tag{3.14}$$

Therefore:

$$A_{\text{total}} = 2(A_1 + A_2) = 2(180\text{a} + 180\text{a}) = 1. \tag{3.15}$$

Finally, solving Eq. 3.15 for a gives:

$$a = \frac{1}{720}. \tag{3.16}$$

An equation for the probability density value, $p(\theta)$, as a function of direction angle, θ, can, now, be derived as follows. For the direction range ($-180° \leq \theta \leq 0°$), the linear probability density function can be expressed as:

$$p_1(\theta) = m_1\theta + b_1. \tag{3.17}$$

From Fig. 3.11:

$$m_1 = \frac{(3a - 1a)}{\left(0° - (-180°)\right)} = \frac{a}{90} = \frac{1/720}{90} = \frac{1}{64800}. \tag{3.18}$$

Also, given that at $\theta = 0°$, $p_1(0°) = 3a$, one can use this information along with Eq. 3.16 to solve for b_1, which gives:

$$b_1 = 3a = \frac{3}{720} = \frac{1}{240}. \tag{3.19}$$

Thus, with $m_1 = 1/64800$ and $b_1 = 1/240$, the final equation for the linear probability density function for likelihood of target travel-direction in the range ($-180° \leq \theta \leq 0°$) becomes:

$$p_1(\theta) = \frac{1}{64800}\theta + \frac{1}{240}. \tag{3.20}$$

Similarly, the equation for this linear probability density function for the direction range ($0° \leq \theta \leq 180°$) is found to be:

$$p_2(\theta) = \frac{-1}{64800}\theta + \frac{1}{240}. \tag{3.21}$$

Thus, one can write:

$$p(\theta) = \begin{cases} p_1(\theta) = \dfrac{1}{64800}\theta + \dfrac{1}{240}, & -180° \leq \theta < 0° \\ p_2(\theta) = \dfrac{-1}{64800}\theta + \dfrac{1}{240}, & 0° \leq \theta \leq 180° \end{cases}. \tag{3.22}$$

Given the derived function, $p(\theta)$, the scaling factor, $q(\theta_i)$, for any ray, θ_i, $i \in [1, N_{rays}]$, for the N_{rays} rays that have been selected to construct the iso-probability curves, can be obtained by taking the probability density ratio relative to that of the high-likelihood direction:

$$q_{dirPref}(\theta_i) = p(\theta_i) \,/\, p(\theta_{hi}). \tag{3.23}$$

The scale factor, $q_{dirPref}(\theta_i)$, is applied to the nominal mean target-speed PDF of ray, θ_i. Thus, all the nominal speeds corresponding to the control points that one wishes to use are multiplied by this scale factor. Scaling the mean target-speed PDF this way, in turn, changes the radial distances of each of the corresponding control points on the target-location PDF accordingly. Continuing with the example above, Fig. 3.12 shows the resulting iso-probability curves for this scenario, using 8 uniformly distributed rays, for the 1800 s time-point.

By applying this approach and scaling the target-location PDF along each of the rays, the net effect is that the iso-probability curves proportionally ‘shrink’ away from the lower-likelihood directions. With the iso-probability curves guiding how the robots move (as mentioned in the robot-path planning method overview in Section 2.2.3, and which will be detailed in Chapter 5), this modification of the curves will cause the search robots to spend more time searching the regions closer to the high-likelihood direction.

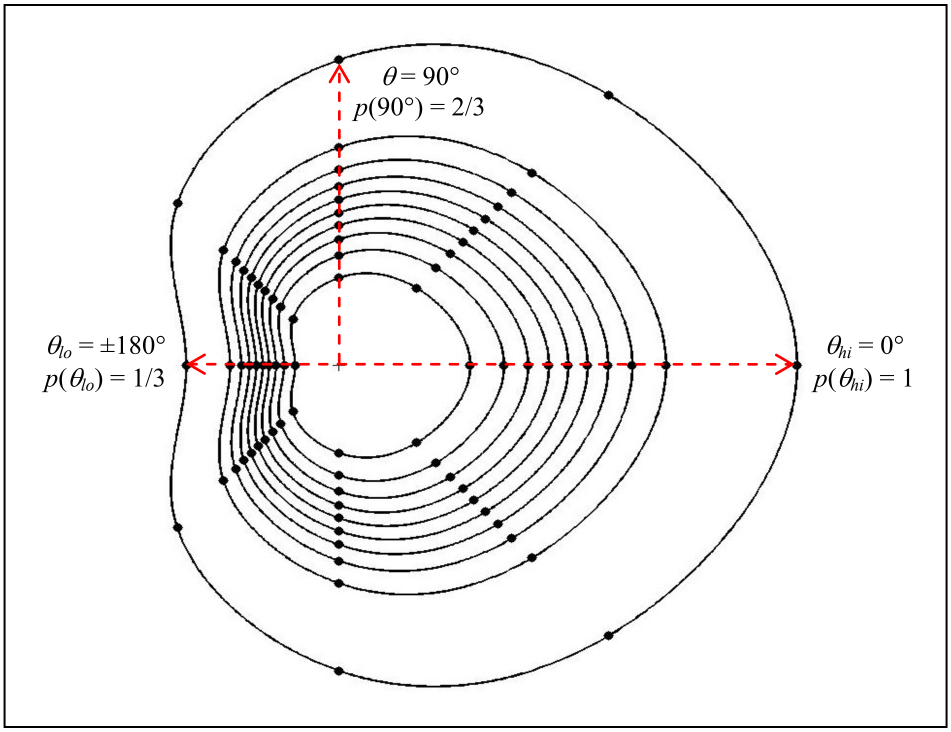


Figure 3.12. Iso-probability curves modified to reflect the effect of a likely travel direction.

3.5 Addressing Iso-Probability Curve-Construction Accuracy

Given the proposed approach of using iso-probability curves for target-location prediction, an important issue that arises is the accuracy of the curves. It is evident from the preceding sections that the control points play a major role in accounting for the probabilistic information on the target’s motion and for the impact of other relevant influences. In particular, it is the positions of the control points that change in response to terrain, obstacles, clues found, and particular lost-person psychologies. Clearly, then, for any specified set of iso-probability curves (i.e., for any set of cumulative probabilities that one wishes to consider), the more control points that are defined, the more accurately the target-location prediction information will be represented by the resulting iso-probability curves. The number and positions of control points per ray are set by the specified cumulative probability points that have been selected. Thus, if it has been decided, for example, that the 10%, 30%, 50%, 70%, and 100% cumulative probabilities for target location are to be considered, then, every ray will have exactly five control points, corresponding, respectively, to those five cumulative probabilities. Determination of the number and position of control points to use (and, therefore, of the number and position of iso-probability curves to establish) is a problem in robot deployment, and will be described in detail in Chapter 4. However, for any specified set of iso-probability curves, the number of *rays* that are defined and used to construct those curves will determine the total number of control points that exist overall, which, in turn, will decide the degree of accuracy of the curves.

The accuracy of iso-probability curves, then, can be increased by utilizing more rays, since this would produce more control points for (curve) interpolation. Simulations were, thus, performed in order to study the impact of the number of rays used to construct iso-probability curves on the accuracy of those curves. However, any measure of the degree of accuracy must be made relative to a benchmark. Since using more rays increases the accuracy of iso-probability curves, this means that constructing the “true” set of iso-probability curves would require an infinite number of rays. Since this is not practically feasible, a suitable alternative benchmark for comparison was required. In the study of accuracy that was conducted, the “true” iso-probability curves were taken to be those resulting from the use of a very large number of rays, but that, nevertheless, were not computationally prohibitive to construct and work with. Thus, an arbitrary selection of 360 rays, distributed uniformly from 0° to 360°, was made. This choice gave a considerably dense set of rays that not only resulted in reasonable curve-computation times, but also allowed for a large range of fewer numbers of rays (i.e., less accurate iso-probability-curve sets) to be compared.

Given a suitable “true” set of iso-probability curves to serve as a benchmark, a measure of the accuracy of a given set of curves had to be defined. This accuracy measure was represented in terms of relative curve-fit errors. For an approximated set of iso-probability curves obtained with a number of rays fewer than 360, the errors in curve-fits were defined as the sum of the distances between corresponding points on the approximated and the true curves at the 360 rays. Thus, let F_k represent the piecewise cubic polynomial set used to construct iso-probability curve *k* (see Eq. 3.6) of the *approximate* set of curves based on a total of N_{rays} rays, where $N_{rays} < 360$. Let $F_{TRUE,k}$ represent the cubic polynomial set for that same curve, but constructed using 360 rays (i.e., the *true* iso-probability curve *k*). Then, the error, E_k, corresponding to this one approximate iso-probability curve would be determined by first computing the positions, $P_{i,k} = F_k(\theta_i)$, on the approximate iso-probability curve *k*, at each angular ray location, θ_i, $i \in [1, 360]$, corresponding to the 360 angular ray locations used to construct the true curves. Next, the true positions, $P_{TRUE,i,k} = F_{TRUE,k}(\theta_i)$, on the true iso-probability curve *k*, are also computed at each of those same angular ray locations, θ_i, $i \in [1, 360]$. These computations would result in a true-position counterpart, $P_{TRUE,i,k}$, to every approximate position, $P_{i,k}$, that was computed. The sum of the absolute differences between every such pair, $\{P_{i,k}, P_{TRUE,i,k}\}$, $i \in [1, 360]$, would yield the error, E_k, corresponding to iso-probability curve *k*:

$$E_k = \sum_{i=1}^{360} \left| P_{i,k} - P_{TRUE,i,k} \right| . \tag{3.24}$$

The total curve-fitting error, E_{TOTAL}, corresponding to the entire set of the approximate iso-probability curves would, therefore, be the sum of the errors computed individually for each curve, $k \in [1, N_{curves}]$, in that set:

$$E_{TOTAL} = \sum_{k=1}^{N_{curves}} E_k . \tag{3.25}$$

The curve-fitting error, E_{TOTAL}, given by Eq. (3.25), then, represents the measure of the accuracy of any given set of iso-probability curves used for the study of iso-probability curve-construction accuracy that was conducted.

The study, in particular, was carried out to determine how curve-fitting errors vary with increasing number of rays. In this study, iso-probability curves were generated for a fixed-time point of 1800 s, for each ray-set size in the range of 3 rays up to 40 rays. For each set, 100 different angular positions of the ray-set were randomly selected, and total curve-fit errors were calculated for each case. This allowed for the computation of the average, minimum, and maximum total-error values for each ray-set size simulated. Based on this data, Fig. 3.13 shows the plot of error ranges, as well as the average error, for each of these ray-set sizes. As can be noted, the errors in curve-fits drop early and rapidly for increasing ray-set sizes. Furthermore, the decrease in the range widths and their increased overlap indicate a lack of significant difference in error-reduction capability for relatively larger number of rays.

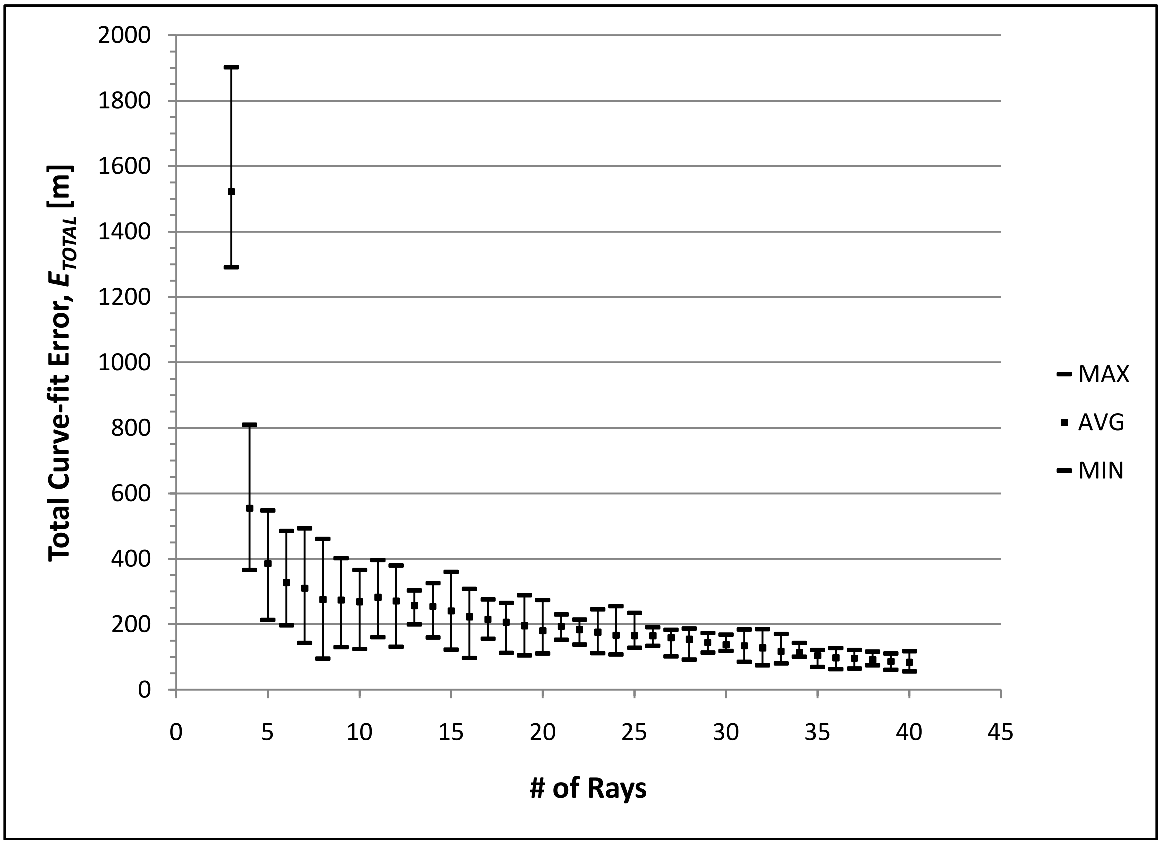
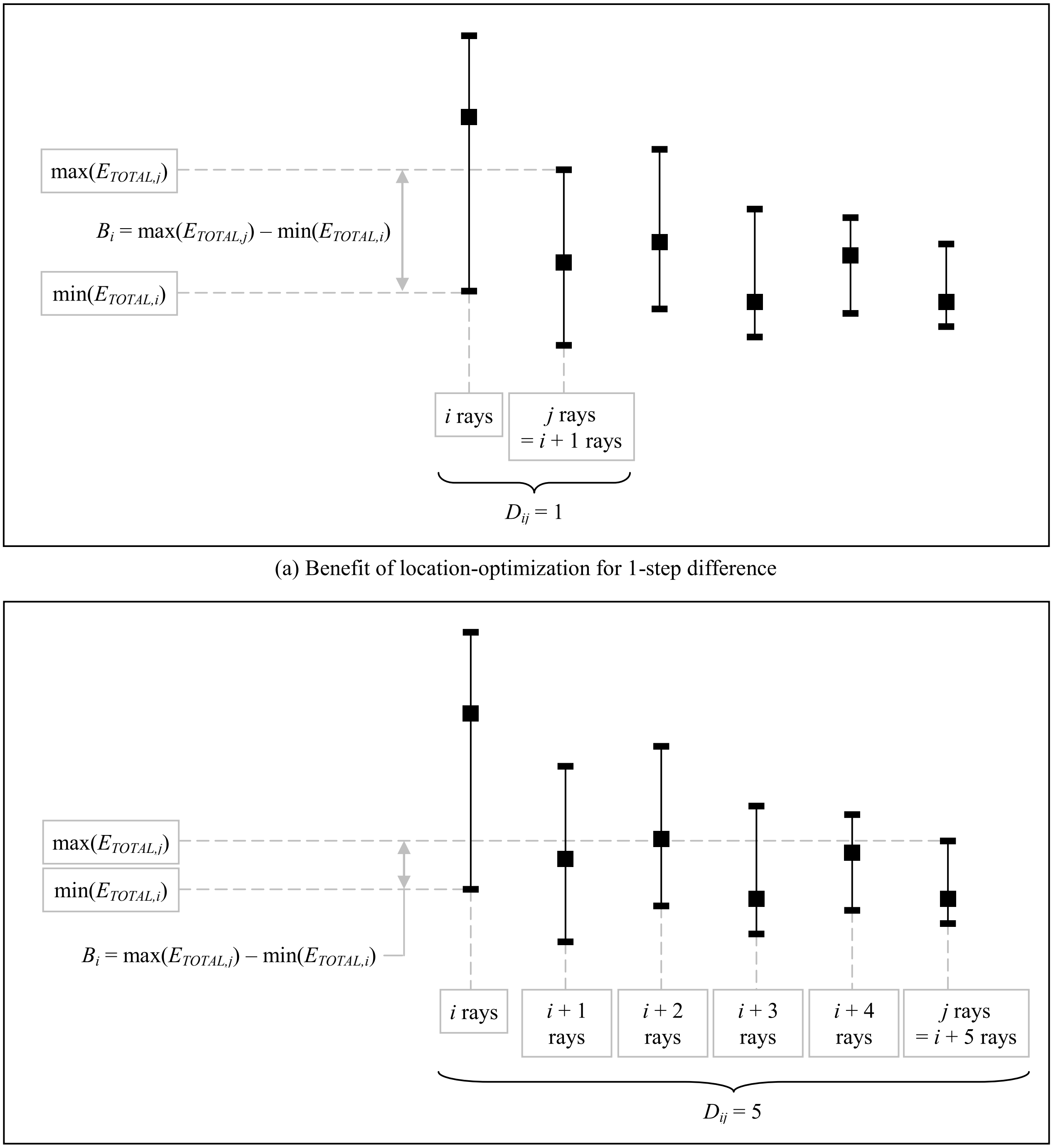


Figure 3.13. Error range variations with respect to ray-set size.

The above findings raise the issue of whether optimal ray placement, (i.e., finding the minimum number of rays and their best locations), could be beneficial. In terms of number of rays, one could conclude that computational time savings, during curve-fitting, from using lesser rays, would be insignificant. Rather, it would be more beneficial to improve curve-fit accuracy through a simple increase in the number of rays. In terms of location of rays (i.e., angular orientation of a ray-set), a measure of potential benefit that can be achieved by using the 'best' location of a set of rays can be taken to be the difference in error between the worst set of curves for a larger number of rays and the best (location-optimized) set of curves for the given (smaller) number of rays. This is illustrated in the example in Fig. 3.14. Consider two given ray-set sizes, i and j, $i < j$. Furthermore, let max($E_{TOTAL,j}$) be the maximum error that can be obtained for j rays, corresponding to a particular location of a uniformly distributed set of j rays, and let min($E_{TOTAL,i}$) be the minimum error that can be obtained for i rays, corresponding to a particular location of a uniformly distributed set of i rays. Then, the benefit of optimizing the location of i rays when comparing ray-set sizes i and j can be taken as $B_i = \max(E_{TOTAL,j}) - \min(E_{TOTAL,i})$. Moreover, the step size for the comparison would be the difference, $D_{ij} = |j - i|$, in the number of rays between i and j rays, so that the benefit of location-optimization can even be calculated for different step sizes. Figure 3.14a illustrates how one would calculate the benefit of location-optimization, B_i, for i rays given the error ranges corresponding to i and j rays for the case where i and j differ by only one ray (i.e., $D_{ij} = 1$). Figure 3.14b illustrates the same calculation for i and j differing by 5 rays (i.e., $D_{ij} = 5$).



(a) Benefit of location-optimization for 1-step difference

(b) Benefit of location-optimization for 5-step difference

Figure 3.14. Illustration of calculating the benefit of location-optimization for a set of rays for different comparison step sizes.

Using such a measure of benefit, studies that were conducted showed that any potential benefit that can be achieved rapidly diminishes as the number of rays increases. Thus, the more efficient and effective approach to constructing iso-probability curves, especially given the need for on-line feasibility of the target-location prediction method, would be to select a sufficiently large number of rays, separate them uniformly, and randomly select the angular orientation of the resulting ray-set. This, then, is the general approach adopted in the proposed method for iso-probability curve construction.

However, for a given number of rays, the distance between rays will increase as the iso-probability curves propagate outwards with time due to the increased circumferential length of any given curve, thereby effectively decreasing the resolution of the curves and their accuracy. Hence, one may have to increase the number of rays over time to maintain a desired level of curve-fit error. In order to determine how many rays are sufficient at any time, it would be necessary to estimate how curve-fit errors vary with time for different ray-set sizes. This may be done using intermediate curve-fit data obtained on-line throughout the search, or through empirical data on typical curve-fit errors for the given search environment, if available. Interpolation may, then, be used to estimate how many rays are required at any given point in time to maintain curve-fit error below a maximum error threshold value.

3.6 Updating / Propagating Iso-Probability Curves

As search time passes, continued possible target motion outward from the LKP will necessitate the propagation of the iso-probability curves outward as well, based on the probable target motion along each ray. Theoretically, the probabilistic target-location information, P_{POS}, would be changing continuously in time. However, practically, it would not be feasible to compute a new P_{POS} for every time-instant during the search. Instead, it is proposed that the target-location information, P_{POS}, be updated at a user-specified, finite-time interval, Δt_{prop}.

The use of the iso-probability curve concept to represent P_{POS} makes the update and propagation of the probabilistic target-location information, after every Δt_{prop} time-period, a straight-forward process of re-computing the positions of the control points for the new point in time specified, based on the amount of time passed since the last propagation. Thus, in general, at some time, t, there would be an existing set of iso-probability curves constructed from some N_{rays} rays, where the nominal target-location PDF, $p(r, t)$, for that point in time has been modified individually for each ray in order to account for the relevant influences discussed above and to compute the positions of the corresponding control points accordingly. Then, at the future point in time, $(t + \Delta t_{prop})$, a new set of iso-probability curves are constructed by translating each control point on each ray a specific amount along its respective ray corresponding to the elapsed time, Δt_{prop}.

Computation of the propagated curves and their new control points begins with obtaining the nominal mean target-speed PDF. This PDF is part of the background information pertaining to the target and is always available. Relevant influences are then incorporated into this PDF, as required, and it is subsequently used to construct the target-location PDF through a transformation of variable from speed, *v*, to position, *r*, for the given time-point of interest, ($t + \Delta t_{prop}$). One can recall from the preceding sections, however, that certain influences are incorporated into the mean target-speed PDF, before using it to construct the target-location PDF, while others are incorporated into the target-location PDF. In particular, accounting for the influence of a certain psychological behaviour, namely, a preferred direction of travel, is accomplished by applying an appropriate scale-factor to the nominal mean target-speed PDF. Thus, during iso-probability curve propagation, the same procedure is conducted, and the nominal mean target-speed PDF is scaled as required. The resulting PDF is, then, converted to a target-location PDF through the appropriate transformation of variable (Eq. 3.1).

So, for example, if a normal distribution, $N(\mu_v, \sigma_v^2)$, with mean, μ_v, and variance, σ_v^2, is assumed for the nominal mean-target speed PDF, then the influence of a preferred direction of travel would result in the application of a unique scale-factor, $q(\theta_i)$ (see Eq. 3.23), to the nominal mean target-speed PDF along each ray θ_i, $i \in [1, N_{rays}]$. This would modify the nominal mean target-speed PDF, resulting in an altered normal distribution, $N\left(\mu'_{v,i}, {\sigma'_{v,i}}^2\right)$, with new mean, $\mu'_{v,i}$, and new variance, ${\sigma'_{v,i}}^2$, for the PDF along each ray θ_i. Next, the appropriate transformation of the variable corresponding to the new time ($t + \Delta t_{prop}$), for which the iso-probability curves need to be constructed, is applied:

$$r = v\left(t + \Delta t_{prop}\right). \tag{3.26}$$

With this transformation, the resulting propagated target-location PDF along any given ray, θ_i, for time ($t + \Delta t_{prop}$) becomes:

$$p\left(r, \left(t + \Delta t_{prop}\right), \theta_i\right) = \frac{1}{\sqrt{2\pi}\left(t + \Delta t_{prop}\right)\sigma'_v} exp\left[-\frac{\left(r - \mu'_v\left(t + \Delta t_{prop}\right)\right)^2}{2\left(t + \Delta t_{prop}\right)^2 (\sigma'_v)^2}\right], \tag{3.27}$$

$$\forall i \in \left[1, N_{rays}\right].$$

With this target-location PDF obtained for each ray, it is, then, further modified to account for terrain influence, as described in Section 3.2. However, accounting for terrain influence requires establishing control points and translating them individually along their respective rays using the iterative process described in Section 3.2.1.1. Hence, for each ray θ_i, $i \in [1, N_{rays}]$, this target-location PDF would be used to delimit the positions of control points $r_{(t+\Delta t_{prop}),Pk\%}$, $k \in [1, N_{curves}]$, corresponding to the N_{curves} cumulative probabilities that one wishes to use in constructing the propagated iso-probability curves. Each of these delimited control points would constitute the initial estimates of the propagated control points for time $(t + \Delta t_{prop})$, which must, then, be re-positioned individually based on the influence of terrain through the iterative process that was described.

However, an important caveat that must be noted is the *portion* of terrain that is used in computing this terrain-influence per control point. When the initial iso-probability curves were established at the start of the search, the portion of terrain that was considered was that corresponding to the terrain height-map cells over which the ray in question lay, from the LKP up to the position of the control point being addressed (e.g., see Fig. 3.6). However, when iso-probability curves are being propagated, the relevant portion of terrain to consider is the *new* portion that would have to be traversed, from the *old* position of the control point being addressed up to its *new* position. This is illustrated in Fig. 3.15. At the time of propagation, $(t + \Delta t_{prop})$, the old control point, $r_{t,Pk\%}$, for example, located on ray θ_i, and which had corresponded to a cumulative probability of $Pk\%$, must now be moved to a new location $r_{(t+\Delta t_{prop}),Pk\%}$ to account for target motion as well as all other influences.

Using the target-location PDF obtained from accounting for target motion up to time $(t + \Delta t_{prop})$ and target psychology influence, as described above (e.g., Eq. 3.27), one can compute an initial estimate for $r_{(t+\Delta t_{prop}),Pk\%}$. In particular, this target-location PDF is used to determine the location of the point on ray θ_i corresponding to a cumulative probability of $Pk\%$. This point is the initial estimate, $r^{[0]}_{(t+\Delta t_{prop}),Pk\%}$, as labelled in Fig. 3.15. The relevant portion of terrain to consider, then, is that indicated by the thick red line in Fig. 3.15, between $r_{t,Pk\%}$ and $r^{[0]}_{(t+\Delta t_{prop}),Pk\%}$ along ray θ_i. Once this initial estimate is obtained, the iterative process for determining the final position of $r_{(t+\Delta t_{prop}),Pk\%}$ based on terrain influence proceeds as described in Section 3.2.1.1. The only differences would be that the height measurements that are used to determine speed scale-factor must be sampled from this new terrain portion only, and the distance of control-point-travel that is iteratively being determined is the distance travelled starting from $r_{t,Pk\%}$, rather than from the LKP, along θ_i. At the end of the iterative process, the correct scaled speed, $v_{Pk\%}$, corresponding to this control point will have been determined and used to compute the correct distance of travel outward from $r_{t,Pk\%}$ along θ_i to obtain the final propagated position, $r_{(t+\Delta t_{prop}),Pk\%}$, of this control point. This procedure is carried out for all control points on ray θ_i, as well as on all the other rays being used. Connecting control points of the same cumulative probability through piecewise cubic spline interpolation as described in Section 3.1 among this new set of control points will, then, produce the propagated set of iso-probability curves. To give a complete picture of what iso-probability curve propagation looks like, Fig. 3.16 shows a set of ten iso-probability curves, initialized at time $t = 1800$ s in Fig. 3.16a, and subsequently propagated up to times $t = 3000$ s and $t = 4200$ s in figures 3.16b and 3.16c, respectively. In this example, $\Delta t_{prop} = 1200$ s.

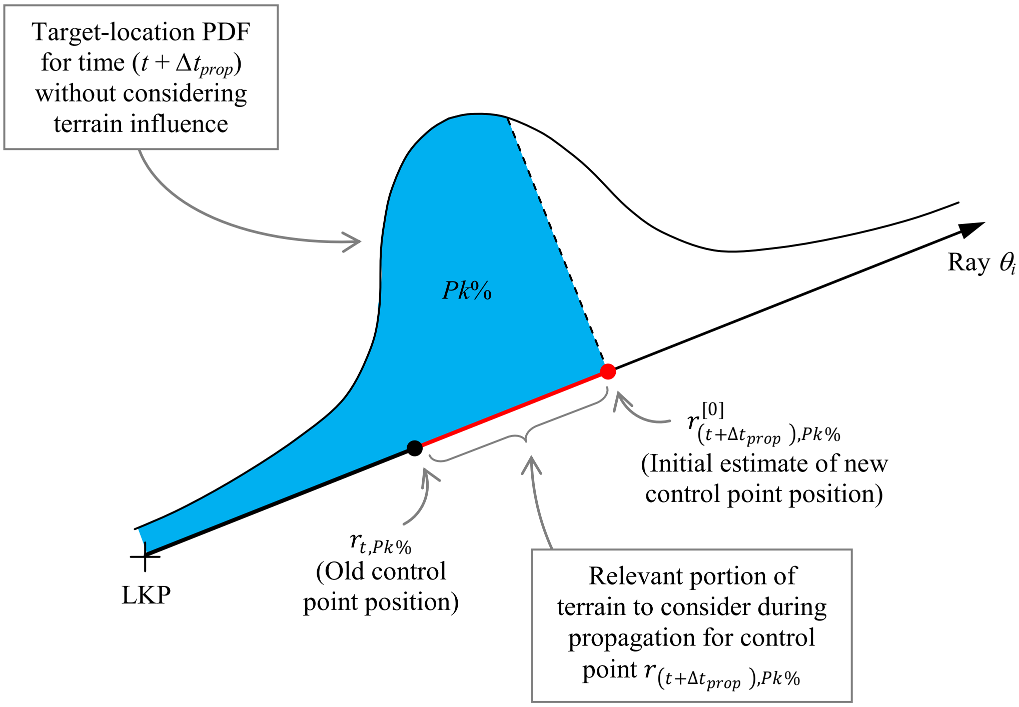
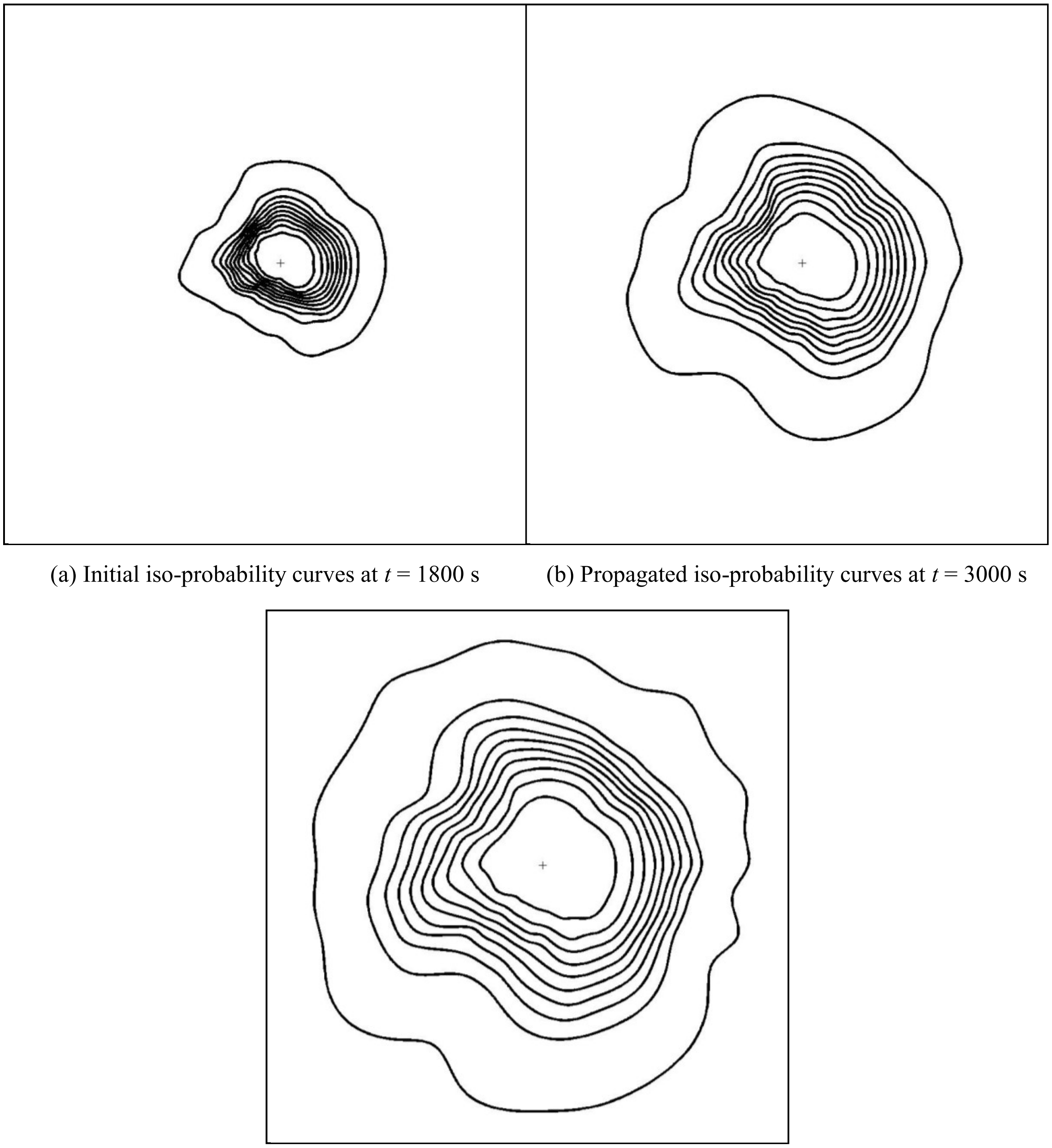

Target-location PDF for time $(t + \Delta t_{prop})$ without considering terrain influence
$Pk\%$
Ray θ_i
$r^{[0]}_{(t+\Delta t_{prop}),Pk\%}$
(Initial estimate of new control point position)
LKP
$r_{t,Pk\%}$
(Old control point position)
Relevant portion of terrain to consider during propagation for control point $r_{(t+\Delta t_{prop}),Pk\%}$

Figure 3.15. Incorporating terrain influence to determine the location of a control point during iso-probability curve propagation.

It is evident from the target-location prediction method described in this Chapter that the proposed approach differs from what would typically be expected in a probabilistic search method. The iso-probability curves essentially provide upper bounds on the target's location, given the influences of terrain, target physiology, target psychology, found clues, and obstacles. Establishing such conservative bounds ensures that one always searches within the area in which the target resides. Thus, the approach does not require a complex motion model specific to the target in question to be given, describing his/her likely motion within different regions of the search area, nor does it involve obtaining a measurement model and the complex computations that go along with it when trying to fuse measurement and motion uncertainty. The result is a method that is quick and efficient for on-line, autonomous implementation, with many iso-probability curves and robots involved, but one that still allows for an intelligent search to be conducted by taking into account probabilistic target information.

Given a method to (*i*) construct the initial iso-probability curves for representing probabilistic target-location prediction information; (*ii*) incorporate important influences relevant to WiSAR into these curves, and; (*iii*) propagate the curves over time and space, they can, then, be used in conducting the other two main tasks in the proposed MRC method. Both robot deployment and robot-path planning require the use of target-location prediction in order to be conducted effectively. In the next chapter, the proposed method for robot deployment will be presented, and it will be shown how the iso-probability curves play an integral part in determining the best deployment that should be implemented.



(a) Initial iso-probability curves at *t* = 1800 s

(b) Propagated iso-probability curves at *t* = 3000 s

(c) Propagated iso-probability curves at *t* = 4200 s

Figure 3.16. Propagation of iso-probability curves.

3.7 Chapter Summary

The chapter has detailed the proposed solution approach to addressing the target-location prediction sub-problem. The novel concept of iso-probability curves was introduced, which serve as a means of representing target-location prediction information through the establishment of probabilistic boundaries for target-presence within the search area. Procedures were introduced for incorporating relevant influences into the iso-probability curves, including the effects of terrain topology, a priori known obstacles, found clues, and certain lost-person psychological behaviours.

Simulations were conducted to determine the impact of the number of rays used to construct iso-probability curves on the curve-construction accuracy. The results indicated that the potential benefit that can be had by increasing the number of rays diminishes rapidly with increasing rays, so that a viable strategy for constructing curves could be to select a sufficiently large number of rays, separated uniformly, at the start of the search and to maintain them throughout the search.

A means of updating and propagating iso-probability curves over time and space to account for potential target motion was also presented. The strategy requires specifying a curve-propagation time-interval for regular curve updates, and moving each individual control point outward along its ray an amount corresponding to the propagation time-interval. All relevant influences are incorporated into the propagated iso-probability curves as usual, with the exception that terrain influence considers only the additional portion of terrain that each control-point traverses due to the propagation.

Chapter 4

4 Robot Deployment

As noted in Chapter 2, the second sub-problem to be addressed in robotic WiSAR is that of robot deployment. The deployment task refers to the distribution and allocation of search-effort within the given search area. In the approach to MRC for WiSAR proposed in this Thesis, the deployment and search tasks are considered to be distinct. Thus, robot deployment entails assigning search robots to specific positions within the search area and having the robots move to those positions. Once these positions are attained, a separate robot-path planning task (to be detailed in Chapter 5) is conducted that determines how the robots should move to perform the search.

The two main tasks in robot deployment are: (*i*) to determine the deployment positions within the search area where search robots should be placed, and (*ii*) to assign specific robots to those positions. The first task, in general, involves the specification of the regions within the search area for search-effort allocation, and specific deployment positions within those regions that must be attained by the robots. This requires the construction of a function that quantifies the benefit that any given selection of such regions and deployment positions would have for the ongoing search. This further requires defining the attributes of importance of a robotic WiSAR operation that one wishes to consider. These individual attributes would serve as measures of the impact of any given selection of search-effort allocation on the quantity represented by that attribute of the search. Given quantitative formulations of these attributes, they would, then, have to be combined to form a single objective function representing the overall benefit to the search entailed by any given selection of search-effort allocation. Such an objective function would allow for optimizing the selection of this allocation. This optimal selection would represent the best positions within the search area where robots should be deployed in order to conduct the search most effectively. However, by-products of this optimization would include the regions within which to apply search-effort (the deployment positions specified would lie in these regions) and the number of robots that must search each region (i.e., the resource-requirements per region).

When deployment positions have been selected, one is still faced with the problem of assigning robots to those positions so that they may subsequently begin searching the corresponding search-effort allocation regions. The determination of which robot to assign to which deployment position constitutes the second task of robot deployment. Since a robot, once assigned to a deployment position, must move and expend some finite effort (e.g., travel-distance or travel-time) to attain that position, one would wish to select the assignments in a way that minimizes such effort. Thus, this problem of assigning robots to deployment positions also requires an optimization. The optimal assignment of robots, then, would result in the specification of deployment positions for each robot such that every optimal search-effort allocation region stipulated is addressed by the required number of robots, and the effort that must be expended by the robots to attain those deployment positions is minimized.

Of course, the selection of optimal deployment positions and the selection of optimal assignments of robots to those deployment positions are intertwined tasks. An optimal assignment of robots to deployment positions will inevitably be affected by the particular deployment positions selected in the first place. Hence, these two tasks must be considered together rather than being treated as two separate optimizations. The quantified effort entailed by any assignment scheme of robots to deployment positions would, in fact, form yet another attribute of the ongoing search to be included in the overall benefit function considered for the first task. This would result in a single optimization that, nevertheless, accounts for both tasks of robot deployment. The decision variables to be optimized would, then, be the deployment positions and the assignment scheme of robots to those deployment positions, all taken together.

Since WiSAR is a dynamic problem by nature, a single optimization of deployment, and the subsequent attainment of those deployment positions by the robots, would not suffice for the entire ongoing search. As the target-location prediction information propagates with time, and as new information about the search is obtained (i.e., clues, modified terrain topography, discovered *a priori* unknown obstacles), a given distribution of search resources within the search area may no longer be optimal and a revised distribution may be required. Thus, robot deployment is, in actuality, a recurring task where, subsequent to an initial optimal deployment of search resources, one must regularly check the optimal deployment solution throughout the ongoing search and re-optimize and re-distribute search resources when required. Any such instance of re-optimization and re-distribution of the search robots is termed ‘re-deployment.’ Re-deployment requires both a specified time-interval dictating when checks of the optimality of the current deployment are conducted, and a decision-making method to decide when the current deployment is sufficiently sub-optimal to warrant implementing a new deployment solution.

In the following sections, the details of the proposed methods for addressing each of the aforementioned two main tasks of robot deployment, as well as the means by which they are combined and addressed together to account for their dependency, will be presented. Subsequently, a method will be proposed by which the optimality of a given deployment, and the benefit of conducting a re-deployment, can be assessed on-line in order to determine whether or not the additional effort in implementing a re-deployment is justifiable. Following this, the results of a study, conducted through simulations to analyze the benefit of using re-deployment during search, will be given, thereby validating the need for re-deployment in the overall MRC methodology for WiSAR.

4.1 Optimal Deployment Method

Firstly, it is important to re-emphasize that in the proposed method for optimal deployment, a distinction is made between 'initial deployment' and 're-deployment.' Initial deployment refers to the distribution of robots within the search area at the start of the search. Thus, when a notification of a missing person arrives, the robots are transported to the site of the LKP and an optimal deployment solution is computed. Optimal deployment positions for all the robots are found and it is assumed that the robots can be directly 'dropped' onto their respective deployment positions. The robots, then, commence the search. In contrast, re-deployment refers to the process of computing a new optimal deployment solution while the search is in progress.

A WiSAR search scenario is dynamic due to the presence of a mobile target (and correspondingly changing probabilistic target-location information), varying terrain (including possible discovery of *a priori* unknown obstacles), and clues found during the search. Due to this dynamic nature, deployment is not simply a singular task that is only solved once and is completed upon determining the initial deployment positions of all robots at the start of the search. Instead, the problem is more complex because, in addition to initial deployment, the new information obtained due to the changing environment and target-state require regular consideration of whether deployment needs to be re-computed in order to change the way search-effort is allocated as the search progresses.

Thus, for autonomous multi-robot WiSAR, search-effort allocation must be dynamic and autonomous, implemented in an on-line mode, where an initial deployment of search resources must be followed by regular re-evaluation and re-optimization to obtain a new optimal deployment solution. This solution *may*, then, be implemented, if it is found that there is sufficient benefit in doing so. If it is decided that the new optimal deployment solution should be implemented, then each robot stops following its current assigned path and moves toward its newly-assigned deployment position, after which the robot continues to perform the search as per given search instructions. This revision to the deployment of the robots, mid-search, and the subsequent detour by the robots to return to their assigned deployment positions before continuing with the search, constitutes a re-deployment.

As discussed in the introduction above, there are two main tasks to robot deployment. First, optimal deployment positions within the search area must be determined. Second, an optimal assignment of robots to those positions must be found. Two by-products of the overall optimization that is conducted to addresses these tasks are the regions within the search area where search-effort should be allocated, and the number of robots required per region. In the proposed method for robot deployment, determination of the optimal deployment positions begins with specifying these regions based on optimizing a combination of certain important attributes of the search. Once regions have been delimited, optimal deployment positions within those regions can be selected, and robots can be assigned to those positions. These three general objectives are summarized in Fig. 4.1 for later reference.

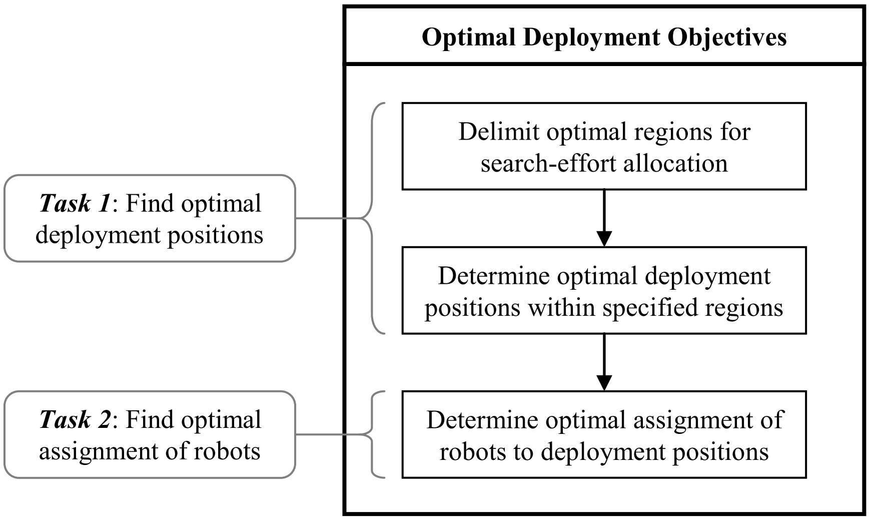

Optimal Deployment Objectives
Delimit optimal regions for search-effort allocation
Determine optimal deployment positions within specified regions
Determine optimal assignment of robots to deployment positions
Task 1: Find optimal deployment positions
Task 2: Find optimal assignment of robots

Figure 4.1. Three main objectives of the general process for optimal robot deployment.

It is clear that in conducting an effective search for a non-trackable target, the target-location prediction information available must play a major role. Indeed, since this is the only information about the target's location at any given time, quantification of the aforementioned 'attributes' of the search that need to be considered in determining the best regions for search-effort allocation, and the optimal deployment positions within them, will undoubtedly have to rely on this prediction. Since target-location prediction is represented using the concept of iso-probability curves, herein, delimitation of the regions for optimal search-effort allocation is conducted via these curves.

Each iso-probability curve bounds a region of the search area within which the target exists with a certain probability. Since these curves visually portray information related to target-location probability, and have a direct physical correspondence to the target's probable location along any given ray at any given time, they allow for a convenient and intuitive means of determining and specifying the regions for search-effort allocation to be considered. In particular, it is proposed in this Thesis that robots be deployed directly onto iso-probability curves. An appropriate selection of curves, then, would represent an appropriate selection of search-effort allocation regions. Deployment positions must, therefore, be specified on those curves, in proper numbers, and the robots must be optimally distributed among those positions.

Upon attaining their assigned positions, the robots commence the search by moving along the curves on which they were deployed. In this sense, the 'regions' for search-effort allocation can be seen as the areas bounded by the iso-probability curves, and the robots search by patrolling the boundaries of those regions. By using such an approach, search-effort is distributed within the search area by virtue of the density of the curves in different parts of the search area – a higher density of curves directly corresponds to areas where there is a relatively high likelihood of target presence, while lower curve-density areas correspond to lower likelihood areas. Appropriately, then, with robots assigned to and moving along curves, the higher likelihood areas will see more search-effort applied relative to the lower likelihood areas.

Referring to Fig. 4.1, since the first objective of the proposed robot deployment method is to select the regions, meeting this objective requires optimally selecting the iso-probability curves. This involves selecting the cumulative probability values to consider per ray (each iso-probability curve corresponds to a unique cumulative probability value), which, in turn, specifies the number and location of iso-probability curves.

The optimal selection of the iso-probability curves would depend on the benefit that any given choice of curves would have on the search in progress. In this Thesis, this benefit is quantified through the evaluation of two attributes of the search: search-time and success-rate. One can recall that search-time quantifies a representative measure of the expected total time that would be required to find the target, and success-rate quantifies a representative measure of the probability of finding the target. Different selections of the numbers and positions of iso-probability curves will evaluate to different values for the search-time and success-rate measures. As such, at any given point in time when an optimal deployment is computed, the iso-probability curves are selected through an optimization that minimizes the measure of search-time and maximizes the measure of success-rate.

The next two objectives in robot deployment, as outlined in Fig. 4.1, are to select optimal deployment positions within the search-effort allocation regions (i.e., on the optimal set of curves), and to optimally assign the robots to those positions. In the proposed method, these two objectives are met simultaneously in the interest of reducing computational complexity and making the optimization on-line-feasible. In particular, rather than finding the optimal deployment positions among the infinite possibilities on each iso-probability curve, the robots are first *assigned* to the selected curves, without specifying any position. This assignment is determined based on a specified robot-quantity requirement (i.e., resource requirement) for each curve, derived through a method that computes a balanced distribution of search-effort among the curves. Once each robot has been assigned to an iso-probability curve, the end-point of the shortest path from each robot to its assigned curve defines the deployment position that must be attained by the robot.

Of course, any given assignment of robots to iso-probability curves will entail the expenditure of some effort by the robots to attain their deployment positions. In this Thesis, this effort is quantified by the time required by the robots to attain their deployment positions, given the shortest-paths from each robot to its assigned curve and an estimate of the traversal-time for each of those paths. To represent this effort in a single quantitative measure, the time required by the *last* robot to attain its deployment position is used. This effort-measure is termed ‘return-time,’ the name being a reference to the fact that robots are essentially ‘returning’ to their assigned curves before they once again commence searching. Given this return-time measure, optimal assignment of robots to curves must be conducted in a manner the minimizes this measure.

Overall, the two main tasks in robot deployment, namely, selecting deployment positions and assigning robots to them, which are executed by meeting the three objectives of (*i*) specifying iso-probability curves, (*ii*) assigning robots to curves, and (*iii*) taking shortest-path end-points as the deployment positions, is conducted in a contiguous manner. A single overall optimization problem is set-up whereby a search engine navigates through the space of possible iso-probability curve selections, and, for each curve-set considered, computes the optimal allocation of robots to curves, thereby also determining optimal deployment positions in the process. Since iso-probability curves must be selected so as to optimize the search-time and success-rate measures, and robot assignments to curves must be selected so as to minimize the return-time measure, the overall objective function of the robot deployment optimization is a weighted sum of these three quantified attributes of the search.

In Section 4.1.1, details of the procedures for formulating metrics for the three attributes used in the overall optimization will be provided, followed by the resulting overall optimization formulation that must be solved each time a deployment solution is required. However, there is a difference between optimal initial deployment and optimal re-deployment. Thus, the general method for deployment that is presented will be further qualified by presenting the details of additional considerations that must be made to account for each of these two situations where optimal robot deployment is computed. In particular, Section 4.1.2 will detail the procedure used to determine whether a re-deployment is warranted *during* the ongoing search, *after* an initial deployment has been found and implemented, while Section 4.1.3 will describe the special case of optimal initial deployment and how the deployment method presented is modified to accommodate this special case.

4.1.1 Optimal Allocation of Search-Effort

In the particular deployment method proposed herein, the two primary tasks for search-effort allocation are (Fig. 4.1): (*i*) selecting optimal deployment positions, and (*ii*) selecting an optimal assignment scheme for robots to those deployment positions. The former task requires delimitation of the optimal regions for search-effort allocation and selection of optimal deployment positions within those regions. The latter task is, then, achieved by optimally assigning specific robots to those positions. In the proposed approach to solve the deployment problem, the optimal regions are established through the selection of the optimal *number* of iso-probability curves and the *positions* of these curves. However, the determination of the optimal deployment positions and assignment of robots to them are achieved together. In particular, each robot is assigned to a specific curve, and the shortest path from each robot to its assigned curve is used to determine the deployment positions for the robots on the curves. In this way, the optimal deployment positions and the assignment of robots to those positions are determined through a single process. This solution approach is summarized in Fig. 4.2 (which builds on the problem-summary from Fig. 4.1).

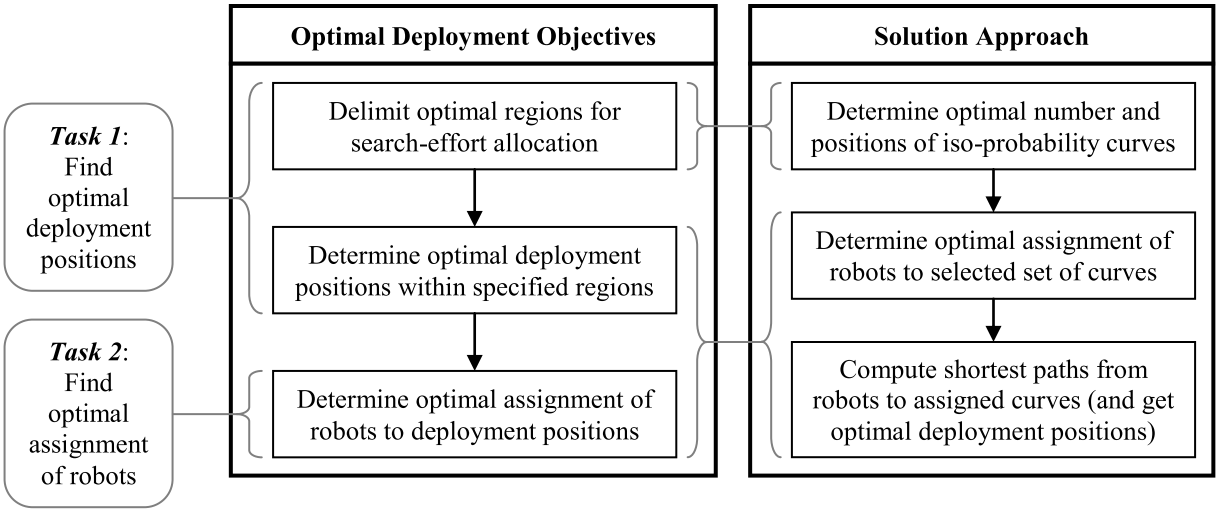

Optimal Deployment Objectives
Solution Approach
Task 1: Find optimal deployment positions
Task 2: Find optimal assignment of robots
Delimit optimal regions for search-effort allocation
Determine optimal deployment positions within specified regions
Determine optimal assignment of robots to deployment positions
Determine optimal number and positions of iso-probability curves
Determine optimal assignment of robots to selected set of curves
Compute shortest paths from robots to assigned curves (and get optimal deployment positions)

Figure 4.2. Summary of solution approach to the robot deployment problem.

Since the iso-probability curves are obtained from the target-location PDF, using the curves to delimit regions for search-effort allocation and select deployment positions in this manner ensures that deployment is conducted in accordance with the available probabilistic target-location information. Furthermore, since the iso-probability curves consider the influences of important factors in WiSAR scenarios, using the curves to conduct deployment allows for these issues to be accounted for during deployment optimization as well.

Thus, as indicated in Fig. 4.2, determining an optimal deployment begins with computing an optimal set of iso-probability curves. For this optimization, the decision variables are the number of iso-probability curves, N_{curves}, and the position, R_p, of each curve, $p \in [1, N_{curves}]$. The position of a curve actually refers to its identity – namely, the cumulative probability of the corresponding control points that form the curve. A curve's position value, therefore, refers to the probability that the target resides anywhere on any given ray, from the LKP up to the point on the ray that the iso-probability curve crosses. Thus, if $R_p = \rho\%$, it implies that, at some time $t = T$, the p^{th} curve passes through the point at $r_{T,\rho\%}$ on each ray, $i \in [1, N_{rays}]$, where:

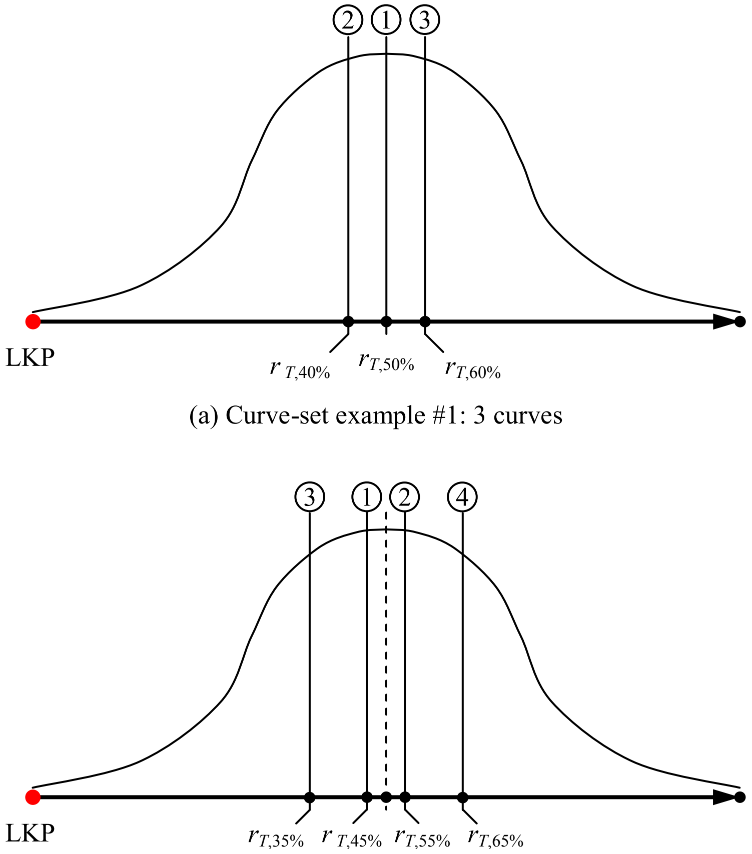
$$\int_0^{r_{T,\rho\%}} p(r, t, \theta_i) dr = \rho\%, \forall i \in [1, N_{rays}]. \tag{4.1}$$

For example, a possible set of decision variable values considered could be $\{N_{curves} = 3; R_1 = 37\%, R_2 = 57\%, R_3 = 77\%\}$, representing three curves with positions R_1, R_2, and R_3, respectively. The first curve bounds an area corresponding to a 37% cumulative probability for target presence along any given ray, the second curve bounds an area corresponding to a 57% cumulative probability for target presence, and the third curve bounds an area corresponding to a 77% cumulative probability for target presence.

Although there exists the flexibility to place a set of N_{curves} iso-probability curves at any positions, R_p, $p \in [1, N_{curves}]$, the most sensible choice would be to begin placing the curves where there is the highest probability of target presence and, then, move into the other regions of the search area in decreasing order of target-presence probability. This strategy would increase the likelihood of finding the target, while decreasing the expected time to target-detection. Thus, in the case of a normal distribution for the target-location PDF, this would imply that the set of curves must be centered on the 50% cumulative probability position (i.e., at the mean of the PDF) along any given ray, since this position corresponds to the highest probability density value. Moreover, since the normal distribution is symmetrical about its mean, and decreases exponentially in both directions as one moves away from the mean, additional curves would, thus, be added symmetrically about the mean position, successively.

Figure 4.3 shows two examples of curves being placed symmetrically about the mean by showing the points at which the curves cross an arbitrary ray. In both examples an inter-curve spacing of 10% is assumed. In Fig. 4.3a, three curves are placed, with the first curve placed at the mean position ($r_{T,50\%}$), the second curve placed to the left of the mean corresponding to 10% lower cumulative probability ($r_{T,40\%}$), and the third curve placed to the right of the mean corresponding to 10% higher cumulative probability ($r_{T,60\%}$). In Fig. 4.3b, an example of a four-curve set is shown, where the first two curves are placed symmetrically about the mean at the

45% and 55% cumulative probability points, respectively, and the next two curves are also placed symmetrically about the mean at the 35% and 65% cumulative probability points, respectively. In both examples, it can be seen that the curve-sets are centered at the 50% cumulative probability position along the ray. However, it should be noted that it is only important that the *centre* of the curve-set lie at the 50% cumulative probability position. Depending on the number of curves that are to be placed, it is not necessary to always have a curve actually placed at the 50% position.



(a) Curve-set example #1: 3 curves

(b) Curve-set example #2: 4 curves

Figure 4.3. Examples of curve-placement (shown in terms of intersection points of the curves along an arbitrary ray).

Use of the infinite-range normal distribution for the target-location PDF requires placing upper and lower truncation points. These points place upper and lower bounds on curve position as well. Since a curve must have a finite distance from the LKP in order to exist, the user must specify a minimum, non-zero position, R_{min}. The R_{min} parameter should be selected as small as is practically possible and useful, so that it maximizes the number of curves that can be placed, thereby allowing for more solution possibilities. However, the maximum curve position, R_{max}, becomes the upper truncation point of the normal target-location PDF.

In order to simplify the specification of the curve positions, an alternative parameter that provides a more concise form of notation, R_c, is used. This parameter specifies the *centre* position of a curve-set with a constant, user-selected, inter-curve cumulative-probability interval (i.e., inter-curve spacing), D_c. This interval is selected such that the maximum number of curves allowable (for the number of robots available) is able to reasonably fill the expanse between the LKP and the "100%" curve position. This selection allows for the most flexibility when searching for the optimal set of curves. The final result is a single decision variable, N_{curves}, for the given parameters R_c and D_c. For example, for $R_c = 0.50$ (i.e., normal target-location PDF) and a specified $D_c = 0.07$, with $N_{curves} = 3$, the set of iso-probability curves considered would be $\{R_1 = 43\%, R_2 = 50\%, R_3 = 57\%\}$. Here, the curves are centered at the highest probability density position, $r_{T,50\%}$, and curves are placed symmetrically about this position. Since there are three curves, one curve ends up exactly at the 50% cumulative probability position (just as in the example curve-set in Fig. 4.3a), and the other two are placed at positions corresponding to 7% below and 7% above this cumulative probability value. Thus, the task of determining the regions for search-effort allocation reduces to a single-variable optimization problem – determining the *number* of iso-probability curves, N_{curves}, where the *positions* of the curves follow naturally from the specified values of the parameters R_c and D_c.

For any specified number (and positions) of iso-probability curves, the proposed method for optimal search-effort allocation dictates the establishment of a certain preferred distribution of resources among the curves. In particular, since iso-probability curves are comprised of a concentric set of curves, each curve will have a different circumferential length than the others, and will, consequently, bound regions of different areas. The main consequence of this is that each curve must be assigned a different number of robots to account for this 'imbalance' in curve-length and bounded areas. Thus, when determining an optimal deployment, the search engine will, in general, increment to a different possible number of curves with each iteration, and will consider that valuation of the decision variable. The moment a certain number of curves is specified for consideration, it is possible to fully specify and represent each curve (i.e., derive the corresponding piecewise cubic polynomial set). This, then, also allows for the computation of the preferred distribution of robots among curves using a method devised in this Thesis, before proceeding any further with the optimization. This preferred distribution is used in the computation of the attributes of the search that are used in finding the optimal deployment solution.

In the overall optimization to find the optimal deployment, three key attributes of the search are considered, herein: (*i*) search-time, A_T (the expected total time taken to find the target), (*ii*) success-rate, A_S (the probability of finding the target), and (*iii*) return-time, A_R (the time taken by the robots to reach their assigned deployment positions). These attributes of the search need to be expressed quantitatively in order to form corresponding metrics by which the optimality of any given selection of iso-probability curves, and the assignment of robots to those curves, can be judged. In deriving these metrics, it is important to note that the intent is to obtain a rough estimate of, respectively, the time it would take to locate the target, the associated probability for success, and the time it would take for the robots to reach their deployment positions. Such estimates are carried out in the interest of computational-timesaving. Using rough estimates is still a viable approach since the proposed method regularly conducts re-deployment throughout the search, whereby the selection of curves and the assignment of robots to them are continually improved as required. Consequently, a poorly performing set of curves would eventually be detected and improved upon. In Section 4.1.1.1 that follows, the proposed procedure for computing the balanced distribution of robots for a given number of curves is presented. The next three sub-sections present the formulations for the proposed metrics for search-time (Section 4.1.1.2), success-rate (Section 4.1.1.3), and return-time (Section 4.1.1.4). Section 4.1.1.5 will present the overall optimization formulation that uses these metrics to determine an optimal search-effort allocation.

4.1.1.1 Balanced Distribution of Search Resources

At any given iteration of the overall optimization, upon specifying a particular number of curves, N_{curves}, to be considered, a certain preferred distribution of robots among the curves must, first, be computed. This entails specifying the number of robots, $N_{r,p}$, for each iso-probability curve $p \in [1, N_{curves}]$ in the curve-set under consideration. The $N_{r,p}$ values are necessary for the evaluation of other attributes used in the overall deployment optimization, as will become evident in later sections when the remaining details of the deployment method are presented.

The robot search-strategy is tied to the search-effort allocation regions to which the robots are assigned. The future position of each region is computed in advance, and robots move toward that new position in order to remain within their assigned regions during the search. A larger region, therefore, would require more resources relative to a smaller region in order to receive the same amount of coverage in a given amount of time. For the proposed method, the search-effort allocation regions are associated with the iso-probability curves, and in a balanced distribution of search resources, no one curve (region) should be given more importance than another. Rather, it is the positions and the relative densities of the curves within the search area that dictate the distribution of search effort within that area. Thus, balancing the search-resources requires proportioning the robots among the curves according to some measure of the size of each curve. As can be noted, curves positioned further out from the LKP would have greater circumferential length and would encompass a larger area, and must, therefore, be assigned proportionately more robots compared to curves closer to the LKP.

In the proposed method for balancing search resources, robots are proportioned according to ratios of circumferential curve-lengths, where the length of the innermost (smallest) curve is taken as the basis of comparison. In particular, it is required that the ratio of the number of robots, $N_{r,p}$ (that are to be assigned to any curve $p \in [1, N_{curves}]$), to the number of robots, $N_{r,1}$ (that are to be assigned to the innermost curve, $p = 1$), must be the same as the ratio of the circumferential length, C_p (of curve $p \in [1, N_{curves}]$), to the circumferential length, C_1 (of the innermost curve, $p = 1$), Eq. 4.2. This gives the formula in Eq. 4.3 for the search-resource requirement, $N_{r,p}$, for any given curve, $p \in [1, N_{curves}]$, with restrictions as stated in Eq. 4.4 and Eq. 4.5. Equation 4.4 requires that the specified number of robots for each curve be an integer, and Eq. 4.5 requires that the sum of the robots assigned to all the curves must be equal to the total number of robots, N_r, available for the search.

$$\frac{N_{r,p}}{N_{r,1}} = \frac{C_p}{C_1}, \forall p \in [1, N_{curves}]. \tag{4.2}$$

$$N_{r,p} = \frac{C_p}{C_1} N_{r,1}, \forall p \in [1, N_{curves}], \tag{4.3}$$

$$N_{r,p} \in \mathbb{Z}, \forall p \in [1, N_{curves}], \tag{4.4}$$

$$\sum_{p=1}^{N_{curves}} N_{r,p} = N_r. \tag{4.5}$$

Since $N_{r,1}$ would not be *a priori* known, computing the resource requirements per curve in order to satisfy Eqs. 4.3 to 4.5 would require an iterative process. This process would be conducted as follows:

Step 1: Compute the circumferential length, C_p, of every iso-probability curve $p \in [1, N_{curves}]$, of the set of curves being considered.

Step 2: Define a scale-factor, K, and initialize its value to '1'.

Step 3: Set the value of $N_{r,1}$ to 'K'.

Step 4: Compute a 'raw' number of robots, $N_{r,p}$, for each iso-probability curve $p \in [2, N_{curves}]$ using Eq. 4.3.

Step 5: Revise each $N_{r,p}$ value computed in Step 3 by rounding the value to the nearest integer.

Step 6: Sum all the $N_{r,p}$ values and check if Eq. 4.5 is satisfied. If it is, then terminate the iterative process: the current $N_{r,p}$ values are taken as the final values for the number of robots to assign to each curve. If the sum in Eq. 4.5 is less than N_r, increment the scale-factor K by an amount ΔK and return to Step 3. If the sum in Eq. 4.5 is greater than N_r, decrement the scale-factor K by an amount ΔK and return to Step 3.

The value of the increment/decrement, ΔK, can be initialized to a convenient and reasonable value (in simulations for this Thesis, a value of 0.01 was used). Its value can, then, be decreased further (i.e., the resolution of the increment/decrement can be increased) if the iterative process above is found to oscillate between incrementing and decrementing ΔK in Step 6 between successive iterations without finding a solution.

It can be noted from this iterative procedure, then, that although the equivalent-ratio requirement of Eq. 4.2 is met by the 'raw' $N_{r,p}$ values computed in Step 4, upon rounding to meet the integer-value requirement of Eq. 4.4, this ratio equivalency may not hold exactly. Thus, Eq. 4.2 is being used here more so as a guide in order to help proportion the robots appropriately among the curves. The requirement that the number of robots assigned to each curve be an integer, and that the sum of all robots assigned be equal to N_r, must prevail to ensure a valid distribution of search resources.

4.1.1.2 Search-Time Metric

A metric that is commensurate with search-time can be devised by considering an alternative yet representative search scenario and strategy. Let $N_{r,p}$ be the number of robots assigned to the p^{th} iso-probability curve, $p \in [1, N_{curves}]$, in a given curve-set. After all robots have achieved their deployment positions, for every curve $p \in [1, N_{curves}]$, the corresponding $N_{r,p}$ robots would be situated *on* the iso-probability curve. Let us assume that the deployment positions specified are such that the robots on any given curve are uniformly distributed along the curve. Let us further assume that, subsequent to deployment, the robots conduct the search by travelling clockwise along the circumference of the iso-probability curve to which they have been assigned, at their constant maximum speed. If the target happens to be located on one of the curves, the robots assigned to that curve would 'find' the target. Based on this simplified scenario, one can derive a metric representative of search-time. Although the actual distribution of the robots on the curves, and the search strategy used, would be different, it can be expected that the trends and derived conclusions based on following the curves in this alternative way and assuming a uniform robot distribution per curve would generally hold true for other more complex robot distributions and search strategies based on those same curves. Indeed in the search strategy proposed in this Thesis (to be presented in Chapter 5), robots search by travelling along their curves, with additional considerations made to address the time-varying nature of the curves.

Given the above assumptions, a measure of the search-time would be equivalent to the average distance that a robot would have to travel on its curve to find the target. A given robot may find the target anywhere along the curve within the segment between its position of deployment and that of the next adjacent (clockwise) robot. Since the target can move along any direction outward from the LKP one can assume a uniform distribution for the likely position of the target along that segment. Consequently, averaging over all the possibilities, the expected distance that a robot would have to travel to 'find' the target, if the target does in fact reside on that segment, would be half the length of this segment (i.e., half the iso-probability curve-length between the given robot's position and its clockwise neighbour). Although this expected (average) distance measure could be divided by the assumed constant maximum speed of the robot to get an actual time value, since all robots are assumed to move at the same speed, such division would be inconsequential as it would simply serve as a scale-factor. Thus, for convenience, the search-time measure is kept as a distance value. Recall that the intent here is to construct a representative measure, not the actual search-time (which cannot be known in advance anyway). Computing the search-times in this manner yields a search-time value for each curve individually. Since the target could reside on any of the curves as well, the average of these individual values is taken to, thereby, yield the overall average search-time measure as:

$$A_T = \frac{1}{N_{curves}} \sum_{p=1}^{N_{curves}} \frac{1}{2} \frac{C_p}{N_{r,p}}, \tag{4.6}$$

where C_p is the total circumference of curve $p \in [1, N_{curves}]$.

In order to allow for the combination of the search-time attribute measure with others, this measure is normalized to lie in the range 1 to 2 using linear scaling between the minimum and maximum search-times. The minimum search-time measure is taken to be that corresponding to a deployment of all N_r available robots on a single curve (which, according to the curve-placement procedure as demonstrated in Fig. 4.3, would be located at the 50% cumulative probability position). This would minimize the distance between adjacent robots on a curve, and would thereby minimize the search-time measure as well. Conversely, the maximum search-time measure is taken to be that corresponding to a deployment of the robots on the maximum possible number of curves, given values for the parameters R_c and D_c. This deployment would require the same number of robots to be distributed among more curves, thereby minimizing the number of robots per curve, and, in turn, maximizing the average distance between adjacent robots on a curve. This would result in the largest possible search-time value for a given number of robots.

In order to conduct the normalization, the minimum search-time measure (most desirable) is given a metric value of 1, while the maximum search-time measure (least desirable) is given a metric value of 2. The final result is a normalized search-time metric, $\tilde{A}_T$, that can be expected to behave in a similar manner as the actual search-time, as N_{curves} is varied. Although it is common to normalize values in the range 0 to 1, the alternative range of 1 to 2 allows for the computation of ratios of objective function values composed of the search-time, success-rate, and return-time metrics without resulting in a possible division by 0; yet a normalization range of magnitude 1 is still maintained. The need for computing such ratios will become evident in the final section of this Chapter (Section 4.2) when objective function values obtained from alternative optimization approaches are compared.

4.1.1.3 Success-Rate Metric

A success-rate metric can be devised to reflect the probability of finding the target for a given selection of curves, using the target-location PDF and knowledge of the amount of coverage provided by each robot's sensors. This amount of coverage is represented by the sensing radius of a robot, and corresponds to a measure of a robot's target-detection effectiveness. For the purposes of formulating the success-rate metric, it is assumed that each robot has a sensing range of r_s from its center, and follows a curve, p, located a distance $L_{p,i}$ from the LKP on a ray, $i \in [1, N_{rays}]$. Since the robots follow their curves in the alternative search scenario described above for the search-time metric (Section 4.1.1.2), a measure of the probability, $\rho_{p,i}$, of search-success for that curve, p, from the perspective of that ray, i, can be taken to be the probability area under the target-location PDF on ray i that would be swept through by that robot, as a result of its r_s sensing range, as it travels along its curve. This probability area, from the perspective of ray i, is shown diagrammatically in Fig. 4.4 for the case of a normal target-location PDF. Mathematically, then, this probability value would be computed as:

$$\rho_{p,i} = \int_{L_{p,i}-r_s}^{L_{p,i}+r_s} p(r, t, \theta_i) dr. \tag{4.7}$$

Since, in general, the position, $L_{p,i}$, of a curve would vary from ray to ray due to different PDF-scaling applied to different rays to account for other influences, the corresponding probability area that is computed would also vary. Thus, the average of the probability values computed for all the rays being used is taken, yielding a measure of the success-rate for a given iso-probability curve, p. Summing this average over all curves, $p \in [1, N_{curves}]$, gives the final success-rate measure as:

$$A_S = \sum_{p=1}^{N_{curves}} \left(\frac{1}{N_{rays}} \sum_{i=1}^{N_{rays}} \rho_{p,i} \right). \tag{4.8}$$

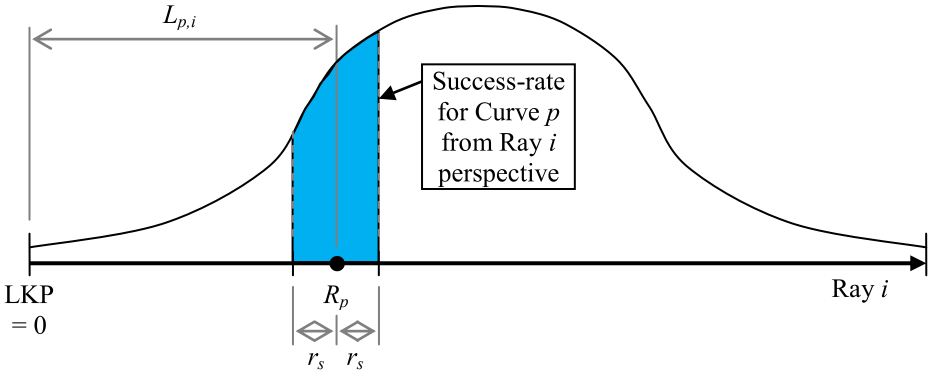


Figure 4.4. Computing success-rate for a curve on a ray.

The success-rate determined from Eq. 4.8 is also normalized to lie in the range 1 to 2 through linear scaling between the maximum and minimum success-rates. The maximum success-rate (most-desirable) is given a value of 1, while the minimum success-rate (least-desirable) is given a value of 2. It is clear that the maximum success-rate occurs when the number of iso-probability curves is maximized, since this results in the most area swept under the target-location PDF per ray. On the other hand, a single curve (which, again, would be positioned at the highest target-location probability density position – corresponding to the 50% cumulative probability position for a normal distribution) would give the minimum success-rate value. The resulting normalized metric, $\tilde{A}_S$, can be expected to behave proportionally to the actual probability of search-success for the selected value of the number of curves, N_{curves}.

4.1.1.4 Return-Time Metric

At any given point in time when an optimal deployment is being evaluated, the robots would, in general, be dispersed in the search area and would not be located on the new set of iso-probability curves that need to be implemented. Each robot would, thus, have to travel some distance to reach its assigned position on a curve. The sooner the robots reach these positions, the sooner they can resume the search for the target. The return-time measure, therefore, is a measure of effort required by the robots to attain their assigned deployment positions on their assigned curves.

Each iteration of the optimization conducted to find an optimal deployment solution will involve successive consideration of a specific number and positions of iso-probability curves by the optimization search engine. The strategy that has been adopted (see Fig. 4.2) is to assign robots to curves based on a preferred robot distribution among the curves, and to, then, use the shortest paths from each robot to its assigned curve to determine the corresponding deployment positions. However, even though the number of robots required per curve can be computed, there is still ambiguity, and, therefore, multiple choices, as to which robot should be assigned to which curve, where one would like to find the best possible assignment.

In particular, an optimal assignment of robots to the given selection of curves under consideration must be found that minimizes the return-time measure based on the use of shortest return-paths, and, in so doing, also finds the corresponding optimal deployment positions on those curves. Namely, a secondary optimization must be conducted to optimize assignments, *within* each loop of the main optimization underway that is trying to optimize the number of curves. This produces a nested optimization structure for solving the optimal deployment problem. Since the objective of this secondary optimization is to minimize return-time, its objective function must be a quantitative measure of return-time. This quantitative measure must be a function of the decision variables for the optimization, which are some form of variables that describe the assignment of robots to curves. The detailed formulation of this optimization problem will be presented below, in Section 4.1.1.5. In the following, however, a means of quantifying return-time is presented.

At any given iteration of the overall robot deployment (main) optimization, a specific number of iso-probability curves, N_{curves}, currently under consideration will be available, and the resource-requirement, $N_{r,p}$, per curve, $p \in [1, N_{curves}]$, can be computed by the procedure described in Section 4.1.1.1. For any specified assignment of robots to curves, a return-time measure can be computed as follows. Let c_{ip} be the time that would be required by robot i to reach its assigned curve, p, via the shortest path. As a conservative measure, a deployment is considered to have been achieved only when *all* robots have reached their assigned deployment positions. Thus, the return-time measure can be expressed as:

$$A_R = max\{c_{ip}\}, \forall i \in [1, N_r], \forall p \in [1, N_{curves}] \tag{4.9}$$

The return-time measure can also be normalized to lie in the range 1 to 2 by scaling its value linearly between the best- and worst-case optimal return-time values. Firstly, one may note that the maximum number of iso-probability curves that can be defined, $N_{curves,max}$, is a finite value dependent upon the number of robots available, the resource-requirements per curve, and the parameters R_c and D_c. Given the possible values for the number of curves, $N_{curves} \in [1, N_{curves,max}]$, there will exist a particular number of curves (indicated by $N_{curves,LO}$) in this range that yields the lowest possible optimal return-time value among all the other values of N_{curves}, as well as another number of curves (indicated by $N_{curves,HI}$) that gives the highest possible optimal return-time value. The optimal return-time value corresponding to $N_{curves,LO}$ (most desirable) is assigned a metric value of 1, while that corresponding to $N_{curves,HI}$ (least desirable) is assigned a metric value of 2.

The final normalized return-time metric, $\tilde{A}_R$, represents the influence of the selected number of curves on the effort and efficiency with which the search robots can attain their assigned deployment positions. It can be expected to behave comparably to the actual maximum robot return-time for any given value of the number of curves, N_{curves}. Return-time, therefore, is one of the attributes of the search that is evaluated in each iteration of the robot deployment optimization procedure, and its value (along with those of other attributes) guides the search engine.

4.1.1.5 Overall Optimization Formulation for Search-Effort Allocation

Returning to the solution approach outlined in Fig. 4.2, there are two distinct stages involved. Firstly, the optimal number of iso-probability curves and the positions of these curves need to be determined. Secondly, robots must be optimally assigned to these curves, from which the corresponding deployment positions can be obtained. However, in order to implement this approach, and, in particular, to address the interdependency that exists between the curves selected and the optimal robot-to-curve assignments that can be obtained, a nested optimization structure is proposed. This structure was summarized in the upper-half of Fig. 2.5 in the method overview in Chapter 2, and is included here in Fig. 4.5 below in order to draw the parallels between the general optimization structure and the solution approach proposed herein.

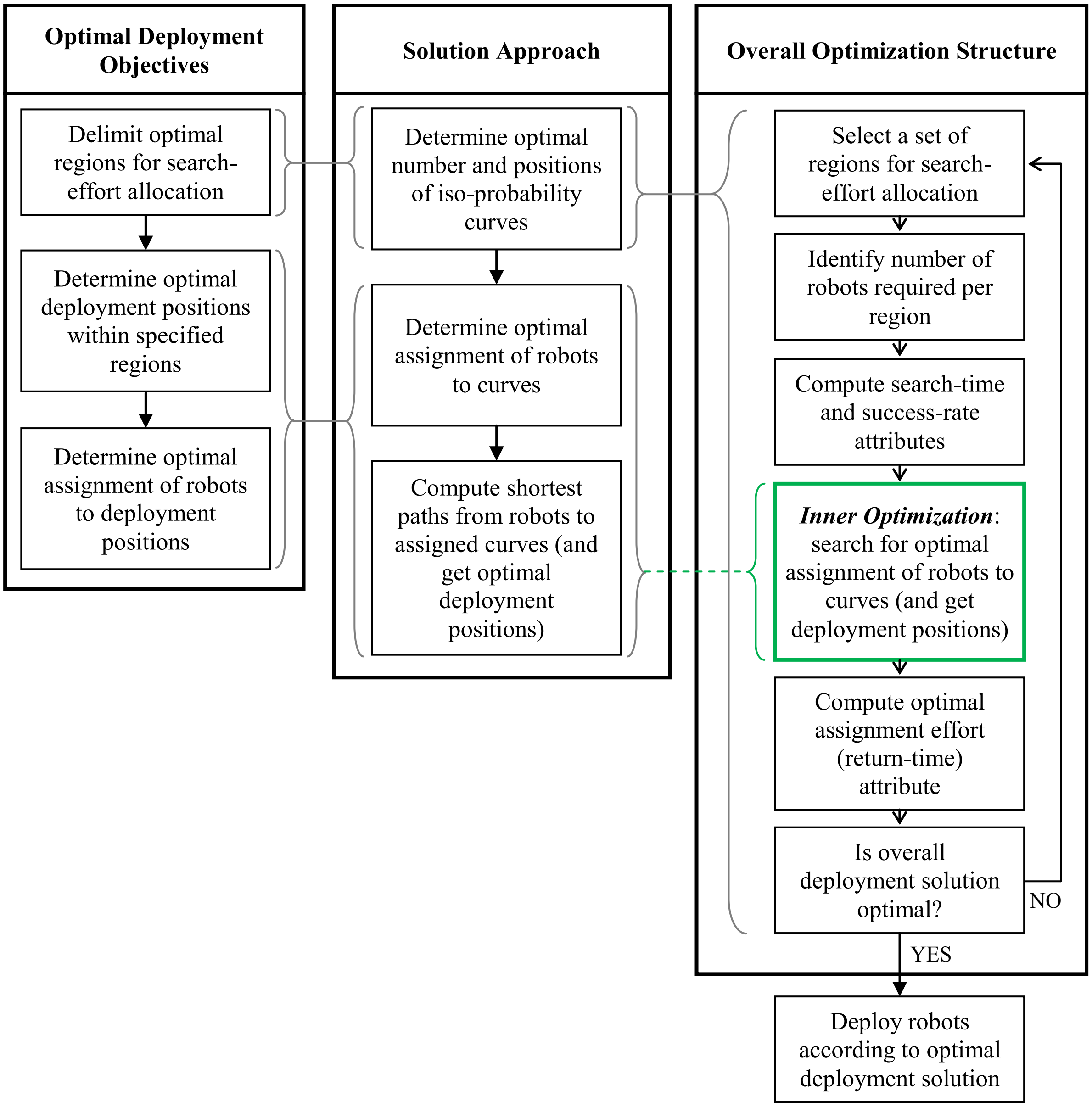

Optimal Deployment Objectives
Delimit optimal regions for search-effort allocation
Determine optimal deployment positions within specified regions
Determine optimal assignment of robots to deployment positions
Solution Approach
Determine optimal number and positions of iso-probability curves
Determine optimal assignment of robots to curves
Compute shortest paths from robots to assigned curves (and get optimal deployment positions)
Overall Optimization Structure
Select a set of regions for search-effort allocation
Identify number of robots required per region
Compute search-time and success-rate attributes
Inner Optimization: search for optimal assignment of robots to curves (and get deployment positions)
Compute optimal assignment effort (return-time) attribute
Is overall deployment solution optimal?
NO
YES
Deploy robots according to optimal deployment solution

Figure 4.5. Overall optimization approach used to solve the optimal robot deployment problem.

The nested optimization structure is evident from the flowchart of the overall optimization shown in Fig. 4.5. In this structure, rather than treating the optimal deployment positions as additional decision variables to be considered along with the number of curves, N_{curves}, in a single optimization, a secondary, ‘inner’ optimization performs the search to find optimal deployment positions based on optimally assigning robots to curves. Such a structure is necessary because of the dependency of the decision variables for the assignment problem on the value of N_{curves}. Since robots are assigned to curves rather than directly to positions, one cannot know how many decision variables pertaining to the assignment problem there must be without first knowing how many curves to use. Consequently, this nested optimization structure is used. This approach, in fact, helps to simplify the problem and allows feasible on-line computation.

The main objective of the overall optimization, of course, is to select the optimal deployment positions for all robots within the search area. This is, still, the main expected output from the overall optimization. The by-products of this optimization are the best regions for search-effort allocation (i.e., the optimal number and positions of iso-probability curves to use). In the process of determining the optimal deployment positions, three attributes of the search are computed, namely, search-time, success-rate, and return-time, and these are combined to form an objective function that evaluates the 'goodness' of any given choice of robot deployment positions.

The overall optimization begins each iteration with a selection of a particular number and positions of iso-probability curves from the corresponding search space of valid curves. At this point, sufficient information is available to compute the search-time and success-rate metrics. With a set of curves specified, this iteration of the main optimization continues with the determination of the optimal deployment positions on those curves and the optimal assignment of robots to those positions. The approach employed to accomplish this, as discussed earlier, is to assign robots optimally to the given set of iso-probability curves based on a preferred distribution of robots among those curves, and to use the shortest paths to determine the deployment positions corresponding to that optimal assignment. This entails a second, 'inner' optimization to be carried out for that particular iteration of the overall, 'outer' optimization. This inner optimization determines an assignment scheme of robots to curves that minimizes the return-time metric.

Upon determining the optimal assignment scheme, the resulting values for the search-time, success-rate, and return-time metrics are used in the overall objective function of the outer optimization loop to evaluate the resulting overall solution for that iteration. The optimization search engine uses this value to determine how to proceed (i.e., whether the current solution is optimal or whether another iteration is required). In this manner, the nested optimization structure computes the optimal robot deployment solution through the evaluation and use of the three key attributes of the search that are proposed in this Thesis (i.e., search-time, success-rate, and return-time).

The overall optimization can, thus, be formulated as follows. Let us assume that at some time $t = T$, the N_r available robots must be optimally deployed within the search area. The main optimization loop involves a single decision variable – the number of iso-probability curves, N_{curves}, to use for the deployment. Based on the approach used to specify curves, the positions of the curves can be determined based on the given values for the parameters R_c and D_c, once the number of curves is indicated.

The three normalized metrics that were formulated above to represent the three important attributes of the search are combined to produce a single objective function for the overall optimization of search-effort allocation. In particular, the objective function is a minimization of the weighted sum of these metrics. However, since both return-time and search-time are normalized time-based metrics, they can be combined into a single metric to which a single weighting can be assigned. This combined time-measure, referred to henceforth as the modified-search-time metric, is constructed by summing the normalized search-time and normalized return-time metrics, and assigning each of them equal weights so as to ensure that the combined result still lies in the range 1 to 2.

The resulting objective function for the overall optimization becomes:

$$Minimize: Z_1 = f(N_{curves}) = w_T\big(0.5\tilde{A}_T + 0.5\tilde{A}_R\big) + w_S\tilde{A}_S. \tag{4.10}$$

Above, w_T and w_S are the weightings for the modified-search-time and success-rate metrics, respectively. These weightings must be specified by the user, and would represent the user's preference on the relative importance of the speed with which the search is to be conducted (w_T) and the thoroughness of the search (w_S), for the particular WiSAR scenario at hand. These weightings can also be changed, on-line, as the search progresses, if the user wishes to change the emphasis of the search. In most cases, one would prefer to give equal importance to modified-search-time and success-rate. Thus, 'default' values for w_T and w_S can be 0.5 and 0.5, respectively.

The constraints for this main (outer) optimization loop are the upper and lower bounds on the single decision variable, N_{curves}. The lower bound on N_{curves} is one curve, but the upper bound depends on two considerations. First, given values for the parameters R_c and D_c, the maximum number of curves is limited by the available space in the range [R_{min}, R_{max}] defined earlier.

Secondly, the available number of robots, N_r, must be allocated to each curve following the balanced robot distribution method described in Section 4.1.1.1. Thus, an iterative process must be used to determine the upper bound, $N_{curves,max}$, where curves are added symmetrically about R_c until any one of the above two conditions cannot be satisfied. The resulting constraint for the main optimization is:

$$1 \leq N_{curves} \leq N_{curves\,,max} \,. \tag{4.11}$$

Since the optimization engine is able to choose as few as one curve, and as many as the $N_{curves,max}$ value that is derived to accommodate the number of robots available, a feasible solution to this optimization will always be possible. Moreover, as this is a single-variable optimization, a non-derivative-based search technique, such as a Golden-Section Search, conducted within the range 1 to $N_{curves,max}$, can be used. Since such methods would rely only on function evaluations, there is no need for concern as to whether or not the objective function in Eq. 4.10 is differentiable.

With each iteration of this outer optimization loop, a particular value of the N_{curves} decision variable will be specified, at which the objective function must be evaluated. Although $\tilde{A}_T$ and $\tilde{A}_S$ can be computed simply by knowing the value of N_{curves} and the resource requirements per curve, the value of $\tilde{A}_R$ cannot. The normalized return-time metric, $\tilde{A}_R$, must be computed through a second 'inner' optimization that optimally assigns robots to the N_{curves} curves specified. The objective is to select an assignment of robots to curves the minimizes the $\tilde{A}_R$ metric. In particular, for any given set of N_{curves} iso-probability curves under consideration in the outer optimization loop, a corresponding return-time metric, $\tilde{A}_R$, would be the optimal value of the objective function of an optimal assignment problem that seeks to determine the best assignment of the N_r robots to these N_{curves} curves. The current position of each robot and the resource-requirements, $N_{r,p}$, per curve $p \in [1, N_{curves}]$ (computed as described in Section 4.1.1.1), would be the constraints. The robots are assumed to take the shortest path to get to their newly assigned curves, and the point at which each robot would reach its curve in this manner becomes the corresponding optimal deployment position for the robot.

For this inner optimization, then, let the binary decision variable $x_{ipj} \in [0, 1]$, represent the yes/no decision to assign Robot $i \in [1, N_r]$ to the j^{th} deployment position, $j \in [1, N_{r,p}]$, on Curve $p \in [1, N_{curves}]$. The index j is used simply as a counter to ensure that the correct number of robots is assigned to each curve. The actual coordinates themselves would depend on the final assignment of each robot to a curve, and the shortest path between them. The time taken by the robot to travel along this path can be computed, and this time is represented by the cost parameter c_{ipj} (this value is the same as 'c_{ip}' in Eq. 4.9), which corresponds to the time it would take robot i to travel to the j^{th} deployment position on iso-probability curve p. Since the total number of deployment positions must equal the total number of robots available, a square cost-matrix results.

One may recall that, as a conservative estimate, a deployment is considered to have been achieved only after *all* robots have reached their assigned positions. Thus, the objective of the optimal assignment problem is to minimize the time taken by the *last* robot to reach its deployment position on its assigned iso-probability curve. The return-time measure of Eq. 4.9 can be re-expressed in terms of the binary decision variables defined above, to produce the objective function for the optimal assignment problem as follows:

$$Minimize\colon Z_2 = f\left(x_{ipj}\right) = \max_{1 \leq i \leq N_r} \left\{ \sum_{p=1}^{N_{curves}} \sum_{j=1}^{N_{r,p}} \left[x_{ipj} \cdot c_{ipj}\right] \right\}. \qquad (4.12)$$

The following five constraints are required for this optimization. First, each robot must be assigned to only one deployment positions (Eq. 4.13). Second, each deployment position on each curve must have a single, unique robot assigned to it so that the resource requirement for each curve is met (Eq. 4.14). Third, the total number of assignments must equal the total number of available robots (Eq. 4.15). Fourth, the decision variables must be integers (Eq. 4.16), and, fifth, they can only take on the values 0 or 1 (Eq. 4.17).

$$\sum_{p=1}^{N_{curves}} \sum_{j=1}^{N_{r,p}} x_{ipj} = 1, \forall i \in [1, N_r], \qquad (4.13)$$

$$\sum_{i=1}^{N_r} x_{ipj} = 1, \forall p \in [1, N_{curves}];\ \forall j \in \left[1, N_{r,p}\right], \qquad (4.14)$$

$$\sum_{i=1}^{N_r} \sum_{p=1}^{N_{curves}} \sum_{j=1}^{N_{r,p}} x_{ipj} = N_r, \tag{4.15}$$

$$x_{ipj} \in \mathbb{Z}, \forall i \in [1, N_r];\ \forall p \in [1, N_{curves}];\ \forall j \in \left[1, N_{r,p}\right], \tag{4.16}$$

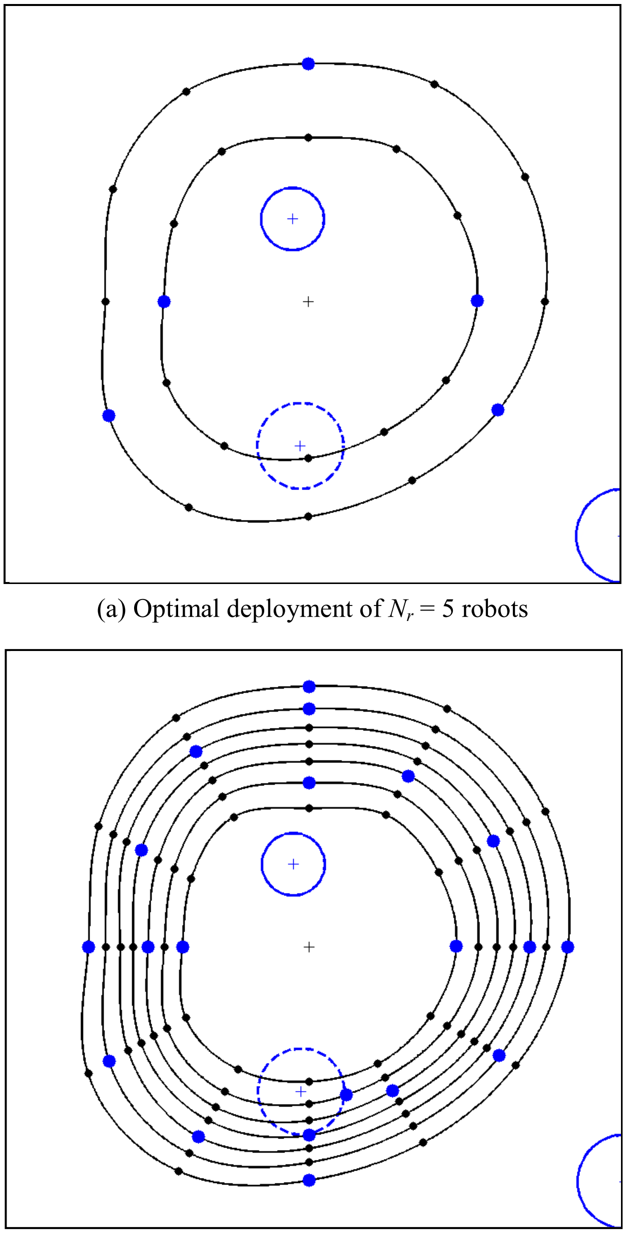
$$x_{ipj} \in [1,0], \forall i \in [1, N_r];\ \forall p \in [1, N_{curves}];\ \forall j \in \left[1, N_{r,p}\right]. \tag{4.17}$$

The formulation of this secondary optimization problem involves binary decision variables with a square cost-matrix, linear constraints, and a Min-Max objective function of a linear sum. This produces a standard Linear Bottleneck Assignment Problem (LBAP) [133]. The LBAP can be solved efficiently via the Threshold Algorithm presented in [133]. In this particular implementation used to solve the LBAP, the threshold algorithm is interpreted as a set of matrix operations applied to the cost-matrix of the LBAP. It iteratively cycles between two phases. The first selects an upper cost threshold value, c_1^*, and a lower cost threshold value, c_0^*, and constructs a threshold matrix, $\overline{C}$, with elements, $\overline{c}_{ipj}$, where $\overline{c}_{ipj} = c_{ipj}$ if $c_0^* < c_{ipj} < c_1^*$ in the original cost matrix, and $\overline{c}_{ipj} = 0$ otherwise. The second phase checks whether a '0'-cost assignment exists in $\overline{C}$, which is defined as that assignment of the robots, Rb_i, $i \in [1, N_r]$, to the positions p_j, $p \in [1, N_{curves}]$, $j \in [1, N_{r,p}]$, such that the objective function value, $Z_2 = f(x_{ipj})$, evaluated using the corresponding cost values $\overline{c}_{ipj}$ from the threshold matrix (instead of c_{ipj} from the original cost-matrix), is '0'. This entails making assignments corresponding only to those elements in the threshold matrix that are '0'.

If a '0'-cost assignment is found, the two-phase cycle repeats using a smaller value for c_1^* equal to the median of the non-zero values in $\overline{C}$. If none is found, the two-phase cycle still repeats, but with a larger value for c_0^*, equal to, again, the median of the non-zero values in $\overline{C}$. Eventually, $c_0^* = c_1^*$, and the c_1^* value at that point is the smallest upper threshold value for which a '0'-cost assignment is possible. This '0'-cost assignment represents the optimal solution to the LBAP.

The optimal objective function value, Z_2^*, yields the minimized maximum robot-return-time, A_R, which is, subsequently, normalized, as explained in Section 4.1.1.4, to produce $\tilde{A}_R$. This value is, then, used in Eq. 4.10 to continue with the current iteration of the outer optimization loop. Once the outer optimization loop determines that the optimal solution has been found and ceases, the resulting optimal decision variable value will yield the optimal number and positions of iso-probability curves to use for the deployment. These curves are indicative of the regions where search-effort should be focused. Furthermore, the normalized return-time metric that will have been computed in order to evaluate the objective function value for this optimal solution will have, associated with it, the solution of the optimal assignment of the search resources to this optimal set of curves. This assignment will indicate which robots are assigned to which curves as well as the destination position coordinates for each robot on its assigned curve. The result of this overall process, therefore, fully specifies the optimal robot deployment to be used for the time point $t = T$ in question.

Figure 4.6 shows two examples of an optimal deployment, both for time $t = 1800$ s. In Fig. 4.6a, a total of $N_r = 5$ robots are available. Performing the optimization, as described above, under equal weighting conditions (i.e., $w_T = 0.5$ and $w_S = 0.5$) and with $R_c = 0.50$ and $D_c = 0.40$, it is found that the optimal set of iso-probability curves consists of $N_{curves} = 2$, with $R_1 = 30\%$ and $R_2 = 70\%$. The resource requirements per curve are computed to be $N_{r,30\%} = 2$ and $N_{r,70\%} = 3$. These two optimal iso-probability curves are shown in Fig. 4.6a with the appropriate number of robots placed on each curve at their assigned deployment positions. The example shown in Fig. 4.6b involves $N_r = 20$ robots, with $w_T = 0.5$, $w_S = 0.5$, $R_c = 0.50$, and $D_c = 0.10$. The optimal deployment solution is found to require $N_{curves} = 7$, where $R_1 = 20\%$, $R_2 = 30\%$, $R_3 = 40\%$, $R_4 = 50\%$, $R_5 = 60\%$, $R_6 = 70\%$, and $R_7 = 80\%$. The per-curve resource requirements are computed to be $N_{r,1} = 2$, $N_{r,2} = 2$, $N_{r,3} = 3$, $N_{r,4} = 3$, $N_{r,5} = 3$, $N_{r,6} = 3$, and $N_{r,7} = 4$. The robots are shown in Fig. 4.6b, allocated in appropriate numbers among the optimal iso-probability curves and each placed at its assigned deployment position on the curves.



(a) Optimal deployment of $N_r = 5$ robots

(b) Optimal deployment of $N_r = 20$ robots

Figure 4.6. Example optimal robot deployments.

4.1.2 Considerations for Optimal Re-Deployment

The robot deployment method described above is a general one, and can be applied any time a deployment is required. Indeed, for a dynamic problem scenario such as robotic WiSAR, deployment may need to be conducted several times throughout the search. As noted earlier, the proposed method differentiates between an initial deployment and a re-deployment. During the ongoing search process, the target-location prediction information changes as it is updated and propagated with time. Moreover, new terrain information may be made available (e.g., modified terrain topography, discovery of *a priori* unknown obstacles), and/or clues may be found, both of which change the target-location prediction as well. Thus, the initial deployment implemented at the start of the search may no longer be optimal and the robots may need to be re-distributed within the search area by computing and implementing a new optimal deployment solution. This process is referred to as re-deployment.

The robot deployment method proposed in this Thesis requires regularly re-checking the optimality of the latest deployment solution throughout the search. Hence, from the moment the search begins, a new deployment solution is computed at pre-determined time-intervals. This time-interval is the same as the time period, Δt_{prop}, between iso-probability curve propagations, so that with each propagation, the optimal set of curves and assignment of robots to those curves is always available in order for an optimality check to be carried out. In the time between these planned optimality-checking events, the robots use the existing set of curves to conduct the search, while a parallel computing process performs the computations required to determine the optimal re-deployment solution for the future planned optimality-checking time-point, T_f, using the method presented in the sections above. The length of this time-interval, Δt_{prop}, must be as short as possible, thereby ensuring that any non-optimality in the used curves is detected and corrected expeditiously. However, the interval must also be long enough to allow sufficient time for the available computational resources to carry out the re-deployment optimization, compute the propagated set of curves, and plan the robot search-paths.

An additional concern in re-deployment is the effort required in re-distributing the search resources within the search area. Time spent by the robots to reach newly-assigned deployment positions is time taken away from conducting the search as per appropriate search-paths computed through the robot-path planning method. Thus, although a new optimal deployment solution, different from the one currently being used, may be determined during re-deployment computation, it must still be associated with sufficient benefit to the search to warrant its implementation. Therefore, once the time approaches T_f, it remains a question whether the *new* deployment solution, determined for time T_f, should be implemented, or if the existing deployment solution should continue to be used. This requires a decision-making method to decide when the current deployment is sufficiently sub-optimal to warrant implementing a new deployment solution.

Such a decision-making procedure can be based on the use of iso-probability curves. Since iso-probability curves determine search-effort allocation (as well as the robot search-paths, as will be seen in Chapter 5), curve-selection has the most significant impact on the optimality of the search. Therefore, the benefit of re-deployment can be gauged using solely the iso-probability curves. The objective function given in Eq. 4.10 that governs the overall (outer) optimization for optimal robot deployment serves as a suitable measure of the effectiveness of the subsequent search for a given set of curves and deployment positions, and is used here to decide whether to implement a re-deployment solution. The flowchart in Figure 4.7 summarizes this decision-making process.

The approach used is to compare the values of the objective function given by Eq. 4.10 for both options: the new deployment solution computed for time $t = T_f$, and the existing deployment solution being used. The new deployment solution is obtained by running the optimal deployment method described above for time $t = T_f$ to determine the (possibly) new optimal set of curves and the new assignment of robots to those curves. The resulting optimal solution yields a value, Z_1^*, for the main objective function, Eq. 4.10. However, in order to compare this to the existing deployment solution, the existing set of curves, $N_{curves}^{[cur]}$, are first propagated to time $t = T_f$ (so that they can be fairly compared with the new deployment solution), and these propagated curves are used to compute a corresponding value for each of the three normalized metrics. This allows for a value, $Z_1^{[cur]}$, to be computed for the main objective function, Eq. 4.10, corresponding to the existing set of $N_{curves}^{[cur]}$ curves. Given values for Z_1^* and $Z_1^{[cur]}$, the percentage difference between these two is computed, and compared to a user-specified threshold value, $\Delta Z_{1,max}$. If the threshold is surpassed, the re-deployment solution is implemented. Otherwise, the existing deployment solution continues to be used. The value for this threshold can be set according to the user's preference. A small threshold would mean a more 'sensitive' decision-making process that is more likely to indicate sub-optimality and trigger a re-deployment. Conversely, a larger threshold would result in a more passive system that would, nonetheless, tend to avoid having too many disruptions to the search that can occur from frequent re-deployments.

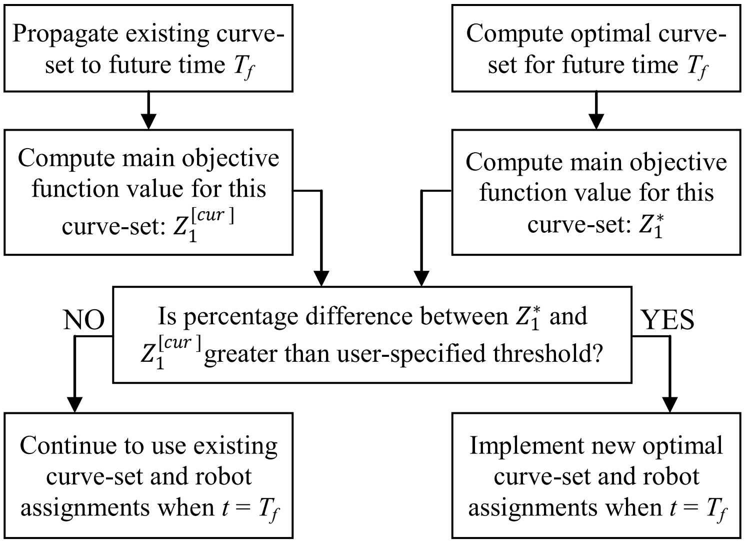

Propagate existing curve-set to future time T_f
Compute optimal curve-set for future time T_f
Compute main objective function value for this curve-set: $Z_1^{[cur]}$
Compute main objective function value for this curve-set: Z_1^*
NO
Is percentage difference between Z_1^* and $Z_1^{[cur]}$ greater than user-specified threshold?
YES
Continue to use existing curve-set and robot assignments when $t = T_f$
Implement new optimal curve-set and robot assignments when $t = T_f$

Figure 4.7. Procedure to determine necessity for re-deployment.

In addition to these regular, planned optimality-checking events, a re-deployment may also be triggered by the discovery of a clue. A clue represents a critical piece of information about the target that cannot be ignored. As such, any clue found is immediately taken to be the new LKP and a new set of iso-probability curves is computed and implemented. Due to this significant change in the probabilistic target-location information, a re-deployment *must* be conducted, and the decision-making process described above is not necessary in this situation. Thus, a new optimal robot deployment is computed and implemented right away.

Figure 4.8 gives a visual representation of an example re-deployment scenario triggered due to a clue-find. Here, $N_r = 5$ robots are available, and the initial deployment solution dictated two iso-probability curves, with two robots assigned to the inner curve ($R_1 = 30\%$) and three robots assigned to the outer curve ($R_2 = 70\%$). At some later time, $t = T_f$, a clue is found at the position (x_{clu}, y_{clu}), and a re-deployment is computed. As shown in Fig. 4.8, the robots (blue dots) are re-assigned (from the original two curves) to new $N_{curves} = 3$ curves, where each curve, $p \in [1, 3]$, are determined to have resource-requirements, $N_{r,p}$, of: $N_{r,1} = 1$, $N_{r,2} = 2$, and $N_{r,3} = 2$, respectively. The magenta lines emanating from the robots indicate their re-location from the earlier set of 2 curves (dashed, magenta lines) to the new set of 3 curves (solid, black lines). The magenta '×'s indicate the new robot deployment positions.

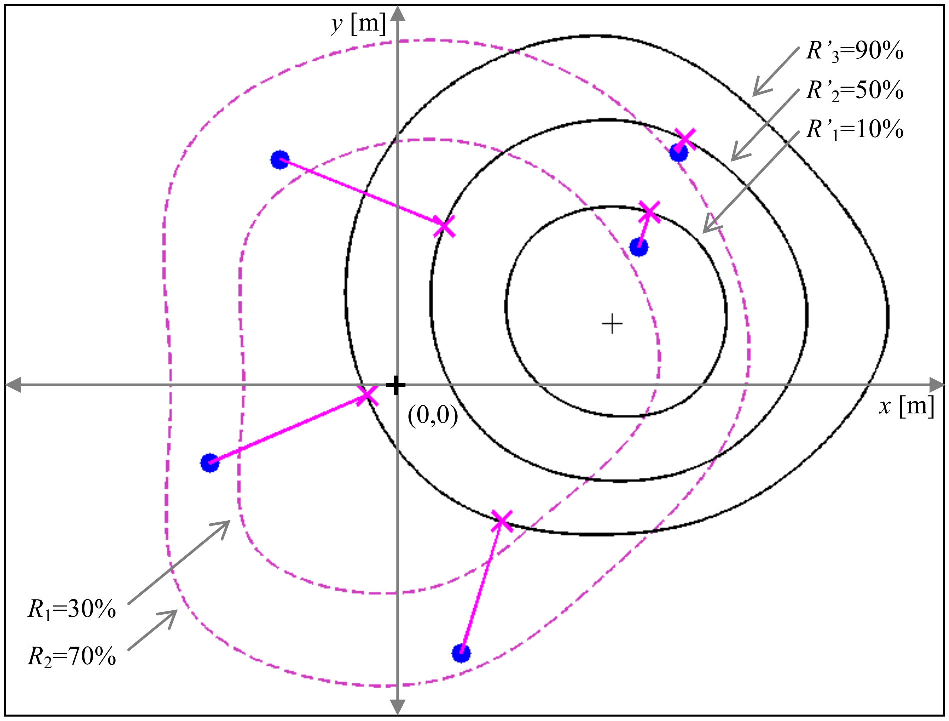


Figure 4.8. An example robot re-deployment ("R_p" = positions of original iso-probability curves, $p \in [1, 2]$; "R_p'" = positions of new iso-probability curves, $p \in [1, 3]$).

When the normal distribution assumption is used for the target-location PDF, as was done for the purposes of simulations in this Thesis, construction of a new set of iso-probability curves due to re-deployment after a clue-find would involve the following modification. Let us assume that due to the discovery of a clue, the LKP changes from its previous position, (x_0, y_0), to some new position, (x_{clu}, y_{clu}). The underlying normal target-location PDF, $p(r, t, \theta_i)$, for each ray θ_i, $i \in [1, N_{rays}]$, is still computed using Eq. 3.5, but with the time parameter, t, adjusted as follows:

$$p(r,t,\theta_i) = \frac{1}{\sqrt{2\pi}\left(t - \frac{\Delta_{LKP}}{v_{max}}\right)\sigma_v} exp\left[-\frac{\left(r - \mu_v\left(t - \frac{\Delta_{LKP}}{v_{max}}\right)\right)^2}{2\left(t - \frac{\Delta_{LKP}}{v_{max}}\right)^2 \sigma_v^2}\right], \forall i \in \left[1, N_{rays}\right], \tag{4.18}$$

where Δ_{LKP} is the distance magnitude $|(x_{clu}, y_{clu}) - (x_0, y_0)|$, and v_{max} is the maximum average target speed, which corresponds to the $\mu_v + 3\sigma_v$ speed on the nominal mean target-speed PDF.

Since the variable, r, in Eq. 3.5 is a relative measure that indicates distance along any ray relative to the most current LKP, the time parameter must also be made relative to the LKP in the case of a re-deployment due to a clue-find. The above adjustment to the time parameter in Eq. 4.18

(compared to using just t as in Eq. 3.5) accomplishes this, and corresponds to the conservative assumption that the target reached the new LKP from the old one via a straight-line path at its maximum speed. This assumption maximizes the outward travel of the target-location PDF resulting from Eq. 4.18 along any given ray, and, thereby, ensures that the curves still bound the area containing the target. The new iso-probability curves that are constructed due to re-deployment are, then, determined based on this modified nominal target-location PDF. The other influences on target-prediction are subsequently applied as usual using this new nominal target-location PDF for each ray.

4.1.3 Considerations for Optimal Initial Deployment

At the onset of a search, it is necessary to conduct *initial* robot deployment, where robots are brought into the search area at the site of the LKP and are deployed for the first time before starting the search. In order to conduct this deployment, an optimal solution must be computed. However, this initial optimal deployment is, in fact, a simplified form of the general robot deployment method presented above in Section 4.1.1.

In particular, in the main (outer) optimization loop, determination of the optimal set of iso-probability curves does not require computation of the normalized return-time metric, $\tilde{A}_R$. This is because the robots do not employ any existing set of curves at the start of a search. They are transported to the search area, and it is assumed that robots are 'dropped' directly onto their initial optimal deployment positions. As a result, there is no expenditure of travel-time (i.e., of effort) by the robots to get onto their curves that needs to be considered, eliminating the need for computing a return-time metric. This removes the inner optimization that computes optimal robot-to-curve assignments. Instead, the resource requirements per curve for a balanced search-effort distribution are determined (Section 4.1.1.1), and the overall robot deployment optimization reduces to a single-loop optimization, as illustrated in the overall optimization flowchart in Figure 4.9 (modified from Fig. 4.5). The objective function for this optimization (given in Eq. 4.19, which is modified from Eq. 4.10) is based only on the search-time and success-rate metrics:

$$Minimize\colon Z_1 = f(N_{curves}) = w_T\tilde{A}_T + w_S\tilde{A}_S. \qquad (4.19)$$

When initially deploying the robots at the start of the search, the proposed method also requires that the assigned deployment positions on each curve be uniformly distributed along the circumference of the curve. This layout may help to maximize the efficiency of the search, as it minimizes overlap of searched area between adjacent robots on a curve.

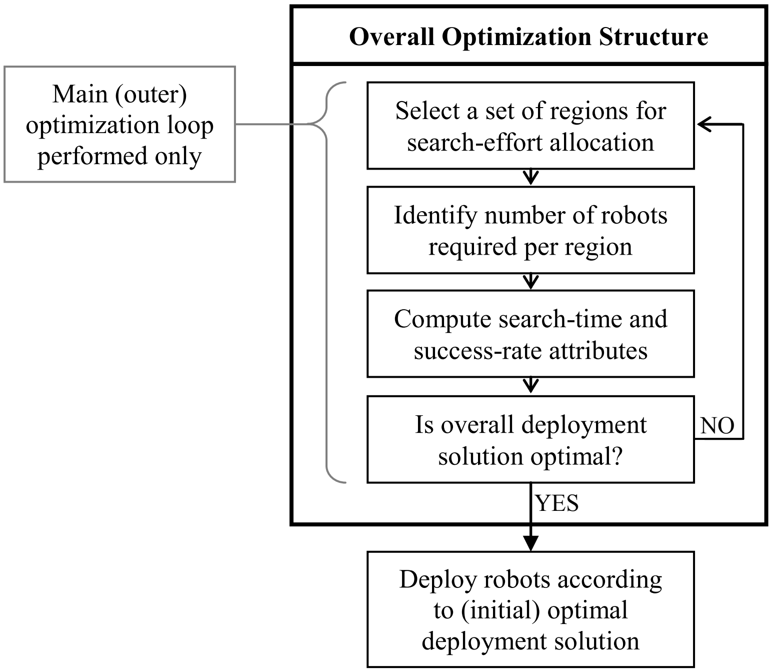

Overall Optimization Structure
Main (outer) optimization loop performed only
Select a set of regions for search-effort allocation
Identify number of robots required per region
Compute search-time and success-rate attributes
Is overall deployment solution optimal?
NO
YES
Deploy robots according to (initial) optimal deployment solution

Figure 4.9. Modified overall optimization approach used for optimal initial robot deployment.

4.2 Analysis of the Benefit of Optimal Re-Deployment

The use of the nested optimization approach for any general re-deployment optimization during an ongoing search provides an optimal solution to the problem of search-effort allocation while maintaining on-line feasibility. To verify that tangible benefit to the search does, indeed, exist by using such an approach, simulations were conducted comparing a random selection of iso-probability curves during a re-deployment scenario with an optimal one obtained from the proposed method.

For this study, 2500 different re-deployment scenarios were simulated, each involving a random selection of the LKP, the time at which a clue is found, and the position of the clue. At the start of the search for each simulation run, the number of iso-probability curves, N_{curves}, for the initial deployment was selected via two approaches: (*i*) the proposed optimal deployment method modified for initial deployment as discussed in Section 4.1.3; and, (*ii*) randomly selecting N_{curves}

from a discrete uniform distribution in the range [1, $N_{curves,max}$], and assigning robots to the curves as per resource-requirements dictated by the procedure in Section 4.1.1.1. Upon discovery of a clue, the clue's position was taken as the new LKP, and a re-deployment solution was computed using two alternative approaches: (*i*) the optimal deployment method as presented in sections 4.1.1 and 4.1.2; and, (*ii*) using the same randomly selected solution from the start of the search and maintaining the same robot-to-curve assignments. Since the re-deployment scenarios were triggered using clue-finds, evaluation of the necessity for implementing the new optimal deployment solution, as summarized in Fig. 4.7, was not necessary. For reference purposes, the proposed approach used for optimal initial deployment and optimal re-deployment in these simulations (i.e., approach listed as (*i*) in the aforementioned) will be referred to as the 'optimization approach,' while the alternative of using randomly-selected iso-probability curves (i.e., approach listed as (*ii*) in the aforementioned) will be referred to as the 'random approach.'

After obtaining the re-deployment solution for each simulation using the two alternative approaches, the objective function value for the main optimization loop (Eq. 4.10) was evaluated for both solutions and was used as a normalized measure, in the range 1 to 2, of the quality of the re-deployment solutions obtained. For the random approach, although the number of iso-probability curves to use are selected randomly, a value for each of the three normalized metrics, search-time, success-rate, and return-time, can still be computed at each re-deployment instance in the simulations using the methods described in sections 4.1.1.2, 4.1.1.3, and 4.1.1.4. Thus, a value for the main objective function (Eq. 4.10) can still be obtained for the random approach as well.

The ratio of the objective function value obtained from the random approach to that obtained from the optimization approach was calculated, which provided a measure of the factor by which the optimal solution performed better than the random one. This 'benefit-ratio' was computed for each simulation run and, then, the minimum, average, and maximum benefit ratios over all the 2500 simulations were determined, respectively. This procedure was repeated for different settings of the number of robots (N_r) and different combinations of the modified-search-time and success-rate weightings (w_T and w_S, respectively). Figure 4.10 shows a plot of the minimum, average, and maximum benefit ratios over the different weight combinations for the case of N_r = 20 robots to provide a visualization of the typical behaviour of these benefit ratios with weight-selection. Table 4.1 summarizes the complete set of results from the simulations.

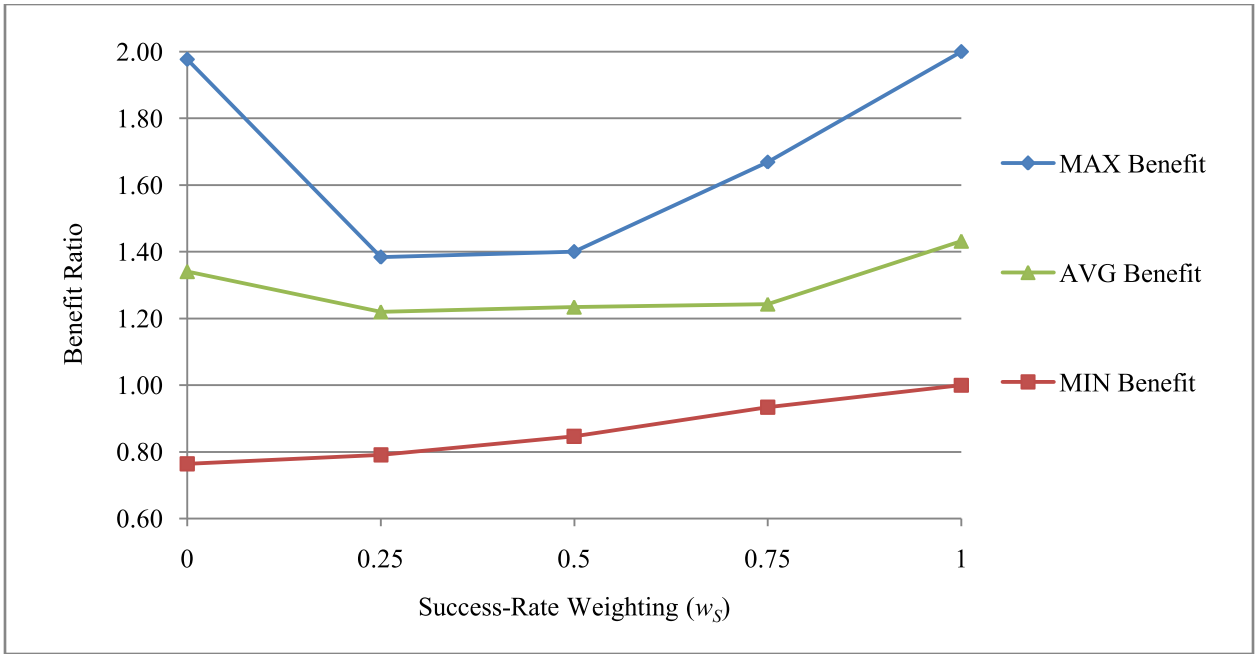


Figure 4.10. Variation of the minimum, average, and maximum re-deployment benefit ratios over different weight combinations for N_r = 20 robots (note: modified-search-time weighting, w_T, is obtained using $(1 - w_S)$).

Table 4.1. Summary of simulation results showing benefit of optimal re-deployment over a random approach.

N_r	w_T	w_S	Minimum Benefit Ratio	Average Benefit Ratio	Maximum Benefit Ratio
20	1	0	0.76	1.34	1.98
	0.75	0.25	0.79	1.22	1.38
	0.50	0.50	0.85	1.23	1.40
	0.25	0.75	0.93	1.24	1.67
	0	1	1.00	1.43	2.00
40	1	0	1.00	1.38	1.97
	0.75	0.25	1.00	1.23	1.40
	0.50	0.50	0.87	1.21	1.35
	0.25	0.75	0.95	1.23	1.59
	0	1	1.00	1.40	2.00
100	1	0	1.00	1.41	1.99
	0.75	0.25	1.00	1.23	1.38
	0.50	0.50	0.89	1.20	1.33
	0.25	0.75	0.90	1.23	1.58
	0	1	1.00	1.41	2.00

The simulation results show that optimization provides the greatest benefit at the extremes of the weighting combinations (i.e., $\{w_T = 1, w_S = 0\}$ and $\{w_T = 0, w_S = 1\}$), and generally tends to decrease as one moves toward the equal-weight setting (i.e., $\{w_T = 0.5, w_S = 0.5\}$). These results are to be expected due to the nature of the typical variations of the normalized search-time and success-rate metrics individually, which are presented in Fig. 4.11 for an example case involving $N_r = 20$ robots. Due to the inverse relationship between these two metrics, a weighted-sum involving *equal* weightings would result in these two metrics effectively cancelling each other out, so that the result has minimal *range* of variation over the different values of N_{curves}. However, the effect of this cancellation is diminished to some extent by the presence of the third metric considered during re-deployment, namely, return-time. Still, the minimal range of variation in the objective function value at the equal-weight setting implies that the *difference* between a given optimal selection of curves and a random one would, in general, be expected to be at its lowest. Conversely, placing all the weight on modified-search-time or on success-rate would maximize the range of variation of the weighted-sum, thereby producing greater difference in objective function value between an optimal and a random selection of curves.

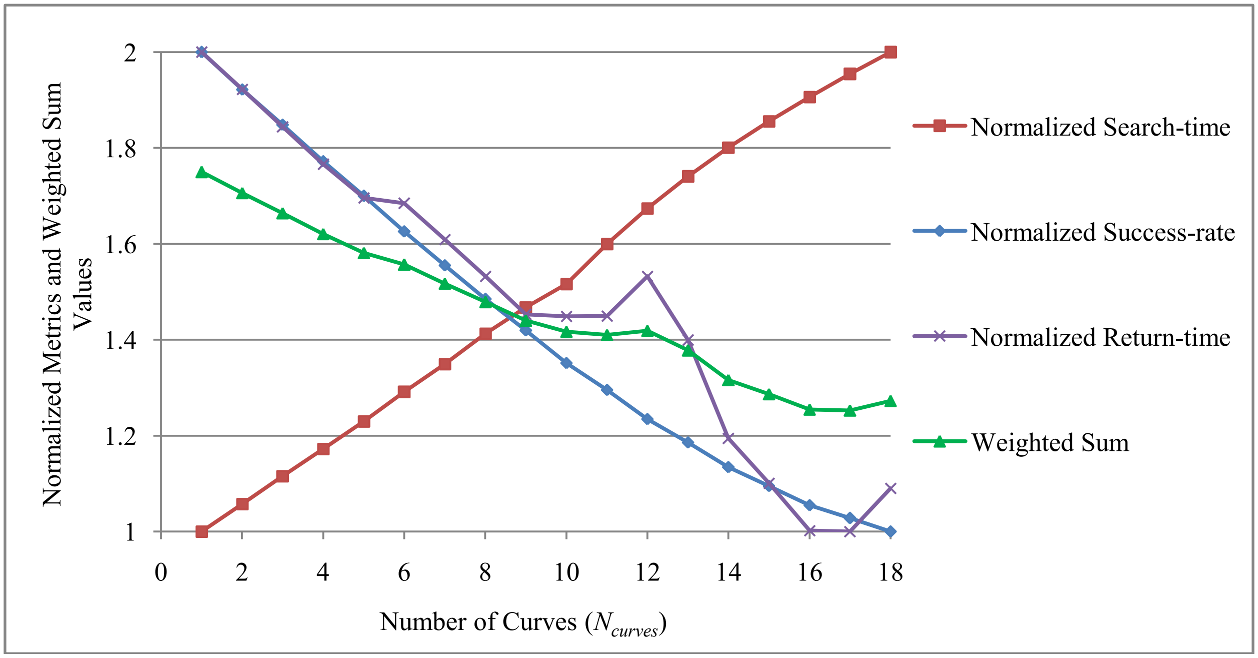


Figure 4.11. Variation of the normalized search-time, success-rate, and return-time metrics, and their equally-weighted sum (using Eq. 4.10) for $N_r = 20$ robots.

Lastly, it should be noted that the minimum benefit ratio can fall below 1, as seen in Table 4.1 and in Fig. 4.10. This situation occurs due to the fact that the different *initial* set of curves selected by the optimization and the random approaches causes the robots to be at different positions when a re-deployment event occurs, since the robots follow their assigned curves as they conduct the search. Thus, it is possible to have occasional cases where the random selection of the initial set of curves, although sub-optimal for initial deployment, allows for a shorter return-time during re-deployment; whereas the corresponding optimal initial deployment curves cannot achieve as low a return-time due to the way in which the robots are distributed within the search area when re-deployment occurs. Nevertheless, the simulation results indicate that, on average, despite the presence of such cases, there is still significant tangible benefit in conducting re-deployment by applying the proposed optimal robot deployment method during the ongoing search, even for equal weightings. This justifies the use of this method in the search process.

4.3 Chapter Summary

This chapter has presented the detailed approach to addressing the robot deployment sub-problem. Deployment is distinguished from searching in that it refers only to the distribution of the robots within the search area in order to achieve a spread of search resource that is most beneficial to the ongoing search. The two main tasks of deployment are to determine deployment positions within the search area and to specify the assignment scheme of robots to those positions.

The general strategy for deployment proposed in this Thesis is to place robots directly on iso-probability curves, thereby allowing the relative density of the curves to dictate the application of search-effort within the search area. A nested optimization formulation was proposed to select the optimal number and position of iso-probability curves, the number of robots to be placed on each curve, the deployment positions on those curves, and the assignment of robots to those positions, during any deployment instance. The outer optimization loop determines the optimal number of iso-probability curves (with their positions subsequently determined based on user-specified parameters) and the inner optimization loop determines the optimal assignment of robots to deployment positions on the set of curves being investigated during any given iteration of the outer optimization. The optimal deployment solution is governed by an objective function representing the benefit that a given deployment solution entails to the ongoing search. This benefit is a weighted combination of normalized metrics devised for search-time, success-rate, and return-time.

The proposed method for deployment also considers the need for re-deployment. A decision-making procedure is proposed that evaluates the optimality of the current deployment during the ongoing search and determines whether a re-optimization of the deployment is warranted.

Simulations were conducted to analyze the benefit of optimal re-deployment by comparing the benefit of using the proposed re-deployment method against the use of a randomly selected initial deployment that is maintained throughout the search. The results indicated tangible benefit to using the proposed re-deployment approach, especially when the relative weightings for search-time and success-rate differ significantly from a balanced setting.

Chapter 5

5 Robot-Path Planning

In the proposed methodology for MRC in WiSAR, the deployment and searching tasks are considered to be distinct phases of the solution process. Thus, searching begins after robots have moved to and attained their assigned deployment positions, whether that was an initial deployment at the start of the search, or a subsequent instance of re-deployment during the ongoing search. Thus, although the optimal deployment solution dictates where robots should be positioned in order to optimally distribute search resources, it does not indicate how robots should move afterwards to optimally perform the search. This necessitates the third main task of the MRC method – robot-path planning. As the term implies, this task involves constructing paths for each robot that guides them to conduct the search in an appropriate manner.

Planning robot (search) paths involves two main considerations. First, since the optimal deployment solution delimits the best way to distribute the search resources, it is necessary to plan paths that allow the robots to maintain this distribution. Thus, the optimal deployment solution constrains path planning. Secondly, the path planning process must be able to deal with uncertainty in the ability of a planned path to guide a robot to the desired destination within the desired time. The main source of this uncertainty is the relative inaccuracy with which the terrain topology information, $\mathcal{M}$, is known. This inaccuracy may arise due to errors in the terrain data and/or low resolution of the terrain information. A general path planning method must, therefore, be devised that accounts for this uncertainty while adhering to constraints imposed by the latest optimal deployment solution.

In addition to these two main considerations, the path planning method must also be able to address the difficulties arising from obstacles in the terrain, as these may obstruct a planned path. Although *a priori* known obstacles can be accounted for during path planning, *a priori* unknown obstacles must be dealt with in real-time as they are encountered. Moreover, as discussed in Chapter 4, any time a re-deployment of the search resources is computed, the robots must attain their newly assigned deployment positions as quickly as possible. Re-deployments may be required during the ongoing search when the current deployment is found to be sub-optimal, and are also required whenever clues are found. In both cases, robots must reach their deployment positions via the shortest-path. Determination of this shortest-path would require a modification of the general path planning method. The devised path planning method must be able to address all these challenges in an on-line fashion.

Due to the presence of the abovementioned uncertainties and unknown events, which can interfere with, and sometimes invalidate, a planned path, path planning must also be a dynamic process. In particular, upon planning an initial path for a robot, regular on-line checks of the feasibility of each path must be made while the path is being implemented by the robot to ensure that it is able to achieve the desired goal. If at any time it is found that a given initial path is unable to guide the robot to achieve its goal, that path must be re-planned. This naturally leads to a path planning process with three phases: (*i*) initial path planning, (*ii*) path implementation and evaluation, and (*iii*) path re-planning (when necessary). One can note that this general approach resembles the one used for the robot deployment problem, where an initial deployment is followed by regular, on-line checking of the optimality of the deployment, and computation and implementation of a re-deployment as required. In this Chapter, a robot-path planning method will be presented that addresses these three phases. Section 5.1 will first present the proposed method by detailing the formulations and procedures for each of the three phases. The three sections that follow will discuss additional considerations in robot-path planning, including, respectively, the concept of a centre-robot and its unique path-planning needs given the proposed approach to coordinating robots (Section 5.2), and the modifications to path planning and implementation that are necessary in order to cope with path-obstructing obstacles (Section 5.3) and instances of re-deployment requiring the planning of shortest-paths for the robots (Section 5.4).

5.1 Robot-Path Planning Method

The solution to the optimal deployment problem indicates the best way to distribute the search resources within the given search area. This is done through the specification of the optimal regions for search-effort allocation as well as the deployment positions within those regions for each robot. In this Thesis, these search-effort allocation regions are delimited by the iso-probability curves, and, furthermore, robots are assigned to deployment positions on these curves. The iso-probability curves, and the distribution of the robots among the curves, are determined through an optimization process that optimizes pertinent attributes of the ongoing search. These attributes characterize the benefit that the implemented deployment solution would have on the search from the perspectives of search-time, success-rate, and return-time.

By virtue of the relative density of the optimally selected set of iso-probability curves in different regions of the search area, as well as the number of robots assigned to each curve, the optimal deployment solution represents the distribution of search robots that would be most beneficial to the search. In addition, the robot deployment method incorporates its own optimality-checking process, whereby the ‘goodness’ of a given set of curves and distribution of robots among them is kept in-check to ensure that it continues to be most beneficial, amidst the variability of the search process caused by time-varying target-location prediction, modifications to the terrain information, and found clues. Naturally, therefore, it is imperative that search paths be planned in such a manner that allow the robots to maintain this search-effort distribution dictated by the iso-probability curves and the robot assignments to those curves. This logic leads to the underlying principle on which robot-path planning is based, namely, that each robot must remain within its assigned search-effort allocation region as it moves to perform the search. Since the search-effort allocation regions are represented by the optimal selection of iso-probability curves, this principle entails that each robot must remain on its assigned iso-probability curve as it performs the search. This would ensure that the robots continue to search within the search-effort allocation regions to which they have been assigned, and also that the relative distribution of the search resources within the search area is always maintained as per that dictated by the latest optimal deployment solution that was implemented. Doing so will ensure that the search progresses in an optimal fashion since the search-effort will continue to be expended in appropriate proportions within the optimal regions.

The iso-probability curves, therefore, once again form a key component that guides the method being proposed. This construct not only portrays the nominal probabilistic information about the location of the target at any given time, but also incorporates the influences of other factors that impact robot-path planning. Thus, by following the curves, the robots not only remain within their assigned search-effort allocation regions but also move in a way that accounts for these other influences that have been incorporated into the curves. However, adhering to the proposed underlying principle of having robots remain on their curves is not a trivial task and poses a significant challenge. This is primarily because the iso-probability curves represent the prediction of the target’s location at any given time, and, as such, they vary continuously with time. In particular, the curves are propagated at regular time-intervals, Δt_{prop}, to account for predicted target motion. Therefore, each robot does not simply have to follow a stationary curve represented by a single, fixed, piecewise cubic-polynomial function, but rather, the robot must travel along a curve that itself moves continuously in time and space.

In order to address this time-varying nature of the curves, the general approach to robot-path planning that is used is to perform a 'look-ahead'. Thus, for a given robot at the current time, t, the iso-probability curve that the robot is assigned to is propagated by some time-interval, Δt, to the future point in time, $(t + \Delta t)$. With the iso-probability curve at time t and its propagated version at time $(t + \Delta t)$ obtained, the objective is to construct a robot path between these two curves. Namely, a path needs to be constructed between the point on the *starting* iso-probability curve on which the robot is currently located at time t, and a point on the propagated *destination* iso-probability curve on which the robot is required to be at time $(t + \Delta t)$, such that the time required by the robot to traverse that path is Δt.

In the following, the iso-probability curve that a given robot must start its path on during any given path-planning instance will be referred to as the 'origin curve,' while the iso-probability curve that is propagated into the future, to which a path must be planned, will be referred to as the 'destination curve.' If such a path can be planned and implemented successfully, then a robot following that path is ensured to remain on its iso-probability both at the start of the path at time t and at the future point in time, $(t + \Delta t)$. As the value of Δt approaches zero, these (short) paths would theoretically allow the robot to remain on its assigned iso-probability curve in a continuous manner as that curve propagates forward with time.

However, the above ideal approach would require the calculation of close to an infinite number of iso-probability curves for every search robot (i.e., a curve at every instant in time), and determination of the corresponding paths that must be continuously 'stitched' together to maintain at least 1st derivative robot-motion continuity. During implementation, there would not be time to compute all such intermediate curves. Practical limitations only allow the computation of a limited number of iso-probability curves, every Δt, where the time-interval Δt is some finite, reasonable value dictated by the available computing resources.

As presented in Chapter 3, the target-location prediction method dictates the propagation of the iso-probability curves at Δt_{prop} time-intervals. Furthermore, according to the robot deployment method, a new deployment solution is computed and considered for implementation at this same time-interval, possibly resulting in a change in the iso-probability curves being used and the assignment of robots to those curves. It would, therefore, be logical and convenient to break-down robot-path planning into paths planned over this same time-interval, Δt_{prop}.

In the method proposed in this Thesis, path planning becomes a task of successively planning paths for each robot from the robot's current position at time t to a future position on its propagated assigned iso-probability curve (destination curve) taken to be the curve at time ($t + \Delta t_{prop}$). Each resulting path will serve as an approximation of the intermediate positions of points on the corresponding iso-probability curve as that curve changes in time and space between individual propagations. In this way, with each propagation of the iso-probability curves, new paths are planned for each robot that continue on from where the last path left-off, resulting in the seamless construction of search-paths throughout the search operation.

The constraints for constructing each robot path, from the corresponding origin curve to the destination curve, are that the end-point of this path must lie on the destination curve and the traversal-time for the path must be Δt_{prop}. However, given the varying influence of terrain over the search area, and the infinite possibilities for destination points on the destination curve, a search process would be required to find an appropriate path. Moreover, there is no guarantee that a path exists which would have a traversal-time of exactly Δt_{prop}. This implies that path planning must be an optimization process, where an optimal path is defined as the one that gives a computed traversal-time as close to Δt_{prop} as possible.

An additional complication that arises in this path planning problem is the uncertainty in the time required to traverse a given path, and, therefore, in the arrival-time of a robot on its destination curve. This uncertainty exists due to the fact that the terrain details would not be known with very high precision. Typically, terrain information is obtained through topographical maps, which give general information about the terrain topology, but would typically not show all terrain details. Although it is possible that more detailed terrain information may be available due to prior exploration of the area, this information would still have to be discretized into a rectangular grid array in order to produce a height map, and this could cause some degradation in accuracy. In general, however, it cannot be assumed that all terrain details are known precisely beforehand, and the proposed method must be designed to address this difficulty.

Due to this terrain uncertainty, the impact of terrain on path-traversal time cannot be determined exactly either. As a result, it is really only possible to compute an *estimate* of the traversal-time and the path-end arrival-time for any given path before its implementation by a robot. The actual arrival-time would only be known once the robot actually traverses the path. This poses a problem because an initially planned (optimal) path may become significantly sub-optimal as the robot begins travelling along it and experiences and reacts to the actual terrain that it passes over. In such cases, the robot could experience an unacceptable amount of discrepancy between the desired and actual arrival-time. Not being able to arrive at the destination curve at the desired time of $(t + \Delta t_{prop})$, either because the robot is too early or too late, would mean that the robot would have strayed too far from its assigned iso-probability curve, thus, failing to adhere to the fundamental principle governing path planning for the proposed method, and thereby decreasing the optimality of the ongoing search.

To address this issue of uncertainty, it is proposed that the suitability of each robot's path be regularly checked, on-line, while the path is being implemented (*in-between* the iso-probability curve propagations). The advantage of using this approach of regular checks during implementation is that as the robot travels along its path, it can gather information relevant to computing the uncertainty in the path's traversal and arrival times. Thus, the further along its path a robot travels, the more information about terrain-influence-based path uncertainty would be obtained, and the more accurately the arrival-time for the current path can be estimated. This can help in deciding whether the current path should still be followed, or if a new optimal path needs to be planned. A method will, therefore, be proposed that uses information obtained by each robot about uncertainty along its path, based on the portion of the path it has traversed so far, to check whether the current path will allow the robot to arrive at its destination curve with acceptable discrepancy from the desired arrival time. The checks are performed at pre-determined positions, referred to as 'check-points,' along the planned path. The method will dictate how these check-points should be designated, the type of information to be gathered as the robot moves along its path, how this information should be analyzed to perform the checks, and the conditions that determine whether the current path is acceptable or if a new path needs to be planned.

If it is determined that a new path must be planned, a re-optimization of the path must be performed, and this constitutes *path re-planning*. The new path would begin at the robot's position at the time of the re-plan and would end on the destination curve. This new path must also be determined according to the same constraints as those used for the original path, namely, that the path must end on the destination curve and that the traversal-time must be as close as possible to the time available.

The proposed overall approach to robot-path planning, then, is a recursive, three-phase method, as illustrated in the summary flowchart in Fig. 5.1. At any given path-planning instance, one begins by planning an initial optimal path for each robot to its destination curve. Each robot begins implementing this path while also gathering relevant path-uncertainty data and using it to check path optimality. If, through this check, a robot's path is found to be sufficiently sub-optimal, then, a new optimal path is computed and, subsequently, implemented. With this new path obtained, this same cyclic process of path-implementation and path-checking continues, until the robot reaches its destination curve. Once a robot completes its path and reaches its destination iso-probability curve, that curve is propagated forward another Δt_{prop} time into the future, and this overall path planning procedure repeats from Phase 1. In this way, each robot can perform a search while staying on its assigned iso-probability curve (i.e., staying within its assigned search-effort allocation region) as best as possible. The detailed formulations and methods for each of these three phases are presented in the next three sections, respectively.

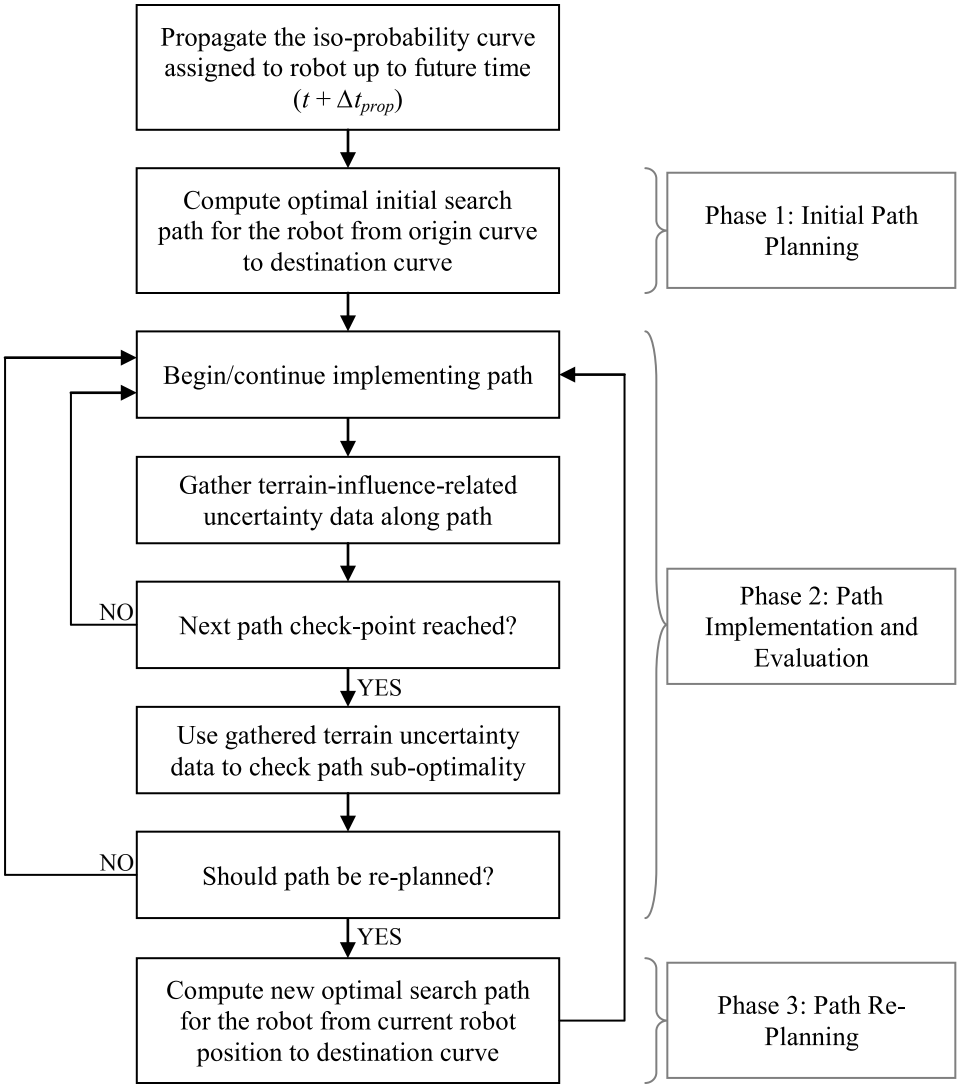

Propagate the iso-probability curve assigned to robot up to future time ($t + \Delta t_{prop}$)
Compute optimal initial search path for the robot from origin curve to destination curve
Phase 1: Initial Path Planning
Begin/continue implementing path
Gather terrain-influence-related uncertainty data along path
NO
Next path check-point reached?
YES
Phase 2: Path Implementation and Evaluation
Use gathered terrain uncertainty data to check path sub-optimality
NO
Should path be re-planned?
YES
Compute new optimal search path for the robot from current robot position to destination curve
Phase 3: Path Re-Planning

Figure 5.1. Proposed overall approach to robot-path planning for any given robot.

5.1.1 Initial Path Planning

Initial path planning refers to the first path that is planned for any given robot at the start of the search, after the robots have been initially deployed. However, it also refers to the first path planned for the robot every time it reaches its destination curve. In particular, once a robot arrives at its destination curve at some time t, the future iso-probability curve for time $(t + \Delta t_{prop})$ is computed through propagation and it becomes the new destination curve. The first path that is, then, planned for the robot to its new destination curve constitutes yet another instance of initial path planning. Thus, a path is ‘initial’ in the sense that it is the first path to be planned for a robot at the start of the latest Δt_{prop} time interval. As mentioned above, multiple path options exist, and one must be found that yields an estimated traversal-time as close to Δt_{prop} as possible, thereby necessitating an optimization process. Presented below are the details of how an initial path is planned. Section 5.1.1.1 will present the overall approach to path-construction proposed in this Thesis, and Section 5.1.1.2 will formulate the optimization problem that must be solved to obtain any given initial robot path.

5.1.1.1 Overall Framework of Path-Construction

Initial path planning for any given robot involves the construction of a path from the robot's origin curve to its destination curve. This path serves as an acceptable approximation to intermediate positions on the iso-probability curve assigned to that robot as the curve propagates in time and space, from the origin curve position to the destination curve position, within an individual curve-propagation time-interval. A typical such scenario for initial path planning for a given robot is shown in Fig. 5.2, and will be used to describe the components of path-construction. In this figure, the black-coloured solid and dashed lines represent the origin and the destination positions of an example iso-probability curve, respectively. The origin curve is assumed to be the iso-probability curve corresponding to time t, and the destination curve is assumed to correspond to the future (propagated) version of that curve for time $(t + \Delta t_{prop})$. The two green-coloured, dashed lines represent two example intermediate curve propagations corresponding to two unique intermediate points in time (t_1 and t_2, respectively), so that: $t < t_1 < t_2 < (t + \Delta t_{prop})$. The red, dashed line represents an example of a chosen path, leading from the robot (blue dot) at its current position on the origin curve at time t to a position on its destination curve corresponding to time $(t + \Delta t_{prop})$.

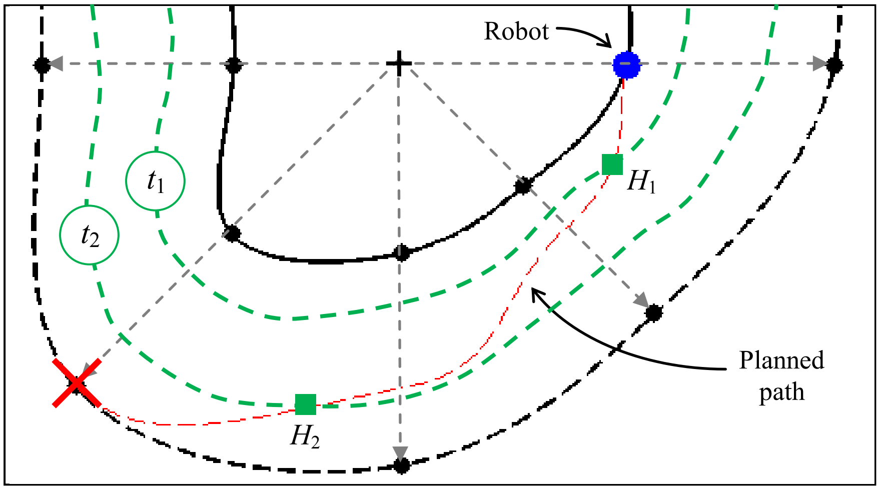


Figure 5.2. Example of an initial planned path.

For this chosen path, if the robot is to remain on its assigned curve at times t_1 and t_2, it would ideally need to be at Points H_1 and H_2 at times t_1 and t_2, respectively, so that it crosses (i.e., attains a position on) the first and second intermediate curves shown, at the appropriate times. In so doing, the robot would adhere to the fundamental principle of path planning, which requires robots to stay on their assigned curves as they perform the search. If these two intermediate points are known, in addition to the initial and destination points, then they can be used to calculate a three-segment, smooth robot-motion path. The first segment would be the portion of the path between the initial point (where the robot is located at time t) and H_1; the second segment would be the portion of the path between points H_1 and H_2; and the third segment would be the portion of the path between H_2 and the destination point (indicate by the red '×' in Fig. 5.2) on the destination curve.

During on-line path planning, it would be computationally prohibitive to determine intermediate iso-probability curves. Instead, one would have the initial curve (on which the robot is currently located) and the single corresponding (future) propagated curve, that was computed at the latest curve-propagation event, and that serves as the destination curve. A suitable path must, therefore, be planned from the origin curve to the destination curve using only the initial point and a single point on the destination curve. Thus, the proposed approach to path construction is to define intermediate path-interpolation points on the individual rays that are used to construct the iso-probability curves. The points are positioned along those rays in a manner proportional to the length of path completed by the time the robot crosses the position of any given ray.

For example, Fig. 5.3 shows Interpolation Point IP_k, which is one of the intermediate interpolation points used to construct the path shown in Fig. 5.2. Interpolation Point IP_k is located between Control Point A on the origin curve and Control Point B on the destination curve, both of which lie on one of the rays, θ_k, used for iso-probability curve construction. This interpolation point must appropriately be positioned along this ray, between points A and B, in order to obtain a point to use for constructing the path. To do so, it is required that the ratio, η_k, of $\overline{AIP_k}$ to $\overline{AB}$ be the same as the ratio of the length of path, s_k, traversed so far by the robot when it crosses ray θ_k, relative to the total length of the path, s_{TOT}, it is planning to follow to reach its destination point on the destination curve, Eq. 5.1. The length $\overline{AIP_k}$ refers to the length of the line-segment along ray θ_k between points A and IP_k, while the length $\overline{AB}$ refers to the length of the line-segment along ray θ_k between points A and B. The ratio, η_k, is referred to as the path-segmentation ratio, since it determines the positions of the interpolation points that delimit the segments that form the overall path. Also, since interpolation points are placed on rays between the initial point and the destination point of the path, there would be as many interpolation points as there are rays in this region between those two points. Thus, the total number of interpolation points, N_{IP}, for a chosen path equals the total number of rays that the path crosses, excluding the starting ray on which the initial point lies and the ending ray on which the destination point lies.

$$\eta_k = \frac{\overline{AIP_k}}{\overline{AB}} = \frac{s_k}{s_{TOT}}, k \in [1, N_{IP}] \tag{5.1}$$

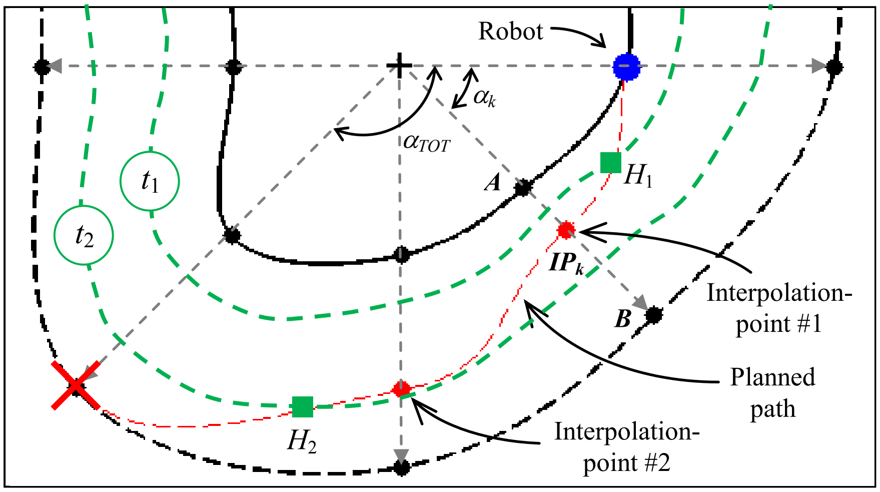


Figure 5.3. Example initial path showing determination of an interpolation point.

However, computation of any given ratio η_k poses another difficulty. Since the path itself cannot be constructed without specifying the exact positions of the interpolation points, and the interpolation points cannot be specified without computing the path-lengths s_k and s_{TOT}, it is apparent that the ratio η_k must be computed in another way; in particular, this ratio is estimated. This can be done by using an angular estimate, $\hat{\eta}_k$, of the ratio η_k, for any given interpolation point $k \in [1, N_{IP}]$.

Let the angular distance α_k represent the angle between a ray passing through the initial point on a chosen path and the ray, θ_k, along which the interpolation point, IP_k, needs to be placed. Similarly, let the angular distance α_{TOT} represent the angle between a ray passing through the initial point on that chosen path and the ray passing though the destination point of that path. Then, the estimated path-segmentation ratio, $\hat{\eta}_k$, can be computed as:

$$\hat{\eta}_k = \frac{\alpha_k}{\alpha_{TOT}}, k \in [1, N_{IP}]. \tag{5.2}$$

Using this approach to delimit interpolation points for path-construction provides a fast and convenient way to ensure that the outward (radial) progress of the path from its initial point has a direct correspondence to the angular progress of the path. This helps to keep the path within the confines of the origin and destination curve, and allows for a smooth transition (i.e., a smooth path) from the initial point to the destination point.

These interpolation points along the path serve as approximations of corresponding intermediate points on the intermediate propagated iso-probability curves. Thus, for instance, let us assume that a robot travelling along the path in Fig. 5.3 arrives at Interpolation Point IP_k at time t_1. This interpolation point would, therefore, represent an approximation to the actual point H_1 on the actual iso-probability curve at that time.

Once all the interpolation points have been determined in this manner, a suitable interpolating curve can be defined that starts at the initial point, passes through all the interpolation points, and ends at the destination point. However, in general, there can be many interpolation points, and to ensure that constraints can be defined requiring the curve to pass through all those points could necessitate a very high order polynomial function that would increase complexity of the path-definition as well as the required computation time. An alternative approach would be to construct a segmented path, where a separate curve is defined between adjacent interpolation points. Each such curve would represent a path segment connecting one interpolation point to the next. Therefore, each segment would connect to its adjacent segments at its endpoints. In this way, constructing each segment would only require appropriate constraints at the two endpoints to ensure a smooth and continuous path, and a relatively lower order, simpler curve would suffice.

In order to implement this alternative approach in this Thesis, piecewise cubic polynomials are used to construct the path segments between path-interpolation points [129]. Position-continuity and 1st-derivative-continuity constraints are defined at every interpolation point to obtain a system of equations, which can, then, be solved to obtain the polynomial coefficients of every path segment. All path segments taken together, then, represent the overall path. One can recall that these are the same constraints that are used at every control-point to construct each iso-probability curve (see Fig. 3.3 in Section 3.1 of Chapter 3). Moreover, at the starting point of the path (where the robot is located), the current position and orientation of the robot provide the necessary constraints for the starting point of the first cubic polynomial segment of the path. Likewise, the desired destination-point coordinates and robot orientation on the destination curve provide the corresponding constraints for the endpoint of the last cubic polynomial segment of the path. In the proposed approach, it is required that the orientation of the robot at the destination point be tangent to the destination curve at that point.

The following steps summarize the overall, general path-construction method:

Step 1: Obtain the initial point of the path on the origin curve (i.e., the robot's current position).

Step 2: Obtain the desired destination point of the path on the destination curve.

Step 3: Identify all rays, θ_k, $k \in [1, N_{IP}]$, that lie between the initial and destination points.

Step 4: Compute the angular distances, α_k, of each ray, θ_k, $k \in [1, N_{IP}]$.

Step 5: Compute all path-segmentation ratio estimates, $\hat{\eta}_k$, $k \in [1, N_{IP}]$, using Eq. 5.2.

Step 6: Use the ratios, $\hat{\eta}_k$, to determine the positions of all the interpolation points, IP_k, $k \in [1, N_{IP}]$

Step 7: Construct the system of equations to determine the cubic polynomial segments between the interpolation points using position-continuity and first-derivative-continuity constraints.

Step 8: Solve the system of equations from Step 7 to fully define the cubic polynomial segments and to, thereby, obtain the complete robot search-path.

The robot search-path, constructed by the method described above, represents an estimate as to where the robot should be throughout the curve-propagation time-interval, Δt_{prop}, so that its position approximately corresponds to a point on its assigned iso-probability curve as that curve propagates in time and space. Therefore, by following this path, the robot should remain close to, or directly on, its assigned iso-probability curve at all times. In this way, each robot is able to adhere to the aforementioned fundamental principle of the proposed path-planning strategy. Of course, the procedure described in this section simply indicates how a path is to be constructed for a *given* destination point on the destination curve. To find a destination point for which the constructed path requires Δt_{prop} time to traverse is an optimization problem that is discussed next.

5.1.1.2 Optimization of Path-Construction

Robot-path planning requires an optimization to find the optimal destination point on the destination curve. For any robot, Rb_i, its respective path needs to be optimized independently, but, concurrently with the paths of the other robots. In each case, the objective of the respective optimization is the same – namely, to construct a smooth path from the robot's current position at time t to an optimal point on its propagated future (destination) iso-probability curve, so that the path-traversal-time is as close as possible to the desired curve-propagation time-interval, Δt_{prop}.

Thus, the single-variable optimization will be one of searching for the best position of the endpoint on the destination curve that will yield an overall path by which the robot arrives at this destination curve at a time as close as possible to ($t + \Delta t_{prop}$). If this endpoint optimization objective is achieved, while also using the method proposed for constructing segmented paths based on appropriate path-segmentation ratios, $\hat{\eta}_k$ (Eq. 5.2), one can assume that all robots would remain approximately on their assigned iso-probability curves throughout the search and, thus, preserve the efficacy of the proposed WiSAR method.

The optimization that is conducted to determine the destination point $P_{dest,i} = \{x_{dest}, y_{dest}\}_i$ for the search-path of any given robot, Rb_i, can be formulated as follows. Let us suppose that at time $t = T_{current}$, the robot Rb_i is located at $P_{Rbi} = \{x_{Rb}, y_{Rb}\}_i$ on its assigned iso-probability curve. This curve is, thus, the origin curve for the purposes of initial path planning. Let the corresponding future, propagated iso-probability curve be defined by the function $F_{dest,i}$. This curve becomes the destination curve on which the destination point must lie. Let the available time until the next curve propagation (i.e., the time available for the robot to reach its destination curve) be Δt_{avail}.

For the initial path to be planned just before implementing the latest curve propagation, Δt_{avail} = Δt_{prop}.

The decision variable for this optimization is the destination position, $P_{dest,i}$. It can be considered a single variable since a function, namely, $F_{dest,i}$, is available that delimits the closed curve representing the possible values for the destination point. As a result, any desired destination point can be indicated by specifying a single, angular value, representing the angular position of a ray extending from the LKP to the destination curve. The point on this iso-probability curve, $F_{dest,i}$, that intersects any such ray of specified position, would be the destination point, $P_{dest,i}$. Thus, the specification of a single angular-position value completely specifies the desired position on the destination curve.

The objective of this optimization, as discussed earlier, is to find the destination position to which a path can be planned that has a traversal-time as close as possible to Δt_{avail}. However, from the preceding section, one can note that upon specifying a destination position, a single, unique path can be planned using the proposed path-construction method, and this path would be associated with a single, unique estimated traversal-time. As such, for any valuation of the decision variable (i.e., for any specified destination position, $P_{dest,i}$, on the destination curve), a single, unique traversal-time can be found. The objective of the optimization, then, becomes to minimize the difference between this estimated path traversal-time (corresponding to a particular destination position) and the time available to complete that path, Δt_{avail}.

Given the above description of the problem, the search for the optimal destination point $P_{dest,i}$ = $\{x_{dest}, y_{dest}\}_i$ for Rb_i can be formulated as:

$$min|\Delta\tau_{trav} - \Delta t_{avail}|, \tag{5.3}$$

subject to:

$$\Delta\tau_{trav} \leq \Delta t_{avail}, \tag{5.4}$$

$$P_{dest,i} \in F_{dest,i}. \tag{5.5}$$

In the above formulation, $\Delta\tau_{trav}$ is the estimated time required to traverse the entire path corresponding to a specified destination point, and is, therefore, a function of the decision variable, $P_{dest,i}$. Thus, for any given potential destination point that is considered during the optimization, a path first has to be constructed from $\{x_{Rb}, y_{Rb}\}_i$ to $P_{dest,i} = \{x_{dest}, y_{dest}\}_i$, before $\Delta\tau_{trav}$ can be computed and before the objective function can be evaluated. The objective function given by Eq. 5.3, therefore, requires the minimization of the absolute difference between the estimated path-traversal time, $\Delta\tau_{trav}$, corresponding to the destination point, and the time available until the next curve-propagation, Δt_{avail}. The first constraint, Eq. 5.4, ensures that a destination point is selected for which the corresponding path's estimated traversal-time does not exceed the available time. The second constraint, Eq. 5.5, ensures that the selected destination point lies on the destination curve.

The path-traversal-time estimate, $\Delta\tau_{trav}$, depends on the influence of the terrain over which the path passes and the speed-capability of the robot when travelling on different slopes. It is computed in a manner similar to that used in computing terrain influence on the position of individual control-points when constructing iso-probability curves. Given knowledge of the speed-capability of a robot, a function similar to that of Eq. 3.8 in Section 3.2.1.1 of Chapter 3 can be set-up that outputs robot speed, or a speed scale-factor, for a specified average terrain slope. With such a function specified, an estimate of the traversal-time, $\Delta\tau_{trav}$, for any chosen path can be determined by dividing the path into a number of pieces, referred to, henceforth, as 'subdivisions,' of equal length.

Namely, if a given path is divided into N_{SD} subdivisions, each subdivision, $j \in [1, N_{SD}]$, is bounded by Node j at its start and Node $(j+1)$ at its end (giving, in total, $(N_{SD} + 1)$ nodes), Fig. 5.4. The average terrain slope, γ_j, along each subdivision, $j \in [1, N_{SD}]$, can be computed using the terrain-heights (obtained from the terrain height-map) at the nodes bounding each subdivision. This slope can be used to obtain an estimate of the robot's speed along each subdivision using the abovementioned function based on robot speed-capability knowledge. Given a speed for each subdivision, as well as the path-length along each subdivision, an estimate of the subdivision traversal-time, $\Delta\tau_{trav,j}$, can be computed for each subdivision, $j \in [1, N_{SD}]$. These subdivision traversal-time estimates can, then, be used to compute an estimate of the overall path traversal-time, $\Delta\tau_{trav}$, as:

$$\Delta\tau_{trav} = \sum_{j=1}^{N_{SD}} \Delta\tau_{trav,j} \, . \tag{5.6}$$

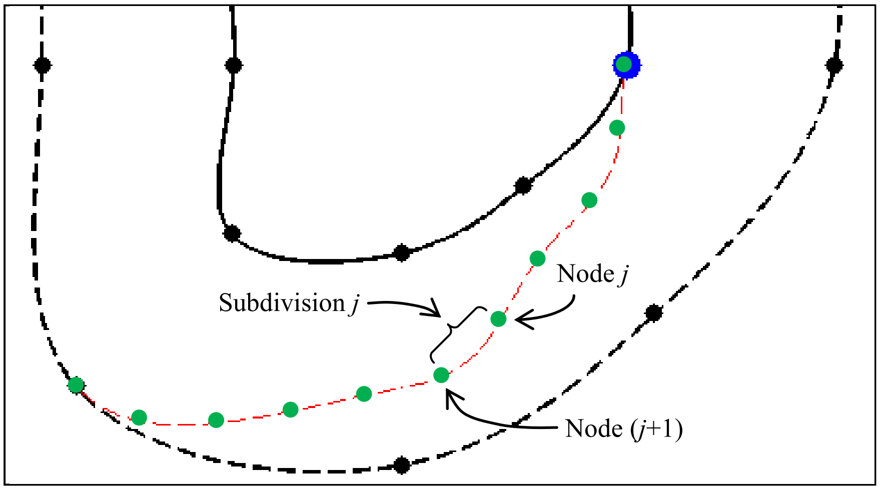


Figure 5.4. Division of a path into subdivisions for estimating path traversal-time.

Since the computation of path traversal-time involves a process of discretization and estimation, and is complicated by factors such as terrain difficulty and impeding obstacles (to be discussed in a later section), the objective function given by Eq. 5.3 would, in general, not be a closed-form differentiable function. Thus, to solve the optimization problem given by Eqs. 5.3 to 5.5, the user may select any derivative-free direct-search technique for constrained optimization, such as the Nelder-Mead method or others, as reviewed in [134].

5.1.2 Path Implementation and Evaluation

Upon solving the optimization problem formulated by equations 5.3 to 5.5, an initial optimal path will have been obtained for each robot, and Phase 1 of the overall robot-path planning approach will be complete (see Fig. 5.1). The second phase of this approach, as summarized in Fig. 5.1, involves a process of implementing the path while also regularly checking the path for sub-optimality. These checks are based on terrain uncertainty information that is gathered by each robot during path implementation. In order to conduct the evaluations of path sub-optimality, a schedule for checking must be established and conditions must be specified indicating what constitutes a sufficient reason for re-planning a path.

As a robot travels along its initially planned path, the (N_{SD} + 1) nodes that were defined for the path subdivisions (excluding the first and last nodes) become the check-points at which the path must be evaluated to ensure that it remains optimal. This forms the 'schedule' for path-checking. However, rather than re-running the entire path-optimization algorithm at each check-point, it is proposed herein that three time-efficient checks be invoked to estimate and examine arrival-time uncertainty via the current path. These uncertainty checks are conducted to gauge whether a feasible path exists that can guide a given robot to its destination curve in the available time, Δt_{avail}, and keep the robot on its assigned iso-probability curve as best as possible during path implementation. The following three sections present these three checks, respectively.

5.1.2.1 Check #1: Next Shortest-Path Feasibility

While a robot is executing its planned path, there may exist an undesirable possibility that not only would the current path be unable to guide the robot to its destination within Δt_{avail}, but that even an alternative (shortest) path to the destination curve, taken upon arriving at the next check-point, may not be feasible (i.e., achievable within Δt_{avail}). In order to check for this extreme possibility in an expeditious manner, an estimate of the shortest possible path from the next check-point to the destination curve is made by assuming a simple straight-line path, Fig. 5.5. Using an estimate, rather than conducting a search to find the actual shortest path (i.e., path with shortest-traversal-time), is necessary because this check must be conducted on-line, where the need for expediency is paramount.

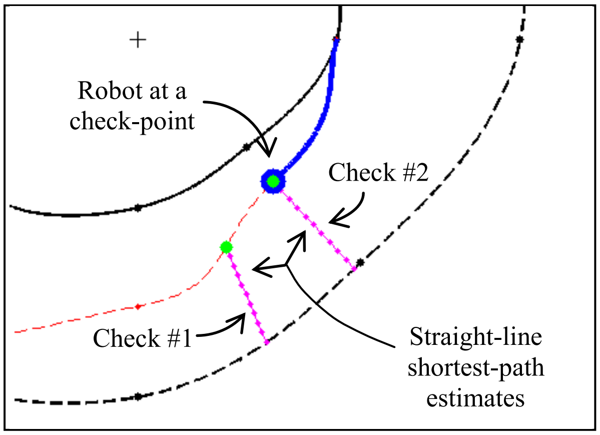


Figure 5.5. Shortest-path estimates for Check #1 and Check #2.

From Fig. 5.5, it is evident that, with this alternative path, the total associated traversal-time that must be compared to Δt_{avail} is the sum of the time taken to traverse the next immediate subdivision up to the next check-point and the time taken to traverse the estimated shortest path. The traversal-time for the next subdivision would already have been estimated during the process of estimating the traversal-time of the overall initial path. Also, the traversal-time associated with the straight-line shortest-path is estimated in the same manner as for any general path as described in Section 5.1.1.2.

However, due to uncertainty in the available terrain information, the robot may end up travelling slower or faster over the next subdivision, compared to what was estimated for that subdivision's traversal-time during initial path planning. This results in uncertainty as to the actual arrival-time at the destination curve, if the robot were to traverse this subdivision up to the next check-point and then take the straight-line shortest-path. Since the arrival-time that results from this alternative path is being used to check path sub-optimality, this uncertainty, in turn, would have an impact on the necessity for re-planning and re-optimizing the robot's path at the current time.

Therefore, rather than obtaining a single estimate of the arrival-time via the combination of the next subdivision and the shortest-path, and comparing it to Δt_{avail}, a more effective comparison would involve constructing a confidence interval for this arrival-time. This confidence interval must be computed based on probabilistic information about the inherent uncertainty in traversal-time per path subdivision, which would be obtained by the robot during path implementation.

As mentioned earlier, information relevant to terrain uncertainty is gathered by each robot as it travels along its assigned path. The more of its path the robot completes, the more uncertainty-related data would be gathered, and the more accurately one can expect to estimate the arrival-time via the current path. In particular, then, as the robot travels along its path, a record is kept of the actual time, $\Delta t_{trav,j}$, spent to traverse each of the past path subdivisions. Based on the initial path-plan, the corresponding estimated traversal-times, $\Delta\tau_{trav,j}$, for each of those subdivisions are also known. This allows for the computation of the traversal-time estimation error, $\varepsilon_{trav,j}$, for each path Subdivision j, that has been traversed up to the robot's current position, where:

$$\varepsilon_{trav,j} = \Delta\tau_{trav,j} - \Delta t_{trav,j} \; . \tag{5.7}$$

With the lack of any other information or error model, it is assumed that the naturally-occurring terrain features have variations that occur randomly according to a normal distribution (i.e., like a type of white, Gaussian noise). As such, the errors in traversal-times based on the influence of this terrain along the current path are also assumed to follow a normal distribution. Thus, once a robot has completed traversing two path subdivisions, there would be enough error data to compute an estimate, $\bar{\varepsilon}_{trav}$, of the average error in the estimated traversal-time per path subdivision of the current path due to terrain variations, as well as a corresponding confidence interval. In particular, let us assume that the robot is currently at check-point j >1, so that it has traversed a total of j subdivisions. In so doing, it will have gathered error data, $\varepsilon_{trav,k}$, $k \in [1, j]$.

Given this data, a $\pm 3\sigma$ confidence interval for the traversal-time estimation error for a single subdivision on the current path can be computed. Computation of a confidence interval for normally distributed data is a standard procedure wherein the *t*-distribution is used for less than 30 error data-points and a *z*-distribution is used for more data, in order to compute a deviation, Δ_{ε}, around $\bar{\varepsilon}_{trav}$ (see [124] for details). The resulting confidence interval will be centred at the mean error, $\bar{\varepsilon}_{trav}$, and will have upper and lower limits, $T_{\varepsilon,HI}$ and $T_{\varepsilon,LO}$, respectively, given by:

$$T_{\varepsilon,HI} = \bar{\varepsilon}_{trav} + \Delta_{\varepsilon}, \tag{5.8}$$

$$T_{\varepsilon,LO} = \bar{\varepsilon}_{trav} - \Delta_{\varepsilon}. \tag{5.9}$$

Next, a $\pm 3\sigma$ confidence interval is constructed for the arrival-time at the destination curve if the robot were to travel along the alternative path consisting of the next subdivision plus the shortest (straight-line) path from the end of that subdivision (i.e., from the next check-point) to the destination curve. Since no subdivision traversal-time estimation error data will have been gathered for the single estimated straight-line shortest path being used for Check #1, no confidence interval for this path is constructed. Instead a single path-traversal-time estimate, $\Delta\tau_{trav,C1}$ (computed as described above for any general path; Eq. 5.6) is determined and used.

The expected arrival-time via this alternative path, then, would be the sum of the current time, $T_{current}$, the expected traversal-time over the next subdivision, $\bar{\varepsilon}_{trav}$, and the expected traversal-time over the straight-line shortest-path portion, $\Delta\tau_{trav,C1}$. The upper and lower limits for the arrival-time confidence interval would be built around this expected time value. Since the only uncertainty information comes from the uncertainty associated with the single subdivision that must be traversed next, the upper and lower arrival-time confidence interval limits, $T_{C1,HI}$ and $T_{C1,LO}$, respectively, would, thus, be given by:

$$T_{C1,HI} = T_{current} + \bar{\varepsilon}_{trav} + \Delta\tau_{trav,C1} + \Delta_{\varepsilon} = T_{current} + \Delta\tau_{trav,C1} + T_{\varepsilon,HI}, \tag{5.10}$$

$$T_{C1,LO} = T_{current} + \bar{\varepsilon}_{trav} + \Delta\tau_{trav,C1} - \Delta_{\varepsilon} = T_{current} + \Delta\tau_{trav,C1} + T_{\varepsilon,LO}. \tag{5.11}$$

These arrival-time confidence interval limits must, therefore, be computed every time a robot arrives at a check-point and performs Check #1.

With $T_{C1,HI}$ and $T_{C1,LO}$ computed, the verification to be performed in Check #1 is that the robot does not run out of time to reach its destination curve even via its shortest-path from the next check-point. Thus, the upper-bound, $T_{C1,HI}$, of the confidence interval for the alternative path arrival-time is compared with a threshold on maximum arrival-time, $T_{C1,Max}$, given by:

$$T_{C1,Max} = T_{current} + \Delta t_{avail} + M\Delta t_{avail}, \tag{5.12}$$

where M is a user-specified percentage margin-of-error, so that $M \cdot \Delta t_{avail}$ represents a safety margin added over and above the time available to reach the destination curve. The condition to be satisfied in Check #1, then, is:

$$T_{C1,HI} \leq T_{C1,Max}. \tag{5.13}$$

Thus, each time a robot reaches a check-point on its initial path (after having passed the first check-point), the values for $T_{C1,HI}$ and $T_{C1,Max}$ are computed and the check given by Eq. 5.13 is performed first. Moreover, the actual time taken to traverse the subdivision that was just completed is recorded for inclusion in future computations of subdivision traversal-time uncertainty along the current path.

5.1.2.2 Check #2: Current Shortest-Path Feasibility

The purpose of a secondary check is to determine whether a feasible shortest path to the destination curve also exists from the robot's current position. This shortest path is estimated via a straight-line path from the robot's current position to its destination curve, Fig. 5.5, in the same manner as the straight-line shortest path computed in Check #1 from the next check-point to the destination curve. Thus, Check #2 serves as a counterpart to the shortest-path alternative determined in Check #1, with the exception that the next subdivision is not first taken. Gauging the feasibility of a shortest path from the robot's current position serves as yet another check of how critical a situation the robot is currently in.

To perform Check #2, the traversal-time, $\Delta\tau_{trav,C2}$, associated with this shortest path is estimated and compared with a suitable threshold. Computation of this traversal-time estimate is done in the same manner as for any general path as described in Section 5.1.1.2. As for Check #1, this threshold is based on the available time, Δt_{avail}, as well as a user-specified margin of error. Thus, rather than comparing the traversal-time estimate, $\Delta\tau_{trav,C2}$, directly with Δt_{avail}, the comparison is made instead with the threshold $\Delta t_{C2,Max}$, given by:

$$\Delta t_{C2,Max} = \Delta t_{avail} + M\Delta t_{avail} \ , \tag{5.14}$$

where *M* is the same user-specified percentage margin-of-error as that used in Eq. 5.12.

Since $\Delta\tau_{trav,C2}$ is only an estimate based on a straight-line path, incorporating such an additional margin is a sensible approach in order to avoid being overly strict for this check. The condition to be satisfied in Check #2, then, is:

$$\Delta\tau_{trav,C2} \leq \Delta t_{C2,Max} \ . \tag{5.15}$$

Thus, each time a robot reaches a check-point (after the first one) on its initial path, the value for the threshold $\Delta t_{C2,Max}$, and for the traversal-time estimate for the current-position-based shortest-path, $\Delta\tau_{trav,C2}$, are also computed. These values are used to perform the check given by Eq. 5.15.

5.1.2.3 Check #3: Destination Arrival-Time Feasibility

Both Checks #1 and #2 evaluate a sense of criticality of the situation that the robot is in by gauging the existence of feasible shortest paths. If both these checks fail, this may be an indication that the robot is in a critical situation where it has possibly deviated significantly from its assigned iso-probability curve so that, soon, even the shortest path may not allow the robot to reach its destination curve within the available time. In this case, the robot's path is re-planned, where the shortest path (i.e., the one that has minimum estimated traversal-time, $\Delta\tau_{trav}$) is accepted as the optimal path rather than the path that comes closest to Δt_{avail}. More details on path re-planning will be given in the next section.

If either Check #1 or #2 fails (but, not both), this is an indication of a relatively lower level of criticality. Thus, a regular re-optimization and re-planning for the remainder of the path is conducted, as will be explained in Section 5.1.3.

In the above two scenarios, Check #3 is not performed. However, if both Checks #1 and #2 pass, then Check #3 must be conducted. The purpose of Check #3 is to determine how early or late a robot can be expected to arrive at its destination curve, relative to the arrival-time originally estimated, if it were to continue to follow its current path, Fig. 5.6. A ±3σ confidence interval can be constructed for the estimated arrival-time error, relative to the estimated arrival-time of ($t + \Delta\tau_{trav}$), Eq. 5.6, that can be expected by travelling along the remainder of the path.

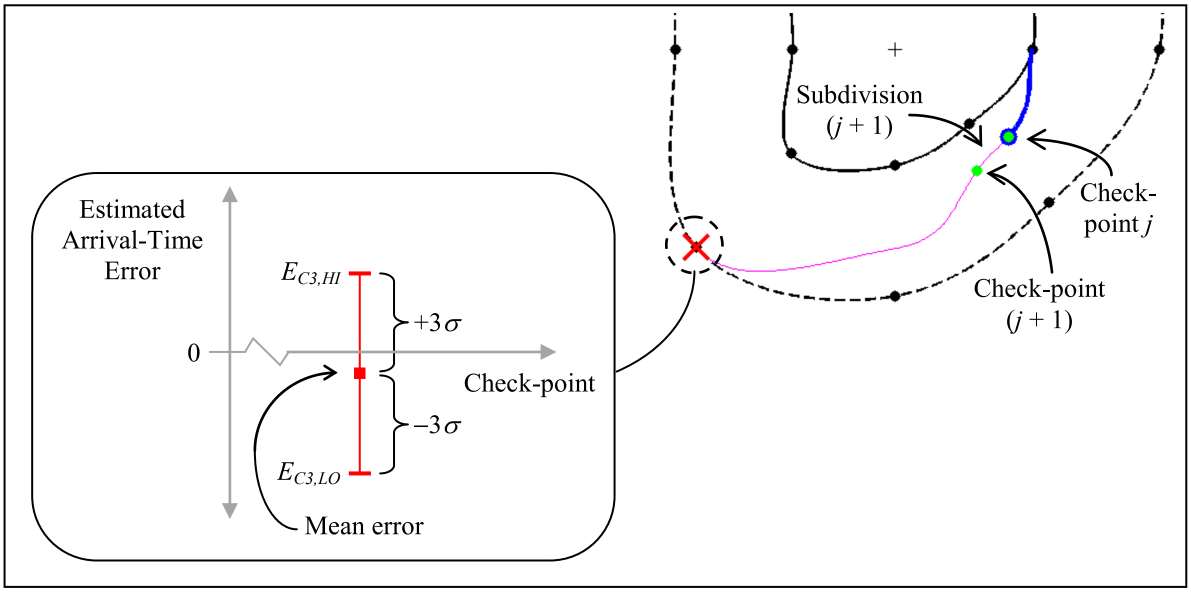


Figure 5.6. Arrival-time error confidence interval computed for Check #3.

To construct this confidence interval, the subdivision traversal-time error data that is gathered as the robot passes through each check-point on its path is used. However, although this error-data yields an approximation of the error in traversal-time estimate per path subdivision, it would be unrealistic to assume that such an approximation will continue to be valid for all the remaining subdivisions on the robot's current path. Thus, the uncertainty in the arrival-time estimation error over the remaining path is computed based on the uncertainty in the subdivision traversal-time estimation error over the next immediate path subdivision only. This computed uncertainty is represented in terms of a corresponding ±3σ confidence interval.

Referring to Fig. 5.6, let us assume that robot Rb_i has just reached Check-point j along its initially planned path, where $j > 1$. At this point, the robot would have gathered individual subdivision traversal-time error data, $\varepsilon_{trav,k}$, $k \in [1, j]$. Let $\bar{\varepsilon}_{trav}$ and s_ε represent the mean and standard deviation of this error data, respectively. The total traversal-time error that would have been accumulated up to Check-point *j* is given by the summation of all the individual traversal-time errors computed for all the subdivisions that have been traversed by the robot so far. Moreover, one can expect there to be some traversal-time error that will be experienced over the next subdivision, Subdivision (*j*+1), that the robot must traverse. The traversal-time error that will be experienced over this next subdivision can be estimated by $\bar{\varepsilon}_{trav}$, and the uncertainty in this estimate can be represented by the confidence interval $\bar{\varepsilon}_{trav} \pm \kappa \frac{s_{\varepsilon}}{\sqrt{j}}$, [124]. For a confidence level of $(1-\alpha)$, the value of κ is given by:

$$\kappa = \begin{cases} t_{\nu,\alpha}, if\ j < 30 \\ z_{\alpha}, if\ j \geq 30 \end{cases}, \tag{5.16}$$

where $t_{\nu,\alpha}$ is the *t*-value corresponding to a *t*-distribution with $\nu = (j-1)$ degrees of freedom and which leaves an area of α to the right, and z_{α} is the *z*-value corresponding to a *z*-distribution which leaves an area of α to the right. For the purposes of obtaining a $\pm 3\sigma$ confidence interval for a normal distribution assumption for the traversal-time errors, a confidence level of $(1 - \alpha) = 0.9973$ must be used [124].

Thus, in the proposed approach to computing the error in the arrival-time estimate for the current path for Check #3, it is assumed that only Subdivision (*j*+1) contributes traversal-time error, and the rest of the subdivisions along the remainder of the path will have a traversal-time equal to $\Delta\tau_{trav,k}$, $k \in [(j+2), N_{SD}]$ as originally estimated during initial path planning, without error. With these assumptions, a confidence interval for the arrival-time estimation error for the remaining path can be computed, where the upper and lower limits $E_{C3,HI}$ and $E_{C3,LO}$, respectively, are given by:

$$E_{C3,HI} = \left(\sum_{k=1}^{j} \varepsilon_{trav,k}\right) + \bar{\varepsilon}_{trav} + \kappa \frac{s_{\varepsilon}}{\sqrt{j}}, \tag{5.17}$$

$$E_{C3,LO} = \left(\sum_{k=1}^{j} \varepsilon_{trav,k}\right) + \bar{\varepsilon}_{trav} - \kappa \frac{s_{\varepsilon}}{\sqrt{j}}. \tag{5.18}$$

Upon computing the upper and lower confidence interval limits, the evaluation that must be performed for Check #3 is to compare these limits to corresponding upper and lower arrival-time error thresholds, respectively. Since the approach used above is to compute arrival-time estimation error based only on the predicted traversal-time error and uncertainty over the next subdivision, while making no such prediction over all the other subdivisions thereafter, it is clear that the accuracy of such an assumption would be poorer near the beginning of the path (where relatively more subdivisions are still pending to be traversed) than near the end.

Furthermore, when the robot is at check-points near the beginning of the path, it would have traversed relatively fewer subdivisions and would, therefore, have gathered relatively fewer traversal-time error data, compared to check-points later on along the path. Thus, there would be considerably more uncertainty in the arrival-time error estimates made early on along a path, which would manifest itself as a relatively wider confidence interval (computed according to Eqs. 5.17 and 5.18).

As the robot travels further along its path, it can be expected that this confidence interval computed at each successive check-point would gradually decrease in width as more data is gathered and the uncertainty decreases. Indeed, the equations for $E_{C3,HI}$ and $E_{C3,LO}$ indicate this with their inverse relation to 'j'. This is a result of the fact that, with the assumption that the distribution of traversal-time errors is normal, the sampling distribution of average errors, $\bar{\varepsilon}_{trav}$, would also be normal with a mean of $\mu_{\bar{\varepsilon}} = \mu_{\varepsilon}$ and a variance of $\sigma_{\bar{\varepsilon}}^2 = \frac{\sigma_{\varepsilon}^2}{n}$, where n represents the number of data-points gathered.

Therefore, the variance of the sampling distribution, and, hence, the uncertainty in the arrival-time estimation error, would decrease according to ($1/n$). It would be reasonable, then, to construct upper and lower confidence-interval thresholds that narrow correspondingly (i.e., according to the function $1/n$). These upper and lower thresholds would indicate the maximum and minimum allowable errors, respectively, on the arrival-time estimates that are made at each check-point along a robot's path. A typical plot of the arrival-time estimation error confidence intervals, and corresponding upper and lower thresholds, for a path having ten check-points, would look like that shown in Fig. 5.7. In this plot, n refers to the number of check-points passed (and, therefore, to the number of subdivisions traversed, and also the number of error data-points gathered).

To compute upper and lower thresholds conforming to the form $1/n$ in order to conduct Check #3, a threshold function, $f(j)$, is defined, having the general form:

$$f(j) = \frac{a}{j^b}. \tag{5.19}$$

The upper and lower thresholds would be symmetric about the zero-error mark, and would each be given by its own function, $E_{C3,Max}(j)$ and $E_{C3,Min}(j)$, respectively, as:

$$E_{C3,Max}(j) = f(j) = \frac{a}{j^b}, \tag{5.20}$$

$$E_{C3,Min}(j) = -f(j) = -\frac{a}{j^b} = -E_{C3,Max}(j) \tag{5.21}$$

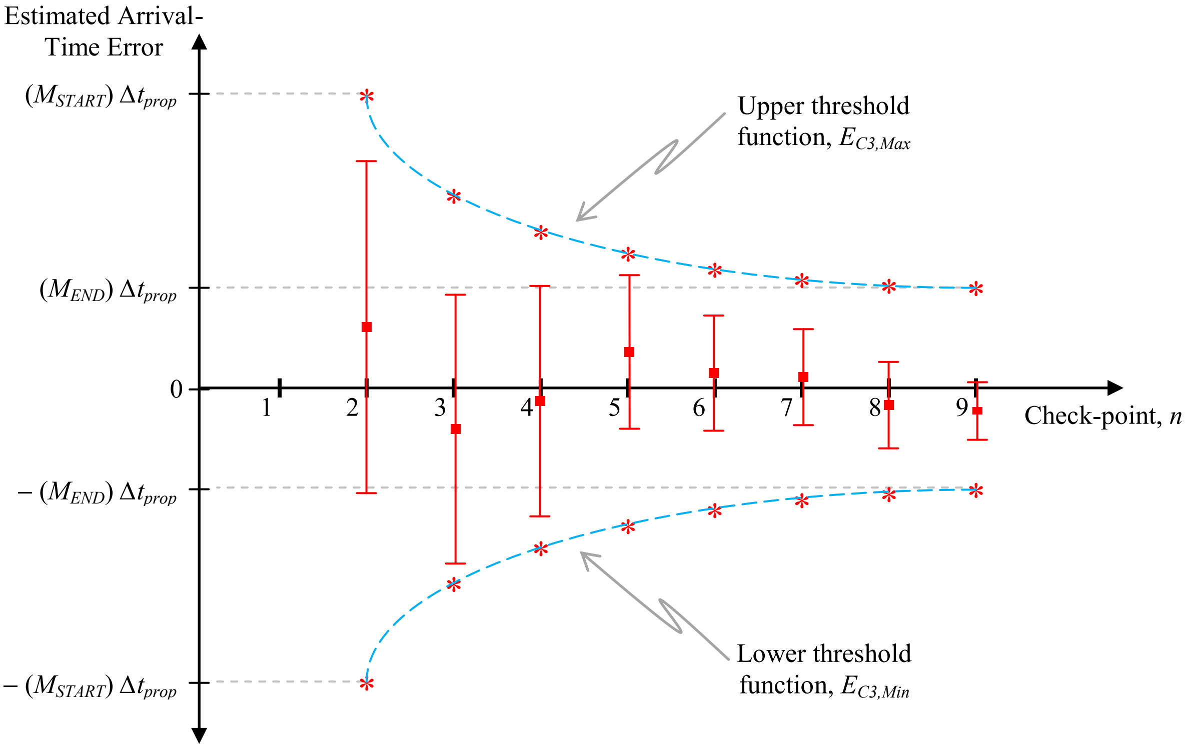


Figure 5.7. Plot of typical arrival-time error confidence intervals and corresponding upper and lower thresholds.

These thresholds are interpreted as allowable errors in the arrival-time estimate for a given path, and are expressed in terms of fractions of the curve-propagation time-interval, Δt_{prop}, which is the total time that a robot has to reach its destination curve from the start of its initially planned path.

To compute the parameters a and b, and to, thereby, obtain a specific functional definition for the upper and lower thresholds, two constraints are dictated.

First, at Check-point j = 2, only two path-subdivisions would have been traversed, resulting in only two error data-points. As such, a user-specified percentage margin of error, M_{START}, for the confidence interval derived based on these two error data-points is allowed. In the simulations for this Thesis, a value of M_{START} = 100% was used, since a large uncertainty can be expected at the beginning, given that only two data points are available, and that the underlying error distribution being normal is also an assumption. This implies that the maximum and minimum allowable arrival-time estimation errors are taken to be:

$$E_{C3,Max}(2) = (M_{START})\Delta t_{prop} \;=\; (100\%)\Delta t_{prop} \;= \Delta t_{prop}\,, \tag{5.22}$$

$$E_{C3,Min}(2) = -E_{C3,Max}(2) = -\Delta t_{prop}\,. \tag{5.23}$$

Next, at the second-last Check-point j = (N_{SD} – 1) (the last check-point before arrival at the destination curve), another user-specified percentage margin of error, M_{END}, is allowed. Since uncertainty should have decreased near the end of the path, M_{END} must be less than M_{START}. In this Thesis, a value of M_{END} = 10% was used, giving the additional constraints:

$$E_{C3,Max}(N_{SD}-1) = (M_{END})\Delta t_{prop} \;=\; (10\%)\Delta t_{prop} \;= 0.1\Delta t_{prop}\,, \tag{5.24}$$

$$E_{C3,Min}(N_{SD}-1) = -E_{C3,Max}(N_{SD}-1) = -0.1\Delta t_{prop}\,. \tag{5.25}$$

Given the constraints in Eqs. 5.22 to 5.25, the values for parameters a and b in Eqs. 5.20 and 5.21 can be computed. This results in fully-defined functions, $E_{C2,Max}(j)$ and $E_{C3,Min}(j)$, which output the upper and lower arrival-time error thresholds, respectively, to be used for Check #3 when the robot arrives at any given check-point j at the end of path Subdivision j. Thus, the conditions given below must be satisfied for Check #3 to pass:

$$E_{C3,HI} \leq E_{C3,Max}(j), \tag{5.26}$$

$$E_{C3,LO} \geq E_{C3,Min}(j). \tag{5.27}$$

The user would select the values of M_{START} and M_{END} based on how conservative they wish to be with Check #3. Specifying smaller values represents a more conservative outlook where the corresponding tighter thresholds would likely trigger more Check #3 fails due to the resulting increased ‘sensitivity’ of this check. In contrast, larger values for these percentages would correspond to greater liberality and likely fewer failed checks.

If Check #3 passes, namely, the robot is expected to arrive at its destination within an acceptable estimated arrival-time error confidence interval, the robot is allowed to continue on its current path. Otherwise, the path is re-planned as described in Section 5.1.3 below.

5.1.3 Path Re-Planning

Since external factors in a WiSAR search scenario result in uncertainty in the arrival-time for a robot at its destination curve using its initially planned path, regular path-optimality checks are required. Path optimality is checked every time a robot completes a subdivision and reaches a check-point along its current path, Fig. 5.5. Only when the path evaluation conducted at these check-points indicates a possibly sub-optimal path, path re-planning is initiated.

Path re-planning represents the third phase of the proposed robot-path planning method, and can be triggered under two circumstances: (*i*) ‘non-critical failure’ (i.e., when Check #1 or Check #2 fail individually, or when both Check #1 and Check #2 pass, but Check #3 fails), and (*ii*) ‘critical failure’ (i.e., when both Check #1 and Check #2 fail). When path re-planning is triggered, a new optimal path for the robot is planned leading from the robot’s current position to its assigned destination curve. The objective of path re-planning is to find a new such path that can be traversed within the available time, Δt_{avail}. Since check-points are placed at intermediate points along the path, it is clear that Δt_{avail} will be less than Δt_{prop} whenever a path re-plan is triggered after the robot has left its origin curve.

5.1.3.1 Path Re-Planning Following a Non-Critical Failure

A Check #3 failure, or a failure in either Check #1 or Check #2 individually, requires a regular path re-optimization and re-planning. Namely, a new path must be found which can be traversed within Δt_{avail} time. However, the scenario for path re-planning is nearly identical to that for initial path planning. The robot’s current position is known, a destination curve has been assigned and established, and a destination position on this destination curve must be found such that the corresponding path that is constructed between the robot and this destination position can be traversed within an estimated time as close as possible to a specified desired traversal-time of Δt_{avail}.

Moreover, for any given destination position, a path can be constructed in exactly the same manner as the path construction procedure described in Section 5.1.1.1, with the only caveat that the angular distances, α_k and α_{TOT}, must continue to be measured from the ray passing through the initial point on the origin curve from which the robot began its path in the current curve-propagation time-interval, Fig. 5.3. Therefore, an optimal path is re-planned in the same manner as the initial (optimal) path is planned, except that the starting point of the path is taken to be the current position of the robot, and the first-derivative constraint to be used for the first cubic-polynomial segment of the path is taken to be the current orientation of the robot. With these modifications, the optimization defined by equations 5.3 to 5.5 is simply re-applied, where Δt_{avail} is the *remaining* available time until the next curve propagation.

Thus, if a robot is at check-point j along its initial path, and $\Delta t_{trav,k}$ represents the actual traversal time experienced for the subdivisions $k \in [1, j]$ completed so far by the robot, then:

$$\Delta t_{avail} = \Delta t_{prop} - \left(\sum_{k=1}^{j} \Delta t_{trav,k} \right). \tag{5.28}$$

This, then, is the value of Δt_{avail} that is used in the objective function in Eq. 5.3 for the path optimization during path re-planning. In addition, $\Delta \tau_{trav}$ in Eq. 5.4 would represent the estimated time required to traverse the new, re-planned path only, corresponding to the new destination point determined. Once the robot's path has been re-planned, the robot continues searching by travelling along this new path. As per the recursive approach outlined in Fig. 5.1, the method returns to Phase 2, where path implementation and evaluation resume as discussed in Section 5.1.2.

In order to be able to continue using all the subdivision traversal-time data gathered up to the path re-planning point, the length of each subdivision along the new path is kept the same as that used during initial path planning. Hence, the new check-points along the re-planned path are positioned based on this existing subdivision length, starting from the robot's current position.

When conducting Check #3 at all subsequent check-points along the re-planned path, the past subdivision traversal-time error data is used only to help in computing the statistics $\bar{\varepsilon}_{trav}$ and s_{ε} in Eqs. 5.17 and 5.18. Since the re-planned path will have a new associated traversal-time estimate, $\Delta\tau_{trav}$, only the new traversal-time errors that are experienced as the robot traverses the new subdivisions along the re-planned path are used to compute the term $\sum_{k=1}^{j} \varepsilon_{trav,k}$ in these two equations. After a path re-planning, then, Check #3 would estimate the arrival-time error, and corresponding confidence interval, for the most recent path planned, and the term $\sum_{k=1}^{j} \varepsilon_{trav,k}$ in Eqs. 5.17 and 5.18 would represent the total traversal-time errors that have accumulated up to the current check-point since the start of the latest path planned.

5.1.3.2 Path Re-Planning Following a Critical Failure

When both Check #1 and Check #2 fail, it represents a possible critical situation where, in the near future, even the shortest-path may not be sufficient to allow the robot to reach its destination curve within Δt_{avail} time. In this situation, the robot's path must be re-planned, but the shortest path is accepted as the optimal one, rather than the one that has a traversal-time closest to Δt_{avail}. This is done as a conservative approach to avoid risking the situation where the robot is unable to reach its destination curve on time. Namely, it is preferable that the robot arrives at its destination curve earlier than ideally desired, so that it is in the preferred position for the subsequent path-planning at the next curve-propagation, rather than arriving late.

As for the former case presented above, path re-planning in this latter case of critical failure resembles that for initial path planning very closely. Once again, a destination position on the destination curve must be found such that the corresponding path that is constructed between the robot's current position and this destination position can be traversed within an estimated time that must be optimized; in particular this traversal-time must be minimized. Additionally, for a specified destination position, path construction is still conducted as described in Section 5.1.1.1, with the angular distances α_k and α_{TOT} still measured from the ray passing through the initial point on the origin curve from which the robot began its path in the current curve-propagation time-interval, Fig. 5.3.

Therefore, the basic path-optimization formulation given by Eqs. 5.3 to 5.5 still applies. One modification that must be made is that the objective function must minimize only the estimated path traversal-time, $\Delta\tau_{trav}$. Second, since the shortest path is being found, the constraint given by Eq. 5.4 is no longer necessary – the traversal-time of the optimal path will simply be the minimum possible, and no upper bound needs to be specified. This, then, gives the modified optimization formulation for this latter case of (shortest) path re-planning as:

$$min(\Delta\tau_{trav}), \tag{5.29}$$

subject to:

$$P_{dest,i} \in F_{dest,i}. \tag{5.30}$$

Upon obtaining the optimal solution, the robot continues to search by travelling along the resulting shortest path, and Phase 2 resumes, Fig. 5.1. Once again, the subdivisions along the shortest path would also remain of the same length as that determined during initial path planning, so that previous subdivision traversal-time error data can be used. Moreover, all subsequent evaluations of Check #3 must be conducted exactly as described in the former case of path re-planning in Section 5.1.3.1 above.

It is important to note, however, that even though the initial path is optimized, and the remaining path upon reaching a check-point is re-optimized when required, there is still no guarantee that the robot will remain precisely on its iso-probability curve at all times and arrive at its destination point in exactly Δt_{avail} time. If the robot arrives early, it would simply continue to travel along its destination curve. This way, when the next curve-propagation occurs, the robot will be in the ideal location (i.e., on its assigned iso-probability curve propagated up to the current time).

If the robot is not able to arrive at its destination curve within the available time, then, the next path planned to the subsequent propagated destination curve is constructed from the robot's current position, wherever that may be. This ensures that although the robot was unsuccessful in arriving at its destination curve during the last curve-propagation, it could still reach its destination curve over the course of the current curve-propagation time interval, thereby 'catching-up' to its assigned, propagating iso-probability curve.

5.2 Path Planning for Centre Robot

Since iso-probability curves propagate outward from the LKP with time, and robots search by remaining on their assigned curves, an unaddressed region is formed by the search area bounded by the innermost curve. Since it is possible that the target could be present in this region during the search, the proposed MRC methodology designates one robot to this central area, referred to as the 'centre robot'.

When the search commences, the centre robot is initially deployed at the LKP. Just as for the other robots, its search path is planned one curve-propagation time-interval at a time. However, this path is composed of two parts. The first part is a path leading from the LKP to a destination point on the current, innermost, iso-probability curve, and the second part is a path leading from this destination point back to the LKP, Fig. 5.8. Construction of the first part of the path proceeds in the same manner as described in Section 5.1.1.1, with the initial position being the robot's current position at the LKP, the current innermost curve serving as the centre robot's destination curve, and a destination point on the destination curve found through an optimization.

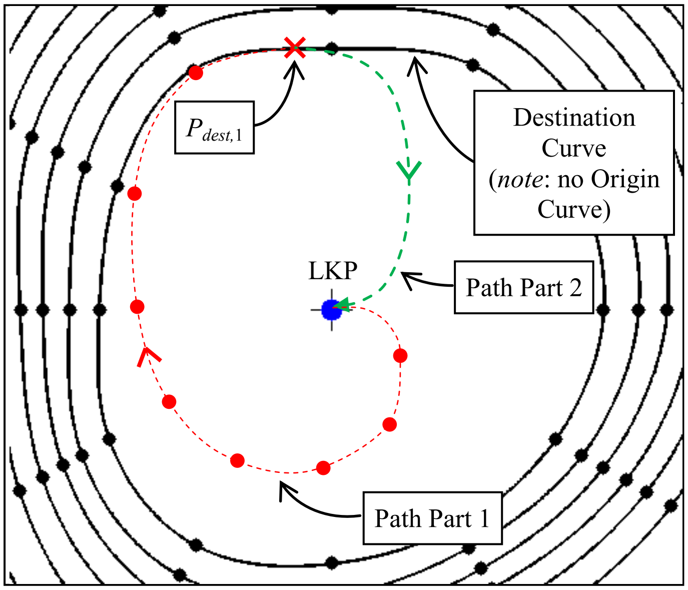


Figure 5.8. Path-planning for centre robot.

Since there is no origin curve for this robot, a destination curve cannot be found through the propagation of an iso-probability curve Δt_{prop} time into the future. Thus, the *current* innermost curve serves as the destination curve. Namely, the destination curve for the centre robot is the position of the innermost curve at the start of the most recent curve-propagation time-interval, which is also the origin curve of all the robots that have been assigned to the innermost curve. Nevertheless, a piecewise cubic polynomial path for the centre robot, from its current position to its destination curve, can still be planned, with path interpolation points still placed using the standard method described earlier involving the use of segmentation ratios.

Since an origin curve does not exist for the centre robot, the interpolation points are placed on the portion of the rays lying between the LKP and the corresponding control points on the destination curve. Thus, referring to Fig. 5.3, for any given ray used for locating a path interpolation point, the LKP would be point 'A' and the corresponding control point on the centre robot's destination curve would be point 'B'.

For the second part of the path, a path must be planned leading from the destination point on the destination curve back to the LKP, Fig. 5.8. Since the destination for this second part is the known LKP, no optimization is required. Instead, a smooth, single-polynomial path is planned back to the LKP, dependent upon the position and the heading-angle at the end of the first part of the path. In particular, when the centre robot reaches its destination point, $P_{dest,1}$, it would have an orientation in the direction defined by some unit vector, $\vec{u}_{P_{dest,1}}$. The single cubic-polynomial representing the second part of the path is computed with the conditions that the start of the polynomial must lie on $P_{dest,1}$ and must have a first-derivative tangent to $\vec{u}_{P_{dest,1}}$, while the end of the polynomial must lie on the LKP and must have a first-derivative tangent to $-\vec{u}_{P_{dest,1}}$. Using an ending first-derivative pointing in the opposite direction to the starting first derivative ensures that the resulting polynomial has a smooth, curved shape, without ripples, as it connects $P_{dest,1}$ back to the LKP.

Finding the optimal path for the centre robot entails determining the optimal destination point on the innermost iso-probability (destination) curve such that the total estimated time required to traverse *both* parts of the path is as close as possible to the available time, Δt_{avail}, which for the initial path will be curve-propagation time-interval, Δt_{prop}. This can be accomplished via the same optimization formulation given in Eqs. 5.3 to 5.5. In the objective function given by Eq. 5.3 and in the constraint in Eq. 5.4, however, $\Delta\tau_{trav}$ would represent the sum of the estimated traversal-times of both parts of the centre robot's path. Moreover, $F_{dest,i}$ in the constraint in Eq. 5.5 would represent the current innermost iso-probability curve that serves as the destination curve for the centre robot. With these changes, the optimization to find the path for the centre robot proceeds as described in Section 5.1.1.2. Path implementation and evaluation as well as path re-planning, when required, also proceed in the same manner as described above for any general robot path.

Upon returning to the LKP, the centre robot repeats this two-part path-planning process, using the heading-angle at which it arrived at the LKP to determine the starting trajectory of its next path. Since, in general, this heading-angle will be different from the one used for the previous path-plan, the end-result will be that the centre robot will travel along different paths each time, thereby exploring different parts of the central region within which it moves. However, it is important to note that the intent here is not to conduct exhaustive coverage of the central region, but rather to implement a type of random search of this region so that it does not remain completely unaddressed. If the user wishes to search this region in a more structured manner, a finer resolution of iso-probability curves combined with more robots would be required so that this region can be populated with iso-probability curves and searched according to the standard procedure described in Section 5.1.

5.3 Coping with Obstacles

One of the other complications that must be dealt with during path-planning is the presence of obstacles along a given path. Obstacles refer to smaller-scale obstructions within the terrain, such as trees, large boulders and rocks, small areas of non-traversable terrain (e.g., a small body of water, a swampy region, or a section of quick-sand), etc., which may not be indicated in the available terrain topology map. Such obstacles would affect a path's traversal-time and corresponding arrival-time because the robot would be required to navigate around them. If the location and boundary of an obstacle are known at the beginning of a path-planning endeavour, they can be used to appropriately plan the path to circumvent the obstruction and to account for the increased traversal-time and arrival-time that this circumvention would entail. However, when obstacles exist in the search area that are *a priori* unknown, it is entirely possible that a path may have been planned that passes directly through such obstacles, thereby creating a path-implementation problem. Namely, the robot must be provided with a strategy to make its way around such obstacles until it can return to its planned path.

Since *a priori* unknown obstacles would not have been accounted for during path planning, the delay caused by having to circumvent them would manifest itself as a large error in one or more subdivision traversal-time estimates, which, in turn, would affect the upper and lower arrival-time estimation error confidence interval limits that are computed at subsequent check-points (Eqs. 5.17 and 5.18). If these delays are significant enough, they would cause the upper and/or lower limits to surpass their corresponding thresholds, thereby triggering a path re-plan during Check #3 (in extreme cases, Check #1 and/or Check #2 would fail as well). Such instances of re-planning, then, would give the system a chance to account for the *a priori* unknown obstacle that was just discovered, and to create a new optimal path for the robot in question that can still allow that robot to arrive at its destination curve within the available time. Such situations, therefore, further justify the importance and necessity of Phase 2, namely, regular path checking and re-planning when necessary. This approach allows for a type of 'self-correction' mechanism embedded within the path planning method, striving to always keep each robot on its assigned curve in order to help ensure that the robots search in an optimal fashion with minimal disruption. *A priori* unknown obstacles, therefore, serve as another major source of arrival-time uncertainty associated with any given path, and the use of the Phase 2 and Phase 3 procedures in the path planning method allow for this uncertainty to be addressed effectively.

What is required in order to cope with obstacles, then, is a procedure for planning paths around *a priori* known obstacles, and a procedure for any given robot to circumvent, on-line, an *a priori* unknown obstacle it may happen to encounter along its current path. In both situations, the proposed approach is to implement a fixed detour strategy to circumnavigate an obstacle. For the case of an *a priori* known obstacle, this strategy can be implemented directly during path-planning. In particular, a path is first constructed to the selected destination point ignoring the known obstacle(s). Next, the points at which the path enters and exits each obstacle, as shown in Fig. 5.9 for an example circular obstacle, are determined. A path is planned such that it traces the obstacle boundary on the side leading to the shortest path around the obstacle until it rejoins the original path. In the example in Fig. 5.9, this side is clearly along the bottom of this particular obstacle, resulting in a path being planned that wraps around the boundary of the obstacle in a clockwise direction (relative to the centre of the obstacle).

When an obstacle is fully known, the correct side to choose and the circumvention path can be calculated precisely in order to plan the circumvention motion in its entirety before implementing. One may note that any check-points designated on the path that would have fallen within the obstacle would also get moved to the obstacle's boundary in such a way as to maintain the length of all path subdivisions the same. Figure 5.10 shows an example of a robot implementing a path that intersects multiple *a priori* known obstacles. The dashed red line indicates the original path before accounting for the obstacles to indicate how the side of each obstacle selected for circumvention is always the side leading to the shortest detour around the obstacle.

When a robot encounters an *a priori* unknown obstacle, a similar fixed detour strategy is implemented. The robot begins by following the boundary of the obstacle in the clockwise direction (relative to the centre of the obstacle). This direction is an arbitrary choice used to allow the robot to keep moving so that it may eventually circumvent the obstacle and return to its planned path. Since no information about the obstacle is available, this is still a reasonable approach. If, in implementing this detour strategy, the robot is able to return to its originally-planned search-path, then the robot continues along this path. Subsequent path-optimality checks conducted at check-points will dictate whether the path needs to be re-planned due to the impact of the obstacle circumvention.

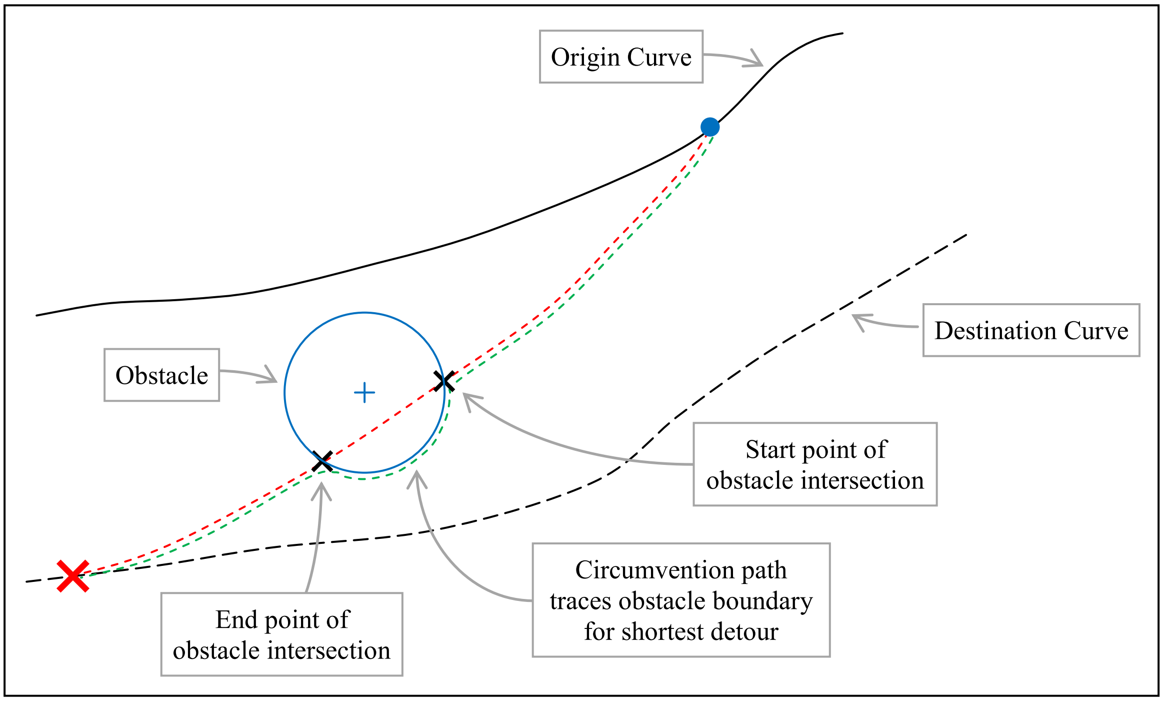

Origin Curve
Destination Curve
Obstacle
Start point of obstacle intersection
End point of obstacle intersection
Circumvention path traces obstacle boundary for shortest detour

Figure 5.9. Illustration of path planning around an *a priori* known circular obstacle.

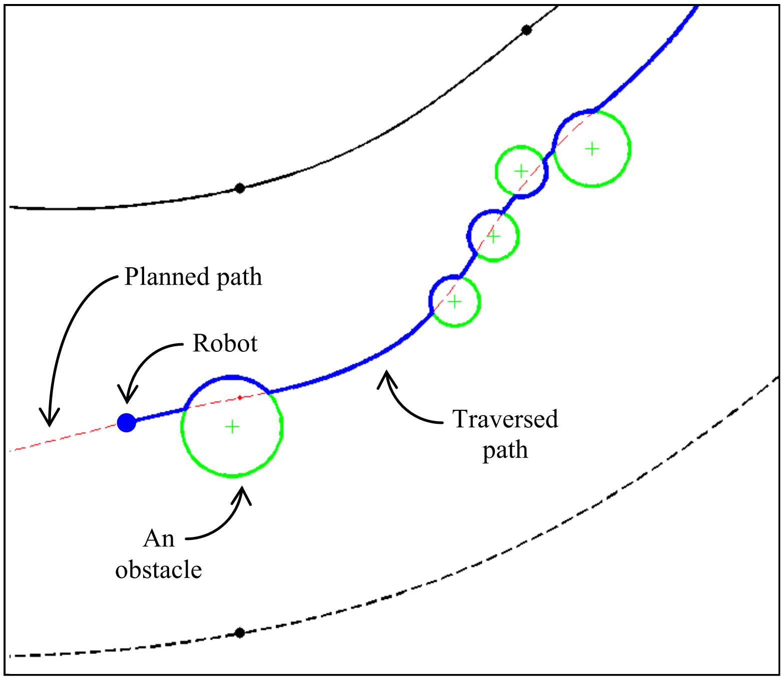


Figure 5.10. Example of robot circumventing *a priori* known circular obstacles.

A special case of *a priori* unknown obstacles arises when such an obstacle intersects the destination curve. When this happens, a robot, in the process of attempting to bypass the obstacle by traveling along its boundary in the clockwise direction, may end up reaching its destination curve, as illustrated in Fig. 5.11. However, once a robot reaches its destination curve in this manner, it would be wise to simply keep the robot on its curve. Since the robot has already reached its destination curve, there would be little sense in risking a possible chance of failure in path-planning by having the robot move away from its destination curve in an attempt to rejoin its original path, not knowing when or if that path will ever be reached in the available time. As such, the strategy employed in this situation is to simply keep the robot on its destination curve. Namely, once the robot reaches its destination curve and subsequently completes bypassing the obstacle, it continues by travelling along its destination curve until the next curve-propagation. This strategy helps to ensure that the robot will be on its destination curve when it comes time to propagate the curves once again and plan new paths. This is illustrated in Fig. 5.11. Here, the dashed red line indicates the originally planned path, while the dashed green line indicates the final path that the robot actually follows.

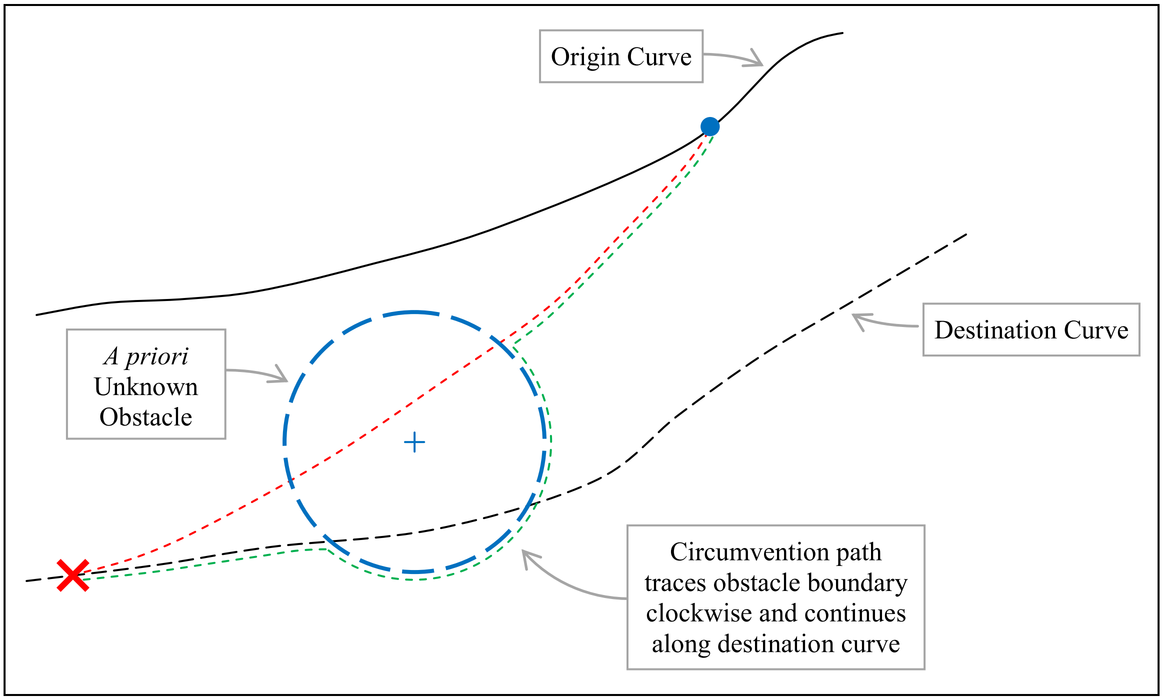


Figure 5.11. Illustration of a robot reaching its destination curve prematurely while attempting to bypass an *a priori* unknown obstacle.

5.4 Shortest-Path Planning for Re-Deployment

As discussed in Chapter 4, a re-deployment represents a change in the number and positions of iso-probability curves, and a re-assignment of robots to curves. Re-deployment can occur for two main reasons. First, the measure of benefit in adopting a new optimal deployment solution, computed for the next curve-propagation at the end of the most recent Δt_{prop} time-interval, could indicate that a re-deployment is necessary. Second, a clue can be found, requiring a change in the LKP to that of the clue's position, and the mandatory re-computation and implementation of the optimal deployment solution. However, having to stop the robots from following their current paths and return to their re-deployment positions before continuing the search represents a disruption to the search process. As such, to minimize any negative impact that this time spent in returning to curves would have, and to help ensure overall optimality of the ongoing search, robots are required to move to their newly assigned deployment positions on the curves as quickly as possible after a re-deployment. This necessitates the planning of the shortest paths from each robot's current position to its destination position.

In both cases where re-deployment occurs, robots may be located anywhere in the search area relative to their assigned iso-probability curves (i.e., inside the bounds of that curve, or outside).

Furthermore, since the number and positions of the iso-probability curves for the re-deployment solution would, in general, be different from the current deployment solution being used, the destination curve to which a path must be planned for any given robot may not represent a propagated position of that robot's current (origin) curve.

Moreover, for the case of a clue-find, a robot's current curve and its newly-assigned curve would, in general, not be concentric either. Therefore, interpolation points between the initial and destination positions cannot be placed along rays lining-up with control-points, as was described in Section 5.1.1.1 for the case of regular robot-path planning, Fig. 5.3. Instead, the approach proposed is to construct the shortest-path for each robot during instances of re-deployment using only a single cubic polynomial, which precludes the need to define any intermediate interpolation points. A third-order polynomial provides sufficient degrees of freedom to specify only initial and final position and first-derivative constraints, yet results in a smooth path.

However, determination of the shortest path for any given robot still requires an optimization. The objective is to find a destination position on the robot's assigned curve (i.e., the new destination curve) such that the estimated path traversal-time is minimized. Whether conducting a regular re-deployment during curve-propagation, or conducting re-deployment due to a clue-find, the deployment position to be assigned to each robot is taken to be the point at which the shortest-path from the robot to its assigned curve intersects that curve. Thus, an optimal re-deployment solution would indicate which curve each robot should be assigned to, and the shortest-path determined through optimization will indicate the deployment position that each robot must attain on its assigned curve. During this optimization, a search is conducted where the search engine moves through the search space of possible deployment positions in an attempt to find the one that minimizes the corresponding path's traversal-time.

For any given deployment position that is considered during this search, a path must be constructed using a single cubic polynomial in order to compute the corresponding traversal-time. To construct this path for a given deployment positions, the following constraints are applied to compute the single cubic polynomial: (*i*) the initial position of the path must be the robot's current position; (*ii*) the destination position of the path must be the deployment position being considered (that lies on the destination curve); (*iii*) the first derivative at the start of the path must be tangent to the current orientation of the robot, and; (*iv*) the first derivative at the end of the path must be tangent to the iso-probability curve at the destination position. With these constraints, the coefficients of the single cubic-polynomial path can be computed. Given this path, its traversal-time can be estimated as usual.

The optimization to determine the shortest path, then, is conducted using the same basic optimization formulation given by Eqs. 5.3 to 5.5 in Section 5.1.1.2, with the modifications that the objective function simply minimizes the estimated path traversal-time, and that the constraint limiting the traversal-time to the available time (Eq. 5.4) is not required. Hence, the formulation for path optimization given by Eqs. 5.3 to 5.5 is modified to the following for the purposes of finding the shortest path for a given robot:

$$min(\Delta\tau_{trav}), \tag{5.31}$$

subject to:

$$P_{dest,i} \in F_{dest,i}, \tag{5.32}$$

where $P_{dest,i}$ is the destination position that must be optimized for Robot Rb_i (i.e., the decision variable), and $F_{dest,i}$ is the destination curve that has been assigned to this robot. It is important to note that if a failure of both Check #1 and Check #2 occurs when an evaluation of path optimality is being conducted during Phase 2 of robot-path planning, the shortest-path that must necessarily be planned for the robot in question also uses the above optimization formulation.

Once robots reach their new deployment positions after completing their shortest-paths, the robots continue by travelling along their assigned curves. Path-planning subsequently resumes as usual at the next curve-propagation time-point (i.e., at the end of the current curve-propagation time-interval).

5.5 Chapter Summary

This chapter has presented a solution approach to address the robot-path planning sub-problem. The general strategy proposed is to plan search-paths that keep the robots on their assigned iso-probability curves in order to maintain the distribution of search resources dictated by the deployment solution that has been implemented.

A search-path is planned for each robot individually, one curve-propagation time-interval at a time. An optimization method is devised to plan each robot's path such that the estimated time taken to traverse the path is as close as possible to the curve-propagation time-interval. If, at the end of each curve-propagation time-interval, a robot is on its propagated iso-probability curve, then the objective of keeping each robot on its assigned curve will be maintained.

In order to address terrain uncertainties, including situations where the actual terrain topology is different from what is indicated on the available height-map as well as the presence of a priori unknown obstacles, check-points are delimited along an initially planned path. A series of three time-efficient checks are conducted at each check-point using traversal time information obtained from traversing each path sub-division up to a given check-point. These checks are used to gauge whether the remaining path will allow a robot to reach its destination iso-probability curve (i.e., its assigned iso-probability curve propagated one curve-propagation time-interval into the future) at the desired time. If the checks indicate that this is not likely, a new time-optimal path is planned with new check-points along it, and this path implementation/checking process repeats.

Path-planning for the centre robot proceeds in a similar fashion, expect that its starting point is the LKP and the destination iso-probability curve is the current, innermost curve in the curve-set that is being used. Upon reaching this destination curve, the centre robot returns to the LKP and this path-planning process repeats. The goal for path-planning for the centre robot is to ensure that the combined traversal-time of the path toward the destination iso-probability curve and the return-path to the LKP is as close as possible to the curve-propagation time-interval.

Path-planning during re-deployment entails planning the shortest path from a given robot's current position to its assigned iso-probability curve. The same optimization formulation as for regular path-planning is used, with the only modifications being that the objective function is to minimize the estimated path traversal-time and that the constraint limiting the traversal-time to the available time is eliminated.

Chapter 6

6 Testing Via Simulations

Throughout the process of developing the proposed MRC methodology for WiSAR, numerous realistic search simulations were performed. In order to demonstrate the overall tangible effectiveness of this devised method herein, one such illustrative example is detailed in Section 6.1. In addition to these simulations, the proposed methodology was further validated through its comparison with a traditional WiSAR method.

One of the novelties of the methodology that has been developed in this Thesis is its ability to address several complicating issues specific to the autonomous coordination of robots for WiSAR. Through the use of a probabilistic approach with expedited procedures for target-location prediction, robot deployment, and robot-path planning, the method is able to coordinate a robot team to find a mobile, non-trackable target, in an on-line manner. In doing so, it addresses the unique physiology and lost-person psychology of the target, while dealing with varying, partially-known, potentially boundless terrain containing *a priori* known and unknown obstacles and found clues. Due to the lack of an existing method in the literature that addresses these issues concurrently for autonomous robotic WiSAR, a comparable alternative approach had to be devised. This alternative method will be presented, along with the results of the comparison tests, in Section 6.2.

A test of robustness for the proposed methodology was also conducted. It is clear that the concept of iso-probability curves plays an important part in each of the three main tasks of the MRC method. However, the probabilistic information represented by these curves depends to a large extent on the nominal mean target-speed PDF available. Although this PDF can be derived from empirical, historical data on different categories of lost-persons, one can still expect there to be some deviation between what is assumed and what exists in actuality for the particular target being sought. Therefore, simulations were conducted to gauge the robustness of the methodology to an incorrectly-estimated nominal mean target-speed PDF. Section 6.3 presents the results of these simulations and discusses the effect of such discrepancies.

6.1 An Example Search Simulation

This section presents one example simulated robotic WiSAR search scenario, in which the target was assumed to be a hiker, lost in an environment where the terrain only permits a ground-based search. The terrain consisted of a rugged, mountainous region, which was generated using the Terragen Classic scenery generator program developed by Planetside Software [131]. In addition to the ability to randomly generate terrain with user-specifiable sizes of prominent features (i.e., hills and mountains), this software is also capable of generating a corresponding terrain height-map matrix. This matrix is formed by discretizing the terrain using a rectangular grid-array overlay of cells. For this simulation, a square height-map matrix containing 1025 × 1025 cells, representing a 5000 m × 5000 m area of terrain, was used, resulting in a cell width of approximately 4.88 m. The origin of the global coordinate system $(x_0, y_0, z_0) = (0, 0, 0)$, was taken to be at the centre of the terrain area. The origin was also assumed to be the position of the initial LKP. This choice maximized the amount of terrain available for the simulation in all directions outward from the LKP. The total time allotted for the search was 9000 s. Sections 6.1.1 to 6.1.4 below discuss the parameter settings used in this particular simulation, which are summarized in Table 6.1. Section 6.1.5 presents the results of the simulation.

6.1.1 Target Parameters

For the lost hiker, the nominal mean target-speed PDF was assumed to be given by a normal distribution with a mean of 0.14 m/s (0.50 km/h) and a standard deviation of 0.046 m/s. A nominal mean speed of 0.16 m/s was randomly selected for the target from this normal distribution by the simulator. Moreover, the target was assigned an initial heading-angle selected randomly from a uniform distribution ranging from 0° to 360°. The simulator randomly chose a heading-angle of 37° from this distribution. The simulation began with the target positioned at the LKP. The target subsequently began moving outward from the LKP at its nominal mean speed along its initial heading-angle.

Table 6.1. Summary of simulation parameter settings.

#	Parameter	Setting
1	Terrain height-map matrix size	1025 × 1025 cells
2	Available terrain area	5000 m × 5000 m
3	Coordinates of initial LKP	(0, 0, 0)
4	Maximum Search Time-limit, $T_{MaxLimit}$	9000 s
5	Mean of nominal mean target-speed PDF, μ_v	0.14 m/s
6	Standard deviation of nominal mean target-speed PDF, σ_v	0.046 m/s
7	Starting nominal mean target speed	0.16 m/s
8	Random target speed variations, $\pm 3\sigma_{\Delta v}$	±0.016 m/s
9	Starting target heading-angle	37°
10	Random target heading-angle variations, $\pm 3\sigma_{\Delta\theta}$	±30°
11	Time-period between target speed and heading-angle changes	120 s
12	Time-period between clue-drops	600 s
13	Total number of search robots, N_r	21
14	Maximum attainable robot speed	1.39 m/s
15	Maximum incline angle traversable by robot, $\gamma_{max,inc}$	60°
16	Maximum decline angle traversable by robot, $\gamma_{max,dec}$	–60°
17	Robot speed scale factor for maximum incline, $q_{max,inc}$	0.7
18	Robot speed scale factor for maximum decline, $q_{max,dec}$	1.0
19	Clue-detection radius (for each robot)	3 m
20	Target-detection radius (for each robot)	10 m
21	Target head-start	1800 s
22	Centre position of iso-probability curves, R_c	0.50
23	Inter-curve spacing of iso-probability curves, D_c	0.10
24	Minimum iso-probability curve position, R_{min}	0.050
25	Maximum iso-probability curve position, R_{max}	0.9987
26	Iso-probability curve propagation time-interval, Δt_{prop}	600 s
27	Maximum deployment-objective-function percentage difference threshold for triggering re-deployment, $\Delta Z_{1,max}$	4.5%
28	Number of check-points per robot path, N_{CP}	10
29	Percentage margin-of-error for path-optimality Check #1, M	10%
30	Percentage margin-of-error for path-optimality Check #2, M	10%
31	Confidence level for arrival-time estimation error confidence interval for path-optimality Check #3, $(1-\alpha)$	0.9987
32	Percentage margin-of-error at start of path for path-optimality Check #3, M_{START}	100%
33	Percentage margin-of-error at end of path for path-optimality Check #3, M_{END}	10%

In order to mimic drift in the target's motion, the target was subjected to random speed and heading-angle variations at regular time-intervals throughout the simulation. The random speed perturbations that were applied conformed to a normal distribution having variations of $\pm 3\sigma_{\Delta v}$ = ±0.016 m/s about the nominal mean speed. Thus, for an actual nominal mean target speed of v at any given time during the simulation, a speed change of Δv was selected randomly from this normal distribution of speed variations and applied to the target, resulting in a new target speed of $(v + \Delta v)$ m/s. Similarly, random direction perturbations were selected from a normal distribution having $\pm 3\sigma_{\Delta\theta}$ = ±30°. In particular, for an actual target heading-angle of θ at any given time during the simulation, an angle change of $\Delta\theta$ was selected randomly from the corresponding normal distribution of angle variations and applied to the target, changing the target's heading angle to $(\theta + \Delta\theta)°$. These speed and heading-angle changes were invoked every 120 s.

The simulation was also programmed such that the target would leave behind clues every 600 s. Clues were assumed to have a known position but an unspecified drop-time, requiring the system to approximate the time at which the clue was dropped using a straight-line-path conservative estimate, as described in Chapter 3 (see Fig. 3.10). For the purposes of this estimate, a maximum possible mean target speed of $v_{max} = \mu_v + 3\sigma_v = 0.14 + 3(0.046) = 0.28\ m/s$ was used to compute the estimate of the time at which the clue was dropped.

6.1.2 Search-Robot Parameters

A total of 21 ground-based robots were used in this simulation. One robot was assigned as the centre-robot, while the others were assigned to iso-probability curves. All robots had a maximum attainable speed of 1.39 m/s (5 km/h). Actual robot speeds varied throughout the simulation as a result of the influence of the terrain over which the robots travelled. In order to scale robot speeds based on terrain influence, a linear relationship between average ground surface slope angle, γ, and speed scale-factor, q, was assumed, as formulated in Eq. 3.8, with reasonable values selected for the relevant parameters. In particular, a maximum incline angle of $\gamma_{max,inc}$ = 60°, a maximum decline angle of $\gamma_{max,dec}$ = –60°, and corresponding speed scale-factors, $q_{max,inc}$ = 0.7 and $q_{max,dec}$ = 1.0, respectively, were used. One can note that with $q_{max,dec}$ = 1.0, it was assumed that the search robots could maintain a constant desired speed on all declines of grades less-steep than $\gamma_{max,dec}$ through the appropriate use of braking. In this way, it was possible for robots to move slower than their desired or maximum speed, but not faster.

All robots were assumed to carry on-board sensors capable of detecting clues as well as the target. During the simulation, all clues were assumed to be pieces of information indicating a past position of the target since the start of the search. These clues could be 'detected' by the robots when a robot approached within a particular distance around a clue. In particular, the robots were assumed to have a clue-detection radius of 3 m (extending in all directions from the position of a clue). Similarly, all robots were assumed to have a target-detection radius of 10 m.

6.1.3 Robot Deployment Parameters

Initial deployment of the robots was assumed to take place 1800 s after the target left the LKP, thereby giving the target an 1800 s head-start. This time-period represented the time taken to transport the robots to the search area and for them to be initially deployed. Since the nominal mean target-speed PDF was assumed to be a normal distribution, the centre position of any given set of iso-probability curves, for the purposes of implementing the deployment method, was R_c = 0.50. Moreover, the inter-curve cumulative-probability interval (i.e., inter-curve spacing), was taken to be D_c = 0.10. The minimum allowable curve position was taken to be R_{min} = 0.050 and the maximum allowable curve position R_{max} (i.e., the '100%' position) was taken to be the cumulative probability corresponding to $(\mu_v + 3\sigma_v)$, namely, 0.9987. The iso-probability curves were constructed using twelve uniformly-distributed rays, and the robots were assumed to be dropped directly onto their initial deployment positions on the initial, optimal set of iso-probability curves.

Throughout the simulation, a curve-propagation time-interval of Δt_{prop} = 600 s was used. At the end of each curve-propagation time-interval, the re-computed optimal deployment solution was compared to the existing deployment solution to determine if a re-deployment was necessary. A re-deployment solution was implemented only if the percentage improvement in the deployment optimization objective function (Eq. 4.6) exceeded a maximum percentage difference threshold, set arbitrarily to $\Delta Z_{1,max}$ = 4.5%.

6.1.4 Robot-Path Planning Parameters

All robot search-paths were constructed using piecewise cubic polynomials. With each iso-probability curve-propagation, a new, optimal path was planned for each robot from its current position to a destination point on its respective assigned destination curve, with traversal-time as close as possible to $\Delta t_{avail} = \Delta t_{prop} = 600$ s. All paths were planned with 10 check-points per path.

The three path-optimality checks were conducted every time a robot reached a check-point (after the first one) along its path. The path subdivision traversal-time estimation errors that were computed and gathered by each robot along its respective path were assumed to conform to a normal distribution. For Check #1, a percentage margin-of-error of $M = 10\%$ was used to compute the maximum allowable upper limit for the confidence interval on the estimated arrival-time via the straight-line shortest path (taken from the next check-point). This same percentage margin-of-error was also used for Check #2 to compute the maximum allowable arrival-time threshold for the straight-line shortest path (taken from the robot's current position).

With the normal distribution assumption for the estimated subdivision traversal-time errors, the upper and lower confidence interval limits for the destination point arrival-time estimation error (via the robot's current path) were computed using Eqs. 5.17 and 5.18, respectively. A confidence level of $(1 - \alpha) = 0.9987$ was used in order to establish a $\pm 3\sigma$ confidence interval at each check-point. Since 10 check-points were used for each robot path, the multiplier, κ, used in computing these limits (see Eq. 5.16) was given by the t-statistic, $t_{\nu,\alpha} = t_{(j-1),0.0013}$, where j refers to the check-point at which the computation was conducted. Moreover, the corresponding upper and lower threshold functions used for computing the maximum and minimum arrival-time errors in order to check the upper and lower confidence interval limits were computed using an inverse functional form, as given by Eqs. 5.20 and 5.21. The percentage margin-of-error parameters M_{START} and M_{END} used to compute the threshold functions were set to 100% and 10%, respectively.

6.1.5 Search-Simulation Results

At the start of the search, optimal initial deployment was carried out to determine the initial number and positions of the iso-probability curves, as well as the deployment positions of the robots on these curves. The resulting optimal solution dictated the use of six iso-probability curves, positioned at $R_1 = 25\%$, $R_2 = 35\%$, $R_3 = 45\%$, $R_4 = 55\%$, $R_5 = 65\%$, and $R_6 = 75\%$, as shown in Fig. 6.1. The optimal solution also dictated per-curve resource requirements of: $N_{1,25\%} = 3$, $N_{2,35\%} = 3$, $N_{3,45\%} = 3$, $N_{4,55\%} = 3$, $N_{5,65\%} = 4$, and $N_{6,75\%} = 4$. The optimal deployment, thus, consisted of one centre robot and 20 robots distributed among the six iso-probability curves based on the dictated resource requirements. The robots placed on each iso-probability curve were also spread out uniformly along the length of the respective curve, resulting in the final optimal initial deployment shown in Fig. 6.1. The robots were assumed to be dropped directly onto their initial deployment positions, and they were considered to be deployed and ready to commence searching at time t = 1800 s (i.e., the head-start of the target).

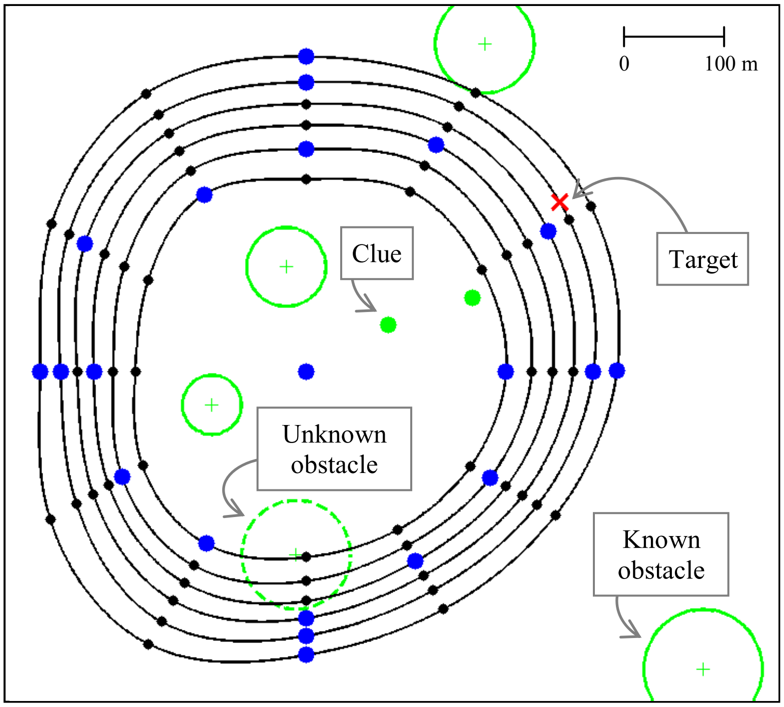


Figure 6.1. Optimal initial deployment of the search-robots at *t* = 1800 s.

During the search, eight iso-probability curve propagations were required. As well, only two clues were found. Re-deployment was executed three times, once after each of the clues was found and once during a regular iso-probability curve propagation. To illustrate the impact of re-deployment on the distribution of the robots, Fig. 6.2 (drawn at a lower scale than Fig. 6.1) shows the state of the system before the first case of re-deployment (i.e., before the first clue-find) at time t = 3000 s. The robots can be noted following their assigned paths that have been constructed based on the current optimal deployment solution being used, namely, the six aforementioned iso-probability curves.

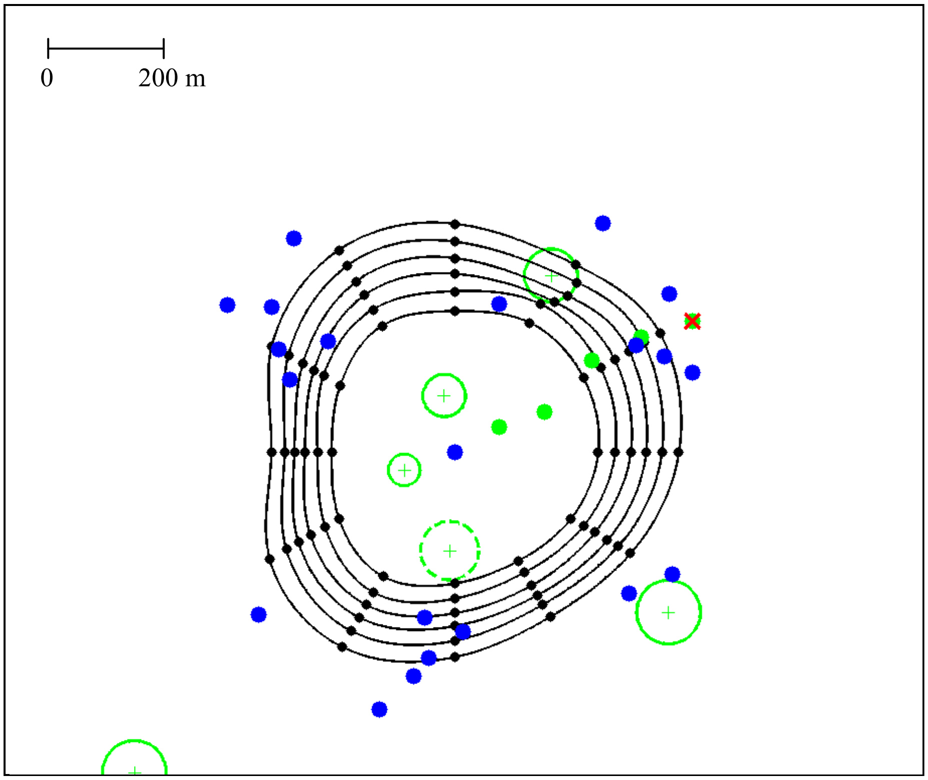


Figure 6.2. System state before first clue-find (*t* = 3000 s).

At time $t = 3267$ s, a clue is found and a re-deployment is conducted. The new optimal deployment solution is found to be ten iso-probability curves, positioned at $R_1 = 5\%$, $R_2 = 15\%$, $R_3 = 25\%$, $R_4 = 35\%$, $R_5 = 45\%$, $R_6 = 55\%$, $R_7 = 65\%$, $R_8 = 75\%$, $R_9 = 85\%$, and $R_{10} = 95\%$, respectively, with the new LKP at the position of the clue. Figure 6.3 shows the system state at $t = 3300$ s, a few seconds after the clue is found and the new deployment solution has been implemented. The robots can be seen moving towards their newly-assigned deployment positions. Figure 6.4 shows the system state at time $t = 3600$ s, at which point most of the robots have reached the general vicinity of the new set of iso-probability curves, and some have already reached their new deployment positions and have resumed searching by travelling along their respective assigned curves.

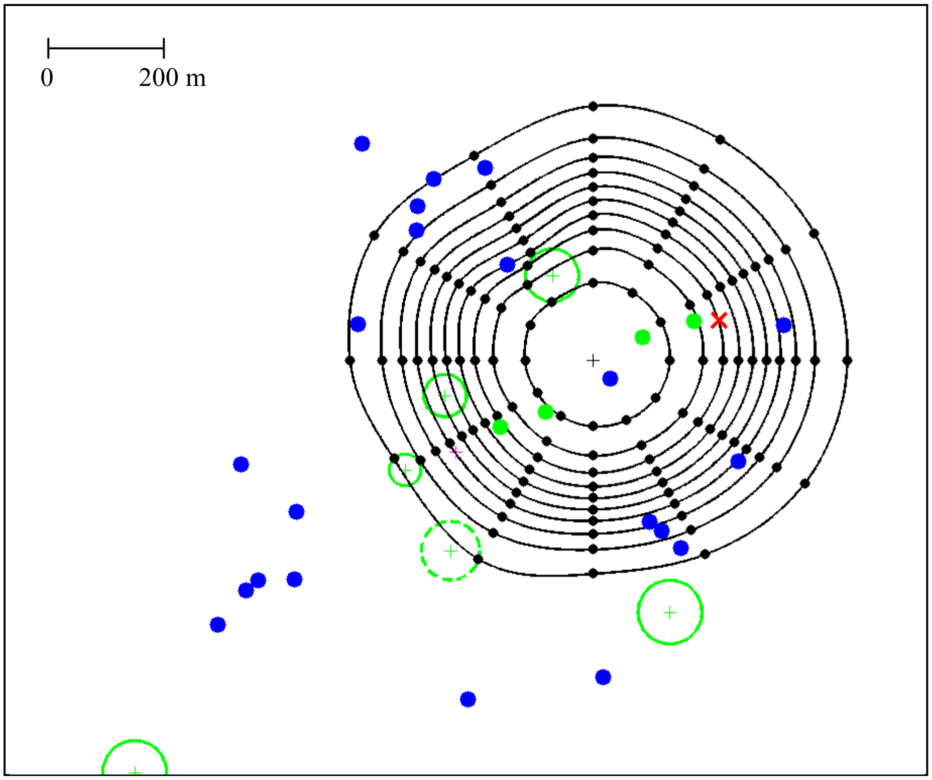


Figure 6.3. System state shortly after first clue-find, with new optimal set of iso-probability curves shown (*t* = 3300 s).

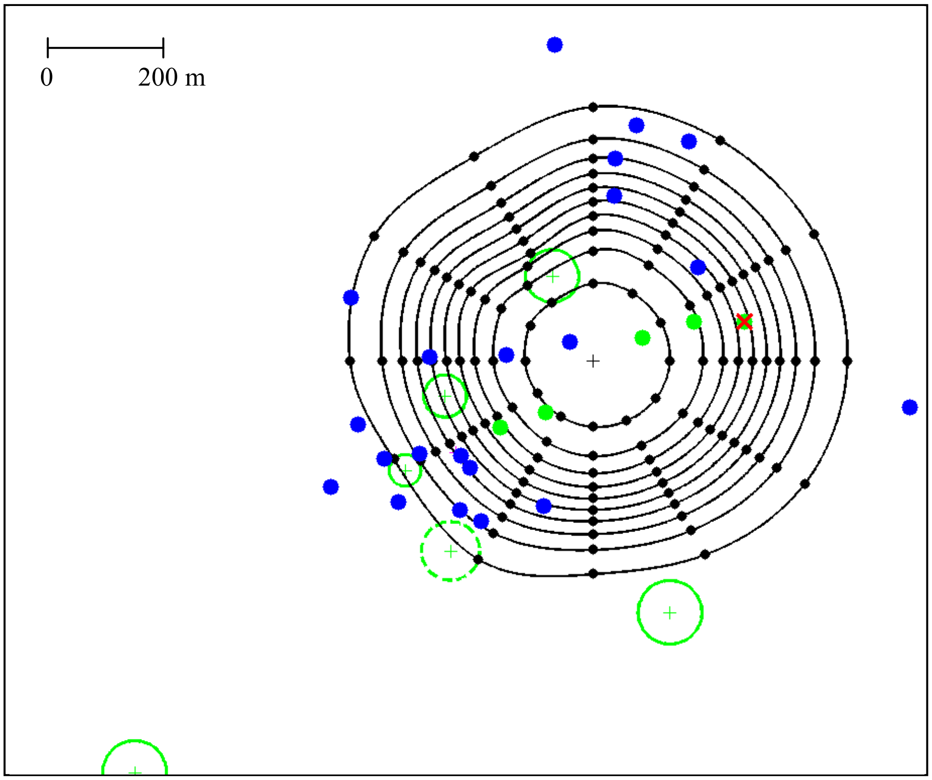


Figure 6.4. System state shortly after first clue-find, with new optimal set of iso-probability curves shown (*t* = 3600 s).

This example simulation illustrates how a search would typically progress. Upon implementing an optimal initial deployment solution, the robots commence the search via the paths that are planned through the recursive three-phase path-planning method. Paths are planned for each robot individually, in successive pieces, each corresponding to one curve-propagation time-interval. Position and first-derivative continuity constraints imposed on the piecewise cubic-polynomial paths ensure a smooth and connected search path throughout the search.

This particular search was successfully completed at time $t = 6937$ s, when the target was located by Robot #2 (Rb_2). For illustrative purposes, Fig. 6.5 shows the path (in blue) taken by Rb_2, along with the path (in red) taken by the target, throughout the entire search simulation. Key points of interest along the path have been indicated by a magenta '×', and labelled as 'P#' on the figure. Table 6.2 provides a walk-through of Rb_2's search-path, summarizing the relevant information corresponding to each of these key points.

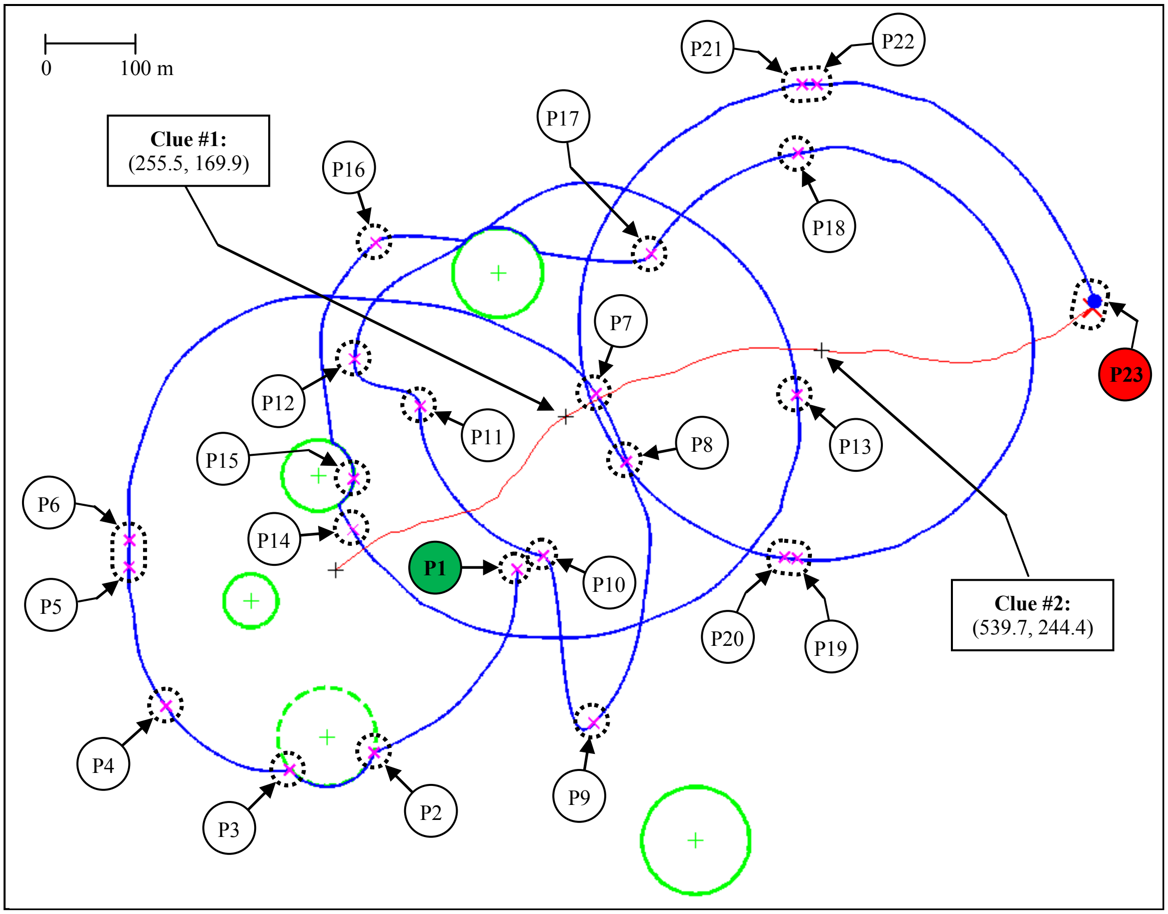


Figure 6.5. Search path traced by Robot #2 throughout the search simulation.

Table 6.2. Summary information at key points along Robot #2's path.

P#	Time [s]	Notes
P1	1800	Optimal initial deployment = 6 curves Rb_2 is assigned to curve #1 Path planned to a destination point on curve #1, 600 s into the future
P2	2021	Unknown obstacle encountered Rb_2 circumvents obstacle by following its boundary along clockwise direction
P3	2108	Rb_2 completes circumventing obstacle Original path is resumed
P4	2232	Rb_2 experiences Check #3 failure at check-point #7 Regular path re-planning is conducted and new optimal path is found
P5	2377	Rb_2 completes planned path Path along destination curve is, then, planned
P6	2400	**Curve-update #1** occurs Current deployment solution is still optimal and remains unchanged New path to next destination curve is planned
P7	2934	Rb_2 completes planned path Path along destination curve is, then, planned
P8	3000	**Curve-update #2** occurs Current deployment solution is still optimal New path to next destination curve is planned
P9	3267	**Clue** found at (255.5 m, 169.9 m) Re-deployment occurs; new optimal deployment solution = 10 curves Rb_2 remains assigned to curve #1 **Shortest-path** to future curve #1 is planned
P10	3426	Rb_2 completes planned shortest path Path along destination curve is, then, planned
P11	3600	**Curve-update #3** occurs Re-deployment occurs; new optimal deployment solution = 9 curves Rb_2 remains assigned to curve #1 **Shortest-path** to future curve #1 is planned
P12	3678	Rb_2 completes planned shortest path Path along destination curve is, then, planned
P13	4200	**Curve-update #4** occurs Current deployment solution is still optimal New path to next destination curve is planned
P14	4750	Rb_2 completes planned path Path along destination curve is, then, planned
P15	4800	**Curve-update #5** occurs Current deployment solution is still optimal New path to next destination curve is planned
P16	5012	**Clue** found at (539.7 m, 244.4 m) Re-deployment occurs; new optimal deployment solution = 10 curves Rb_2 remains assigned to curve #1 **Shortest-path** to future curve #1 is planned
P17	5249	Rb_2 completes planned shortest path Path along destination curve is, then, planned
P18	5400	**Curve-update #6** occurs Current deployment solution is still optimal New path to next destination curve is planned
P19	5986	Rb_2 completes planned path Path along destination curve is, then, planned
P20	6000	**Curve-update #7** occurs Current deployment solution is still optimal New path to next destination curve is planned
P21	6589	Rb_2 completes planned path Path along destination curve is, then, planned
P22	6600	**Curve-update #8** occurs Current deployment solution is still optimal New path to next destination curve is planned
P23	6937	**Target is found** by Rb_2 Search is complete

Throughout the search, the proposed path-planning method was able to address unforeseen situations, including encounters with a known and an unknown obstacle, and two clue-finds, while still ensuring that the robot arrived at its assigned curve at the start of each curve-propagation time-interval, thereby maintaining synchronicity with its propagating iso-probability curve. In this way, all 21 robots were able to conduct the search in a coordinated and on-line manner, responding appropriately with the continuously changing probabilistic target-location prediction information. Figures 6.2-6.4 further illustrate how the re-deployment method is able to effectively re-distribute the search resources, when required, while the search is in progress.

This and other similar simulations that were conducted, therefore, demonstrate the ability of the proposed overall MRC methodology for WiSAR to function effectively in an on-line manner. Based on data from other such simulations that were performed, Table 6.3 summarizes typical computing times associated with the different components of the methodology. The computation-time data has been organized according to the number of rays employed as well as the number of robots used. These times are based on the use of a single computer equipped with an Intel Pentium E5200, 2.50 GHz, Dual-Core processor. The simulations were conducted with non-optimized code programmed in Matlab, in which all processes were performed sequentially. As expected, computation times for initial curve construction, curve propagation, initial deployment, re-deployment, and path planning, all generally increase as the number of rays and the number of robots increase. More rays correspond to more control points that need to be positioned in order to construct and propagate the iso-probability curves, thus increasing computation time. Greater numbers of rays also mean more piecewise segments required for constructing and optimizing the path for each robot. Under the modest computing conditions used for these simulations, the computing times tend to range from a few seconds up to a minute when 12 rays are used. Even when as many as 100 rays are used, computation times for most tasks are still around a few seconds, with certain tasks such as re-deployment understandably requiring around 3 minutes due to the optimizations involved.

During practical implementation, one would employ parallel computing through multiple, high-speed processors to further expedite the computation tasks involved, which can be expected to further reduce these observed times. Moreover, tasks such as re-deployment and path planning that must be conducted at the pre-designated Δt_{prop} time-intervals to match curve-propagation could also be computed ahead of time using the expected final positions of the robots based on their current paths. It should also be noted that although Matlab is a convenient programming tool to use due to the presence of pre-built functions for many typical routine computation tasks, it does not have the efficacy of more preferable, object-oriented programming languages such as C++, which would be a better choice for implementation purposes. All in all, the proposed methodology is still viable as an on-line approach.

The simulation presented in this section, and others that were conducted, have demonstrated how the procedures corresponding to target-location prediction, robot deployment, and robot path planning are all able to work together seamlessly to create an overall search methodology that intelligently and autonomously coordinates a robot team to find an unobserved and mobile target. In the next section, performance validation of the methodology is conducted via comparison with an alternative robotic search method.

Table 6.3. Typical computation times for different components of the proposed methodology.

Number of Rays	Number of Robots	Computation Times [s]									
		Initial Deployment	Initial Curve Construction	Robot Path Planning						Curve Propagation	Re-Deployment
				Average (Per Robot)	Maximum (Per Robot)	Minimum (Per Robot)	Path Optimality Check #1	Path Optimality Check #2	Path Optimality Check #3		
12	6	1.13	0.49	3.52	10.59	1.58	0.0053	0.0022	5.00×10^{-5}	0.16	2.24
	11	7.47	0.92	3.12	9.86	1.47	0.0050	0.0020	3.75×10^{-5}	0.32	13.34
	21	11.83	1.36	3.48	13.77	1.60	0.0053	0.0021	7.50×10^{-5}	0.45	21.73
	41	11.06	1.23	3.09	17.60	1.48	0.0040	0.0021	9.90×10^{-5}	0.42	24.52
40	6	3.59	1.42	10.81	35.83	4.48	0.0056	0.0022	1.50×10^{-4}	0.47	6.72
	11	26.08	3.11	10.07	33.48	4.50	0.0051	0.0022	5.00×10^{-5}	1.04	42.74
	21	38.16	4.75	12.99	63.05	4.94	0.0050	0.0022	5.00×10^{-5}	1.61	63.79
	41	39.26	4.51	16.34	272.46	4.48	0.0052	0.0026	3.83×10^{-5}	1.42	68.00
100	6	9.18	3.29	23.57	75.14	10.33	0.0054	0.0032	4.00×10^{-4}	1.17	16.63
	11	60.82	7.52	23.60	75.43	11.00	0.0054	0.0023	5.00×10^{-5}	2.81	107.77
	21	90.10	10.47	24.81	98.00	11.59	0.0053	0.0021	5.88×10^{-5}	3.53	158.40
	41	89.30	10.33	22.24	106.20	11.42	0.0052	0.0023	4.15×10^{-5}	4.06	175.31

6.2 Comparison to Non-Probabilistic Approach

In order to further demonstrate the benefit that the proposed MRC methodology provides, its performance was compared to that of an alternative method. A key novelty of the proposed method that distinguishes it from other MRC methods is its use of the concept of iso-probability curves to represent probabilistic information on the target's location at any given time. Moreover, the method has been designed specifically for autonomous WiSAR, thereby addressing pertinent issues including: the influence of terrain based on target physiology, the influence of target psychology, the impact of found clues, the effect of *a priori* known and unknown obstacles, and the incorporation of all these influences into a time-varying and propagating target-location prediction over the search area. Many attributes of the design of this method have also been tailored to facilitate on-line implementation. In this way, the proposed method explicitly addresses needs for autonomous WiSAR that other MRC methods do not. Therefore, due to the lack of existence of a comparable method for autonomous robotic WiSAR in the literature, an alternative method had to be developed with which to compare the one devised in this Thesis.

Since the proposed use and representation of probabilistic information for the purposes of autonomous MRC for WiSAR constitutes the most unique feature here, to best reveal the advantage provided by implementing this approach, the alternative method that was developed for comparison was designed as a non-probabilistic one. By comparing performance to a method that does not use probabilistic information in the manner developed in this Thesis, the relative amount of benefit provided by the proposed means of incorporating probability can be made evident. In the following sections, the alternative method that was devised will first be presented, followed by a description of the comparison simulations conducted, and presentation and discussion of the results obtained.

6.2.1 An Alternative Search Method

The non-probabilistic method that was devised resembles a type of grid-search technique. Without obtaining probabilistic information about the target's motion and explicitly performing target-location prediction during a search, the most logical approach would be to use a search technique that attempts to apply uniform coverage over the search area. In so doing, one effectively makes the assumption that the probability of target presence is uniform over the search area – a sensible choice when no other information is available. The evenly-spaced, sweeping search of the area that a grid-search provides would, therefore, be the ideal approach in such a scenario. Indeed, grid-searching is a traditional approach used in SAR operations with live personnel [135], and served as inspiration in the development of this alternative search method.

The method starts the search by placing all the robots at the LKP. The robots, then, move radially outward with uniformly distributed heading-angles, as shown in Fig. 6.6 for the case involving 5 robots. Clearly, the more robots that are used, the better (i.e., more thorough) coverage of the search area can be made. The search area that the robots must cover through this grid-search technique is designated by the area bounded by the circle representing the conservative estimate of the maximum distance that the target would travel within the allotted search time. Thus, for a specified total available search time of $t = T_{MaxLimit}$, a circle is delimited with a radius, r_{SA}, equal to the distance that the target would travel in time $T_{MaxLimit}$ at its constant maximum speed. This maximum speed, v_{max}, is taken to be the speed corresponding to $(\mu_v + 3\sigma_v)$ on the nominal mean target-speed PDF, giving a search area radius of:

$$r_{SA} = (v_{max})(T_{MaxLimit}). \tag{6.1}$$

The region contained within this circular boundary represents the conservative estimate of the search area that the robots must cover.

Of course, it is entirely possible that the robots could reach this search-area boundary before the end of the available search time. If this happens, the strategy employed dictates that the robots travel along this boundary a specific amount and, then, return to the LKP along new radial paths. The amount of travel along the boundary that is performed by a robot, and the portion of the boundary along which it travels, depends on the initial trajectory of the robot and the total number of robots used (which influences the portions of the boundary that the other robots would cover).

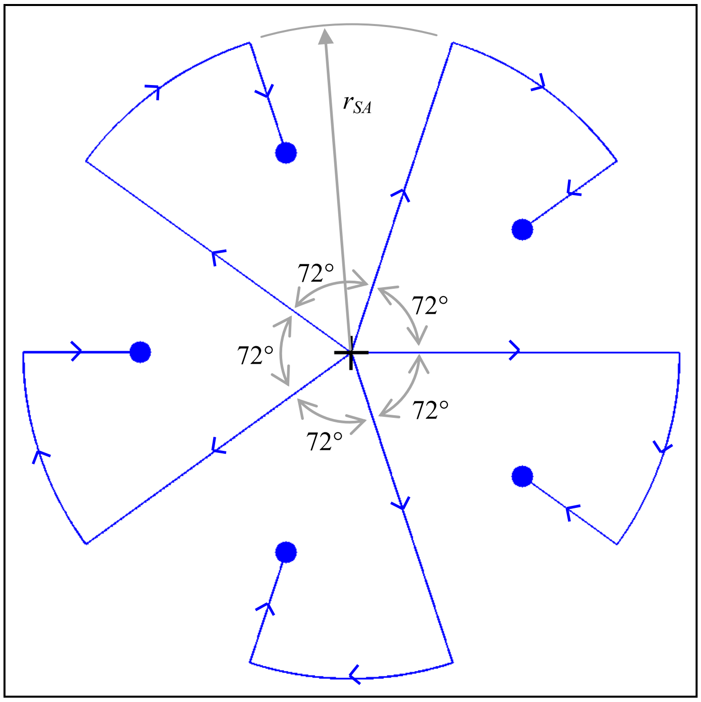


Figure 6.6. Illustration of the alternative search method with 5 robots.

If a given robot, Rb_i, begins at the LKP and travels in the direction having angular position θ_i (i.e., along an imaginary ray at angle θ_i), it would reach the boundary at some point $P_{dest,i}$, as illustrated in Fig. 6.7. If N_r robots are being used, then, the uniformly-distributed heading-angles established at the start of the search would result in an angular spacing, $\Delta\theta$, between adjacent heading-angles of:

$$\Delta\theta = \frac{360°}{N_r}. \tag{6.2}$$

For example, in the case of N_r = 5 robots shown in Fig. 6.6, $\Delta\theta$ is 72°. To continue creating a uniform coverage pattern after reaching position $P_{dest,i}$, robot Rb_i would travel clockwise (matching the general direction of travel used in the proposed method) along the boundary by an amount corresponding to an angular distance of $\Delta\theta/2$, Fig. 6.7. At the point $P'_{dest,i}$ that robot Rb_i reaches after covering an angular distance of $\Delta\theta/2$, the robot returns along a straight-line path back to the LKP. This results in a return trajectory along an imaginary ray at angle θ_i', given by:

$$\theta'_i = \theta_i - \frac{\Delta\theta}{2}. \tag{6.3}$$

When all robots conform to this search strategy, the end result is the search pattern shown in Fig. 6.6.

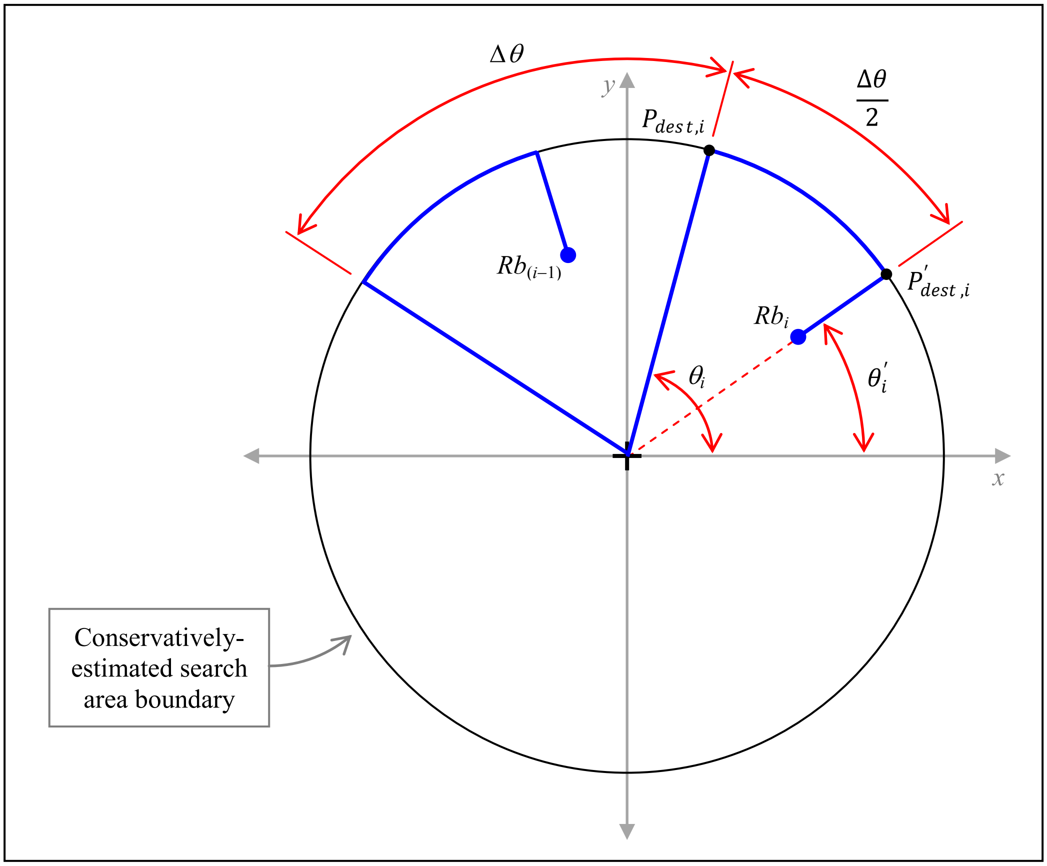


Figure 6.7. Computation of return trajectory for a robot under the alternative search method.

If a robot reaches the LKP through its return-trajectory before the available search time, $T_{MaxLimit}$, has elapsed, then this same search pattern commences again. However, each robot, Rb_i, would move out from the LKP along a different trajectory than what it used previously. Thus, if, at the start of the search, robot Rb_i travelled along heading-angle $\theta_i^{[1]}$, then the next outward trajectory after returning to the LKP would be along the heading-angle $\theta_i^{[2]} = \left(\theta_i^{[1]} - \beta\right)$, where β is computed based on an estimate of how many 'return-trips' between the LKP and the search area boundary a robot can be expected to make within $T_{MaxLimit}$ time. In particular, if a robot can attain a maximum speed of $v_{Rb,max}$, then, as a conservative estimate, it could travel a total distance of:

$$d_{Rb,total} = (v_{Rb,max})(T_{MaxLimit}), \tag{6.4}$$

within $T_{MaxLimit}$ time.

Given that the radial distance of the circular search area boundary is r_{SA} (Eq. 6.1), a return-trip would have total length of ($2r_{SA}$). A conservative estimate of the total number of return-trips of length ($2r_{SA}$) that any given robot Rb_i can be expected to make throughout the search can be taken as:

$$N_{returnTrips} = roundUp\left(\frac{d_{Rb,total}}{2r_{SA}}\right), \tag{6.5}$$

where the function '*roundUp*(·)' indicates that the computed quantity must be rounded up to the next integer. To enable a uniform coverage grid-search pattern, then, the heading-angle increment, β, can be computed as:

$$\beta = \frac{\Delta\theta}{N_{returnTrips}}. \tag{6.6}$$

In general, then, for a robot Rb_i starting its k^{th} return-trip from the LKP, $k \in [1, N_{returnTrips}]$, its heading-angle outward from the LKP would be given by:

$$\theta_i^{[k]} = \theta_i^{[1]} - (k-1)\beta. \tag{6.7}$$

The initial heading-angle, $\theta_i^{[1]}$, can be chosen arbitrarily, with the constraint that the angular spacing between the heading-angles of adjacent robots must be $\Delta\theta$. Thus, if the first robot, Rb_1, is arbitrarily given an initial heading angle of $\theta_1^{[1]} = 0°$, then this constraint would fix the initial heading angles of all other robots as well.

All robot paths, therefore, are planned according to this fixed, *a priori* known pattern, for this alternative method. In this sense, this method can be considered to be a 'deterministic' one, and will be referred to as such in the following.

6.2.2 Simulation Set-Up and Procedure

In this comparison, the proposed MRC method will be referred to as the 'Probabilistic Method,' while the alternative approach described above will be referred to as the 'Deterministic Method.' To conduct this comparison, a total of 1000 random simulations were performed for each of these two methods. For fairness, both the probabilistic method and the deterministic method were implemented using the same number of robots, maximum search area, and search-time limit. Furthermore, no clues were used in the probabilistic method, in order to avoid any incidental advantage that the discovery of a clue may provide to either method by virtue of the chance-distribution of the robots at the time of the clue-find. In addition, for a similar reason, no terrain was used in either method. This was done to avoid any possible disadvantage to either method that may result from the influence of terrain on the robots' motion. If, for example, due to their assigned paths, robots happen to be distributed over a particularly rugged portion of the search area in either method, the robots would tend to travel slower due to the added effort required to traverse the terrain, possibly compromising search effectiveness relative to the other method which may not experience the same scenario. Thus, all robots were assumed to move on a flat, planar surface.

Both methods also used the same 21 search robots moving with a maximum attainable speed of 1.39 m/s. All robots were assumed to have a target-detection radius of 20 m. The target was given a head-start of 7200 s and moved with a nominal mean speed of 0.14 m/s, with random variations following a normal distribution with $\pm 3\sigma_{\Delta v}$ = ±0.014 m/s about the nominal mean speed. The initial target heading-angle outward from the LKP was selected randomly at the start of each simulation based on a uniform distribution ranging from 0° to 360°.

Throughout each search the target's heading-angle was also subject to random normal variations with a mean of 0° and $\pm 3\sigma_{\Delta\theta}$ = ±15°. The maximum available search time for each simulation run was set to 4000 s, beginning after the 7200 s head-start of the target, for a total possible target motion of $T_{MaxLimit}$ = 11,200 s. A search was considered to be unsuccessful if the target was not found within this time. All other simulation parameters were identical to those used in the example search simulation from Section 6.1, summarized in Table 6.2.

For the deterministic method, a total of 21 robots resulted in an initial heading-angle spacing of $\Delta\theta$= 17.14°. The initial heading-angle for robot Rb_1 was selected arbitrarily to be 0°, thereby fixing the initial heading-angles of all other robots accordingly. By Eq. 6.6 the heading-angle increment, β, was computed to be 3.43°, with the relevant parameters being: v_{max} = 0.15 m/s; r_{SA} = 1712.48 m (Eq. 6.1); $v_{Rb,max}$ = 1.39 m/s; $d_{Rb,total}$ = 15,555.68 m (Eq. 6.4); and $N_{returnTrips}$ = 5 (Eq. 6.5).

6.2.3 Simulation Results and Discussion

In earlier chapters it was explained how search-time and success-rate comprise two of the main attributes used to gauge the effectiveness of a search. As such, the comparison of the probabilistic and deterministic methods was based on observing the relative differences in the resulting average search-time and success-rate values for each method over the 1000 simulations. The average search-time is defined as the arithmetic mean of the search-times of all the successful runs for a method, and the success-rate is the fraction of runs that were successful with that method.

Table 6.4 summarizes the results, which show that the probabilistic method was able to attain a success-rate in locating the target over three times that of the deterministic one. As well, a nearly 30% relative improvement in average search-time for successful searches was observed.

Table 6.4. Summary of comparative search-simulation results.

	Probabilistic Method	**Deterministic Method**
Success-rate	684/1000	201/1000
Average Search-time [s]	1271	1811

A two-sample *t*-test was also conducted to test the statistical significance of this difference in search-times. The null hypothesis used for this test was to assume that the mean of the distribution of search-times under the probabilistic method and the deterministic method were the same. With this assumption, and using only the search-time data corresponding to the successful searches for the two methods, a *t*-value of 5.3882 was found, which corresponds to a near-zero *p*-value of about 7.48×10^{-6} %. This indicates a highly unlikely situation and, thus, refutes the null hypothesis, thereby indicating that the observed difference in average search-times was, indeed, statistically significant.

In particular, the average search-times experienced under the probabilistic method were, on average, significantly better (lower) than those observed under the deterministic method. With the main difference between these methods being the utilization of probabilistic information on target location, these significant observed benefits in search-time and success-rate can be attributed to the positive impact resulting from the use of iso-probability curves to represent this information. The results indicate that the iso-probability curves are able to effectively incorporate the relevant influences into the target-location prediction, and guide the construction of the search paths of the robots, thereby producing tangible benefits to the search.

As noted above, the two methods achieved different percentages of successful searches. Since a larger proportion of the 1000 simulations were successful runs for the probabilistic method compared to the deterministic one, the average search-time computed for the probabilistic method could be unfairly influenced due to an over-representation of successes with longer search-times. Namely, due to the greater efficacy of this method, it may have been able to achieve success even in scenarios where the target was found much later within the available search time, but where the deterministic method was unsuccessful.

Since such unsuccessful searches would have been ignored in computing the average search-time for the deterministic method, but the corresponding longer search-time for the probabilistic method would still have been counted in the determination of its average search-time, the final set of search-time data for the probabilistic method could include many longer search-times. As a result, the computed average would tend to be relatively higher for the probabilistic method, giving the impression of lower method effectiveness. Thus, perhaps a more unbiased means of comparing search-times that can mitigate such over-representation could be to compute search-time averages over the same number and relative quality of successful runs between the two methods.

In particular, since the deterministic method had the fewer total number of successful runs of 201 out of 1000, one could compare the average of the top 201 search-times of the two methods, respectively. To do so, the 684 search-time data-points for successful searches corresponding to the probabilistic method were ordered from smallest to largest. The average of the first 201 search-times from this list was computed and compared with the average of the 201 search-times corresponding to the deterministic method. These computations yielded average search-times of 276 s and 1811 s for the probabilistic and deterministic methods, respectively. The resulting 6.5 times search-time advantage for the probabilistic method lends further credence to the significance of the benefit provided by the proposed MRC methodology.

Although the simulations verify the expected conclusion that there is indeed significant advantage provided by conducting the search according to the overall proposed MRC methodology, it is important to realize that, given the probabilistic nature of the WiSAR problem scenario, there cannot be any guarantees for always finding the target. In the WiSAR problem, the target is not observed during the search, intermittently or otherwise, and, at best, one only has a probability distribution of average speeds of typical targets falling under different categories.

The best strategy under such circumstances can only be to estimate the boundaries within which the target could happen to be, as best as possible, and search within this area. The available probabilistic information must be used to logically distribute search-effort according to the relative likelihood of target presence among different regions of that area. The proposed MRC method successfully implements such a strategy. The observed better success-rate and better search-time achieved by this method validate the iso-probability curve concept used to handle probabilistic target information, and justify the strategies proposed for robot deployment and robot-path planning that make use of this construct.

6.3 Robustness to Discrepancy in Assumed Mean Target-Speed PDF

The iso-probability curves represent the up-to-date, available probabilistic information about the target's location at any given time, and play a key role in determining the optimal robot deployment and path-planning. As such, they have a strong influence on the effectiveness of the search that is conducted via the proposed MRC method. According to the proposed approach to target-location prediction, iso-probability curves are constructed based on available empirical data from which a nominal mean target-speed PDF is derived.

Therefore, one concern that can be raised is that circumstances may arise whereby a significant discrepancy exists between the actual, or 'true,' mean target-speed PDF that pertains to the target being sought and the assumed one that is used to compute the iso-probability curves. However, since the type of target being sought would always be known (e.g., an adolescent child, an experienced, healthy hiker in his/her mid-20s, etc.), and since all raw data used to construct the mean target-speed PDF comes from historical data that is categorized by target-type, such differences can be expected to be minimal. Nevertheless, it is instructive to consider how the proposed methodology would behave if the assumed and true mean target-speed PDFs were to, indeed, differ.

Robustness tests were, therefore, performed, to analyze the behaviour of the method under such discrepancies. In particular, search simulations were performed where the true (normal) mean target-speed PDF governing target-motion was intentionally scaled relative to the assumed (normal) mean target-speed PDF used to construct the iso-probability curves. To implement this test, a 'default' nominal mean target-speed PDF was, first, established to represent the assumed nominal mean target-speed PDF. This PDF was the same as that used in the example search simulation presented above in Section 6.1 – a normal distribution with a mean of $\mu_{v,assumed}$ = 0.14 m/s and a standard deviation of $\sigma_{v,assumed}$ = 0.046 m/s. The true nominal mean target-speed PDF was, then, constructed by multiplying the speed random variable, v, of the assumed PDF by a particular scale factor, q_{true}, and re-normalizing the resulting distribution. Thus, for a given scale-factor, q_{true}, the true PDF would be a normal distribution with a mean of $\mu_{v,true} = q\left(\mu_{v,assumed}\right)$ and a variance of $\sigma_{v,true}^2 = q_{true}^2\left(\sigma_{v,assumed}^2\right)$ [124].

For the robustness tests conducted, three scale factors were considered: $q_{\text{true},1}$ = 0.75, $q_{\text{true},2}$ = 1.00, and $q_{\text{true},3}$ = 1.50; each representing the multiplication factor by which the true mean target-speed PDF was scaled relative to the assumed PDF. A scale factor of 0.75 meant that the true mean target-speed PDF was scaled down (i.e., had a smaller $\pm 3\sigma_v$ interval range) relative to the assumed PDF, resulting in the assumed PDF being over-estimated relative to the true PDF. Likewise, a scale factor of 1.50 meant that the true mean target-speed PDF was scaled up (i.e., had a larger $\pm 3\sigma_v$ interval range) relative to the assumed PDF, resulting in the assumed PDF being under-estimated relative to the true PDF. The 1.00 scale-factor served as the benchmark case where the assumed and true PDFs were the same.

For each scale-factor scenario, 1000 simulation runs were conducted. The same parameter settings as those for the comparative simulations (Section 6.2) were used, except that the target was given an 1800 s head-start, and the available search time thereafter was set to 5400 s, resulting in a total available search time of $T_{MaxLimit}$ = 7200 s. Table 6.5 summarizes the test results.

When the mean target-speed PDF is significantly over-estimated (i.e., the true PDF is scaled down by 0.75 relative to the assumed PDF), the true PDF has a smaller mean and variance relative to the assumed one and is, therefore, contained within the assumed PDF. Hence, for these circumstances, one would still expect a similar success-rate value as for the ideal case since the target would remain within the boundaries of the area being searched at any given time. Furthermore, it is also observed that the corresponding average search-time is about the same as that for the benchmark case. With the true mean target-speed PDF scaled down relative to the assumed PDF, the resulting lower actual target speeds would mean that the target would tend to reside, on average, in the regions toward the LKP, where lesser search-effort is placed relative to the region near the mean of the assumed PDF. This is indicated in the illustration in Fig. 6.8 where the true PDF (shown in red) has the bulk of its probability mass to the left of the mean of the assumed PDF (shown in blue), where the density of iso-probability curves and the number of robots assigned to those curves would be lower than near the mean of the assumed PDF. This accounts for the somewhat increased search-time for the over-estimation scenario relative to the benchmark case.

Table 6.5. Summary of robustness test results.

Scale-factor	**0.75 (Over-estimation)**	**1.00 (Benchmark)**	**1.50 (Under-estimation)**
Success-rate	925/1000	932/1000	603/1000
Average Search-time [s]	263	237	333

A two-sample *t*-test confirms that this difference in average search-times is statistically significant to a minor degree. In particular, under the null hypothesis that the mean of the distributions of search-time for the over-estimation and benchmark cases are the same, a two-sample *t*-test reveals that the observed difference in average search-times between these two cases (i.e., 263 s – 237 s = 26 s) corresponds to a *t*-value of 1.6032, and the probability of this difference being greater than or equal to this observed value corresponds to a *p*-value of about 5.45 %.

As also noted in Table 6.5, when the mean target-speed PDF is significantly under-estimated (i.e., the true PDF is scaled-up by 1.50 relative to the assumed PDF), there is the possibility that the target could have a mean speed much greater than the maximum assumed, and may travel so fast as to move out of the area being searched. This would imply an increased number of failed searches among the 1000 simulations conducted (compared to the over-estimation and benchmark scenarios). With respect to search-time, since the actual mean target-speed PDF is scaled up, the resulting higher actual target speeds mean that, among the successful runs, the target would tend to reside, on average, in the regions further from the mean of the assumed PDF, where, again, lesser search-effort is placed relative to the region near that mean.

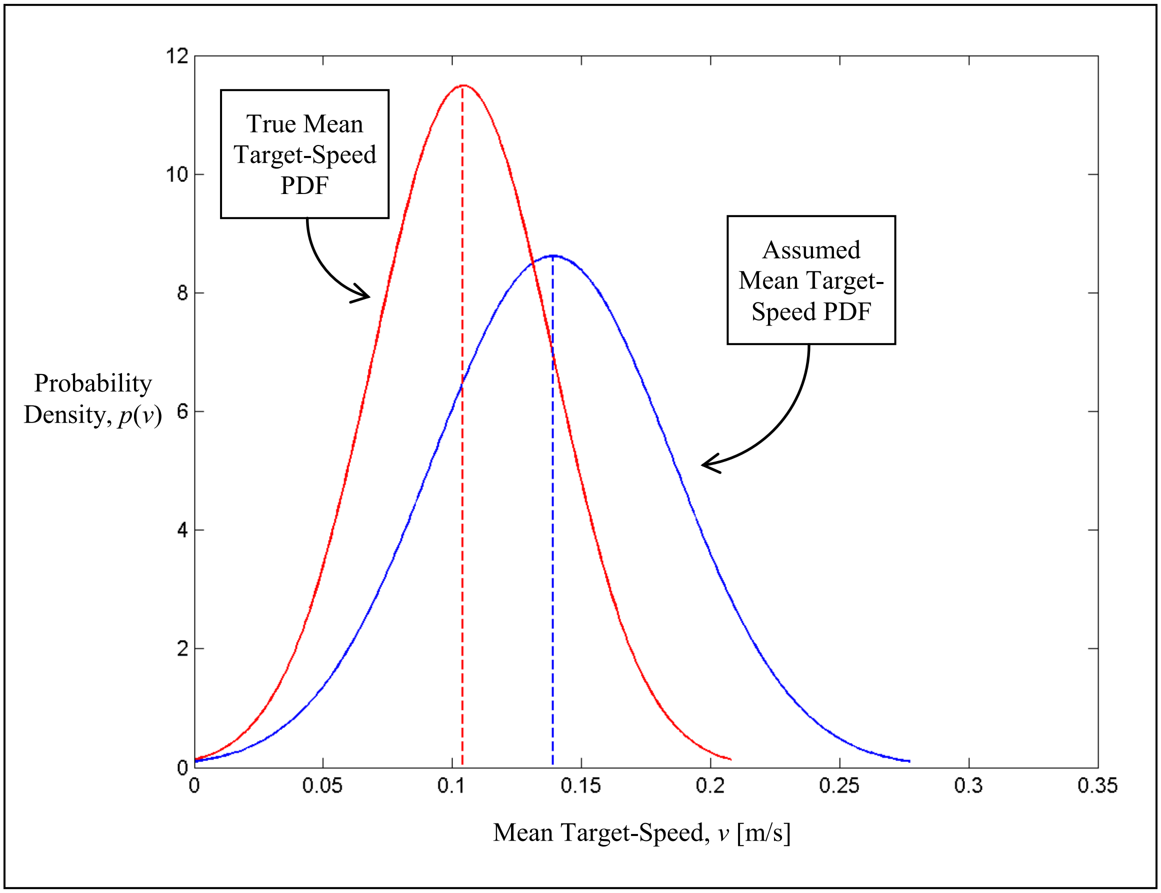


Figure 6.8. Relative positions of the true and assumed nominal mean target-speed PDFs for the over-estimation scenario (scale-factor 0.75).

Figure 6.9 illustrates the situation described above, where the bulk of the probability mass of the true PDF (shown in red) is to the right of the mean of the assumed PDF (shown in blue). Thus, the target would tend to have speeds that would place it in regions where relatively sparser iso-probability curves are established, resulting in relatively lesser search-effort and a corresponding increase in search-time. This expectation is confirmed by the results shown in Table 6.5, where the tests showed greater average search-time for the under-estimation scenario relative to that for the benchmark and the over-estimation cases. Also, as expected, this difference was found to be statistically significant based on a two-sample *t*-test. Under the null hypothesis that the mean of the distributions of search-time for the over-estimation and benchmark cases are the same, a two-sample *t*-test showed that the observed difference in average search-times for these two cases (i.e., 333 s – 237 s = 96 s) corresponds to a *t*-value of 3.5266 with an associated *p*-value of about 0.02 %.

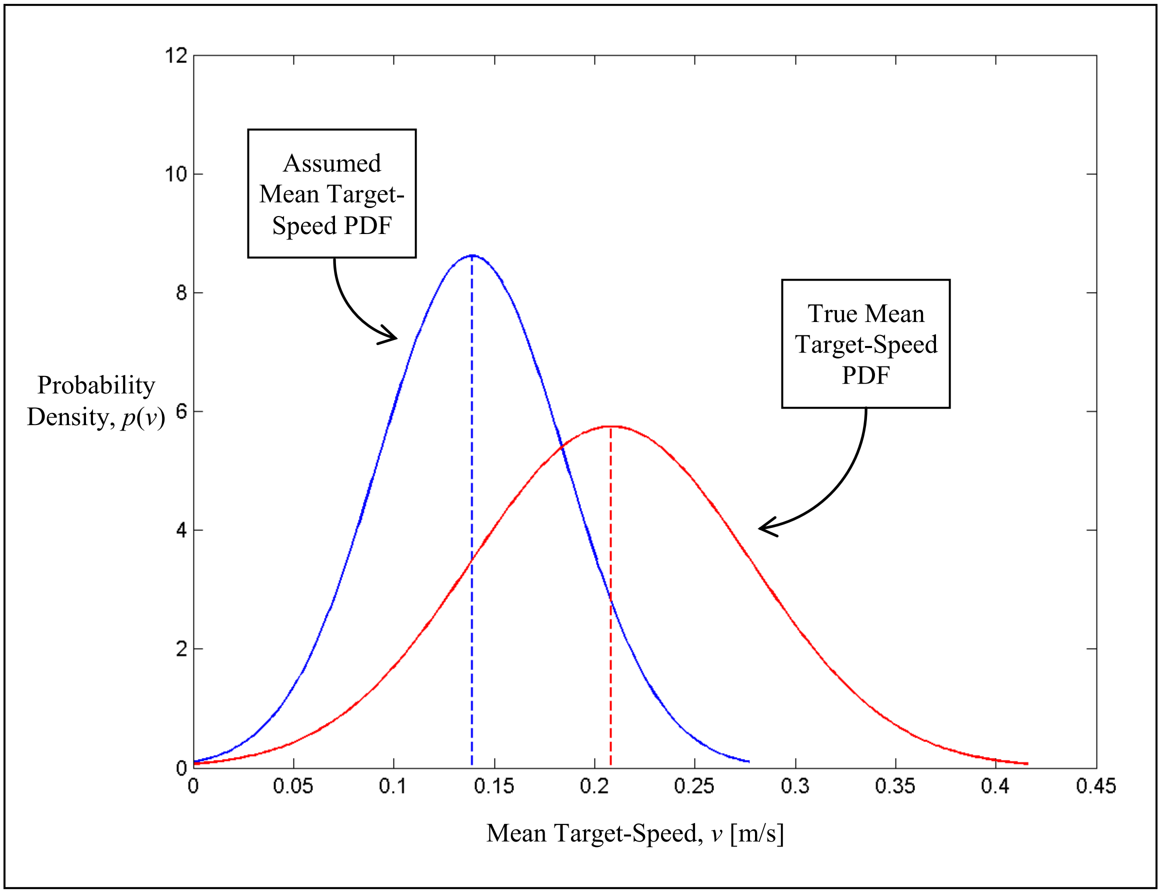


Figure 6.9. Relative positions of the true and assumed mean target-speed PDFs for the under-estimation scenario (scale-factor 1.50).

Hence, the results show that even when the true mean target-speed PDF differs from the assumed one by as much as 25% (for the over-estimation cases), the proposed method is robust, producing negligible changes in success-rates and search-times. For greater discrepancies between the true and assumed mean target-speed PDFs (especially for the under-estimation cases), the performance of the method, not surprisingly, can be expected to drop.

6.4 Chapter Summary

In this chapter, the results of simulations were presented to validate the overall MRC methodology proposed. Firstly, a detailed example search simulation was presented, illustrating all the different aspects of the methodology, including initial deployment, re-deployment, addressing found clues, path planning and re-planning, and coping with a priori known and unknown obstacles. The search-path of a single robot throughout the search simulation was depicted in its entirety to illustrate how, typically, a robot would move to conduct the search. Sample computation times for the different components of the methodology were also presented to indicate the on-line feasible of the proposed method.

A comparison to an alternative, grid-based, deterministic approach was conducted as well. By no using the iso-probability curves in the deterministic method, the comparison was able to illustrate the advantage of using this novel concept in the proposed probabilistic approach to autonomous robotic WiSAR. A total of 1000 search simulations were conducted for the proposed (probabilistic) method and the alternative (deterministic) method, with all simulation parameters being identical between the two method for each simulation run. The simulation results showed significant benefit in terms of search-time and success-rate when using the probabilistic method, thereby validating the proposed methodology.

Finally, a robustness test was also conducted to investigate the impact of using an incorrect mean target-speed PDF when constructing the target-location prediction. These simulations involved creating an intentional discrepancy between the true and assumed mean-target speed PDFs and studying the impact on search-time and success-rate. It was found that for cases where the true PDF is over-estimated by as much as 25% relative to the assumed PDF, the proposed methodology remains robust, producing negligible changes in success-rates and search-times.

Chapter 7

7 Conclusions and Recommendations for Future Work

The main research challenge addressed in this Thesis is summarized below, along with a brief review of the approaches proposed for the three main issues that were considered, namely, target-location prediction, robot deployment, and robot path planning. The scientific contributions from the perspectives of each of these three devised approaches are also summarized. Following this, some directions for future research that would help to further augment the existing methodology are suggested.

7.1 Summary of Contributions

This Thesis has addressed autonomous robotic WiSAR that involves the search for a non-trackable, potentially mobile target (i.e., the lost-person) within in-land wilderness environments. Rugged terrains, harsh weather, and large search areas can make it difficult and dangerous for live SAR personnel to perform searches in WiSAR scenarios. Robot teams capable of performing the search process autonomously can, therefore, aid greatly in this application domain.

The search problem in autonomous robotic WiSAR presents a number of challenges, including: a mobile target whose time-varying state must be predicted at any given time using available probabilistic information; unique physiology and lost-person psychology that influences target motion and behaviour; a partially-known environment with varying terrain topology; the possible existence of *a priori* known and unknown obstacles; and, clues left behind by the target. In this Thesis, a MRC methodology has been proposed for the on-line, autonomous coordination of a team of robots to search for a target in WiSAR scenarios. This methodology has been shown to be capable of successfully addressing the above challenges.

The overall approach proposed for this methodology involves the derivation of probabilistic information describing the probable motion of the target, and, in turn, the probable location of the target at any given time during the search. Methods were devised to create, update, and propagate this probabilistic information over time and space, and to use it to optimally deploy search robots and plan robot search paths. This methodology, therefore, involves three main tasks: target-location prediction; robot deployment; and, robot-path planning. Novel procedures were developed to address each of these tasks, resulting in substantial theoretical contributions as summarized in the following sub-sections.

7.1.1 Target-Location Prediction

A probabilistic approach to the problem of autonomous search has been investigated in this Thesis, where the target's location is predicted and updated with time to guide the search. Information for constructing this prediction comes from empirical data on past lost-person search incidents that is gathered and stored by SAR organizations, and further organized into relevant lost-person categories. Alternatively, empirical physiological data on average walking speeds, available from the literature in fields such as Biomechanics and Human Biology and categorized by age group, may also be used. The available data is analyzed and used to derive a 1D probability distribution describing the mean speed of the target along a single straight line of travel. When combined with a specified point in time, this PDF can be used to compute a 1D distribution for the location of the target along that line of travel for that point in time.

The novel concept of iso-probability curves was introduced, which uses the probabilistic information contained within the 1D target-location PDF to construct a 2D representation of the target-location prediction over the search area. By considering multiple possible directions of target travel (i.e., rays) outward from the LKP, and associating the computed 1D target-location PDF with each ray, control-points corresponding to different cumulative probabilities for target presence are established along each ray. The use of piecewise cubic polynomials was proposed for the construction of continuous, closed curves that interpolate between control-points of the same cumulative probability among the rays. Each such curve delimits the boundary of a region and represents the maximum distance that the target, with mean-speed corresponding to the cumulative probability associated with that boundary, would travel.

The iso-probability curves have a direct correspondence to the possible locations of the target within the search area at any given time. By using them in target-location prediction, the density of the curves in different regions of the search area would yield the relative likelihood of target presence in those regions. Namely, regions of high curve-density would correspond to regions of high probability for target presence, while low curve-density regions would indicate relatively lower target-presence likelihood.

Moreover, the use of iso-probability curves provides an efficient means for target-location prediction representation. Closed, piecewise cubic polynomials can be computed rapidly from the control points established on rays in order to form the curves. This precludes the need for constructing and maintaining a 2D continuous (or discretized) bivariate probability distribution through computationally intensive means, as is typically done in search theory techniques.

The iso-probability curves can also be propagated in time expeditiously through: (*i*) the update of the target-location PDF on each ray; (*ii*) the subsequent re-establishment of the control-point positions on the rays; and, (*iii*) the connection of those control-points through piecewise cubic polynomial interpolation; for any new time-value specified.

The iso-probability curve concept also allows for efficient methods to be devised for incorporating other influences relevant to WiSAR into the target-location prediction. The influence of terrain on the probable location of the target can be incorporated by manipulating the positions of the control-points based on knowledge of the terrain topology and a given target physiological model. The impact of target psychology can be incorporated as well.

This Thesis considered the influence of one particular class of lost-person psychological behaviour, namely, a possible preferred direction of travel that can be presumed based on elementary knowledge of the target and his/her purpose in the environment. Based on the estimated likelihoods of different directions of travel, the control-points along the rays used to construct the iso-probability curves are re-positioned in a corresponding manner through scaling of the associated PDF, allowing the resulting curves to be modified to reflect the influence of this psychological behaviour.

Procedures were also devised to modify the locations of the control points when they intersect obstacles. As well, the iso-probability curve concept allows for the incorporation of the impact of clues found by shifting the position of the curves and reconstructing them based on the location of the clue and the estimated time at which it was left behind.

All these influences can be incorporated into the curves in an efficient manner, thereby, facilitating an on-line implementation of the proposed WiSAR methodology. This novel representation mechanism of the iso-probability curves, thus, forms the main contribution of this Thesis from the perspective of the target-location prediction task – an efficient means of computing and representing probabilistic target-location information, incorporating influences on target location relevant to WiSAR, and updating and propagating this information over time and space.

7.1.2 Robot Deployment

The robot deployment task refers to the optimal distribution of search effort within the search area. In the approach proposed, optimal regions for search effort allocation are delimited and robots are assigned to each region in appropriate numbers. In particular, since the iso-probability curves indicate the relative likelihood of target presence within the search area by virtue of their numbers and their positions, deployment is conducted by assigning robots to specific positions on optimally selected iso-probability curves. Determination of an optimal deployment, therefore, requires the determination of the optimal number and positions of iso-probability curves to use during any given deployment endeavour and the optimal assignment of robots to those curves.

In order to determine the optimal deployment, a nested optimization structure was proposed. The outer optimization loop searches through the space of the feasible numbers and positions of iso-probability curves to use. With each possible valuation of the corresponding decision variables, the inner loop, on the other hand, determines an optimal assignment of robots to those curves by solving a corresponding Linear Bottleneck Assignment Problem. Per-curve robot quantity requirements for the selected iso-probability curves are determined based on a procedure that proportions the resource requirements for each curve relative to the sizes of the curves. These resource requirements serve as inputs for computing the optimal assignment solution in the inner optimization.

Upon determining the optimal assignment scheme of robots to curves, the shortest paths from each robot to its assigned curve indicates, through the point of intersection, the deployment position corresponding to that robot. The selection of curves combined with the assignment of robots to curves and the shortest-paths corresponding to those assignments constitutes a single possible deployment solution within each loop of the outer optimization.

Three normalized metrics were formulated to quantify the optimality of any given deployment solution: search-time, success-rate, and return-time. A weighted sum of these normalized metrics served as the objective function for the outer optimization loop, guiding the overall optimization to an optimal deployment solution.

The proposed approach also differentiates initial deployment from re-deployment. Given a mobile target and a continuously changing target-location prediction, it may be necessary to re-distribute the search resources multiple times throughout the search after having conducted an initial optimal deployment at the start of the search. These intermediate re-distributions are instances of re-deployment. A procedure was proposed whereby the optimal deployment solution is re-computed at regular time intervals and the relative optimality of the current deployment solution being used is gauged. This is done through the computation of a measure of the benefit to the search entailed by adopting the new optimal deployment solution compared to using the current one. A sufficiently large benefit justifies conducting re-deployment and implementing the new optimal deployment solution.

This use of a nested optimization structure, and the decision to optimally assign robots to curves while using the shortest paths to determine the deployment positions, enables the re-deployment computations to be conducted multiple times on-line. It bypasses the complexity involved in modelling all aspects of the deployment problem into a single optimization, thereby expediting the process. The subsequent regular optimality checks, and the re-deployments (when necessary), address the compromise to global optimality that this expedited approach entails, yet allows the method to successfully deal with the dynamic nature of WiSAR by re-deploying the search robots appropriately, when required. In this way, an effective distribution of search resources is maintained throughout the ongoing search.

Thus, the overall contribution in addressing this second main task of the MRC methodology is a novel, generic approach for the autonomous deployment of multiple robots in coordinated robotic WiSAR. The approach addresses the need for considering dynamically-changing probabilistic information on a mobile lost-person in order to optimally distribute search resources within a growing search area.

7.1.3 Robot-Path Planning

In the proposed MRC methodology, the task of robot deployment is distinguished from the task of searching. Upon attaining their deployment positions, robots must perform the search by following different, optimally determined search paths. With optimal deployment prescribing the most effective distribution of search resources, it was dictated that the robots must remain on their assigned curves even as they move about to perform the search. This allows the search to be conducted while maintaining the optimal deployment. A method was, therefore, devised to construct search paths that follow this fundamental directive of path planning.

To contend with the inability to compute propagated iso-probability curves at every instant in time for the purposes of satisfying the abovementioned directive, propagated curves corresponding to a finite time-interval, Δt_{prop}, into the future are constructed instead. Path planning was, thus, proposed as a recursive application of an optimization algorithm that constructs time-optimal paths from each robot's current position on its origin iso-probability curve to an optimal position on the propagated version of this curve Δt_{prop} into the future (i.e., the destination iso-probability curve).

Each path is constructed using piecewise cubic polynomials, with interpolation points defined along the established rays used to construct the curves. The objective of each path-planning endeavour is to find the position on the destination curve such that the path constructed to that position can be traversed within an estimated time-interval as close as possible to Δt_{prop}. This ensures that the robot is still on its assigned curve at the end of this time-interval. In so doing, each robot would adhere to the fundamental principle at the end of each such time-interval. Furthermore, the smooth piecewise cubic polynomial path that the robot must follow to reach its destination position would keep the robot close to, or directly on, its assigned curve as that curve propagates from its origin position to its destination position.

Once a given robot reaches its destination curve, this path-planning process is repeated. In this way, a continuous search-path is constructed for each robot throughout the search, one Δt_{prop} time-interval at a time.

Once again, to address the dynamic nature of WiSAR, the method dictates path optimality checks at pre-designated, evenly-spaced check-points along a robot's path. Procedures were devised to perform three time-efficient checks at each check-point, during path-implementation, to gauge path optimality. The first two checks serve as measures of criticality wherein the purpose is to obtain an indication that at least an alternative shortest path to the destination curve from the robot's current position remains feasible. The third check gauges the feasibility of the current planned path that the robot is following by ensuring that a confidence interval for the error in the estimated arrival-time at the destination curve is within acceptable limits. This check is computed based on path uncertainty data gathered and evaluated by each robot as it travels along its path.

A procedure was devised that uses the results of these checks to determine if a path must be re-planned at any check-point. An affirmative indication to the path re-planning query would require re-running the path optimization algorithm to find the new optimal path from the robot's current position to a new optimal destination position on the destination curve.

The contribution with respect to this third main task of the proposed MRC methodology, thus, is an on-line-feasible method to construct continuous, effective search-paths for all robots throughout the search process. This method utilizes target-location prediction information in determining these paths, and maintains search effectiveness through regular path optimality checks, along with path re-planning when required. At the same time, this method is also able to maintain the latest optimal search-effort distribution by ensuring that robots remain on their curves as best as possible as they perform the search.

The use of the iso-probability curves as the underlying component for all three main tasks of the overall methodology allows for a unified approach to MRC for WiSAR, where the various influences that need to be considered in WiSAR scenarios can be accounted for concurrently. Thus, path planning is able to work in tandem with optimal robot deployment and accordingly respond to the continuously changing target-location prediction.

7.1.4 Method Validation

The overall proposed MRC methodology was validated through numerous simulations in Matlab involving large-scale, team-based, optimal WiSAR searches for a moving target. The use of simulations allowed for comprehensive tests to be conducted that modelled the various influences relative to WiSAR which the methodology was designed to address. Furthermore, it allowed the methodology to be tested under the load of large robot teams within expanding terrains.

One example WiSAR search simulation was presented in this Thesis involving the use of 21 robots within a 5000 m × 5000 m terrain map and a maximum search time limit of 9000 s. This simulation involved varying terrain topology as well as *a priori* known and unknown obstacles. The mobile target moved at a randomly-selected initial heading-angle and nominal initial speed, but was subjected to random speed and direction perturbations throughout the simulation to mimic drift in its motion. The target also left behind clues. All simulations were performed under modest computing conditions (i.e., using non-optimized code written in Matlab v7, and running on a 2.50 Ghz Pentium Dual-Core processor).

This and other similar simulations demonstrated the ability of the proposed overall MRC methodology for WiSAR to function effectively in an on-line manner. The solution methods that were devised for each of the three main tasks, namely, target-location prediction, robot deployment, and robot-path planning, were all able to successfully distribute and guide the search resources to perform an intelligent search that effectively utilized the time-varying probabilistic information on the target's location. These simulations have, thus, validated the ability of the methodology to intelligently and autonomously coordinate a team of robots to find a mobile yet non-trackable target in WiSAR scenarios.

Further validation of the methodology was also conducted through its comparison with an alternative robotic search method. Due to the lack of a comparable method in the literature that addresses the WiSAR-specific issues for which the proposed methodology was developed, a suitable alternative method had to be devised. This alternative method was based on the grid-search technique used in SAR missions with live search personnel, and employs a strategy of uniform coverage of the search area.

The particular method devised for the comparison in this Thesis was deterministic in the sense that it used a pre-determined pattern to delimit and construct the robot paths, rather than modelling and utilizing target-location prediction to guide the search. In so doing, the simulations were able to more explicitly reveal the benefit provided by the use of the iso-probability curve concept to represent the modelled probabilistic target-location information, as well as the effectiveness of the use of this concept in the methods for deployment and path planning.

The comparison simulations consisted of 1000 simulation runs for each of the two methods. Each simulation run involved a randomly-selected initial target heading-angle, as well as random speed and heading-angle perturbations to the target's motion throughout the search. Each simulation run used a total of 21 robots, and was given a maximum search time limit of 11,200 s. The simulation results revealed a significantly higher success-rate and shorter average search-time for the proposed (probabilistic) method over the alternative (deterministic) one.

In particular, the proposed method achieved a 3 times higher success-rate and a 30% relative improvement in average search-time (6.5 times relative average search-time improvement when equal numbers of the top simulation runs were compared). These results further validate the proposed probabilistic, autonomous MRC method for WiSAR by demonstrating its ability to provide significant tangible benefit, both in terms of search-time and success-rate.

Finally, numerous other simulations demonstrated the proposed method's robustness to discrepancies in the assumed mean target-speed PDF relative to the true distribution. Simulations were conducted where the distribution that determined the actual mean speed of the target (the true mean target-speed PDF) was scaled relative to the distribution used to construct the iso-probability curves (the assumed mean target-speed PDF).

A total of 1000 simulation runs were performed for each of three different scaling scenarios: (*i*) over-estimation (true PDF scaled down by 0.75 relative to assumed PDF); (*ii*) benchmark (true PDF un-scaled); and (*iii*) under-estimation (true PDF scaled up by 1.50 relative to assumed PDF). In each simulation run, the actual mean target-speed was randomly selected from the true PDF, while iso-probability curves were constructed according to the assumed PDF. The results showed that the proposed method remained robust for the over-estimation cases with minimal differences in average search-time and success-rate compared to the benchmark cases.

As expected, though, robustness was compromised for the under-estimation cases, where a significantly scaled-up true mean target-speed PDF resulted in situations where the mean target speeds were so high as to place the target outside the search area at any given time, thereby reducing overall success-rate and increasing average search-time. Nevertheless, in general, discrepancies between the assumed and true mean target-speed PDFs can be expected to be minimal. Namely, situations involving large discrepancies can be expected to be few, since the mean target-speed PDF is obtained based on empirical data organized into appropriate categories of target and should, therefore, reflect reality closely.

7.2 Recommendations for Future Work

The overall MRC methodology proposed in this Thesis addresses the problem of autonomous robotic WiSAR, an application area for robotics that has received little attention in the literature up to now. As such, the research carried out for this Thesis represents a pioneering effort to addressing this problem. One of the main challenges in this work has been to concurrently address WiSAR-specific issues in a single method, while maintaining its on-line feasibility. Although the method that has been devised successfully meets this challenge and provides tangible benefits to the search process, there is still room for improvement and advancement in order to make this method more comprehensive. Some suggestions for future work are provided below.

7.2.1 Extending Lost-Person Psychology Modelling

One beneficial avenue of method improvement would be an expansion on the treatment of lost-person psychology. The present methodology considers the influence of a preferred direction of target travel. However, lost-persons in different categories also display a number of other psychological behaviours, which, if modelled and addressed explicitly, could further improve the effectiveness of the MRC methodology for different target types. The challenge, as mentioned previously, is that typical lost-person psychological models tend to be qualitative descriptions of behavioural tendencies that make them difficult to utilize in a quantitative fashion. Nevertheless, with reasonable assumptions on available input information, certain behaviours can still be addressed effectively.

As an example, a typical behaviour often displayed by lost children, 1 to 6 years of age, is that of passively seeking shelter [119]. After an initial period of random motion, the lost child would stop at the first location of shelter they may happen upon (e.g., an over-hanging rock, under thick brush, etc.), and lie down there to sleep. If possible shelter locations can be identified *a priori* and indicated on the topological map, they could be used during the search. One possible means to utilize such information is to take each shelter location as a possible check-point along a robot’s path. If a shelter location lies within the region through which a robot’s path is to be planned, that shelter location could be designated as an intermediate check-point through which that robot’s path is required to pass on its way to its destination position. This would allow the robot to ‘check’ the shelter location as it travels towards its destination curve, thereby, locating the target if he/she happens to be resting there.

Alternatively, robot search paths could be initially planned without considering the shelter-locations, and standard detour procedures can be devised through which a robot would move off its planned path to investigate a shelter location when the robot comes within a certain pre-determined distance of the shelter. Subsequently, the robot would return to its planned path, allowing the current method of checking path-optimality to take care of path re-planning, if necessary, due to the disruption to that robot caused by the detour. Each possible shelter location could also be associated with a particular probability value, indicating the likelihood of that location being used for shelter by the target being sought, which in turn could be used to determine how often that location is ‘checked’ by a robot throughout the search.

Another possible psychological behaviour is that of trail-running [11]. This is a strategy often utilized by children 12 years of age or younger, whereby the lost person follows a continuous feature, such as a drainage pipe, an old dirt trail, a shore-line, etc., in the hopes that it will lead them to a familiar location or to a populated area where they can find help. Addressing this behaviour in the search methodology would also require the assumption that such trails can be identified and designated on the map, either *a priori* or during the search process when such features are found.

Given the general solution framework that has been used in this Thesis, it is clear that the iso-probability curves would have to be modified in some fashion to account for the influence of trail-running behaviour. However, incorporating the influence of a given trail on the iso-probability curves poses a considerable challenge.

The concept behind the iso-probability curves is to delineate multiple conservative boundaries of target presence within the search area based on the assumption of straight-line target travel outward from the LKP along different rays. This avoids the difficult (if not impossible) task of having to obtain a 2D PDF for target-presence at all points within the search area, or particular probability values for each cell within a discretized grid overlaid onto the search area. As a result, any modification to the iso-probability curves would have to be done through modification of the control-points along the rays that are being used.

One possible approach that can be employed to address trail-running behaviour, then, is to control the outward expansion of control-points in the vicinity of any given trail in such a way as to cause the iso-probability curves to slow down their outward progress when they cross trails. With robot paths constrained by the need to keep robots on their assigned curves, this approach would, indirectly, cause the robots to spend more time searching near trails as well. This modification to the control-points could be done through the use of appropriate scaling of individual control-points, or through the modification of the mean target-speed PDF along rays intersecting trails. In the latter strategy, for any given ray that passes over a trail, the portion of the corresponding mean target-speed PDF that lies over the trail could be altered. If such PDFs could be modified to reflect the higher likelihood of target presence over trails, they would directly control the density of control-points, and thus, of curves, near trails. Further research would be required to determine what, exactly, these modifications should be, and how they should change with time as the affected PDFs propagate outward.

7.2.2 Improving the Target Physiological Model

In order to incorporate the effect of terrain on the iso-probability curves, the proposed method scales the positions of the control-points along rays based on a particular target physiological model. This model is a function that outputs a mean speed scale-factor for a specified average terrain slope over which the target is expected to travel. The current method uses a static physiological model throughout the search. However, a more realistic model could be time-varying to reflect relevant physiological changes in the lost person over time. Since iso-probability curves play an important role in all aspects of the overall MRC methodology, devising a more realistic model would be a worthwhile endeavour to improving the methodology’s effectiveness.

As an example, one possible time-varying influence that could be incorporated into the target’s physiological model is the gradual tiring that one would expect to observe in a target that is moving. This tiring should manifest itself as slower mean target speeds over time for a given average terrain slope over which the target must travel. Thus, from the perspective of the physiological model, stronger scale-factors would have to be used, resulting in greater scaling-down of the target’s mean speed over inclined slopes in the terrain as time passes. The same may be necessary over declines as well. To determine the appropriate functional relationship to use for such a model, one would have to obtain empirical data on changes in mean walking speeds over time for persons of different age-groups walking over different inclines. If such data can be obtained, it can be used to construct separate relationships between mean speed scale-factor and average terrain slope for different points in time.

Each age-group for which data is obtained would, thus, have a corresponding set of time-phased functional relationships that constitute the physiological model for that group. Depending on the age-group to which a given target belongs, the appropriate physiological model, consisting of a corresponding set of functional relationships (each suited for a different point in time), can be specified for use. As the search progresses, the appropriate functional relationship can be called upon at the appropriate point in time, and used to properly scale the positions of the control-points in order to construct the iso-probability curves. In this way, the iso-probability curves would be able to reflect the lost person's speed variation over time due to the gradual onset of fatigue.

7.2.3 Addressing Multiple LKPs

The current MRC methodology deals with one LKP at a time. A single LKP is assumed to be given at the start of the search, from which the iso-probability curves are constructed and propagated over time. However, there may be scenarios when multiple possibilities for the LKP exist at the start of a search, with each having a different associated probability. For instance, an abandoned car belonging to the target, combined with evidence of a recently-used campsite at another location, as well as a possible target-sighting in yet a third location, may be available at the beginning of the search, providing three possibilities for the LKP. Although the search manager could identify the most likely piece of information to select a single LKP to use, a more thorough approach would be to incorporate the impact of all three pieces of information so that they may be considered simultaneously in order to construct the iso-probability curves.

One possible approach by which to address this problem would be to divide the available robots into sub-teams. Each sub-team could be assigned to a different LKP, where the relative sizes of the sub-teams would have to be proportioned according to the relative likelihoods of the possible LKPs. With this approach, each LKP can be treated separately, where iso-probability curves could be constructed around each possible LKP independent of the others. The sub-team assigned to each LKP would, then, carry out a separate search using the corresponding iso-probability curves as per the methodology proposed in this Thesis. However, additional research would still be required to determine how iso-probability curves constructed from different LKPs should be modified (if at all) when they intersect, and if any corresponding robot search path modifications are necessary.

Alternatively, separate iso-probability curves could first be constructed for each LKP independently and, then, amalgamated in some fashion to produce a single conservative set of iso-probability curves. This conservative set of curves could use, as its LKP, a weighted sum of the alternative LKP possibilities that exist, with weights made proportional to the probabilities associated with those LKPs. The mean target-speed PDFs along each ray that emanates from this weighted-sum LKP would have to be modified such that each resulting iso-probability curve encompasses each of its corresponding curves constructed from the different LKPs individually. An appropriate method of modifying each ray's PDF would have to be devised to accomplish this. This could range from a simple approach of scaling the individual PDFs, to something more complex, depending on what addresses the problem in the most appropriate manner.

7.2.4 Incorporating a Measure of Search-Resource Suitability

The optimization formulation for the robot deployment task in the current methodology considers the impact of a given deployment solution on measures of search-time, success-rate, and return-time. However, in finding an optimal distribution and assignment of search resources, the current method does not differentiate between the available robots. Depending on the terrain topology and associated traversal difficulty, as well as the sensing requirements in different regions of the search area, certain robots may be better suited to search certain regions than others if one is using a heterogeneous team comprised of robots with differing capabilities. Therefore, another possible improvement to the methodology could be to devise a measure of robot suitability for searching different regions, and to incorporate that measure into the robot deployment method.

A robot with better mobility over rough terrain (e.g., due to its size and/or locomotion system design) would be better suited to search a particularly rugged region of the search area compared to others on the team. Similarly, a robot that has, for example, thermal sensing capability might be the better choice to assign to regions containing dense vegetation that obstruct vision-based sensors in order to allow for target detection through body-heat readings. When assigning robots to iso-probability curves, then, a measure of suitability of the robot to each curve, given the requirements associated with each curve and the relevant capabilities of each robot, can be computed. Formulation of such a measure would first require a procedure to quantify the ‘difficulty’ associated with an iso-probability curve. This difficulty must be a function of relevant attributes that one wishes to consider.

Correspondingly, procedures would also be required to quantify the ‘capability’ of each robot with respect to the attributes considered. Based on a robot’s capability and the difficultly associated with each iso-probability curve, a normalized measure of suitability of each robot to each iso-probability curve would have to be formulated. This suitability measure would, then, have to be considered alongside the return-time measure when conducting the inner optimization of the overall robot deployment solution approach. The optimal assignment solution obtained from this revised formulation of the inner optimization would provide the values for the normalized return-time measure as well as the normalized suitability measure. These would, in turn, be used in the objective function of the main (outer) deployment optimization loop which searches for the optimal set of iso-probability curves. Incorporating such a suitability measure, therefore, would allow for a better distribution of search resources due to a more appropriate allocation of search effort within the search area.

References

[1] N. Kalra, D. Ferguson, and A. Stentz, "Constrained Exploration for Studies in Multirobot Coordination," *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2006.

[2] D. Milutinovic and P. Lima, "Modeling and Optimal Centralized Control of a Large-Size Robotic Population," *IEEE Transactions on Robotics*, Vol. 22, No. 6, pp. 1280-1285, Dec. 2006.

[3] M. Moors, T. Röhling, and D. Schulz, "A probabilistic approach to coordinated multi-robot indoor surveillance," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3447-3452, Aug. 2-6, 2005.

[4] Madhavan, R., Fregene, K., Parker, L.E. "Distributed Cooperative Outdoor Multirobot Localization and Mapping," *Autonomous Robots*, vol. 17, no. 1, pp. 23-39, Jul. 2004.

[5] T. Balch, "Communication, Diversity and Learning: Cornerstones of Swarm Behavior," *LNCS 3342, Swarm Robotics: SAB 2004 International Workshop*, Santa Monica, pp. 21-30, 2004.

[6] A. ElGindy and A. M. Khamis, "Team-theoretic Approach to Cooperative Multirobot Systems," *International Conference on Engineering and Technology*, pp. 1-6, Oct. 10-11, 2012.

[7] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. M. Erkmen, "Search and rescue robotics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin: Springer-Verlag, 2008, ch. 50, pp.1151-1173.

[8] G. Dissanayake, J. Paxman, J. V. Miro, O. Thane, and H.-T. Thi, "Robotics for urban search and rescue," *International Conference on Industrial and Information Systems*, pp. 294-298, Aug. 11, 2006.

[9] M. A. Goodrich, J. L. Cooper, J. A. Adams, C. Humphrey, R. Zeeman, and B. G. Buss, "Using a mini-UAV to support wilderness search and rescue: practices for human-robot teaming," *International Workshop on Safety, Security and Rescue Robotics*, Sept. 2007.

[10] T. J. Setnicka, *Wilderness Search and Rescue*. Appalachian Mountain Club, 1980.

[11] K. A. Hill, "The psychology of lost," in *Lost Person Behavior*. National SAR Secretariat, Ottawa, Canada, 1998.

[12] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini UAV," *J. Field Rob.*, vol. 25, no. 1-2, pp. 89-110, Jan. 2008.

[13] A. Wolf, H. B. Brown, R. Casciola, A. Costa, M. Schwerin, E. Shamas, H. Choset, "A mobile hyper redundant mechanism for search and rescue tasks," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2889-2895, Oct. 2003.

[14] N. Ruangpayoongsak, H. Roth, and J. Chudoba, "Mobile robots for search and rescue," *IEEE Int. Workshop on Safety, Security and Rescue Robotics*, pp. 212-217, Kobe, Japan, Jun. 2005.

[15] S. P. N. Singh and S. M. Thayer, “Kilorobot search and rescue using an immunologically inspired approach,” in *Distributed Autonomous Robotic Systems 5*, H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, Eds. Tokyo, Japan: Springer-Verlag, 2002, ch. 11, pp. 424-433.

[16] J. Borenstein, M. Hansen, and A. Borrell, “The omni-tread ot-4 serpentine robot – design and performance”, *J. Field Rob.*, vol. 24, no. 7, pp. 601-621, Jul. 2007.

[17] M. Arai, Y. Tanaka, S. Hirose, H. Kuwahara, and S. Tsukui, “Development of “Souryu-IV” and “Souryu-V:” serially connected crawler vehicles for in-rubble searching operations,” *J. Field Rob.*, vol. 25, no. 1-2, pp. 31-65, Jan.-Feb. 2008.

[18] J. Casper and R. R. Murphy, “Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center,” *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 33, no. 3, pp. 367-385, Jun. 2003.

[19] J. L. Drury, J. Scholtz, and H. A. Yanco, “Awareness in human-robot interactions,” *IEEE Conf. on Systems, Man and Cybernetics*, vol. 1, pp. 912-918, Washington, D. C., Oct. 2003.

[20] Y. Liu, G. Nejat, and B. Doroodgar, “Learning based semi-autonomous control for robots in urban search and rescue,” *IEEE International Symposium on Safety, Security, and Rescue Robotics*, College Station, Texas, USA, November 5-8, 2012.

[21] R. Murphy, “Marsupial and shape-shifting robots for urban search and rescue,” *IEEE Intelligent Systems*, vol. 15, no. 3, pp. 14-19, 2000.

[22] R. Murphy and J. J. Martinez, “Lessons learned from the NSF REU site grant: multiple autonomous mobile robots for search and rescue applications,” *Proceedings of the 21st Annual ASEE/IEEE Conference on Frontiers in Education (CD-ROM)*, 1997.

[23] S. Hirose, “Snake, walking and group robots for super mechano-system,” *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 129-133, Oct. 12-15, 1999.

[24] A. Kobayashi and K. Nakamura, “Rescue robots for fire hazards,” *Proceedings of the 1st International Conference on Advanced Robotics*, pp. 91-98, Sept. 12-13, 1983.

[25] H. Amano, K. Osuka, and T.-J. J. Tarn, “Development of vertically moving robot with gripping handrails for fire fighting,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 661-667, Oct. 29-Nov. 3, 2001.

[26] A. Castano, W. M. Shen, and P. Will, “CONRO: toward deployable robots with inter-robot metamorphic capabilities,” *Autonomous Robots*, vol. 8, no. 3, pp. 309-324, 2000.

[27] R. M. Voyles, “TerminatorBot: a robot with dual-use arms for manipulation and locomotion,” *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 61-66, April 24-28, 2000.

[28] J. S. Jennings, G. Whelan, and W. F. Evans, “Cooperative search and rescue with a team of mobile robots,” *Proceedings of the 8th International Conference on Advanced Robotics*, pp. 193-200, Jul. 7-9, 1997.

[29] R. Murphy, J. Casper, H. Hyams, and M. Micire, "Mixed-initiative control of multiple heterogeneous robots for search and rescue," Technical report, University of South Florida, 2002.

[30] R. Masuda, T. Oinuma, and A. Muramatsu, "Multi-sensor control system for rescue robot," *IEEE/SICE/RJS International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 381-387, Dec. 8-11, 1996.

[31] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 343-354, 1994.

[32] S. Hirose, *Biologically inspired robots: snake-like locomotors and manipulators*. Oxford, U. K.: Oxford University Press, 1993.

[33] V. A. Ziparo, A. Kleiner, L. Marchetti, A. Farinelli, and D. Nardi, "Cooperative exploration for USAR robots with indirect communication," 6th IFAC Symposium on Intelligent Autonomous Vehicles, Toulouse, France, Sept. 3-5, 2007.

[34] J. Valls Miro, W. Zhou, and G. Dissanayake, "Towards vision based navigation in large indoor environments," IEEE/RSJ Int. Conf. on Intell. Rob. Syst., Beijing, China, Oct. 9-15, 2006.

[35] Z. Zhang, G. Nejat, H. Guo, and P. Huang, "A Sensory System for Robot-Assisted 3D Mapping of Urban Search and Rescue Environments," Intelligent Service Robots, Vol. 4, No. 2, pp. 119-134, 2011.

[36] Z. Zhang and G. Nejat, "Robot-assisted 3D mapping of unknown cluttered USAR environments," Disaster Robots Special Issue, Advanced Robotics Journal, vol. 23, no. 9, pp. 1179–1198, 2009.

[37] J. Cooper and M. Goodrich, "Towards Combining UAV and Sensor Operator Roles in UAV-Enabled Visual Search," *ACM/IEEE Int. Conf. on Human-Robot Interaction*, pp. 351-358, Mar. 12-15, 2008.

[38] D. Kingston, R. Beard, T. McLain, M. Larsen, W. Ren, "Autonomous vehicle technologies for small fixed wing UAVs," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 1, pp. 92-108, Jan. 2005.

[39] M. Quigley, B. Barber, S. Griffiths, and M. A. Goodrich, "Towards real-world searching with fixed-wing mini-UAVs," *IEEE/RSJ Intelligent Robots and Systems*, pp. 3028-3033, Aug. 2-6, 2005.

[40] L. Lin, M. Roscheck, M. A. Goodrich, and B. S. Morse, "Supporting wilderness search and rescue with integrated intelligence: autonomy and information at the right time and the right place," *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, GA, July 11-15, 2010.

[41] B. S. Morse, C. H. Engh, and M. A. Goodrich, "UAV video coverage quality maps and prioritized indexing for wilderness search and rescue," *ACM/IEEE International Conference on Human-Robot Interaction*, Osaka, Japan, March 2-5, 2010.

[42] B. O. Koopman, *Search and screening: general principles with historical applications*. Toronto: Pergamon Press, 1980.

[43] L. D. Stone, “Generalized search optimization,” *Statistical Signal Processing*, vol. 53, E. J. Wegman and J. G. Smith, Eds. New York: Marcel Dekker, Inc., 1984, pp. 265-272.

[44] D.V. Chudnovsky and G.V. Chudnovsky (Eds.), *Search theory: some recent developments*. New York: Marcel Dekker Inc., 1989.

[45] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, “Deployment of Mobile Robots with Energy and Timing Constraints,” *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 507-521, Jun. 2006.

[46] G. Antonelli and S. Chiaverini, “Kinematic Control of Platoons of Autonomous Vehicles,” *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1285-1292, Dec. 2006.

[47] N. Kalra, D. Ferguson, and A. Stentz, “Constrained Exploration for Studies in Multirobot Coordination,” *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2006.

[48] R. Fierro, C. Branca, and J. R. Spletzer, “On-line Optimization-based Coordination of Multi Unmanned Vehicles,” *IEEE International Conference on Networking, Sensing and Control*, Tucson, AZ, March 19-22, pp. 716-721, 2005.

[49] B. Brumitt, and A. Stentz, “Dynamic Mission Planning for Multiple Mobile Robots,” *IEEE International Conference on Robotics and Automation*, vol. 3, Minneapolis, MN, April 22-28, pp. 2396-2401, 1996.

[50] S. J. Benkoski, M. G. Monticino, and J. R. Weisinger, “A survey of the search theory literature,” *Naval Research Logistics*, vol. 38, no. 4, pp. 469-494, August 1991.

[51] O. Hellman, “On the effect of a search upon the probability distribution of a target whose motion is a diffusion process,” *Ann. Math. Stat.*, vol. 41, pp. 1717-1724, October 1970.

[52] S. S. Brown, “Optimal search for a moving target in discrete time and space,” *Oper. Res.*, vol. 28, no. 6, pp. 1275-1289, Nov.-Dec. 1980.

[53] A. R. Washburn, “On search for a moving target,” *Naval Res. Logist. Quart.*, vol. 27, no. 2, pp. 315-322, 1980.

[54] L. D. Stone, “Necessary and sufficient conditions for optimal search plans for moving targets,” *Mathematics of Operations Research*, vol. 4, no. 4, pp. 431-440, 1979.

[55] W. R. Stromquist and L. D. Stone, “Constrained optimization of functional with search theory applications,” *Mathematics of Operations Research*, vol. 6, no. 4, pp. 518-529, 1981.

[56] L. D. Stone, S. S. Brown, R. P. Buemi, and C. R. Hopkins, “Numerical optimization of search for a moving target,” Report to Office of Naval Research. Daniel H. Wagner, Associates, 1978.

[57] J. H. Discenza and L. D. Stone, “Optimal survivor search with multiple states,” *Operations Research*, vol. 29, no. 2, pp. 309-323, 1981.

[58] A. R. Washburn, “Search for a moving target: the FAB algorithm,” *Operations Research*, vol. 31, no. 4, pp. 739-751, 1983.

[59] H. R. Richardson, D. H. Wagner, and J. H. Discenza, “The United States coast guard computer-assisted search planning system,” *Naval Research Logistics Quarterly*, vol. 27, no. 4, pp. 659-680, 1980.

[60] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, “Optimal search for a lost target in a Bayesian world,” in *Field and Service Robotics, STAR 24*, S. Yuta et al. (Eds.), pp. 209-222, Berlin/Heidelberg: Springer-Verlag, 2006.

[61] B. Lavis, T. Furukawa, and H.F.D. Whyte, “Dynamic space reconfiguration for Bayesian search and tracking with moving targets,” *Auton. Robots*, vol. 24, pp. 387-399, Jan. 2008.

[62] G. Hollinger, J. Djugash, and S. Singh, “Coordinated search in cluttered environments using range from multiple robots,” *STAR: Field and Service Robotics*, C. Laugier and R. Siegwart (Eds.), vol. 42, pp. 433-442, Berlin / Heidelberg: Springer, 2008.

[63] D.C. Cooper, J.R. Frost, and R.Q. Robe, *Compatibility of Land SAR Procedures with Search Theory*, Technical Report DTCG32-02-F-000032, Department of Homeland Security, U.S. Coast Guard Operations. Washington, DC: Potomac Management Group Inc., 2003.

[64] J. N. Eagle and J. R. Yee, “An optimal branch-and-bound procedure for the constrained path, moving target search problem,” *Operations Research*, vol. 38, no. 1, pp. 110-114, 1990.

[65] K. E. Trummel and J. R. Weisinger, “The complexity of the optimal searcher path problem,” *Operations Research*, vol. 34, no. 2, pp. 324-327, 1986.

[66] I. Wegener, “Optimal search with positive switch cost is NP-Hard,” *Information Processing Letters*, vol. 21, no. 1, pp. 49-52, 1985.

[67] A. Gautam and S. Mohan, “A review of research in multi-robot systems,” *IEEE International Conference on Industrial and Information Systems*, Chennai, India, August 6-9, 2012.

[68] Z. Yao, X. Dai, and H. Ge, “Quantitative and qualitative coordination for multi-robot systems,” in *Artificial Intelligence and Computational Intelligence, Lecture Notes in Computer Science*, J. Lei et al., Eds. Berlin: Springer-Verlag, 2012, pp. 755-761.

[69] Parker L., “Current State of the Art in Distributed Robot Systems,” *Distributed Autonomous Robotic Systems 4*, Parker L., Bekey G., Barhen J. (eds.), Springer, pp. 3-12, 2000.

[70] Rus D., Donald B., Jennings J., “Moving Furniture with Teams of Autonomous Robots,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, August 5-9, pp. 235-242, 1995.

[71] Donald B., Gariepy L., and Rus D., “Distributed Manipulation of Multiple Objects Using Ropes,” *IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 24-28, pp. 450-457, 2000.

[72] M. B. Dias and A. Stentz, *Enhanced Negotiation and Opportunistic Optimization for Market-Based Multi-robot Coordination*, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Aug. 2002.

[73] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, "Deployment of Mobile Robots with Energy and Timing Constraints," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 507-521, Jun. 2006.

[74] G. Antonelli and S. Chiaverini, "Kinematic Control of Platoons of Autonomous Vehicles," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1285-1292, Dec. 2006.

[75] D. Milutinovic and P. Lima, "Modeling and Optimal Centralized Control of a Large-Size Robotic Population," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1280-1285, Dec. 2006.

[76] L. Chaimowicz, T. Sugar, V. Kumar, and M. Campos, "An architecture for tightly coupled multi-robot cooperation," *Proceedings of the International Conference on Robotics and Automation*, Seoul, Korea, May 21-26, 2001.

[77] Z. Jin, T. Shima, and C. J. Schumacher, "Optimal Scheduling for Refueling Multiple Autonomous Aerial Vehicles," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 682-693, Aug. 2006.

[78] H. Jianghai, J. Lygeros, and S. Sastry, *Towards a Theory of Stochastic Hybrid Systems*, HSCC 2000. New York: Springer-Verlag, 2000, pp. 160-173.

[79] N. Kalra and A. Stentz, "A Market Approach to Tightly-Coupled Multi-Robot Coordination: First Results," *Proceedings of the ARL Collaborative Technologies Alliance Symposium*, May, 2003.

[80] R. Alur, A. Das, J. Esposito, R. Fierro, G. Grudic, Y. Hur, V. Kumar, I. Lee, J. Ostrowski, G. Pappas, B. Southall, J. Spletzer, and C. J. Taylor, "A Framework and Architecture for Multi-robot Coordination," *International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 977-995, Oct.-Nov. 2002.

[81] G. Wieczerzak and K. Kozlowski, "Agents that Live in Robots: How are successful applications built?," *Fourth International Workshop on Robot Motion and Control*, pp. 97-101, Jun. 17-20, 2004.

[82] Z. Cao, M. Tan, L. Li, N. Gu, and S. Wang, "Cooperative Hunting by Distributed Mobile Robots Based on Local Interaction," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 403-407, Apr. 2006.

[83] M. Ani Hsieh, A. Halasz, S. Berman, and V. Kumar, "Biologically Inspired Redistribution of a Swarm of Robots Among Multiple Sites," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 121-141, Sept. 2008.

[84] R. A. Cortez, R. Fierro, and J. E. Wood, "Prioritized Sensor Detection via Dynamic Voronoi-Based Navigation," *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, pp. 5815-5820, Oct. 11-15, 2009.

[85] W. Sheng, Q. Yang, J. Tan, and N. Xi, "Distributed Multi-robot Coordination in Area Exploration," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 945-955, Dec. 31, 2006.

[86] M. Wooldridge and N. R. Jennings, "Pitfalls of Agent-Oriented Development," *International Conference of Autonomous Agents*, Minneapolis/Saint Paul, MN, May 9-13, pp. 385-391, 1998.

[87] M. Wellman and P. R. Wurman, “Market-Aware Agents for a Multi-agent World,” *Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp. 115-125, Sept. 1998.

[88] A. Symington, S. Waharte, S. Julier, and N. Trigoni, “Probabilistic target detection by camera-equipped UAVs,” *IEEE Int. Conf. on Robotics and Automation*, pp. 4076-4081, May 3-8, 2010.

[89] Z. Zhang, H. Guo, G. Nejat, and P. Huang, “Finding disaster victims: a sensory system for robot assisted 3D mapping of urban search and rescue environments,” *IEEE Int. Conf. on Robotics and Automation*, pp. 3889-3894, Rome, Italy, Apr. 2007.

[90] Z. Zhang and G. Nejat, “Robot-assisted 3D mapping of unknown cluttered USAR environments,” Disaster Robots Special Issue, *Advanced Robotics Journal*, vol. 23, no. 9, pp. 1179–1198, 2009.

[91] Y. M. Chan, S. Wong, M. C. Foo, and R. Teo, “Engineering intuition for designing multi-robot search and rescue solutions,” *IEEE Conf. Cybern. Intell. Syst.*, vol. 2, pp. 1238-1242, Singapore, Dec. 1-3, 2004.

[92] V. A. Ziparo, A. Kleiner, L. Marchetti, A. Farinelli, and D. Nardi, “Cooperative exploration for USAR robots with indirect communication,” *6th IFAC Symposium on Intelligent Autonomous Vehicles*, Toulouse, France, Sept. 3-5, 2007.

[93] G. A. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, “Distributed search and rescue with robot and sensor teams,” in *STAR: 2003 Int. Conf. on Field and Service Robotics, FSR 2003*, vol. 24, S. Yuta, H. Asama, S. Thrun, E. Prassler, and T. Tsubouchi, Eds. Berlin / Heidelberg: Springer, 2006, pp. 529-538.

[94] J.L. Baxter, E. K. Burke, J. M. Garibaldi, and M. Norman, “Multi-robot search and rescue: a potential field based approach,” in *Autonomous Robots and Agents Series: Studies in Computational Intelligence*, S. Mukhopadhyay and G. S. Gupta, Eds. Berlin: Springer-Verlag, 2007, pp. 9-16.

[95] P. Lima, L. Custodio, M. I. Ribeiro, and J. Santos-Victor, “The RESCUE project – cooperative navigation for rescue robots,” *Int. Workshop on Advances in Service Robotics*, Bardolino, Italy, Mar. 2003.

[96] I. Suzuki and M. Yamashita, “Searching for a mobile intruder in a polygonal region,” *SIAM J. Comput.*, vol. 21, no. 5, pp. 863-888, Oct. 1992.

[97] B. P. Gerkey, S. Thrun, and G. Gordon, “Visibility-based pursuit-evasion with limited field of view,” *Int. J. Rob. Res.*, vol. 25, no. 4, pp. 299-315, Apr. 2006.

[98] M. T. Khan, B. Chan, A. Khan, Z. Haq, and J. Iqbal, “Optimized dynamic task allocation and priority assignments in an immunized autonomous multi-robot search and rescue operation,” *Romanian Journal of Information Science and Technology*, vol. 15, no. 2, pp. 129-145, 2012.

[99] L. Lin and M. A. Goodrich, “A Bayesian approach to modeling lost person behaviors based on terrain features in Wilderness Search and Rescue,” *Conf. on Behavior Representation in Modeling and Simulation*, pp. 49-56, Sundance, UT, Mar.-Apr. 2009.

[100] C. D. Heth and E. H. Cornell, “Characteristics of travel by persons lost in Albertan wilderness areas,” *J. Environ. Psychol.*, vol. 18, no. 3, pp. 223-235, Sept. 1998.

[101] J. R. M. Hosking and J. R. Wallis, *Regional frequency analysis: an approach based on L-moments*. New York: Cambridge University Press, 1997.

[102] E. Schneider and D. Wildermuth, "A potential field based approach to multi-robot formation navigation," *IEEE Int. Conf. Rob., Intell. Syst. Signal Process.*, pp. 680-685, Oct. 8-13, 2003.

[103] V. Gazi and K. M. Passion, "A class of attractions/repulsion functions for stable swarm aggregations," *Int. J. Control*, vol. 77, no. 18, pp. 1567-1579, 2004.

[104] R. Simmons et al., "Coordinated deployment of multiple, heterogeneous robots," *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, vol. 3, pp. 2254-2260, Oct.-Nov. 2000.

[105] K. R. Guruprasad and D. Ghose, "Performance of a class of multi-robot deploy and search strategies based on centroidal Voronoi configurations," arXiv:0908.1485v1 [math.OC], Cornell University Library, Aug. 2009.

[106] M. Schwager, D. Rus, and J.-J. Slotine, "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *Int. J. Rob. Res.*, vol. 30, no. 3, pp. 371-383, 2011.

[107] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Rob. Autom.*, vol. 20, no. 2, pp. 243-255, Apr. 2004.

[108] L. Dale and N. Amato, "Probabilistic roadmaps – putting it all together," *IEEE Int. Conf. on Robotics and Automation*, pp. 1940-1947, May 21-26, 2001.

[109] G. Wagner, M. Kang, and H. Choset, "Probabilistic path planning for multiple robots with subdimensional expansion," *IEEE Int. Conf. on Robotics and Automation*, pp. 2886-2892, May 14-18, 2012.

[110] M. Bennewitz, W. Burgard, and S. Thrun, "Finding and optimizing solvable priority schemes for decoupled path planning for mobile robots," *Robotics and Autonomous Systems*, vol. 41, no. 2-3, pp. 89-99, Nov. 2002.

[111] K. E. Bekris, K. I. Tsianos, and L. E. Kavraki, "A decentralized planner that guarantees the safety of communicating vehicles with complex dynamics that replan online," *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, pp. 3784-3790, Oct. 29-Nov. 2, 2007.

[112] S. Carpin and E. Pagello, "On parallel RRTs for multi-robot systems," *Proc. 8th Conf. Italian Association for Artificial Intelligence*, pp. 834-841, 2002.

[113] J. P. van den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots" IEEE/RSJ Int. Conf. Intell. Rob. Syst., pp. 2217-2222, Aug. 2-6, 2005.

[114] M. Saha and P. Isto, "Multi-robot motion planning by incremental coordination," *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, pp. 5960-5963, Oct. 9-15, 2006.

[115] L. Lin and M. A. Goodrich, "UAV intelligent path planning for wilderness search and rescue," IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, Oct. 11-15, 2009.

[116] A. Karimoddini, H. Lin, B. M. Chen, and T. H. Lee, "Hybrid formation control of the unmanned aerial vehicles," *Mechatronics*, vol. 21, no. 5, pp. 886-898, Aug. 2011.

[117] J.R. Frost, "Principles of Search Theory, Parts I & II," *Response, J. of NASAR*, vol. 17, no. 2, pp 1-15, 1999.

[118] J.R. Frost, "Principles of Search Theory, Parts III & IV," *Response, J. of NASAR*, vol. 17, no. 3, pp 1-23, 1999.

[119] D. Perkins, P. Roberts, and G. Feeney, *Missing Person Behaviour – An Aid to the Search Manager*, 1st ed. Northumberland, UK: The Centre for Search Research, 2003.

[120] R. Engelmore and T. Morgan, *Blackboard Systems*, 1st ed. Don Mills: Addison-Wesley, 1988.

[121] K. Krishnamoorthy, *Handbook of Statistical Distributions with Applications*. Boca Raton: Chapman & Hall/CRC, 2006.

[122] Z. A. Karian and E. J. Dudewicz, *Handbook of Fitting Statistical Distributions with R*. Boca Raton: CRC Press/Taylor & Francis, 2011.

[123] Z. A. Karian and E. J. Dudewicz, *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. Boca Raton : CRC Press, 2000.

[124] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists*, 9th ed. Upper Saddle River: Prentice Hall, 2000.

[125] A. Sen and M. Srivastava, *Regression Analysis: Theory, Methods and Applications*. New York: Springer-Verlag, 1990.

[126] M. Schimpl, C. Moore, C. Lederer, A. Neuhaus, J. Sambrook, J. Danesh, W. Ouwehand, M. Daumer, "Association between Walking Speed and Age in Healthy, Free-Living Individuals Using Mobile Accelerometry – A Cross-Sectional Study," *PLoS ONE*, vol. 6, No. 8, Aug. 2011. e23299. doi:10.1371/journal.pone.0023299.

[127] T. Öberg, A. Karsznia, and K. Öberg, "Basic Gait Parameters: Reference Data for Normal Subjects, 10-79 Years of Age," *Journal of Rehabilitation Research and Development*, vol. 30, no. 2, pp. 210-223, 1993.

[128] F. J. Imms and O. G. Edholm, "Studies of Gait and Mobility in the Elderly," *Age and Ageing*, vol. 10, no. 3, pp. 147-156, Jan. 1981.

[129] G. D. Knot, *Interpolating Cubic Splines*. Boston: Birkhauser, 2000.

[130] K. Aminian, P. Robert, E. Jéquier, and Y. Schutz, "Estimation of Speed and Incline of Walking Using Neural Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 44, no. 3, pp. 743-746, 1995.

[131] M. P. Fairclough. (2005). Terragen Classic (Version 0.9.43) [Software]. Available from http://planetside.co.uk/index.php?option=com_content &view=article&id=38&Itemid=174

[132] R. J. Koester, *Lost person behavior, a search and rescue guide on where to look – for land, air, and water*. Charlottesville: dbS Productions, 2008.

[133] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Philadelphia: SIAM, Society for Industrial and Applied Mathematics, 2009.

[134] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: new perspectives on some classical and modern methods," *SIAM Review*, vol. 45, no. 3, pp. 385-482, Aug. 2003.

[135] Deputy Minister of National Defence, the Chief of Defence Staff, the Commissioner, and the Canadian Coast Guard, *National search and rescue manual*, B-GA-209-001/FP-001 – DFO 5449. Ottawa, Ontario, Canada: Department of National Defence / Canadian Coast Guard, 2000.