The Value of Information in Multi-Objective Missions

Shaun Brown

A thesis submitted in fulfillment
of the requirements for the degree of
Master of Engineering (Research)

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

October 2008

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Shaun Brown

October , 2008

Shaun Brown
The University of Sydney

Master of Science
October 2008

The Value of Information in Multi-Objective Missions

In many multi-objective missions there are situations when actions based on maximum information gain may not be the ‘best’ given the overall mission objectives. In addition to properties such as entropy, information also has value, which is situationally dependent. This thesis examines the concept of information value in a multi-objective mission from an information theory perspective.

A derivation of information value is presented that considers both the context of information, via a fused world belief state, and a system mission. The derived information value is used as part of the objective function for control of autonomous platforms within a framework developed for human robot cooperative control.

A simulated security operation in a structured environment is implemented to test both the framework, and information value based control. The simulation involves a system of heterogeneous, sensor equipped *Unmanned Aerial Vehicles* (UAVs), tasked with gathering information regarding ground vehicles. The UAVs support an effort to protect a number of important buildings in the area of operation. Thus, the purpose of the information is to aid the security operation by ensuring that security forces can deploy efficiently to counter any threat.

A number of different local controllers using information based control are implemented and compared to a task based control scheme. The relative performance of each is examined with respect to a number of performance metrics with conclusions drawn regarding the performance and flexibility of information value based control.

Acknowledgements

Salah Sukkarieh for his time and patience, above and beyond the call of duty.

Jake Toh with whom much of the background work was a collaborative effort.

All the friends and family who have provided support and encouragement.

To my parents

Contents

List of Figures

List of Tables

Nomenclature

Notation

$\backslash(\cdot)$	Without
$(\cdot)^T$	Matrix transpose
$(\cdot)^{-1}$	Matrix inverse
$(\dot{\cdot})$	Derivative with respect to time
$(\bar{\cdot})$	Length of $(\cdot)$
$\triangleq$	Is defined

Subscripts

0	Initial
b	Body frame
c	Current
d	Desired
e	Error
f	Future
f	Corner set index
g	Observation set index
h	Horizontal
h	Human input
i	General set index
j	Action set index
k	At the present time
l	Undetected target location set index
m	Classification set index
n	Classification set index
q	Tracked ground vehicle set index
r	Sensor/Platform set index
s	Set of x points
s	Sensor frame

t	True state
t	Target frame
v	Velocity component
v	Vertical
x	At point x
w	Priority weighted
w	World frame
z	With observation
α	Hellinger-Bhattacharya affinity component
ϕ	Bank angle component

Superscripts

b	Body frame
c	Related to classification of tracked ground vehicles
d	Related to classification of the undetected ground vehicle
l	Related to proximity of tracked ground vehicles
s	Related to the position of the undetected ground vehicle
s	Sensor frame
t	Related to the position of tracked ground vehicles
t	Target frame
U	Related to the uniform distribution
w	World frame
z	With observation
λ	Related to proximity of the undetected ground vehicle

General Symbols

a	Task
$\mathbb{A}$	Action set
a	Asset
A	Asset set
b	Belief
B	Belief set
$\mathcal{B}$	Belief space
C	Transformation matrix
C	Information cost
$\mathbb{C}$	Set of classifications
C	Cost coefficient
$\mathcal{C}$	In the Coordination domain
δ	Hellinger-Bhattachrya distance
E	Expected value

$f()$	Function
$G(.)$	Gaussian function
$G(.)$	Transfer function
h	Height
h	Hypothesis particle
$H(.)$	Entropy of distribution
Hz	Hertz
$\mathbb{H}$	Hypothesis particle set
$I(.)$	Mutual information
J	Johnson criteria
$\mathcal{J}$	Objective function
k	Current time
k	Controller gain
k	Runge-Kutta coefficient
K	Normalising value
$\mathcal{L}3$	Level 3 fusion domain
m	Meters
m	Mission objective
M	Complete mission description
$\mathcal{M}$	Mission state
n	Unit vector
n	Number of pixels
N	Set of ground vehicle types
$\mathcal{O}$	In the Organization domain
p	Task priority
$\mathbb{P}$	Task priority set
$P(.)$	Probability distribution
P	Coordinate Vector
q	Tracked target
Q	Set of tracked targets
R	Information preference
s	Seconds
s	Node
s	Laplace transform
$\mathbb{S}$	Node set
S	Node subset
S	Sensor suitability
T	Threat metric
u	Action
U	Set of all actions
$\mathcal{U}$	Utility function

v	UAV airspeed
v	Value
V	Value set
w	Task weight
W	Task weight set
x	Map point
$\mathbf{x}$	State variable
X	Platform state
z	Observation
Z	Observation set
Z_g	Ground plane
α	Angle
α	Hellinger-Bhattacharya affinity
β	Projected area
η	Vulnerability limit distance
γ	Flight path angle
Γ	Sensor suitability
κ	Pixel density $pixels/m^2$
μ	Pixel count $pixels/m$
ν	Vulnerability
Π	Platform suitability
ϕ, θ, ψ	Euler roll, pitch and yaw angles
x, y, z	Map coordinate directions
σ^2	Distribution variance
τ	Information decay constant
Υ	Set of allocated tasks

Typefaces

`MyComponent`	Executable component name

Abbreviations

CG	Centre of Gravity
DoF	Degree of Freedom
GUI	Graphical User Interface
FHF	Finite Horizon Footprint
FHP	Finite Horizon Peak-chasing
FOV	Field Of View
HB	Hellinger-Bhattacharya
HRI	Human Robot Interface
IHP	Infinite Horizon Peak-chasing
IPDI	Increasing Precision with Decreasing Intelligence

IR	Infra-Red
JDL	Joint Directorate Laboratories
ODE	Ordinary Differential Equation
PI	Proportional Integral
ROC	Rate Of Climb
UAV	Unmanned Aerial Vehicle

Chapter 1

Introduction

1.1 Motivation

Autonomous vehicles are being increasingly utilised for information gathering tasks in military operations, particularly for those missions considered dull, dirty or dangerous. A wide range of both Uninhabited Aerial Vehicle (UAV) and Uninhabited Ground Vehicle (UGV) types have been successfully deployed around the world, adding significant capability to the militaries that field them. Much recent research has focused on teams and the co-ordintation/cooperation between both the autonomous vehicles themselves, and the human users. These developments have enhanced the ability of autonomous systems to accomplish complex missions and extend the capability of simple platforms with limited sensing ability. These considerations are particularly important for emerging applications such as law enforcement, search and rescue, bush fire fighting and environmental survey.

Autonomous platforms currently in service are primarily employed in the information gathering role as an aid to situational awareness, where manned systems may be too costly, the situation considered too dangerous or mission persistence insufficient. Being a situational awareness tool, the information gathered by the autonomous system will commonly serve as an enabler for action. This action may be performed by either the information gatherer or some external agent. For example information gathered by a camera equipped UAV in aid of bush fire fighting will be used by the fire chief to direct fire crews and refine resource allocation.

In the example applications of autonomous vehicles in the civilian and military domains, the information requirements are varied and dynamic. Further the information may be used for a number of purposes and directed by agents external to the system. This gives rise to information objectives which may be dynamic with respect to external mission demands and current information state. In addition, the demands of the final information customer or customers may result in multiple conflicting information objectives. As the information gathering resources are limited, the problem for the system becomes a decision problem of how to autonomously allocate the available resources to ensure that the information gathered is most beneficial to the objectives of a wider mission, even as the mission changes.

This recognition of the dynamic nature of the wider mission and the need to adapt to the wider mission requirements within a multi-objective scenario has not been comprehensively examined within the robotics community. This gap in research between operations research and autonomous control of systems has been the motivation for this research.

1.2 Thesis Objectives

This thesis aims to address the control and resource allocation problem outlined by extending information theoretical control concepts to account for multiple objectives and, via a framework for examining information value, to enable different information types to be used in determining cooperative control actions. This extension of information theoretic concepts also aims to provide a link between the information requirements of a wider mission and the cooperative control scheme, through human level input from a single operator.

The implementation in the simulation environment is aimed at demonstrating control of a system of autonomous information gatherers, based on the concept of information value.

1.3 Thesis Contributions

The contributions of this thesis are:

- Presentation of a flexible framework for human-robot cooperation that allows for a wider mission objective to flow through the system and be incorporated into the control of autonomous platforms. The framework is designed for the implementation of different control architectures, different fusion components as well as centralised and distributed control architectures.

- Presentation of a consistent formulation of information value, based on information theoretic concepts that takes into account the wider mission objectives as well as the current information state. The formulation is capable of dealing with multiple objectives and dynamic information states and mission objectives.

1.4 Thesis Structure

Chapter 2 examines some relevant work from the economics community in the area of valuing market goods and information as a tradeable good. The difficulty of defining information value, due to a number of unique attributes, is highlighted along with the relationship between information value and the agent decision problem.

Chapter 3 details a method of determining information value for a mission with multiple objectives. The formulation captures both the current information state along with a preference ordering for different information types that is situation dependant.

Chapter 4 describes the architecture of the simulator used to investigate information value based control. The architecture is generic to a multi-agent information gathering task and can be applied to both centralised and distributed control and information fusion schemes.

Chapter 5 details the implementation of information value for an urban security scenario. This scenario consists of a number airborne sensor platforms finding and tracking ground vehicles in an urban road network where static assets are to be protected. A number of different control schemes are described along with an outline of the code used for simulation and the relevant operator interfaces.

Chapter 6 provides detail on the setup of the simulated system and includes the situations:

- baseline with no modification to the information gain rate utility/preference ordering
- modification to information gain rate utility/preference ordering based on human input
- addition of external information into the system

Data from running each situation in simulation is presented with associated evaluation metrics.

Chapter 7 draws some conclusions from the simulation data and identifies potential directions for future work.

Chapter 2

Related Work

2.1 Valuing Goods in a Market

There are many economic theories and models which provide ways of valuing and predicting the value of market goods. These can be broadly described by the concepts of intrinsic and subjective value.

Intrinsic value theory relates value to some intrinsic property of the good. This implies that a market value is determined irrespective of the individual value judgements of the agents participating in the market [7]. An important incarnation of the concept of intrinsic value, proposed by Smith, Ricardo and Marx among others, is the labour value theory [30]. This states that if supply and demand are in equilibrium, the value of a good is related only to the labour or effort expended in producing the good [35].

In the body of current economic theory, intrinsic value is not generally accepted as being useful in free markets, but is often used by proponents to decry profiteering and to demonstrate labour exploitation, thus justifying a command economy [7]. One of the issues pointed to as a flaw in intrinsic value, is that for trade to occur each agent must have an incentive to trade. If the value of a good were constant for all agents, then an agent will have little incentive to trade, or alternatively once a good is acquired there is as much incentive to on-trade the good to another agent as there is to keep the good [7]. This implies that for market trade, there must be some discrepancy in the value of the good as judged by the agent's performing the transaction and hence the value of a good is not solely dependant on intrinsic properties.

This subjective view of value has traditionally been illustrated by the so called diamond/water paradox. A diamond does not fill any base human need to survive, unlike water, which is arguably more useful. However diamonds are far more costly and by implication more valuable [39]. If offered either diamonds or a bottle of water for the same small price a rational human would, in normal circumstances choose the diamonds. If the same person found themselves stranded and dying of thirst the rational choice would be reversed. Subjective value theory thus relates value to the ability of the good to satisfy some desire or utility [28]. In this model each agent has some internal method for assessing the value, or for preference ordering goods, with respect to a satisfaction criteria. Hence through trading, the individual value judgements of all agents participating in the market can influence the value of a good.

The concept of scarcity and its relationship to utility has also been illustrated using the diamond/water paradox. In the case of a person dying of thirst, if the water/diamond offer is made again and again there will be a point at which the utility for water decreases below that of diamonds and thus the diamonds will be chosen. As an agent acquires goods, the value of having more of the same or similar goods may decrease to a point where there is no net value for any further acquisition. This behaviour is described by the marginal value theory, where each agent considers the goods currently owned in the subjective valuation mechanism [11], [19]. This is the basis of utility or indifference curves, where the utility of acquiring examples of a good tends toward zero as more are acquired. If an agent has limited buying power in a market where two goods, A and B, are available for purchase, the relative utility of acquiring good A over good B will reduce as the agent's stocks of good A increase. Even if the utility of good A is initially higher than that of good B, at some point the utility gain of acquiring more of good A will be less than that of acquiring good B [18]. Thus for an agent to maximise utility this effect of marginal utility must be considered.

The effect of changes to agent's preference ordering due to introduction of new goods, scarcity, marginalisation or other influences will be reflected in the market price. This has been used by a variety of multi-agent systems as a mechanism for cooperation and task allocation. In general the agents receive rewards for tasks, which require expenditure of finite agent resources. By forming an economy and trading tasks, global good is done through each agent's self interest while the competitive element enables competing local information to be dealt with in context of the team objective [9]. Typically the trade will take the form of an auction such as in [3], [13], [17]and [20] with various mechanisms used to conduct the auction. In [2], the auction is centralised with a master auctioneer setting the reward for each task, the agents then bid to perform the tasks based on their estimated cost of action. An extension to the task based auction is presented in [42] which allows auctioning of roles or task abstractions in addition to task fragments. In [37] the centralised auctioneer is used only to enter new tasks into the system with each agent able to hold a subsequent auction for all or part of any task it is contracted to complete.

However, simply observing the value of the traded good does not provide details of the internal valuation mechanism applied by an individual agent. Many techniques for learning an agents utility are available such as indifference curves [18] or graph based methods including the active utility elicitation in [4], [5]. Alternatives, which use market based schemes such as [22], the attention bond used by [24] or the techniques based on buyer collusion [27] also provide ways of finding agent utility. These methods use market observations to uncover utility, however for the agent to participate in the market, or in the case of a multi-agent system to also coordinate actions, the agent must still be designed with some suitable internal valuation mechanism.

2.2 The Market Value of Information

While some research has suggested that in information markets with human agents trade and value show similar trends to that of more material goods [34], traditional economic models often have difficulty dealing with information as a commodity. This is due to the intangible nature of information as a good and a number of unique attributes.

1. *Information is abstract.* While data is corporeal and readily measurable, information relies on the understanding and interpretation of the consumer. A large amount of data can be summarised by a concept and a single ideogram can stand in for an entire collection of ideas.

2. *The marginal cost of information gain may be low.* Information can be cheap to produce, have a negligible replication and distribution costs, and be cheap to store.

Hence cost of information in isolation is unlikely to provide a good yard stick for measuring the value of information to an agent.

3. *Information is an experience good.* The true value of a piece of information cannot be known until it has been consumed or processed. Once consumed there is no further value to be obtained from acquiring the same information and in general it cannot be 'unconsumed'. This leads to the so called *inspection paradox* [41]. Additionally acquired information may be sold or distributed by the purchasing agent without information loss.

4. *Acquiring new information can invalidate information already owned.* A physical tool will continue to be able to perform its designed function even after the purchase of newer or more efficient tools and thus will still have a measurable value in the market. However if information is made obsolete due to the addition of new information, the residual market value will be zero.

In [40] these characteristics are addressed by decomposing information value into three components. The *relevance*, which indicates the extent the information relates to a decision problem, *utility* the time function relating to the ability of the information to be able to influence a decision and *acceptance* which relates to the agent's ability to process information given the current situation. Note that all three components have a temporal element.

Both the relevance and value of information to an agent have conditionality and limit characteristics. The conditionality characteristic implies that information may only be relevant and valuable in certain conditions, which may be defined by context or events. As an example of event based relevance, if a security system can deploy a sensor to check the integrity of doors, this information will be irrelevant if the building has burnt down. For context based relevance in a homeland security type scenario, a person's profession will, in general, have little relevance or value in determining the terrorist threat posed. However given that a person has bought a large quantity of ammonium nitrate the person's profession will be highly relevant in determining threat. The concept of conditionality is partially captured in the objective theory of value through the marginal contribution of the information to the agent's goals.

Information relevance and value may have limits where the marginal value of information may be zero above or below a certain level. For example, if a convoy of critical supplies has a particular pre-planned route, information may indicate the presence of hostile parties. If the extent of the information only extends to presence then this may have no influence on the configuration of the convoy or the route taken. However if the information indicates the size and armament of hostile forces that may be encountered, then the route may be altered. As an example of an upper limit of information value, localizing a target to greater accuracy than the blast effects of the weapon being employed against it, will have no influence on the decision of where to aim.

In [41] the author extends objective and marginal value theories to instrumental goods, such as information. In the proposed valuation mechanism, a value is placed on the state the agent is trying to reach and a path of intermediate states identified. If multiple paths exist between states, then the Shapley value of coalition contribution can be used to allocate the total value of the final state to all intermediate state. This implicitly captures conditionality in information value, by formalising the relationship between different pieces of information and allowing the formation of 'infomation coalitions' without which the desired state cannot be reached. To be useful the goal and body of knowledge to be acquired must be well defined, needing decomposition to a level that is manageable for computation. For scenarios where highly abstract goal states may exist or the path between information states may be non-linear, this may be problematic.

2.3 The Decision Problem and the Value of Information

If information is not free and an agent has finite resources, then the question of whether to acquire new information becomes part of the decision agent's decision problem. In classical decision making problems the relative preference for action is encoded by a utility function. This defines a gain for each outcome resulting from actions taken by the decision maker. Except in the case of perfect knowledge, the deterministic utility function $\mathcal{U}(\mathbf{x}, u)$, with u the utility gain associated with a particular action when the world is in state $\mathbf{x}$, is not especially useful as the true state of the world is not known with precision and the true utility gain is uncertain [16]. If a distribution $P(\mathbf{x})$ summarises all the probabilistic information available about the state $\mathbf{x}$, then expected utility is the probabilistic estimate of the deterministic utility.

$$\mathcal{J}(u) \triangleq E\{\mathcal{U}(\mathbf{x},u)\} = \sum\nolimits_{u\in U}\mathcal{U}(\mathbf{x},u)P(\mathbf{x}|u) \tag{2.1}$$

A rational decision maker will always select actions that maximise expected utility. The utility function form, however says nothing about the nature of the actions or how they become available to the decision maker, only the impact of an action on the state x is of interest [8].

If an agent uses expected utility in decision making, then both the belief about the state $\mathbf{x}$ and the impact of the action taken $P(\mathbf{x}|u)$ will influence the desirability of that action. Thus any information which modifies the agents belief on $\mathbf{x}$ can be considered as having value with respect to maximising utility. At the agent level, information value can then be defined as the expected gain having the information compared to the expected gain not having the information and then subtracting the cost of acquiring the information. This concept is used as the basis for valuing information in [14], [31] and [38].

In the investigation of the utility of price information in [26] the utility is defined as the expected cost saving in having a retailers price information. This is extended in [25] to account for more abstract concepts like brand by applying monetary value when calculating utility. This approach determines the expected best price before and after price information is purchased and uses the marginal utility as the decision variable for purchase. However the limitation of using such a marginalisation approach is that characteristics of price distributions in the market are assumed.

In information theoretic control the utility is related to the certainty of the estimate $P(\mathbf{x})$ [15], [31], thus the value of new information is implicitly correlated to its ability to reduce the uncertainty about the true state of $\mathbf{x}$. Using mutual information as the expected change in the distribution after observation provides a way of predicting how the utility will be affected prior to gaining information. Thus nothing is assumed *a priori* regarding the shape of the distribution.

2.4 Other Information Value Metrics

Methods of valuing information based on content are fundamental in modern data mining systems, notably internet search engines. Typically approaches such as word frequency [40], a learned model [1] or heuristic templates such as described in [10] are used by these search engines to rank new information. While many of these algorithms are highly sophisticated, the ranking applied is subjective, with the ranking of the information retrieved reflecting the operator's internal valuation if he or she has sufficient skill in phrasing the search query for the given algorithm. In general, whatever the search engine algorithm, the objective is to minimise the time taken to find information while maximising the probability that the information returned is what the operator is looking for. This approach is not readily extendable to multiple competing objectives or dynamic objectives.

In some applications the quality of the information is considered synonymous with value. In [29] an operator ranks information sources based on quality, with this weighting used in the fusion process. In other systems such as [12] and [32] the value of information is judged according to its reliability based on trust criteria or corroboration with other information. These information valuations do not consider the marginal or situational behaviour of the value of information value.

2.5 Summary

Determining the value of market goods has been well researched in economics, however many existing theories have some difficulty in relating value to an intangible good such as information. Where the value of intangible goods has been examined, it has been suggested that the value of information to an agent follows similar principles to other goods and is subjective. The value being a function of both the agents internal priorities and the perceived state of the world.

If an autonomous agent's control mechanism can be represented by a gain based objective function, then at each decision point, the reward gain will be impacted by both the agent's belief about the world, and the objective variables. Thus the value of information can be described in terms of an objective function. While this is not new, it has not been well understood in regard to autonomous control of multiple agents with competing objectives and dynamic missions.

Chapter 3

The Value of Information

3.1 Introduction

In this chapter a value formulation based on utility and information entropy is applied to the information gathering mission and extended to allow different information states to be dealt with. The fundamental utility based value of information proposed by Sheridan is outlined in Section 3.2 and then extended to the multivariate case in section 3.3. In Section 3.4, the information gathering scenario is introduced with the relevant information value formulations shown.

3.2 Utility Based Value of Information

In [38], Sheridan formally defines information value as '*the difference between the gain in taking the best action given each specific state of a random variable* $\mathbf{x}$ *and the gain in taking the best action knowing only* $P(\mathbf{x})$*, minus the effort or other cost of discovering the truth from the initial uncertain state.*'

If a finite set of j actions is available to a decision making agent, the utility for performing action u_j conditioned on the random variable of interest being in state i is given by the utility function $\mathcal{U}$.

$$\mathcal{U}(u_j|\mathbf{x}_i) \tag{3.1}$$

If $\mathbf{x}$ has a finite set of possible states, the average utility gain for an agent taking the best action for each occurrence of the state i is the average utility $\mathcal{U}_{avg}$.

$$\mathcal{U}_{avg} = \sum_i P(\mathbf{x}_i)\{\max_j[\mathcal{U}(u_j|\mathbf{x}_i)]\} \tag{3.2}$$

For an agent to realise the average utility gain in Equation 3.2, the true state of $\mathbf{x}$ must be known at each decision point. If the agent does not have perfect knowledge and $P(\mathbf{x})$ summarises all the probabilistic information available, then the agent's gain for each occurrence of the state i is the average expected utility gain $\mathcal{U}'_{avg}$.

$$\mathcal{U}'_{avg} = \max_j\{\sum_i P(\mathbf{x}_i)\mathcal{U}(u_j|\mathbf{x}_i)\} \tag{3.3}$$

The difference between the average utility and average expected utility represents the value in utility units of determining the true state of $\mathbf{x}$ from the current $P(\mathbf{x})$.

$$V^* = \mathcal{U}_{avg} - \mathcal{U}'_{avg} \tag{3.4}$$

If the agent updates the belief $P(\mathbf{x})$ by observing $\mathbf{x}$ then the additional information has value with respect to utility gain. If a single observation z of $\mathbf{x}$ is made, then value of the information in z is the difference between V^* before, $\backslash z$, and after the observation.

$$V_z = V^*_{\backslash z} - V^*_z \tag{3.5}$$

The terms V^* and V_z provide general measures of information value that can be applied whenever there is probabilistic representation of the variable of interest and a utility function $\mathcal{U}$ is available.

In general, for an agent to make an observation and update $P(\mathbf{x})$ requires the expenditure of resources. Following the value of information in Equation 3.4 being described as the difference in expected gain when knowing the true state of a variable and having an estimate of its state, a natural description for the cost of information is the cost of determining the true state of $\mathbf{x}$ from the current estimate $P(\mathbf{x})$. If the cost of obtaining or processing new information can be defined in terms of resource units expended per *bit* of information, then the expected cost of determining the true state of $\mathbf{x}$ or C_{avg} can be found using the entropy of the distribution $P(\mathbf{x})$.

$$\begin{aligned} C_{avg} &= \mathrm{C}H(\mathbf{x}) && (3.6) \\ &= -\mathrm{C}\sum_i P(\mathbf{x}_i)\log P(\mathbf{x}_i) && (3.7) \end{aligned}$$

The cost coefficient C will define the resources expended, such as time, fuel, dollars or bandwidth. For resources other than bandwidth or computation time, the coefficient is unlikely to be static and may be a function of some external variable or even the state of interest itself C($\mathbf{x}$).

The net value of determining the true state of $\mathbf{x}$ is then found by subtracting the cost from the value.

$$V_{net} = V^* - \mathrm{C}_{avg} \qquad (3.8)$$

In an uncertain world, multiple observations are likely to be required to provide a good estimate of the true state of $\mathbf{x}$. This makes defining the cost coefficient C difficult. A more practical approach is to determine the cost of an individual or small set of observations. In Equation 3.5 a portion of the total information value is apportioned to the information from a single observation. Similarly the cost of a single observation can be defined as a function of the difference between the cost of determining the true state of $\mathbf{x}$ before and after using the information in z. The single observation cost coefficient, C_z can then be used.

$$\begin{aligned} C_z &= \mathrm{C}_z\left(H(\mathbf{x}^{\backslash z}) - H(\mathbf{x}^z)\right) && (3.9) \\ &= -\mathrm{C}_z\left(\sum_i P(\mathbf{x}_i^{\backslash z})\log P(\mathbf{x}_i^{\backslash z}) - \sum_i P(\mathbf{x}_i^z)\log P(\mathbf{x}_i^z)\right) && (3.10) \end{aligned}$$

Equation 3.9 shows that for a constant C_z, only the change in entropy of the distribution $P(\mathbf{x})$ influences the cost of information.

$$\begin{aligned} H(\mathbf{x}^{\backslash z}) - H(\mathbf{x}^{z}) &= H(\mathbf{x}) - H(\mathbf{x}|z) \\ &\triangleq I(\mathbf{x};z) \end{aligned} \tag{3.11}$$

The mutual information, $I(\mathbf{x};z)$, of the distributions $P(\mathbf{x}|z)$ and $P(\mathbf{x})$ represents the expected change in the entropy after adding information from the observation.

$$\begin{aligned} I(\mathbf{x};z) &= H(\mathbf{x}) - H(\mathbf{x}|z) \\ &= H(z) - H(z|\mathbf{x}) \\ &= H(\mathbf{x}) + H(z) - H(\mathbf{x},z) \end{aligned} \tag{3.12}$$

Where the distribution entropies can be found from the standard entropy equations.

$$H(.) = -\sum_i P(._i) \log P(._i) \tag{3.13}$$

$$H(\mathbf{x},z) = -\sum_i \sum_j P(\mathbf{x}_i,z) \log P(\mathbf{x}_i,z) \tag{3.14}$$

$$H(\mathbf{x}|z) = -\sum_i P(\mathbf{x}_i|z) \log P(\mathbf{x}_i|z) \tag{3.15}$$

$$H(z|\mathbf{x}) = -\sum_i P(z|\mathbf{x}_i) \log P(z|\mathbf{x}_i) \tag{3.16}$$

Hence if the sensor can be modelled by $P(z|\mathbf{x})$, the mutual information can be determined prior to the observation through the difference $H(z) - H(z|\mathbf{x})$. In this case $P(z)$ can be determined through the total probability theorem shown in Equation 3.17.

$$P_z(z) = \sum_{\mathbf{x}} P_{\mathbf{x}|z}(z|\mathbf{x}) P_{\mathbf{x}}(\mathbf{x}) \tag{3.17}$$

Substituting the mutual information into equation 3.9 gives a cost form that can be readily determined.

$$C_z = \mathrm{C}_z I(\mathbf{x}; z) \tag{3.18}$$

For each observation z that may be made, the expected net value of the information added to the system is found by subtracting the single observation cost from the single observation value.

$$V_{net}^{z} = V_z - C_z \tag{3.19}$$

3.3 Value of Information for Multiple Variables

In Section 3.2 the state of interest in the agent's utility function was represented by a single random variable $\mathbf{x}$. In a situation where the state of interest is best understood as a joint distribution of multiple random variables, the agent's utility will become a function of this distribution. If four variables $\mathbf{x}^t$, $\mathbf{x}^s$, $\mathbf{x}^c$ and $\mathbf{x}^d$ determine the state of interest $\mathbf{x}$ then the utility function from Equation 3.1 will become:.

$$\mathcal{U}(u_j|\mathbf{x}_i^t, \mathbf{x}_l^s, \mathbf{x}_m^c, \mathbf{x}_n^d) \tag{3.20}$$

Similarly the agents average utility and expected average utility will take the joint distribution form.

$$\mathcal{U}_{avg} = \sum_i \sum_l \sum_m \sum_n P(\mathbf{x}_i^t, \mathbf{x}_l^s, \mathbf{x}_m^c, \mathbf{x}_n^d)\{\max_j[\mathcal{U}(u_j|\mathbf{x}_i^t, \mathbf{x}_l^s, \mathbf{x}_m^c, \mathbf{x}_n^d)]\} \tag{3.21}$$

$$\mathcal{U}_{avg}^{'} = \max_l\{\sum_i \sum_j \sum_m \sum_n P(\mathbf{x}_i^t, \mathbf{x}_l^s, \mathbf{x}_m^c, \mathbf{x}_n^d)\mathcal{U}(u_j|\mathbf{x}_i^t, \mathbf{x}_l^s, \mathbf{x}_m^c, \mathbf{x}_n^d)\} \tag{3.22}$$

For the multivariate case, the information value functions retain the forms shown in Equations 3.4 and 3.5. The information cost will take the joint distribution form of Equation 3.9 and the information cost for a single observation will use the mutual information of the joint distribution.

$$\begin{aligned} C_z &= \mathrm{C}_z \Bigg(\sum_i \sum_l \sum_m \sum_n P(\mathbf{x}_i^{tz}, \mathbf{x}_l^{sz}, \mathbf{x}_m^{cz}, \mathbf{x}_n^{dz}) \log\{P(\mathbf{x}_i^{tz}, \mathbf{x}_l^{sz}, \mathbf{x}_m^{cz}, \mathbf{x}_n^{dz})\} \\ &\quad - \sum_i \sum_l \sum_m \sum_n P(\mathbf{x}_i^{t\backslash z}, \mathbf{x}_l^{s\backslash z}, \mathbf{x}_m^{c\backslash z}, \mathbf{x}_n^{d\backslash z}) \log\{P(\mathbf{x}_i^{\backslash z}, \mathbf{x}_l^{s\backslash z}, \mathbf{x}_m^{c\backslash z}, \mathbf{x}_n^{d\backslash z})\} \Bigg) \end{aligned} \tag{3.23}$$

$$C_z = \mathrm{C}_z I(\mathbf{x}_i^t, \mathbf{x}_l^s, \mathbf{x}_m^c, \mathbf{x}_n^d; z) \tag{3.24}$$

Equation 3.24 assumes that the function defining the cost per bit of information is the same across all the states being observed. If this is not the case then the cost coefficient should reflect this by being a function of all the variables of interest $C_z(\mathbf{x}_i^t, \mathbf{x}_l^s, \mathbf{x}_m^c, \mathbf{x}_n^d)$.

3.4 The Multi-Objective Scenario

In an example scenario, there are a number of assets, both mobile and static, that need to be protected as part of ongoing security operation in an urban environment. An unknown number of ground vehicles are in the area, some of which may pose a threat to the assets. An information gathering system consisting of a number of sensor equipped mobile agents is available to provide information about vehicles within the area of operation. This information is provided to security resources, which can then deploy to counter any threats. If security resources are limited, being unable to provide protection to all assets at once, then the efficiency of the security response will be determined by what is known prior to deployment regarding the nature and location of any threats.

As an information gathering resource, the agent's primary objective is to maximise information regarding threats in order to facilitate an efficient security response. In support of this objective the agent's tasks can be decomposed: to find targets, determine whether they are a threat and once identified as a threat, to maintain an estimate of current location. This can be considered as the multi-objective mission: search for targets, classify targets and track targets. For each planning cycle the agent's decision problem is to determine which observation/s should be made in order to provide the most informative threat picture.

The current threat picture will be some function of the proximity of all ground vehicles to the protected assets, the types of target and hence the threat posed, and the uncertainty regarding the vehicles true position. As the number and type of ground vehicles is initially unknown, the utility gain for the observation z will contain the variables:

- $\mathbf{x}^t$, indicating the existence of a known or tracked vehicle in the observation
- $\mathbf{x}^c$, the classification of the tracked vehicle observed
- $\mathbf{x}^l$, the relative proximity of the tracked vehicle to all assets
- $\mathbf{x}^s$, indicating the existence of a previously unknown vehicle in the observation
- $\mathbf{x}^d$, the classification of the unknown vehicle
- $\mathbf{x}^\lambda$, the relative proximity of the unknown vehicle to all assets

Intuitively it may be expected that vehicles considered to pose a significant threat will try to reach the assets being protected and thus are more likely to be found near or heading towards these assets. Further, if the vehicles are cooperating then there will be a correlation between the locations of all those cooperating.

$$\begin{aligned} P(\mathbf{x}^t|\mathbf{x}^c) &\neq P(\mathbf{x}^t) \qquad (3.25) \\ P(\mathbf{x}^s|\mathbf{x}^d) &\neq P(\mathbf{x}^s) \\ P(\mathbf{x}^t|\mathbf{x}^s) &\neq P(\mathbf{x}^t) \qquad (3.26) \\ P(\mathbf{x}^s|\mathbf{x}^t) &\neq P(\mathbf{x}^s) \end{aligned}$$

This has implications for both the tracking algorithm and the information value functions. To avoid the complex inference required to determine the level of co-operation between targets, for the purposes of this study it has been assumed that ground vehicle actions are independent and that vehicle type and location are independent given the existence of the vehicle. Where the threat primarily comes from a single vehicle which is difficult to distinguish from civilian traffic, this assumption is reasonable. The independence assumption allows the utility function in Equation 3.20 to be the sum of the independent utilities as in Equation 3.27. In this case the action u is taking the observation z and $1..j$ is the set of possible points in the area that may be observed.

$$\mathcal{U}(u_j|\mathbf{x}^t,\mathbf{x}^s,\mathbf{x}^c,\mathbf{x}^d,\mathbf{x}^l,\mathbf{x}^\lambda) = \mathcal{U}(u_j|\mathbf{x}^t,\mathbf{x}^c) + \mathcal{U}(u_j|\mathbf{x}^t,\mathbf{x}^l) + \mathcal{U}(u_j|\mathbf{x}^s,\mathbf{x}^d) + \mathcal{U}(u_j|\mathbf{x}^s,\mathbf{x}^\lambda) \quad (3.27)$$

For a set of q tracked targets and a single undetected target in the observation, the utility of observing this state can be found via the value, v, which is assigned to each of the true states of $\mathbf{x}$ as part of the mission setup. Equation 3.28 shows the observation utility determined from value of the $\mathbf{x}$ states observed.

$$\mathcal{U}(u_j|\mathbf{x}^t,\mathbf{x}^s,\mathbf{x}^c,\mathbf{x}^d,\mathbf{x}^l,\mathbf{x}^\lambda) = \sum_q v^l_q + \sum_q v^c_q + v^\lambda + v^d \quad (3.28)$$

The current belief on each of the variables of interest, either maintained directly or accessed by the agent, is used to calculate the average and expected utility for the next observation z. Due to the independence assumption, the Equations 3.21 and 3.22 can be applied to find the utility of z, as per Equations 3.29 and 3.30 with the value for each state found from Equation 3.28.

$$\mathcal{U}_{avg} = \sum_i \sum_m P(\mathbf{x}^l_i, \mathbf{x}^c_m, \mathbf{x}^t_m) \max_j \mathcal{U}(u_j|\mathbf{x}^l_i, \mathbf{x}^c_m, \mathbf{x}^t_m) + \sum_l \sum_n P(\mathbf{x}^\lambda_l, \mathbf{x}^d_n, \mathbf{x}^s_n) \max_j \mathcal{U}(u_j|\mathbf{x}^\lambda_l, \mathbf{x}^d_n, \mathbf{x}^s_n) \quad (3.29)$$

$$\mathcal{U}'_{avg} = \max_j \sum_i \sum_m P(\mathbf{x}^l_i, \mathbf{x}^c_m, \mathbf{x}^t_m) \mathcal{U}(u_j|\mathbf{x}^l_i, \mathbf{x}^c_m, \mathbf{x}^t_m) + \max_j \sum_l \sum_n P(\mathbf{x}^\lambda_l, \mathbf{x}^d_n, \mathbf{x}^s_n) \mathcal{U}(u_j|\mathbf{x}^\lambda_l, \mathbf{x}^d_n, \mathbf{x}^s_n) \quad (3.30)$$

Where i and l are the summations across all possible locations and m and n the summations across all possible types for the tracked and untracked vehicles respectively.

The probability of the classification of an undetected target cannot be extracted from the observable belief states unless a prior belief on the target type distribution is available. In line with the assumption that nothing is known *a priori* regarding the targets in the area or indeed the existence of potential targets, a flat distribution across all target types can be used. Once a target has been observed a distribution of classification probabilities for the target can be generated and hence the undetected target distribution over N possible target types will remain naive.

$$P(\mathbf{x}^d) = \frac{1}{N} \tag{3.31}$$

Where N is the number of distinct target types.

For the case of multiple known targets being present in the observation, the Equations 3.29 and 3.30 need to reflect the joint distribution of the tracked target classifications. If the assumption of independent target behaviour holds then Equations 3.21 and 3.33 will apply for q targets which have a probability of appearing in the observation.

$$\begin{aligned}\mathcal{U}_{avg} &= \sum_i \sum_m P(\mathbf{x}^l_{i_1}, \mathbf{x}^c_{m_1}, \mathbf{x}^t_{m_1}) \max_j \mathcal{U}(u_j | \mathbf{x}^l_{i_1}, \mathbf{x}^c_{m_1}, \mathbf{x}^t_{m_1}) + \ldots \\ &\quad + \sum_i \sum_m P(\mathbf{x}^l_{i_q}, \mathbf{x}^c_{m_q}, \mathbf{x}^t_{m_q}) \max_j \mathcal{U}(u_j | \mathbf{x}^l_{i_q}, \mathbf{x}^c_{m_q}, \mathbf{x}^t_{m_q}) \\ &\quad + \sum_l \sum_n P(\mathbf{x}^\lambda_l, \mathbf{x}^d_n, \mathbf{x}^s_n) \max_j \mathcal{U}(u_j | \mathbf{x}^\lambda_l, \mathbf{x}^d_n, \mathbf{x}^s_n)\end{aligned} \tag{3.32}$$

$$\begin{aligned}\mathcal{U}^{'}_{avg} &= \max_j \sum_i \sum_m P(\mathbf{x}^l_{i_1}, \mathbf{x}^c_{m_1}, \mathbf{x}^t_{m_1}) \mathcal{U}(u_j | \mathbf{x}^l_{i_1}, \mathbf{x}^c_{m_1}, \mathbf{x}^t_{m_1}) + \ldots \\ &\quad + \max_j \sum_i \sum_m P(\mathbf{x}^l_{i_q}, \mathbf{x}^c_{m_q}, \mathbf{x}^t_{m_q}) \mathcal{U}(u_j | \mathbf{x}^l_{i_q}, \mathbf{x}^c_{m_q}, \mathbf{x}^t_{m_q}) \\ &\quad + \max_j \sum_l \sum_n P(\mathbf{x}^\lambda_l, \mathbf{x}^d_n, \mathbf{x}^s_n) \mathcal{U}(u_j | \mathbf{x}^\lambda_l, \mathbf{x}^d_n, \mathbf{x}^s_n)\end{aligned} \tag{3.33}$$

In the scenario presented, the primary cost to the agent in making an observation is the resources expended in manoeuvering to a position to allow the best observation z. In this case C_z is a function of the spatial parameters of z and is most appropriately measured in time, fuel or distance units. As the cost is independent of the information gained, the cost of the observation is used directly rather than the cost per bit formulation shown in Equation 3.18. If the agent performs an observation or set of observation actions each planning cycle then only the relative cost of each potential action is of interest in determining V^z_{net}.

3.5 Summary

In this chapter a scheme for valuing different information types has been presented. The value process considers both human judgements, the state of the world observed and the current information state. The information valuation scheme is applied to the specific multi-objective scenario of security and asset protection in an urban environment. Chapter 4 outlines a complete system architecture for applying information value to the control of a heterogeneous sensor network for the security scenario.

Chapter 4

Architecture

4.1 Introduction

Control of a number of sensor platforms based on the value of information outlined in Section 3.4 was implemented in a mid-fidelity simulation environment. The system was originally developed for testing algorithms and architectures for the co-operative control of a system of heterogeneous airborne sensor platforms tracking ground targets in an urban environment. The architecture of the simulator is modular allowing components complying with the interface standard to be inserted or removed with relative ease.

This section provides a brief outline of the original architecture, followed by the modifications for decentralised control and the implementation of information value in controlling the platforms. The implementation of the modules relevant to the information value based control scheme is then detailed.

4.2 Multi-Objective System Architecture

4.2.1 Task Based Control

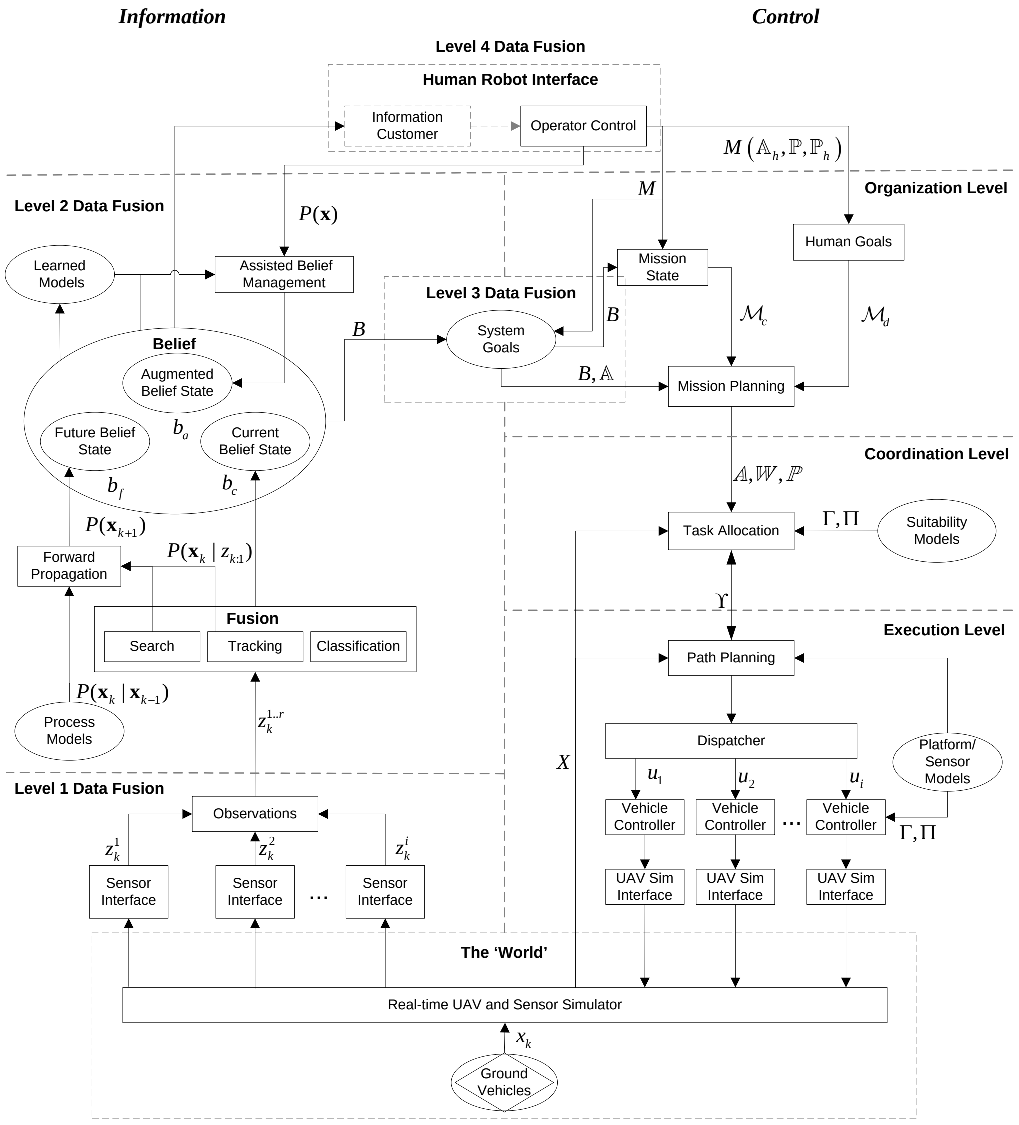


Figure 4.1: The basic simulator framework.

The simulator and architecture were conceived as a possible solution to the problem of single operator control of multiple autonomous platforms supporting an urban security mission. The original architecture is based on centralised processing of sensor information with a central control loop to determine waypoints that are then passed to each platform for autonomous execution.

The complete system framework, shown in Figure 4.1, is an amalgamation of distinct control and fusion functionalities. The control domain is based on Saradis's framework for the principle of *Increasing Precision with Decreasing Intelligence* (IPDI)[36]. Control is task based, with discrete tasks being allocated to available resources. In line with the concept of IPDI, the higher the level of control, the more abstract the task description. Passing a high level task such as *Search* down through the *Organize, Coordinate, Execute* refinement processes, results in the task being described at the lowest level in terms of a path for a platform/sensor combination. Execution of the path by the agents causes information gain which feeds into the fusion domain as sensor returns or observations.

The fusion domain is layered according to the levels described in the JDL Data fusion model [23]. Information enters the fusion domain at the lowest level as raw sensor observations and is refined to generate an estimate of the state of the world. The fusion and control domains interact at two points, corresponding to the JDL descriptions of Level 3 and 4 Data Fusion. At *Level 3* or '*situation assessment*' a mission state is derived from the estimate of the world state. This can be used to evaluate the utility of performing any task. In this framework *Level 4 Data Fusion*, or '*process refinement*', is an implicit part of the *Human Robot Interface.* The operator has access to the output of the fusion components via the *Information Customer Interface*, thus human level information assessment and abstract reasoning for control refinement may be performed by the operator. This information is implicit in the operator's mission specification and enters the system via the situation assessment level.

4.2.2 Distributed Control

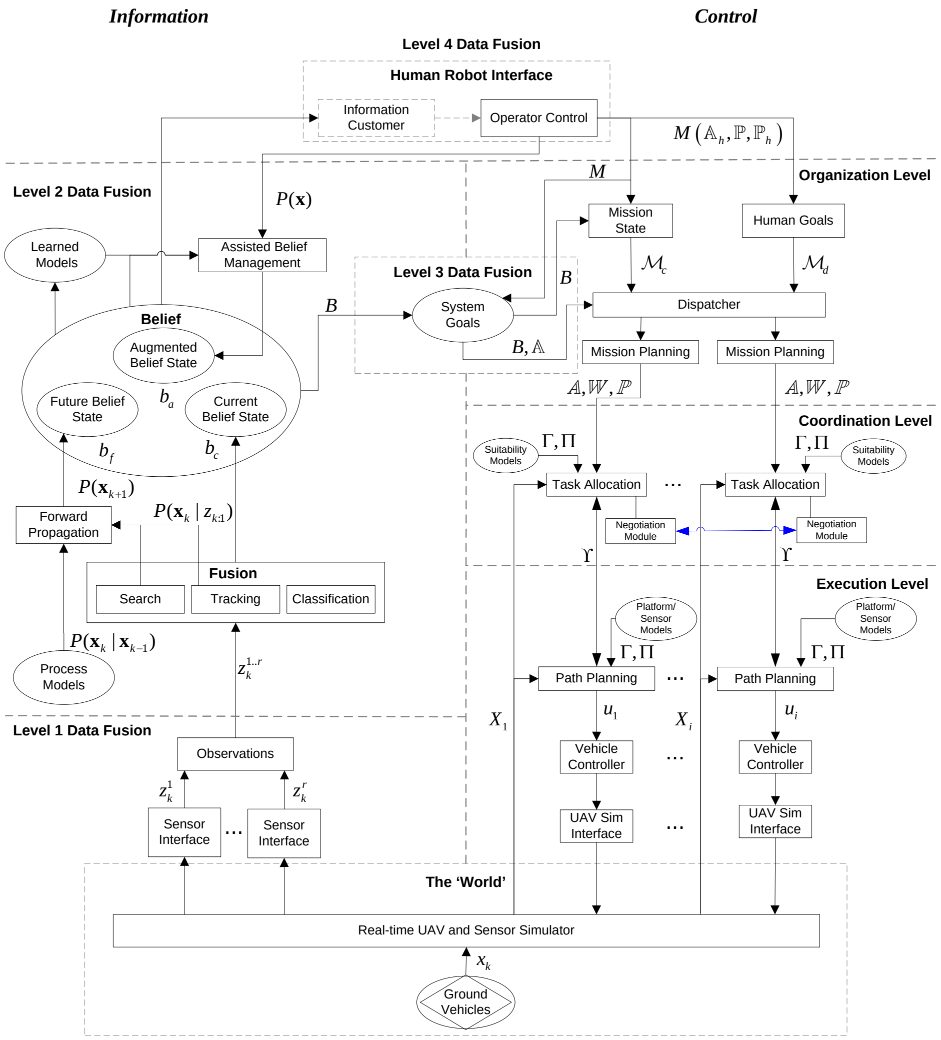


Figure 4.2: The distributed control simulator framework.

The Distributed Control architecture shown in Figure 4.2 is a modification of that in Figure 4.1, where higher level control decisions are deferred to the agents. In order to support this, the Mission Planning and Task Allocation components are on board each agent. In this case for the team to be coordinated a negotiation module is added to each agent to permit inter-agent communication. If each agent communicates it's best task assignment and the utility to other agents, then the allocation for maximum team utility can be negotiated.

The fusion domain is shared with the centralised architecture in Figure 4.1. The single point of fusion and dissemination ensures that agents have a common belief state on which to make decisions. However for each agent planning cycle the Belief, B, needs to be communicated to each agent via the Dispatcher along with the Mission.

4.2.3 Information Value Based Control

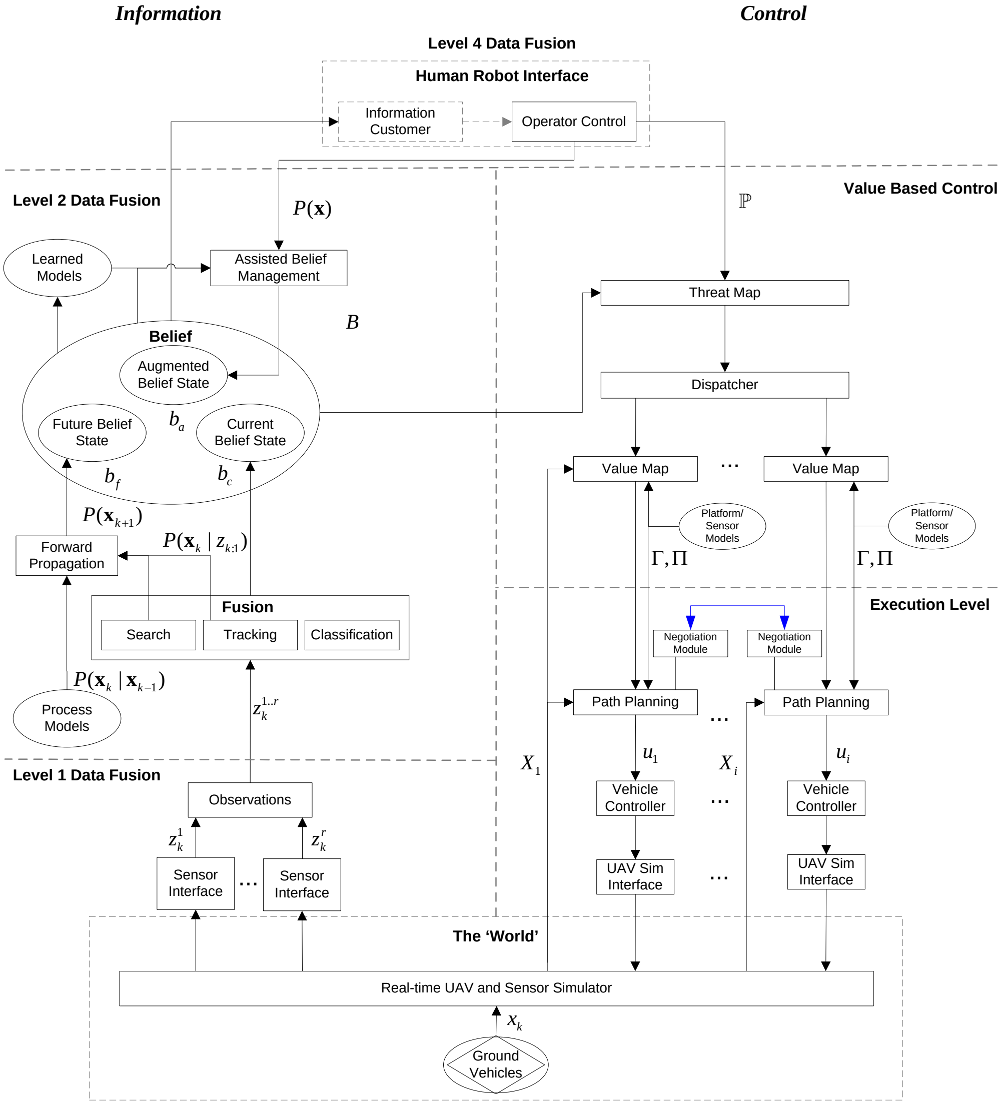


Figure 4.3: The information value based control simulator framework.

The Information Value architecture shown in Figure 4.3 is a further modification of the Distributed Control architecture. Again fusion components are shared and centralised, however Level 3 Fusion, Organisation and Coordination are collapsed into a single level. The belief state is used to generate a common 'Threat Map', which implicitly captures the system goals. The Threat Map is used by each agent to generate an information value map on which all agent decisions are based. A negotiation module is added to maximise the value of information by ensuring information is not duplicated. However negotiation is not used for all information value based control schemes as outlined in Section 5.2.4.

4.3 Components

4.3.1 Human Robot Interface

The *Human Robot Interface* (HRI) consists of two components, the Information Customer Interface and the Operator Control Interface. The architecture separates these functionalities for command and control purposes, allowing information dissemination by distributing copies of the Information Customer Interface while maintaining single point of mission control.

The link between the Operator Control Interface and Information Customer Interface provides a bridge between the information and control domains at the human level. This allows for active inference, planning and refinement at the abstract Level 4 Data Fusion level.

4.3.2 Level 0 and 1 Data Fusion

At the Level 0 Data Fusion level raw sensor signals are associated to target features and Level 1 Data Fusion associates these features to an entity and provides an entity state estimate. In the simulated system both Level 1 and 2 processes are performed within the various sensor modules, producing an observation z as output. Appendix A details this process.

4.3.3 System Belief

The complete system belief state, B, is constructed from a number of components.

$$B = f(b_c, b_f, b_a) \tag{4.1}$$

- b_c is the belief about the current state of the world.
- b_f is the belief about the future state of the world.
- b_a is an augmented belief state resulting from additional high level input from a human or through inference or learning.

Each b is the joint distribution of the variables of interest, which for the security scenario is the components of $P(\mathbf{x})$.

$$b = f(P(\mathbf{x}^t, \mathbf{x}^c, \mathbf{x}^s, \mathbf{x}^d)) \tag{4.2}$$

Where $P(\mathbf{x}^t)$ is the track distribution over the area of operation for all targets being tracked, $P(\mathbf{x}^s)$ is the search state over the area of operation, $P(\mathbf{x}^c)$ is the classification distribution for all targets being tracked and $P(\mathbf{x}^d)$ is the distribution associated with the classification of any undetected target.

According to the JDL fusion model, the complete belief state contains elements belonging to both Level 2 and 3 data fusion. By fusing sensor observations with prior information, the current belief b_c provides situation assessment, or Level 2 fusion, through the track, classification and search modules. These are detailed in appendix A. If the actions of the agent and their effect on the estimate of the world states are considered, then the future belief b_f provides impact assessment which is considered part of Level 3 fusion.

The current belief b_c is characterised at time k by $P(\mathbf{x}_k)$, which summarises all the probabilistic information available about $\mathbf{x}$ from the observation history $z_0, ...z_{k-1}$. In the scenario the observation history applies only to $\mathbf{x}^c$, $\mathbf{x}^t$ and $\mathbf{x}^s$, as $P(\mathbf{x}^d)$ is assumed constant.

If target classification is such that it describes an intrinsic property of the target object then $P(\mathbf{x}^c_k) = P(\mathbf{x}^c_{k-1})$, if z_{k-1} contains no information about $\mathbf{x}^c$. However if the targets are mobile, then this relationship does not hold for $\mathbf{x}^t$ or $\mathbf{x}^s$. Thus, due to the small size of the area associated with a single observation relative to the area of operation, $P(\mathbf{x}^t)$ and $P(\mathbf{x}^s)$ will necessarily be predicted to time k if no observation has been made at $k-1$. The predicted state may be obtained by the Chapman-Kolmogorov equation.

$$P(\mathbf{x}_k^t|z_{1:k-1}^t) = \int P(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)P(\mathbf{x}_{k-1}^t|z_{1:k-1}^t)d\mathbf{x}_{k-1}^t \tag{4.3}$$

Where $P(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)$ is a probabilistic Markov motion model. In the case of $\mathbf{x}^t$ this is the actual target motion model, which will be conditioned on target type $P(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t, \mathbf{x}_{k-1}^c)$, and for $\mathbf{x}^s$ this term will be the motion model for an unknown target type.

With the target classification, $P(\mathbf{x}^c)$, invariant with respect to time if no observation is made, the future belief b_f at time $k+n$ is characterised only by $P(\mathbf{x}_{k+n}^t)$ and $P(\mathbf{x}_{k+n}^s)$. Using the same motion model as Equation 4.3, a prediction of $P(\mathbf{x})$ can be made for some arbitrary future time. The forward predict time is driven by the planning cycle of the agents. In the task based control implementation the world states are predicted forward according to Equation 4.3 as part of the process of determining the best path for the assigned task. Incorporating the impact of agent actions is discussed below.

4.4 Level 3 Data Fusion

The product of Level 3 fusion is the estimate of the situation utility [23]. The Level 3 process involves assessing the likely impact of future agent actions and evaluating the situation with respect to the mission objectives.

For the task based control architecture, both the state of the world and the mission contribute to defining task priority. Thus Level 3 data fusion includes a mapping of the mission goals and current state of the world to prioritised system goals.

$$\mathcal{L}3 : B \times M \rightarrow \mathbb{A}, \mathbb{P} \tag{4.4}$$

- B is the current system belief state generated from all available information.
- M is the mission description for the whole system defined by the human operator.
- $\mathbb{A}$ is a set of system tasks described by M.
- $\mathbb{P}$ is the set of priorities for all tasks.

To assess the impact of the agent's actions on the future belief state, a fixed time horizon corresponding the agent's replan cycle is used. For the task based control system the tasks are prioritised according to the mapping in Equation 4.4, leaving the relative impact of the observation action in completing the task as the remaining component for determining situation utility. This is determined by assessing the mutual information of the agent's sensor and the task's state variable. This is detailed in Appendix B

For the information value based control system, the likely impact of the observation z_{k+1} directly on all states of interest is used to assess future agent actions with respect to the mission. The situation utility is determined by comparing the value of world state prior to the observation $P(\mathbf{x}_k)$ and post the observation $P(\mathbf{x}_{k+2})$, with one step look ahead prediction of target tracks. The future belief state b_f at the planning horizon $k+1$ and the Chapman-Kolmogorov Equation 4.3, provide the basis for determining $P(\mathbf{x}_{k+2}|z_{k+1})$ This is detailed in Section 5.2.

4.4.1 Execution

The Execution Level is where physical control of the platforms and sensors is carried out. The assigned task a or desired observation z is used to generate inputs for the platform controllers, via a path planner or other guidance system. The platform controllers, which in the case of a UAV will be the navigation module and autopilot, then execute the path using feedback from the simulated world regarding the vehicle's motion and position.

The framework shown in Figure 4.1, shows the Path Planner component above the Dispatcher, with only the final path or way point passed to the platform. This arrangement is suitable for sensor platforms having minimal onboard processing and no inter-platform communication. With increased processing power and communications ability the Path Planner may be moved onboard, with centralised control maintained if the Dispatcher simply passes details of a to the relevant platform.

Path Planning

A path appropriate for the assigned platform must be generated that satisfies the requirements for either accomplishing the task a or observing the area specified by z. This applies whether the path planning component is on board the platform or residing in the ground station. The separation of task allocation and path planning allows flexibility in the path planning algorithms used without impacting the higher levels of control. Further, in the case of task based control, it allows the Coordinator to operate with only limited knowledge regarding details of the platforms being controlled.

In the task based systems the assigned task a will normally be associated with an area for the time $k + 1$. Thus the Path Planning component generates a platform specific optimal path in order to provide sensor coverage for a. If a single task requires coordination between multiple platforms, determining the group optimum path may require multiple iterations of the individual path planners. If the planner resides in the platform the ability to send and receive information about the path between platforms is necessary for successful execution.

Various constraints such as fuel usage or manoeuvre limits may result in the path planner being unable to generate a valid path within the cost budget allocated to the task. The framework includes a feed back mechanism between the Task Allocation and Path Planning modules to allow the task to be returned to the Coordinator where the task set A is reallocated.

In the information value framework the observation z_{k+1} is defined by the point s in the area of operation to be observed. Hence the Path Planning component generates an optimal path for the platform to point its sensor at a point of interest. Assuming that all parts of the map are accessible and given that an estimate of the cost of taking the observation is explicitly included in the value formulation, the feed back mechanism, although included in the architecture, should not be used in practice.

4.5 Summary

This chapter has outlined a general architecture for the co-operative control of a system of heterogeneous airborne sensor platforms tracking ground targets. Use is made of the concept of data fusion levels and the principle of Increasing Precision with Decreasing Intelligence. The general framework is modular and flexible with task and information value based control both being able to be used within the same system architecture. Chapter 5 details the implementation of the system architecture with an information value based control scheme derived from the formulations in Chapter 3.

Chapter 5

Implementation

5.1 Introduction

The implementation of the value based control formulation outlined in Chapter 3 using the architecture described in Chapter 4 is presented in 5.2, with the corresponding implementation of the task based control used for comparison detailed in Appendix B.

At every planning cycle, each agent calculates its own value map for the security scenario, as shown in Section 5.2.3, from the common threat map described in Section 5.2.2. The value map is used to determine the value of an observation V^z_{net} according to the control schemes described in section 5.2.4 and their associated cost functions in Section 5.2.5.

Section 5.3 details the metrics used for comparison of the value based control and the task based control schemes. The specifics of the scenarios used for comparison are shown in Section 6.1, with results presented in Section 6.4.

5.2 Value Based Control

5.2.1 Priority

The priority p of either a ground vehicle or asset is used as the basis of preference ordering for both the task based and value based controllers. In the value based control, the priority is used to determine the value of the observation z, while in task based control it is used to determine the weighting for a discrete task as detailed in Appendix B.

The priority of a ground vehicle is given by the set $\mathbb{P}^c$, which differentiates based on the threat posed by the particular type.

$$\mathbb{P}^c = K\left[p_1^c, p_2^c \ldots p_m^c\right] \tag{5.1}$$

Where there are m distinguishable ground vehicle types and K may be a normalising value or other equivalence term such that comparison can be made with other priority sets active in the system. In the simulated system, the target type priority is applied by the operator according to one of the descriptions shown below. A numerical value corresponding to the human level description is then automatically applied.

$$p_m = |\text{No Impact},\ \text{Some Impact},\ \text{Desirable},\ \text{Highly Desirable},\ \text{Mission Critical}| \tag{5.2}$$

It is desirable that the values corresponding to each descriptive level are either assigned by a human with high level knowledge of the situation or extracted from historical data. In the scenarios investigated, the corresponding numerical values are applied as either the linear set $1 \ldots 5$ or the logarithmic set $2^1 \ldots 2^5$. Similar priority descriptions and numerical values are used for asset priority. This same basic priority description is used as a foundation for both task based and value based systems.

5.2.2 Threat Map

As discussed in Section 3.4, in the urban security scenario the information gathering agent's observation utility is a function of the variables corresponding to ground vehicle type, current position estimate and proximity to protected assets. Information on all states of interest is gained through placing the footprint of the agent's sensor over a section of the area of operation and making an observation. This allows the asset proximity variable for the current planning cycle to be fixed if the two dimensional space of the area of operation is used as an anchor for the belief states and value functions. This gives rise to the 'threat map' which describes $\mathcal{U}(u_j|\mathbf{x}^{l,\lambda})$. If the observation action u_j is similarly anchored, with j describing a subset x of the area of operation, then a 'value map' can be created which will indicate the expected value of observing any point in the area of operation. Defining x as a point area, the utility for $u_{j=x}$ is then a function of the variables $\mathbf{x}^t$, $\mathbf{x}^c$, $\mathbf{x}^s$ and $\mathbf{x}^d$.

The threat map is determined by a combination of the asset properties and the relative location of all assets. Each asset will have an associated mission priority p_1^a applied as in Equation 5.2 and a vulnerability to threat v_1. For the point x, the contribution of asset m is found by using one of the threat function shown in Equations 5.3 to 5.6. The total threat at any point x is the sum over all a where a ∈ A is the set of all assets and x_a is the location of asset a.

$$p^{l,\lambda}(x) = p_{\text{a}} e^{-0.5\left(\frac{(x_{\text{a}}-x)^2}{\nu_{\text{a}}}\right)} \tag{5.3}$$

$$p^{l,\lambda}(x) = p_{\text{a}} \left(1 - \frac{1}{\eta\nu_{\text{a}}} \left|x_{\text{a}} - x\right|\right) \tag{5.4}$$

$$p^{l,\lambda}(x) = \begin{cases} p_{\text{a}} & \text{for } |x_{\text{a}} - x| < \eta\nu_{\text{a}} \\ 0 & \text{for } |x_{\text{a}} - x| > \eta\nu_{\text{a}} \end{cases} \tag{5.5}$$

$$p^{l,\lambda}(x) = \begin{cases} p_{\text{a}} & \text{for } x \in \mathbb{S} \\ 0 & \text{for } x \notin \mathbb{S} \end{cases} \tag{5.6}$$

Equations 5.3 and 5.4 allow the operator to assign potential threat based on the distance of a target from the asset. In Equation 5.3 a log decreasing threat function is applied centered on the asset and in 5.4 the threat decreases linearly with distance from the target. Equation 5.5 sets a constant threat within the specified distance from the asset. The η value is the vulnerability limit distance, beyond which a target poses no threat to the asset.

The numerical values for p_a are as for the type priority and the vulnerability of asset, v_a, is applied by the operator according to the human specified levels in Equation 5.7.

In Equation 5.6, the set $\mathbb{S}$ is a group of points or nodes specified by the operator and allows manual setting of the priority of the area around an asset. This provides flexibility for assigning a threat area of any shape, which can be used by the operator to reflect vulnerability for different avenues of approach to the asset. This assignment of threat to a set of points is also used in the Mission setup for task based control and is described in Appendix B.

$$v = |\text{Soft}, \text{ Reinforced}, \text{ Hardened}, \text{ Fortified}, \text{ Protected}| \tag{5.7}$$

The values corresponding to each vulnerability level relate to a 'safe' distance. This is the minimum distance at which a security response will be able to neutralise any threat posed by a ground vehicle without substantial damage to the asset. In this case *Soft* corresponds to 150m and *Protected* to 10m.

5.2.3 State Utility

As the information gathering component of the security mission, the primary task of the agents is to provide information regarding potential threats to ensure they are effectively countered. Hence the observation utility for an agent will be a function of both the state of the ground vehicles observed and the current information state. This is supported by the intuition that for a vehicle of known type, an observation will be more valuable if the current position estimate is poor than if the vehicles position is known precisely.

If maximum entropy provides maximum gain, then combining the information state with the target state provides the observation utility for each of the variables of interest. The observation utility conditioned on the true type and location is shown in Equations 5.8 and 5.9 for the point x.

$$v_x^l = p_x^l \frac{H(\mathbf{x}^t)}{H_{\max}(\mathbf{x}^t)} + p^c \frac{H(\mathbf{x}^t)}{H_{\max}(\mathbf{x}^t)} \tag{5.8}$$

$$v^c = p^c \frac{H(\mathbf{x}^c)}{H_{\max}(\mathbf{x}^c)} \tag{5.9}$$

Where $H(\mathbf{x}^t)$ is the current track entropy associated with target q, which has a track distribution that includes the point x and $H_{\max}(\mathbf{x}^t)$ is the maximum dispersion of $P(\mathbf{x}^t)$ before the target is considered not to be in track. This is detailed in Appendix A. Similarly $H(\mathbf{x}^c)$ is the entropy associated with the current vehicle type estimate, with $H_{\max}(\mathbf{x}^c)$ being the entropy of a flat distribution of all possible target types.

As nothing is know regarding any untracked vehicles in the observation, the current entropy of $\mathbf{x}^t$ and $\mathbf{x}^c$ can be considered as $H_{\max}$. Hence the observation utilities for the untracked target at point x is simply equal to the priority as shown in Equation 5.10 and 5.11.

$$\begin{aligned} v_x^{\lambda} &= p^{\lambda} && (5.10)\\ v^d &= p^d && (5.11)\end{aligned}$$

The maximum utility prior to observation z of point x is then found in Equation 5.12 by substitution.

$$\begin{aligned} \mathcal{U}(u_{j=x}|\mathbf{x}) &= \mathcal{U}(u_{j=x}|\mathbf{x}^t, \mathbf{x}^c, \mathbf{x}^l) + \mathcal{U}(u_{j=x}|\mathbf{x}^s, \mathbf{x}^d, \mathbf{x}^{l=\lambda}) \\ &= \sum_q v_{x,q}^l + \sum_q v_{x,q}^c + v_x^l + v_x^d && (5.12)\end{aligned}$$

The observation utility for point x conditioned on the best knowledge that sensor r can obtain of a target is then found by substituting Equation 5.13 for 5.8 and Equation 5.14 for 5.9.

$$\begin{aligned} {v'}_x^l &= p_x^l P(z_x^t|\mathbf{x}_x^t)\frac{H(\mathbf{x}_x^t)}{H_{\max}(\mathbf{x}_x^t)} + p^c P(z^c|\mathbf{x}^c)\frac{H(\mathbf{x}_x^t)}{H_{\max}(\mathbf{x}_x^t)} && (5.13)\\ {v'}^c &= p_q^c P(z^c|\mathbf{x}^c)\frac{H(\mathbf{x}^c)}{H_{\max}(\mathbf{x}^c)} && (5.14)\end{aligned}$$

Similarly for the unknown vehicle, Equations 5.10 and 5.11 are modified based on the estimated state of the world.

$$\begin{aligned} v_x^{'\lambda} &= p^{\lambda} P(\mathbf{x}_x^t) \qquad (5.15) \\ v_x^{'d} &= p^{d} P(\mathbf{x}^c) \qquad (5.16) \end{aligned}$$

The average utility prior to observation z of point x is then found in Equation 5.17 by substitution.

$$\mathcal{U}'(u_{j=x}|\mathbf{x}) = \sum_q v_{x,q}^{'l} + \sum_q v_{x,q}^{'c} + v_x^{'l} + v_x^{'d} \tag{5.17}$$

The value of $V^*_{\backslash z}$ can then be found for x from Equation 3.4.

$$\begin{aligned} V^*_{\backslash k} &= \mathcal{U}_{avg} - \mathcal{U}'_{avg} \\ &= \sum_i P(\mathbf{x}_i)\mathcal{U}(u_{j=x}|\mathbf{x}) - \sum_i P(\mathbf{x}_i)\mathcal{U}'(u_{j=x}|\mathbf{x}) \qquad (5.18) \end{aligned}$$

Where $\mathcal{U}_{avg}$ and $\mathcal{U}'_{avg}$ can be found from the summation of the observation utility over all possible vehicle types for all tracked and unknown targets.

$$\begin{aligned} \sum_i P(\mathbf{x}_i)\mathcal{U}(u_{j=x}|\mathbf{x}) &= \sum_m P(\mathbf{x}_{x_1}^t)P(\mathbf{x}_{m_1}^c)\mathcal{U}(u_{j=x}|\mathbf{x}_{x_1}^t, \mathbf{x}_{m_1}^c, \mathbf{x}_x^l) + \cdots \\ &\quad + \sum_m P(\mathbf{x}_{x_q}^t)P(\mathbf{x}_{m_q}^c)\mathcal{U}(u_{j=x}|\mathbf{x}_{x_q}^t, \mathbf{x}_{m_q}^c, \mathbf{x}_x^l) \\ &\quad + \sum_n P(\mathbf{x}_x^s)P(\mathbf{x}_n^d)\mathcal{U}(u_{j=x}|\mathbf{x}_x^s, \mathbf{x}_n^d, \mathbf{x}_x^{\lambda}) \qquad (5.19) \end{aligned}$$

The probability of the existence of the tracked target $P(\mathbf{x}^t)$ at x is obtained from the track estimate, the type estimate $P(\mathbf{x}^c)$ for the tracked target is provided by the classification distribution and the probability $P(\mathbf{x}^s)$ of a previously undetected target at x is obtained from the search state. See Appendix A for the implementation of the search, track and classify belief states. Note that with no *a priori* knowledge, $P(\mathbf{x}_n^d)$ will be constant.

If sensor r makes observation z_r that includes information about a target then the resulting observation utilities for the point x can be found by substituting the post observation entropy into Equations 5.8, 5.9, 5.13 and 5.14.

$$\begin{aligned}
v_x^l &= p_x^l \frac{H(\mathbf{x}_x^t|z_r)}{H_{\max}(\mathbf{x}_x^t)} + p^c \frac{H(\mathbf{x}_x^t|z_r)}{H_{\max}(\mathbf{x}_x^t)} && (5.20)\\
v^c &= p^c \frac{H(\mathbf{x}^c|z_r)}{H_{\max}(\mathbf{x}^c)} && (5.21)\\
v_x^{\prime l} &= p_x^l P(z_x^t|\mathbf{x}_x^t, z_r) \frac{H(\mathbf{x}_x^t|z_r)}{H_{\max}(\mathbf{x}_x^t)} + p^c P(z^c|\mathbf{x}^c) \frac{H(\mathbf{x}_x^t|z_r)}{H_{\max}(\mathbf{x}_x^t)} && (5.22)\\
v^{\prime c} &= v_q^c P(z^c|\mathbf{x}^c, z_r) \frac{H(\mathbf{x}^c|z_r)}{H_{\max}(\mathbf{x}^c)} && (5.23)
\end{aligned}$$

The post observation entropy can be found prior to the observation being made via mutual information.

$$\begin{aligned}
H(\mathbf{x}_x^t|z_r) = H(\mathbf{x}_x^t) - I(\mathbf{x}_x^t, z_r^t) && (5.24)\\
H(\mathbf{x}_x^c|z_r) = H(\mathbf{x}_x^c) - I(\mathbf{x}_x^c, z_r^c) && (5.25)
\end{aligned}$$

Where I is found using Equations 3.12, 3.13 and 3.14. The joint probability of target states and the observation is found using the sensor model of sensor r.

$$\begin{aligned}
P(\mathbf{x}^t, z_r^t) P(z_r^t|\mathbf{x}^t) \otimes P(\mathbf{x}^t) && (5.26)\\
P(\mathbf{x}^c, z_r^c) P(z_r^c|\mathbf{x}^c) \otimes P(\mathbf{x}^c) && (5.27)
\end{aligned}$$

For the unknown target v_x^λ and v^d remain as for Equations 5.10 and 5.11, while the v' values for the unknown target are shown in Equations 5.28 and 5.29.

$$\begin{aligned}
v_x^{\prime\lambda} &= p^\lambda P(\mathbf{x}_x^t, z_r) && (5.28)\\
v_x^{\prime d} &= p^d P(\mathbf{x}^c, z_r) && (5.29)
\end{aligned}$$

These values of v found from Equations 5.20 to 5.29 can be substituted into Equations 5.12

and 5.17 to give $\mathcal{U}^z(u_{j=x}|\mathbf{x}, z_g)$ and $\mathcal{U}'^z(u_{j=x}|\mathbf{x}, z_g)$ respectively.

To find V_z^*, the joint distributions in Equations 5.26 and 5.27 are used for the estimate of the state after observation z.

$$\begin{aligned} V_z^* &= \mathcal{U}_{avg}^z - \mathcal{U}_{avg}'^z \\ &= \sum_i \sum_m P(\mathbf{x}_i, z_g)\mathcal{U}^Z(u_{j=x}|\mathbf{x}, z_m) - \sum_i \sum_g P(\mathbf{x}_i, z_g)\mathcal{U}'^z(u_{j=x}|\mathbf{x}, z_g) \qquad (5.30) \end{aligned}$$

Hence for q ground vehicles having a probability of being point x and g possible type observations:

$$\begin{aligned} \sum_i \sum_g P(\mathbf{x}_i, z_g)\mathcal{U}^z(u_{j=x}|\mathbf{x}, z_g) &= \sum_g \sum_m P(\mathbf{x}_{x_1}^t, z_x^t)P(\mathbf{x}_{m_1}^c, z_g^c)\mathcal{U}^z(u_{j=x}|\mathbf{x}_{x_1}^t, \mathbf{x}_{m_1}^c, \mathbf{x}_x^l, z_g) + \cdots \\ &+ \sum_g \sum_m P(\mathbf{x}_{x_q}^t, z_x^t)P(\mathbf{x}_{m_q}^c, z_g^c)\mathcal{U}^z(u_{j=x}|\mathbf{x}_{x_q}^t, \mathbf{x}_{m_q}^c, \mathbf{x}_x^l, z_g) \\ &+ \sum_n P(\mathbf{x}_x^s, z_x^s)P(\mathbf{x}_n^d)\mathcal{U}^z(u_{j=x}|\mathbf{x}_x^s, \mathbf{x}_n^d, \mathbf{x}_x^\lambda) \qquad (5.31) \end{aligned}$$

The complete action set for the agent is the set of observations for all points in the area of operation, or $x \in \mathbb{S}$. By determining V^* for each point, a value map for each sensor can be created. The complete observation value V_{net} can be found from this map by subtracting the cost of placing the sensor to observe the area defined by x. The cost will be dependent on control issues such how the value map is used and the planning horizon, which is discussed in Sections 5.2.5 and 5.2.4.

5.2.4 Planning and Control

Three different control schemes were used in the implementation of value based control, *Infinite Horizon Peak-chasing* (IHP), *Finite Horizon Peak-chasing* (FHP) and *Finite Horizon Footprint* (FHF). These are compared to the Task based control scheme in Appendix B as a performance benchmark.

In the IHP control scheme, at each planning cycle the agent chooses a waypoint which allows observation of the point x where the V_{net} map shows maximum value. This point is passed to all other agents, and if two or more agents have selected an observation point x within $20m$ of each other, new observation points are chosen in a pareto optimal fashion.

In the FHP control scheme, as in IHP, the agent chooses a waypoint which allows observation of the point x where the V_{net} map shows maximum value. However the point chosen is limited by a 30^o arc either side of the agents direction of travel, with a radius equal to the maximum distance the agent can travel before the next planning cycle. This is shown in Figure 5.2. In FHP, nothing is communicated between the agents. The area considered is shown in Figure 5.2, with the point of maximum value within this area chosen as the waypoint for the next planning cycle.

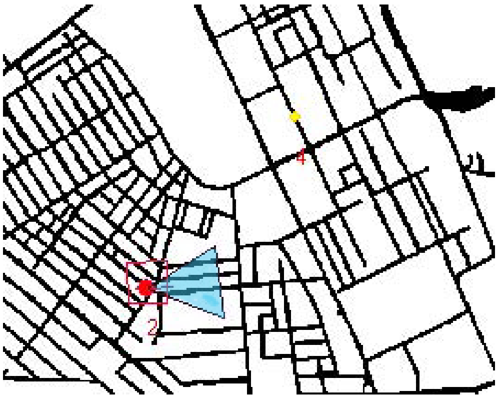


Figure 5.1: The Finite Horizon Peak chasing control scheme for a 5 second look ahead planning horizon. A waypoint for the current planning cycle is chosen from within the 30^o arc highlighted in blue.

The FHF control scheme uses the same path planning mechanism as the task based control system discussed in Appendix B. A set of potential sensor footprints is found at the planning horizon, and the set of points s contained in each footprint found. The value of the observation is then $\sum_{x \in s} V_{x\text{net}}$. The waypoint corresponding to maximum value is then chosen for the current planning cycle. The predicted footprints for a 5 second look ahead planning cycle are shown in Figure 5.2.

The predicted footprints are based on predicting a set of fixed bank angles from the current pose. As a result of the UAV needing to reach the desired bank angle, the actual footprint at the horizon will differ somewhat from that predicted at the beginning of the planning cycle. Figure 5.3 shows the difference between the footprints predicted at the beginning of the planning cycle and the actual footprint at the end of the planning cycle.

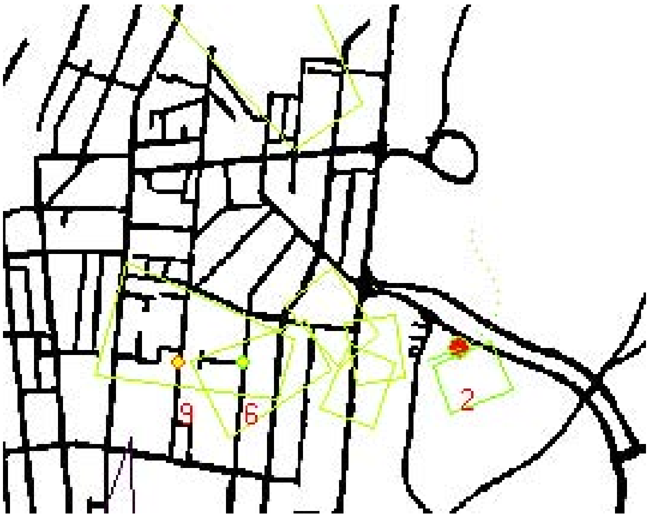


Figure 5.2: The Finite Horizon Footprint chasing control scheme for a 5 second look ahead planning horizon. Predicted footprints are shown in light green with the dark green indicating the current footprint. The UAV bank angle control associated with the footprint providing the best $\sum_{x \in s} V_{x\text{net}}$ is chosen for the current planning cycle.

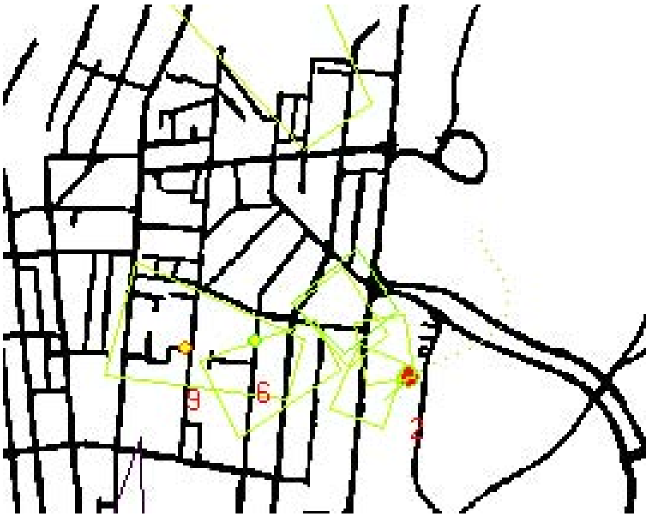


Figure 5.3: Actual footprint at the end of the planning cycle shown in dark green, with predicted footprints shown in light green as per Figure 5.2. Note the actual footprint is similar but not identical to the footprint selected for control.

5.2.5 The Observation Cost

As discussed in Section 3.4 the cost coefficient is independent of the information contained in the observation and dependent on the parameters of z. In the case of IHP, the cost is a function of the distance from the agents current position to the chosen point on the map, x and the agents current speed.

$$C_x^z = \left|\bar{X}_x - \bar{x}\right| X_v \tag{5.32}$$

Where X_v is the agent's velocity along the heading axis and X_x the current location.

At each time step the agent makes an observation. As FHP determines the value of observing points at the planning horizon, ie within a constant number of time steps, it is assumed that each observation action consumes the same resources. Hence no cost is applied and the V^* map used directly.

For FHF control the cost is applied as a function of the agents state at the planning horizon k. This penalises extreme manoeuvres and large control inputs.

$$C_s^z = X_\phi^c - X_\phi^k \tag{5.33}$$

Where X_ϕ is the agent's bank angle.

5.3 Evaluation Metrics

A number of metrics are used for evaluating performance and for comparison between the task based and information value based control schemes.

5.3.1 Threat Measures

The first of these is the threat measure T. The threat is derived from the utility functions outlined in Section 5.2.3. The threat posed by tracked target T is shown in Equation 5.34.

$$T_q = \sum_{i=1}^{\mathbb{H}} \sum_{m=1}^{\mathrm{C}} P(\mathbf{x}_i^t, \mathbf{x}_m^c) v_{i,m}^l + \sum_{m=1}^{\mathrm{C}} P(\mathbf{x}_m^c) v^c \tag{5.34}$$

The overall threat measure can then be found using Equation 5.35.

$$T = \sum_{s=1}^{\mathbb{S}} P(\mathbf{x}_s^s) v_s^{\lambda} + \sum_{m=1}^{\mathrm{C}} P(\mathbf{x}_m^d) v_m^d + \sum_{q=1}^{Q} T_q \tag{5.35}$$

As described in Section 5.2.3, for the information gathering system the threat is a function of both the true state of the world and in the information available. This is reflected in the threat measures in Equations 5.34 and 5.35. The threat conditioned solely on the true world state is the true threat T_t shown in Equation 5.36.

$$T_t = \sum_{q=1}^{Q} \mathbf{x}_x^t p_x^l + \mathbf{x}_m^c p_m^c \tag{5.36}$$

- The set Q is all ground vehicles currently in the area of operation both tracked and untracked.
- x is the true position of ground vehicle q at the current time.
- m is the true ground vehicle type.

5.3.2 Entropy

The entropy of the distributions maintained as part of the belief state B provides a set of metrics on which comparison of the different control schemes can be made. However the entropy alone fails to capture the relative priority of the distributions. As a result both the entropy and a measure termed the *weighted entropy* H_w are used for comparison. The weighted entropy is shown in Equation 5.37.

$$H_w(\mathbf{x}) = \sum_{i} p_i P(\mathbf{x}_i) \log\left(P(\mathbf{x}_i)\right) \tag{5.37}$$

- p_i is the priority associated with the variable $\mathbf{x}$ being in state i.
- For the search distribution $\mathbf{x}^s$, the summation is $i = 1 \ldots \mathbb{S}$ and $p_i = p^l$.
- For each target track distribution $\mathbf{x}_q^t$, the summation is $i = 1 \ldots \mathbb{H}$, where $\mathbb{H}$ is the number of particles making up the track, and $p_i = p^l$.
- For each target classification $\mathbf{x}_q^c$ the summation is $i = 1 \ldots \mathrm{M}$.

5.3.3 Hellinger-Bhattacharya Distance

The task based control system uses the *Hellinger-Bhattacharya* distance between a current and desired distribution to determine the utility for each task as discussed in Section B.5. As a result the HB distances are used as a set of metrics for comparison of the different control schemes. The HB distance δ between two distributions of $\mathbf{x}$ is shown in Equation 5.38.

$$\delta\left(P(\mathbf{x}^a), P(\mathbf{x}^b)\right) = \left[\sum_i \left(P(\mathbf{x}_i^a)^{0.5} - P(\mathbf{x}_i^b)^{0.5}\right)^2\right]^{0.5} \tag{5.38}$$

Where the current distribution $P(\mathbf{x}^a)$ is compared to the flat distribution $P(\mathbf{x}^b)$.

As for the entropy metrics, the HB distance does not capture the relative priority of the distributions. As a result a measure termed the *weighted HB distance* δ_w is used for comparison. The weighted entropy is shown in Equation 5.39.

$$\delta_w = \left[\sum_i \left(p_i \left[P(\mathbf{x}_i^a)^{0.5} - P(\mathbf{x}_i^b)^{0.5}\right]\right)^2\right]^{0.5} \tag{5.39}$$

- p_i is the priority associated with the variable $\mathbf{x}$ being in state i.
- For the search distribution $\mathbf{x}^s$, the summation is $i = 1 \ldots \mathbb{S}$ and $p_i = p^l$. For the uniform distribution $P(\mathbf{x}^s) = |\mathbb{S}^{-1}$ for a single undetected target as discussed in Appendix B.
- For each target track distribution $\mathbf{x}_q^t$, the summation is $i = 1 \ldots \mathbb{H}$, where $\mathbb{H}$ is the number of particles making up the track, and $p_i = p^l$. For the uniform distribution $P(\mathbf{x}^t) = \mathbb{H}^{-1}$.
- For each target classification $\mathbf{x}_q^c$ the summation is $i = 1 \ldots \mathrm{M}$. For the uniform distribution $P(\mathbf{x}^c) = Q^{-1}$.

5.4 The Simulated System

5.4.1 System Overview

The framework shown in Figure 4.3 was programmed in C++ and implemented for a single processor as a single threaded application. The application makes use of modular components to allow for flexibility in setup and the testing of interchangeable elements. The code modules or module blocks closely correspond to the framework components from Chapter 4 as illustrated in Figure 5.4.

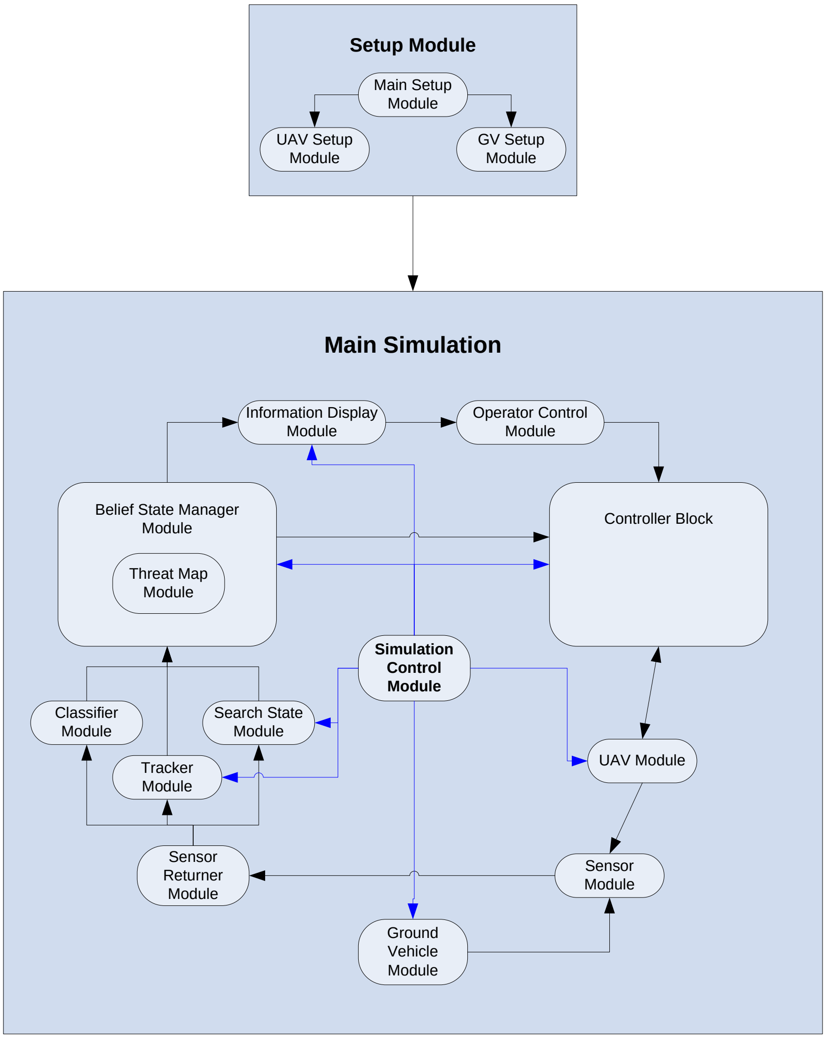


Figure 5.4: Software modules used for simulation.

Both the task and value based control schemes make use of the same structure with many common modules. These common modules and components are described in detail in Appendix A. The modules which make up the control block in Figure 5.4 unique to task based control are described in Appendix B.

5.4.2 Control Modules

For value based control scheme, the control block contains the modules as shown in Figure 5.5.

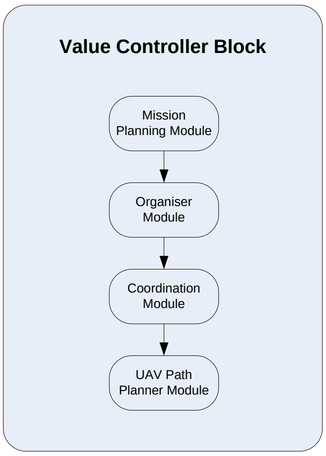


Figure 5.5: Software modules used for information value based control.

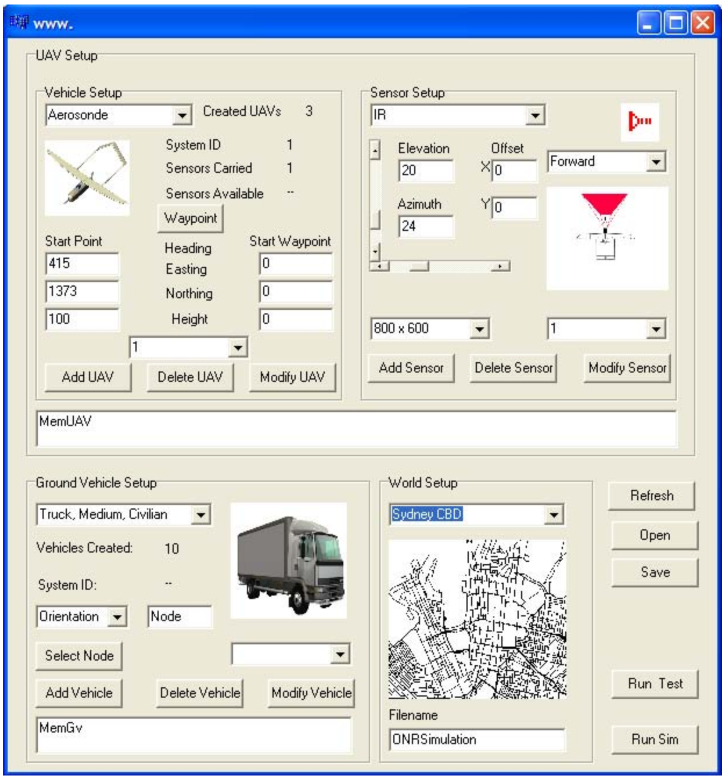
The Value Map for each UAV is derived from the Threat Map and operator mission input as per Section 5.2.3. This map is then used to determine the best path or waypoint using the control schemes in Section 5.2.4 which is then passed to the UAV Controller.

5.4.3 Simulation Setup and Interface

The Simulation Setup module provides the interface for creating the simulation scenario. This can be done manually each time the simulation runs or loaded from a file. Once the scenario is created it is passed to the Main Simulation module which is started by the operator.

Simulation Setup

The Main Setup module displays the screen shown in Figure 5.6. Using this interface the number, type and initial locations of UAV sensor platforms and ground vehicles is be set. Additionally the number, type, orientation and FOV for each sensor carried by the UAV platforms is established.



Aerosonde Created UAVs 3
System ID 1
Sensors Carried 1
Sensors Available --
Start Point 415 1373 100
Heading
Start Waypoint
Easting 0
Northing 0
Height 0
1

IR
Elevation 20
Azimuth 24
Offset X 0 Y 0
Forward
800 x 600
1

MemUAV

Truck, Medium, Civilian
Vehicles Created: 10
System ID: --
Orientation
Node
MemGv

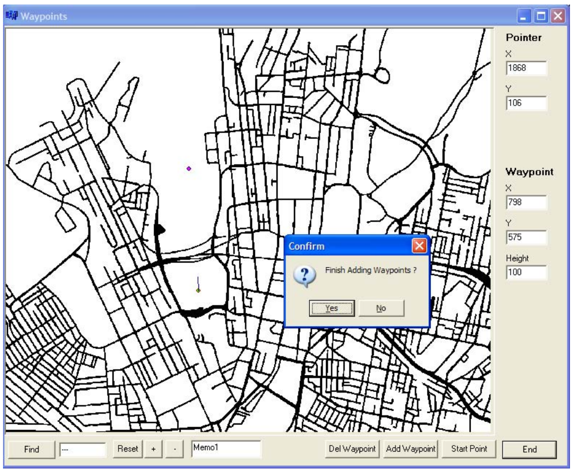
Sydney CBD
Filename ONRSimulation

Figure 5.6: Main simulation setup screen.

The setup module also contains two sub modules for setting additional initial conditions for the UAVs and ground vehicles to be used in simulation. These are discussed below.

UAV Parameters

The UAV Setup module displays the screen shown in Figure 5.7. This screen provides a graphical interface for setting initial UAV locations and heading. It also permits an initial path, consisting of a number of waypoints, to be provided to the UAV controller on simulation start. If a path is provided, the UAV will complete the path before becoming available to the system for consideration in the planning loop.



Pointer
X 1868
Y 106

Waypoint
X 798
Y 575
Height 100

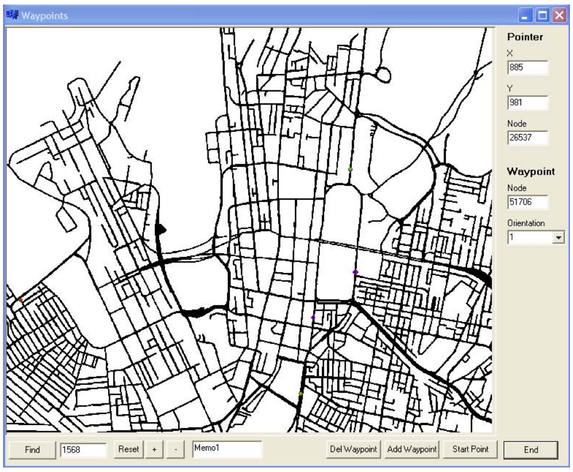
Finish Adding Waypoints ?

--- Memo1

Figure 5.7: Screen for setting up initial UAV parameters.

Ground Vehicle Parameters

The Ground Vehicle Setup module displays the screen shown in Figure 5.8. This screen provides a graphical interface for setting initial ground vehicle locations and headings. It also permits the assignment of any number of waypoints which are provided to the ground vehicle guidance component, as described in Appendix A.



Pointer

X: 885

Y: 981

Node: 26537

Waypoint

Node: 51706

Orientation: 1

Find: 1568

Memo1

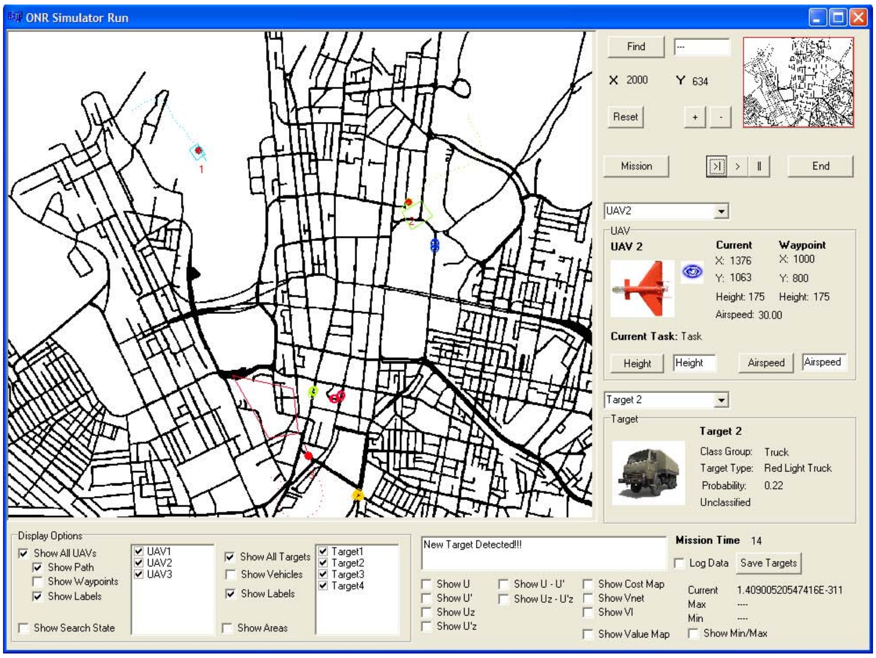
Figure 5.8: Screen for setting up initial ground vehicle parameters.

5.4.4 Operator Interface

Main Screen

Once the simulation scenario setup is complete the operator is presented with the screen shown in Figure 5.9. This is the primary information interface between the operator and the system. All information available to the simulated system, including the belief state and UAV/Sensor status is available through this screen. The control interface is detailed in Section 5.4.4

The map display allows a graphical representation of the location of UAVs, the current footprint as well as a summary of track information for ground vehicles detected and being tracked. The search state, threat map and value map for each UAV can also be displayed graphically. For testing purposes the true location of all ground vehicles may be displayed, however this is normally switched off during a simulation run.



X 2000 Y 634

UAV2

UAV

UAV 2

	Current	Waypoint
X:	1376	1000
Y:	1063	800
Height:	175	175
Airspeed:	30.00	

Current Task: Task

Target 2

Target

Target 2

Class Group: Truck
Target Type: Red Light Truck
Probability: 0.22
Unclassified

Display Options
☑ Show All UAVs
☑ Show Path
☐ Show Waypoints
☑ Show Labels
☐ Show Search State

☑ UAV1
☑ UAV2
☑ UAV3

☑ Show All Targets
☐ Show Vehicles
☑ Show Labels
☐ Show Areas

☑ Target1
☑ Target2
☑ Target3
☑ Target4

New Target Detected!!!

☐ Show U
☐ Show U'
☐ Show Uz
☐ Show U'z
☐ Show U - U'
☐ Show Uz - U'z
☐ Show Cost Map
☐ Show Vnet
☐ Show VI
☐ Show Value Map

Mission Time 14

☐ Log Data

Current 1.40900520547416E-311
Max ----
Min ----
☐ Show Min/Max

Figure 5.9: Main screen for displaying information to operator during simulation.

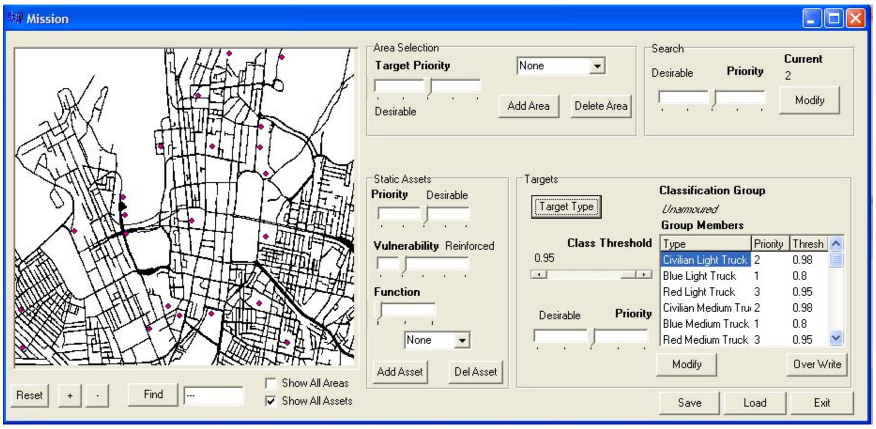
Classification information and detailed UAV/Sensor information is displayed in the dedicated panels.

Mission Parameters

The operator exercises control of the UAV sensor platforms via the Controller or Mission module which interfaces via the screen shown in Figure 5.10.

Using the graphical control, the operator can instantiate assets and search areas in the desired location. A priority and vulnerability rating is then assigned along with the threat function from Section 5.2.2. Existing assets or areas can be modified or removed at any point during the simulation.

The desired track variance, priority and classification threshold for each vehicle type can be assigned and dynamically modified using the ground vehicle pane. The classification threshold indicates the point at which there is no further value in obtaining vehicle type information. For the Task based control scheme, once the threshold is reached the ‘classify vehicle’ task is removed from task list.



Target Priority: None

Search — Current: 2

Static Assets — Function: None

Class Threshold: 0.95

Classification Group: *Unamoured*

Group Members

Type	Priority	Thresh
Civilian Light Truck	2	0.98
Blue Light Truck	1	0.8
Red Light Truck	3	0.95
Civilian Medium Tru	2	0.98
Blue Medium Truck	1	0.8
Red Medium Truck	3	0.95

Figure 5.10: Mission screen for operator input to control sensor system.

Once vehicles are being tracked, the relative priority of searching for undetected vehicles is controlled via the Search pane. The assigned search likelihood indicates the threshold likelihood for searching an area or point for a previously undetected ground vehicle.

5.5 Summary

This chapter has described the software implementation of the the architecture developed in Chapter 4 using information value as described in Chapter 3. Graphical interfaces are available to the operator for both visualisation of the information or belief state and for providing information and control inputs. These are used for the simulation results presented in Chapter 6.

Chapter 6

Simulation

6.1 Simulation Overview

The main execution loop used by the simulation represents 0.5 seconds of elapsed time. As the simulation is single thread and run on a single CPU, the relationship between simulation time and real time is heavily dependant on the number of vehicles being tracked and the complexity of the sensor platform control scheme being used. In general, using standard desktop PC hardware, more than 3 or 4 vehicles in track results in each simulation loop taking more than 0.5 seconds.

During a simulation loop every ground vehicle, UAV and sensor position is recalculated, sensor observations made and the belief state updated. As the UAV module is setup to run at 20Hz, the Guidance/Autopilot/Motion Model loop is run 10 times each loop prior to the sensor observations being triggered.

The planning horizon used is 5 simulation seconds. For the FHF, FHP and IHP control schemes, the UAV waypoints/paths are updated each replan cycle. For the Task based control scheme, tasks are reorganised and reallocated each replan cycle. All data, including the true state of the simulated world and the evaluation metrics, is logged every 5 simulation seconds.

Each scenario and situation outlined in Section 6.2 was run 5 times for a simulated mission time of 800 seconds.

6.2 Scenario Setup

Two example scenarios are used, both based around information gathering in aid of asset protection. The fixed assets are shown in Table 6.1, along with a vulnerability rating. For the purposes of these scenarios it is assumed that vulnerability is constant throughout the simulation. The asset locations are shown in Figure 6.1.

Table 6.1: Assets

Asset	**Classification**	**Vulnerability**
1	Government House	Hardened
2	State Library	Reinforced
3	Hospital	Reinforced
4	Army Barracks	Protected
5	Museum	Soft
6	Town Hall	Reinforced
7	Ferry Terminal	Reinforced
8	Central Post Office	Reinforced
9	Maritime Museum	Reinforced
10	Convention Centre	Reinforced
11	Exhibition Centre	Reinforced
12	World Trade Centre	Hardened
13	Ferry Terminal	Reinforced
14	Stadium	Reinforced
15	Central Station	Reinforced
16	Broadcast Centre	Reinforced
17	Ambulance HQ	Reinforced
18	Police HQ	Hardened
19	Hospital	Reinforced
20	Cargo Terminal	Reinforced
21	School	Soft
22	School	Soft

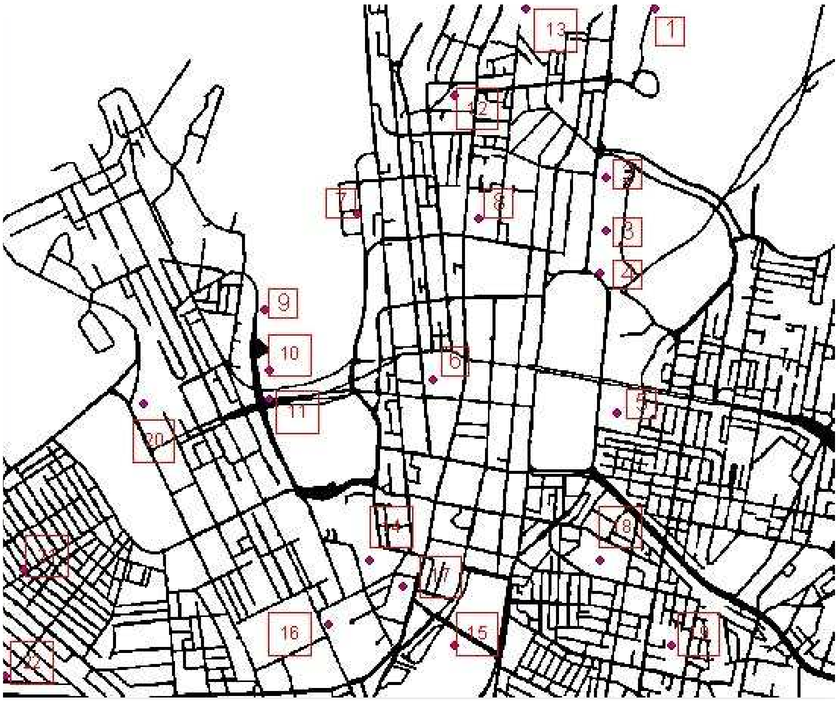


Figure 6.1: Asset location in the Sydney CBD for the security scenario under investigation

For Task based control, a representative area around each asset is identified for the 'Search Area' task discussed in Appendix B. These areas are shown in Figure 6.2, with the size of the area corresponding to the vulnerability of the associated asset. Note that the more vulnerable the asset the larger the search area.

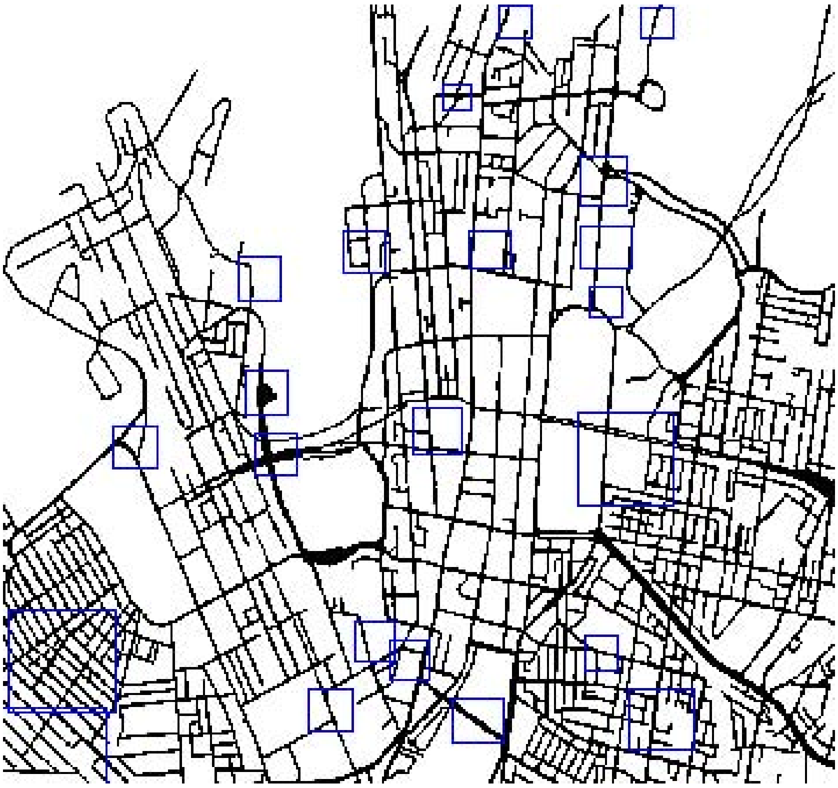


Figure 6.2: Areas corresponding to asset location in the Sydney CBD for the security scenario under investigation

6.2.1 Scenario 1

The first scenario consists of 2 UAV sensor platforms and 5 ground vehicles as outlined in Tables 6.2 and 6.3. Details on the vehicle types can be found in Appendix C.

Table 6.2: Ground Vehicles - Scenario 1

Vehicle	Type
1	4wd, Red Force
2	Truck, Light, Blue Force
3	4wd, Civilian
4	Truck, Light, Civilian
5	Car, Civilian

The initial UAV and ground vehicle locations and heading are shown in Figure 6.3. Note that ground vehicles are indicated by a yellow marker with a coloured border and UAVs by a red marker with coloured border. The coloured line extending from the UAV marker

Table 6.3: UAVs - Scenario 1

UAV	Type	Altitude (m)	Sensor	Orientation
1	Fast Delta	100	CCD	Forward
2	Aerosonde	150	IR	Down

indicates the current heading with the current sensor footprint bounded by the same colour as the marker border.

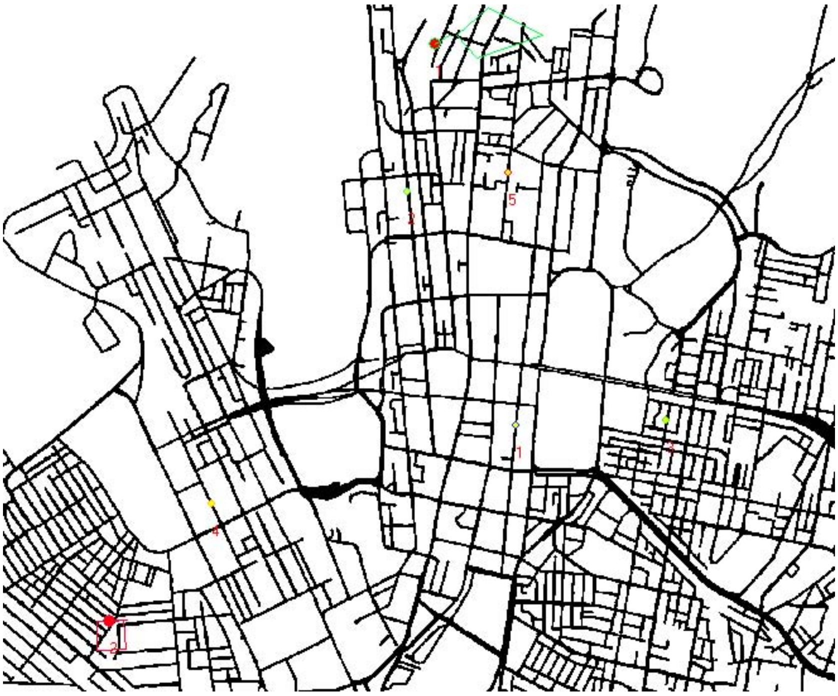


Figure 6.3: Ground Vehicle and UAV initial locations for Scenario 1.

6.2.2 Scenario 2

The second scenario consists of 3 UAV sensor platforms and 10 ground vehicles as outlined in Tables 6.4 and 6.5.

The initial UAV and ground vehicle locations and heading are shown in Figure 6.4. Note that the same symbology applies as for Scenario 1.

Table 6.4: Ground Vehicles - Scenario 2

Vehicle	Type
1	4wd, Militia
2	Truck, Light, Red Force
3	Armour, Light, Blue Force
4	4wd, Blue Force
5	Car, Civilian
6	Car, Civilian
7	Truck, Heavy Civilian
8	Truck, Light, Civilian
9	Truck, Light, Civilian
10	4wd, Civilian

Table 6.5: UAVs - Scenario 2

UAV	Type	Altitude (m)	Sensor	Orientation
1	Fast Delta	100	BW CCD	Down
2	Aerosonde	175	CCD	Down
3	Aerosonde	150	CCD	Forward

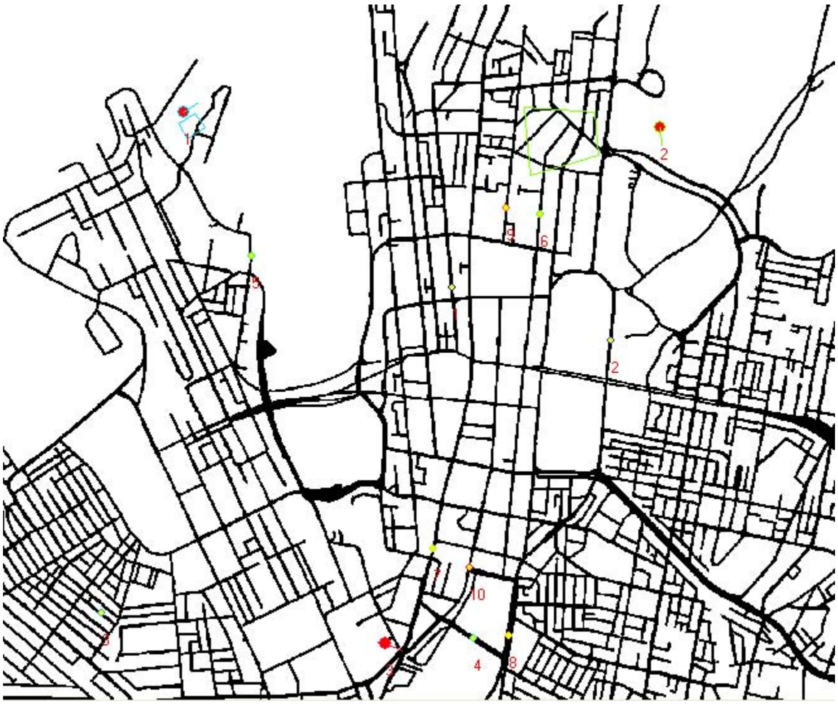


Figure 6.4: Ground Vehicle and UAV initial locations for Scenario 2.

6.3 Situation Setup

Using the Scenarios described in Section 6.2, a number of Situations are developed to examine the impact of: applying the information value scheme presented in Section 5.2, allowing human directed modification of the value scheme and incorporating information from external sources. These are outlined below.

6.3.1 Situation 1 - Baseline

For the 'Baseline' situation, there is no ranking of asset value or preference order for tracking the vehicles detected. Thus the utility function for UAV control maximises information gain with respect to time, without consideration of the information type or value with respect to a larger mission.

6.3.2 Situation 2 - Operator Modified

In the 'Operator Modified' situation, the operator specifies mission parameters prior to the mission start via the Mission Console as described in Section 5.4.4. The priority for each asset is set by the appropriate descriptor from Equation 5.2 chosen by the operator. Table 6.1 is reproduced in Table 6.6 with priorities assigned.

The priorities assigned to each asset are an interpretation of the disruption likely to be caused to the functioning of the city if the asset were targeted for attack.

The system develops a threat map according to the equations in Section 5.2.2 for each vehicle type using the vulnerability and priority assignment in Table 6.6. A representative threat map is shown in 6.5.

The preference ordering for tracking and classifying vehicles is determined through the operator assigned priority, desired track variance and classification threshold. The operator assigned parameters are shown in Table 6.7.

Table 6.6: Assets

Asset	**Classification**	**Priority**	**Vulnerability**
1	Government House	Highly Valuable	Hardened
2	State Library	Useful	Reinforced
3	Hospital	Highly Valuable	Reinforced
4	Army Barracks	Highly Valuable	Protected
5	Museum	Important	Soft
6	Town Hall	Important	Reinforced
7	Ferry Terminal	Important	Reinforced
8	Central Post Office	Important	Reinforced
9	Maritime Museum	Useful	Reinforced
10	Convention Centre	Valuable	Reinforced
11	Exhibition Centre	Important	Reinforced
12	World Trade Centre	Valuable	Hardened
13	Ferry Terminal	Valuable	Reinforced
14	Stadium	Useful	Reinforced
15	Central Station	Valuable	Reinforced
16	Broadcast Centre	Useful	Reinforced
17	Ambulance HQ	Highly Valuable	Reinforced
18	Police HQ	Highly Valuable	Hardened
19	Hospital	Highly Valuable	Reinforced
20	Cargo Terminal	Important	Reinforced
21	School	Useful	Soft
22	School	Useful	Soft

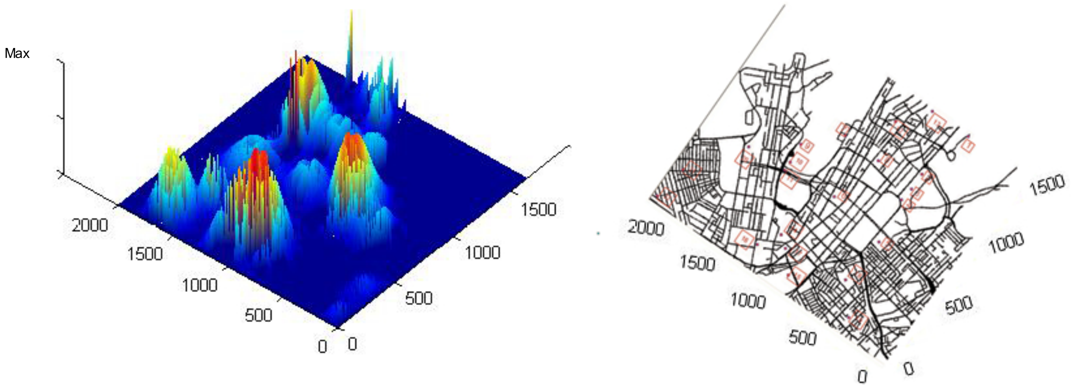


Figure 6.5: Figure on the left shows the relative threat corresponding to the assets outlined in Table 6.1, with the z axis representing the maximum threat from a target located at that point. The x,y scale is in metres, and the map grid is shown in the figure on the right.

The priority ranking of vehicle types largely reflects their ability to cause damage to the assets being protected, with the necessity of knowing where a vehicle is in order to intercept, being reflected in the desired track variances. The application of the classification threshold shown recognises the importance of distinguishing between civilians and irregular or insur-

Table 6.7: Mission Setup

Vehicle	**Desired Variance**	**Classification Threshold**	**Priority**
Civilian Light Truck	35	0.98	Desirable
Blue Force Light Truck	45	0.8	Some Impact
Red Force Light Truck	25	0.95	Highly Desirable
Civilian Medium Truck	35	0.98	Desirable
Blue Force Medium Truck	45	0.8	Some Impact
Red Force Medium Truck	25	0.95	Highly Desirable
Civilian Heavy Truck	35	0.98	Desirable
Blue Force Heavy Truck	60	0.8	Some Impact
Red Force Heavy Truck	10	0.95	Highly Desirable
Civilian 4WD	35	0.98	Some Impact
Blue Force 4WD	60	0.8	Some Impact
Red Force 4WD	10	0.95	Highly Desirable
Militia 4WD	10	0.98	Mission Critical
Civilian Motorbike	35	0.98	Desirable
Militia Motorbike	10	0.98	Mission Critical
Civilian Car	35	0.98	Desirable
Blue Force Car	45	0.8	Some Impact
Red Force Car	25	0.95	Highly Desirable
Militia Car	10	0.98	Mission Critical
Blue Force Light Armour	90	0.8	Some Impact
Red Force Light Armour	10	0.95	Mission Critical
Blue Force Heavy Armour	90	0.8	Some Impact
Blue Force IFV	90	0.8	Some Impact
Search	Likelihood Threshold	0.25	Mission Critical

gent forces prior to intercepting or using force. Thus via the Mission Console much high level reasoning regarding the overall situation is captured from the operator prior to the mission starting, which is then used as part of the utility function for determining allocation of sensor resources.

6.3.3 Situation 3 - Online Operator Modified

In the ‘Online Operator Modified’ situation the operator modifies mission parameters during the mission. This may be done to reflect new mission priorities or other changes in situation that may be external to the system.

The initial mission parameters are identical to Situation 2, with the changes described in Table 6.8.

The modification to the priority assigned to Asset 14, Stadium and Asset 15, Central Station, reflects a situation where the mission priority of an asset or area may change due

Table 6.8: Operator modified mission parameters

Time (s)	Mission Element	Mission Parameter	Original	Modified
250	Stadium	Priority	Useful	Highly Valuable
250	Central Station	Priority	Valuable	Highly Valuable
450	Civilian Light Truck	Priority	Desirable	Highly Desirable
450	Civilian Light Truck	Desired Variance	35	20
450	Civilian 4wd	Priority	Desirable	Highly Desirable
450	Civilian 4wd	Desired Variance	35	20
450	Civilian Motorbike	Priority	Desirable	Highly Desirable
450	Civilian Motorbike	Desired Variance	35	20
450	Civilian Car	Priority	Desirable	Highly Desirable
450	Civilian Car	Desired Variance	35	20

to external factors. For example in this case it may be due to a large event scheduled at the given location or specific information or tip off regarding an imminent threat which is then passed on to the operator. The change in priority and desired variance for all civilian vehicles in this case may represent a situation where it has been found that distinguishing between civilian and militia is more difficult than originally expected.

Using this online modification of mission parameters, the operator's judgement of the situation, both with respect to the larger external mission and events observed through the system, can be captured and used to reallocate sensor resources in response to the current situation.

The 'Online Operator Modified' situation was run only with Scenario 2.

6.3.4 Situation 4 - Observation Points

In the 'Observation Points' situation, observers are located around a number of key installations. These observers can relay information back to the system, with the information provided able to be incorporated into the system's belief state. The initial mission parameters are identical to Situation 2, with the areas in the observers' view shown in Figure 6.6.

The 'Observation Points' situation was run only with Scenario 2.

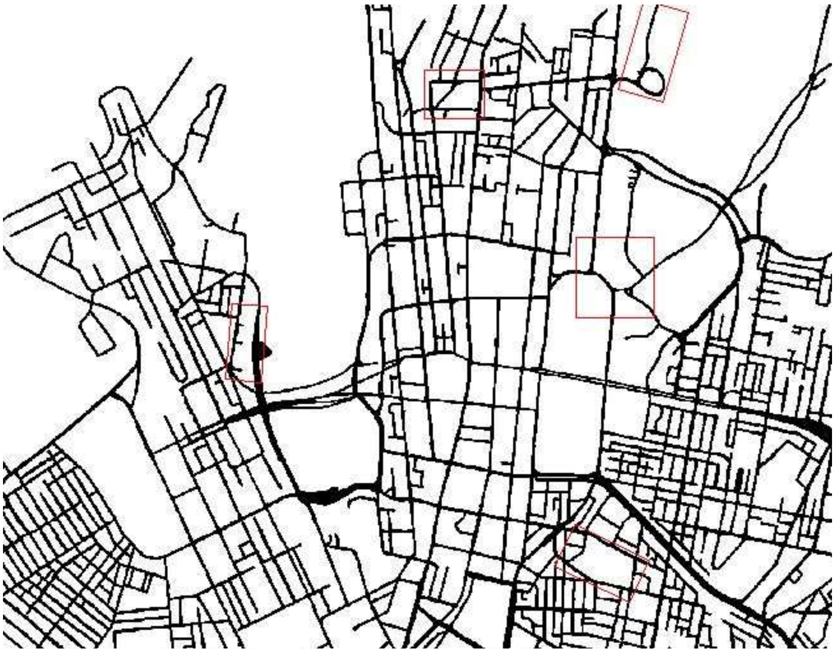


Figure 6.6: Areas observable from the fixed observation points.

6.3.5 Situation 5 - External Observations

The ‘External Observations’ situation is similar to Situation 4, however there are no fixed observers. Observations come through as a single report as they might from a tip off or patrol moving through the area. A summary of the observations is shown in Table 6.9.

Table 6.9: External observations.

Time (s)	Observation
120	Vehicle 1
150	Vehicle 6
450	Vehicle 1
500	Vehicle 2

In this case the external information received is considered reliable and used to update the system’s belief state. The ‘External Observations’ situation was run only with Scenario 2.

6.4 Simulation Results

The following sections present a snap shot of the data collected from running the scenarios and situations from Sections 6.2 and 6.3 in simulation. The data for each situation is the average of five runs. If a target is not detected in at least three of the five runs, the detection, track and classification information for that target is not shown.

For each simulation set, the time to find and classify each of the simulated ground vehicle targets is presented as a comparison of different control types, preference ordering and situations. The threat posed by all simulated ground vehicles, given their location and type is known at each time step is defined as the *true threat*. The true threat for each simulation set is shown graphically along with the system's estimate of the true threat. The absolute difference at each time step between the true and estimated threat is also presented graphically along with the cumulative error in the system's threat estimate.

The results presented are logged at the end of each 5 second planning cycle. Thus new target detections and classifications between planning cycles only appear in the simulation data logs at the end of each cycle. The operator, however sees the new detection or classification as soon as it is fused into the system belief state.

6.4.1 Baseline vs Operator Modified

In this section the simulation is run with no preference ordering, which is referred to as the Baseline, and is compared to the results of adding preference ordering, referred to as Operator Modified. In the Operator Modified scenario the asset and ground vehicle preference data from Tables 6.6 and 6.7 is input prior to the mission start. This is then used to determine the value of information as per Chapter 5. Both Scenario 1 and 2 are examined for each of the four control schemes.

Scenario 1

Examining Table 6.10 it can seen that the two control schemes which have limited waypoint planning, FHP and FHF, show identical detection times for the Baseline and Operator Modified simulations. It appears that by only considering the local threat, the behaviour of the sensor platforms is not sufficiently altered by preference ordering to result in additional

Table 6.10: Targets Tracked - Baseline vs Operator Modified for Scenario 1

	FHP BSL	**FHP OPM**	**FHF BSL**	**FHF OPM**	**IHP BSL**	**IHP OPM**	**Task BSL**	**Task OPM**
Targets Found	3	3	5	5	1	4	5	4
Time to Find (s)								
1	-	-	185.5	185.5	-	75.5	80.5	-
2	15.5	15.5	140.5	140.5	-	200.5	10.5	260.5
3	-	-	100.5	100.5	-	-	105.5	190.5
4	280.5	280.5	15.5	15.5	10.5	10.5	420.5	10.5
5	10.5	10.5	35.5	35.5	-	115.5	10.5	110.5
Targets Classified	1	1	3	5	1	4	4	1
Time to Classify (s)								
1	-	-	-	755.5	-	100.5	160.5	-
2	-	-	310.5	310.5	-	230.5	20.5	265.5
3	-	-	470.5	495.5	-	-	625.5	-
4	365.5	365.5	-	675.5	70.5	170.5	445.5	-
5	-	-	490.5	500.5	-	290.5	-	-

target detections. As preference ordering primarily effects the threat map on a larger scale, the difference in utility between observing 2 nearby nodes will be similar with or without preference ordering. In general preference ordering will only cause a significant difference in detection performance with FHP and FHF when the ordering results in 2 assets close together having very different values or when a target track of high or low value is within the area being considered.

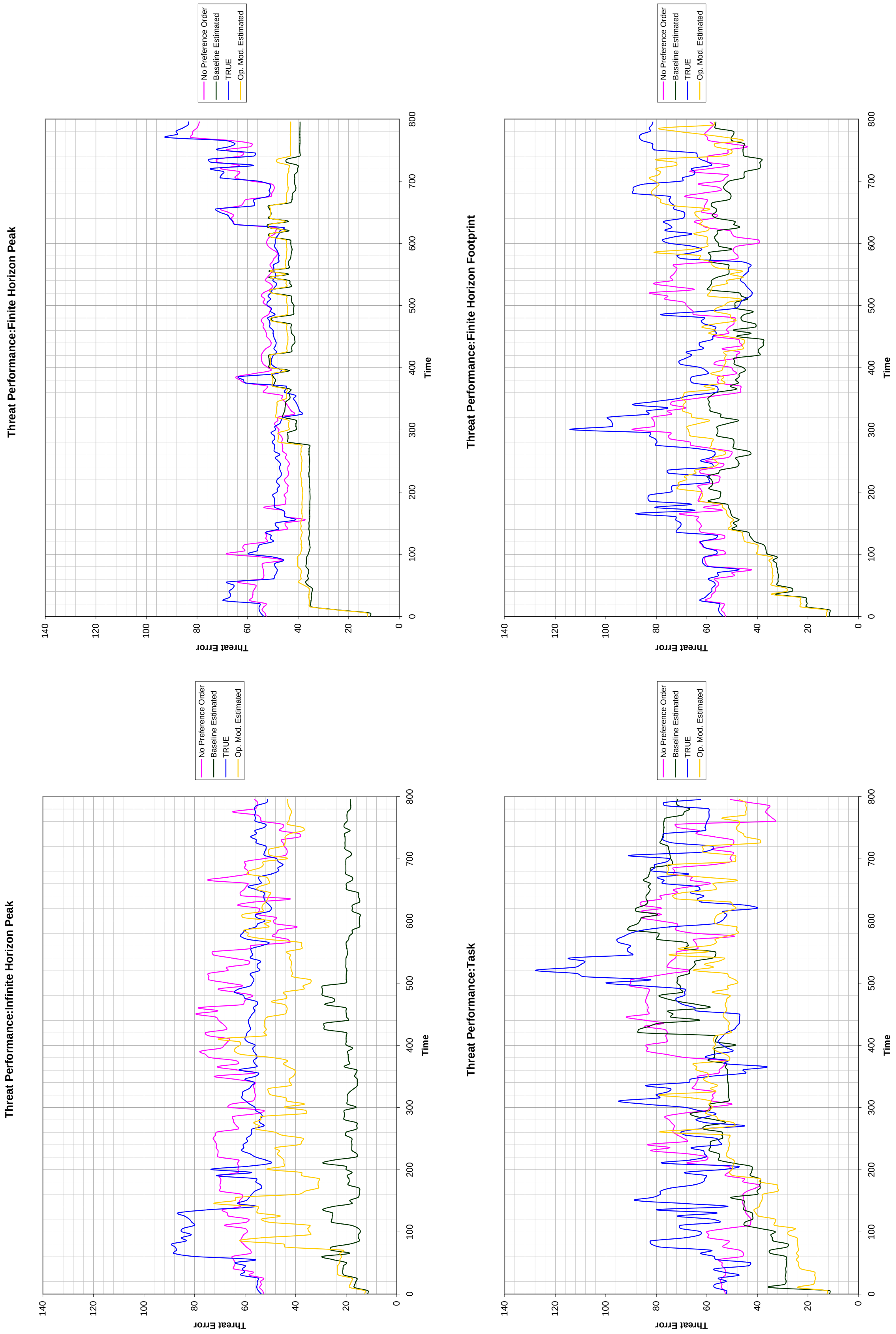
The number of targets detected with the IHP control scheme increased from 1 to 4 using preference ordering, and most significantly resulted in the detection of Target 1 which is the Red Force threat. Unlike FHP and FHF control schemes, IHP considers the entire threat map when determining a new waypoint and thus, changes due to preference ordering have a greater impact. Using Task based control resulted fewer detections with preference ordering and importantly, Target 1 was not detected.

Of the four control schemes only FHF control detected Target 1 with or without preference ordering. Without preference ordering Target 1 was not classified, however with ordering it was classified at 755.5 seconds. Of note is that while target detection times for FHF did not change, all targets were classified.

Examining the FHP and FHF graphs in Figure 6.7 it can be seen that changes to the true and estimated threat plots through the addition of preference ordering were relatively small. Much more significant changes can be seen for IHP and Task based controls. The

IHP graph shows consistently better threat estimates using preference ordering while the Task graph shows marginally better performance for the Baseline estimate. This is better illustrated in Figure 6.8.

In Figure in 6.8 as in Figure 6.7 the difference between the estimated threat, with or without preference ordering, with FHP control is very small. As a result the cumulative error plots are similar with a small improvement seen when applying preference ordering. FHF shows a similar trend however the performance improvement is significantly larger. As is alluded to by Table 6.10 and Figure 6.7, applying preference ordering to IHP control significantly improves the threat estimate. Conversely for Task based control, the average error is greater when using preference ordering. Looking at the Task graph the oscillating behaviour of the threat estimate can be seen with the threat estimate error up to twice the peak error of IHP, particularly when using preference ordering.



Threat Performance:Infinite Horizon Peak

Threat Performance:Finite Horizon Peak

Threat Performance:Task

Threat Performance:Finite Horizon Footprint

Figure 6.7: True and Estimated Threat for Scenario 1 for each control scheme.

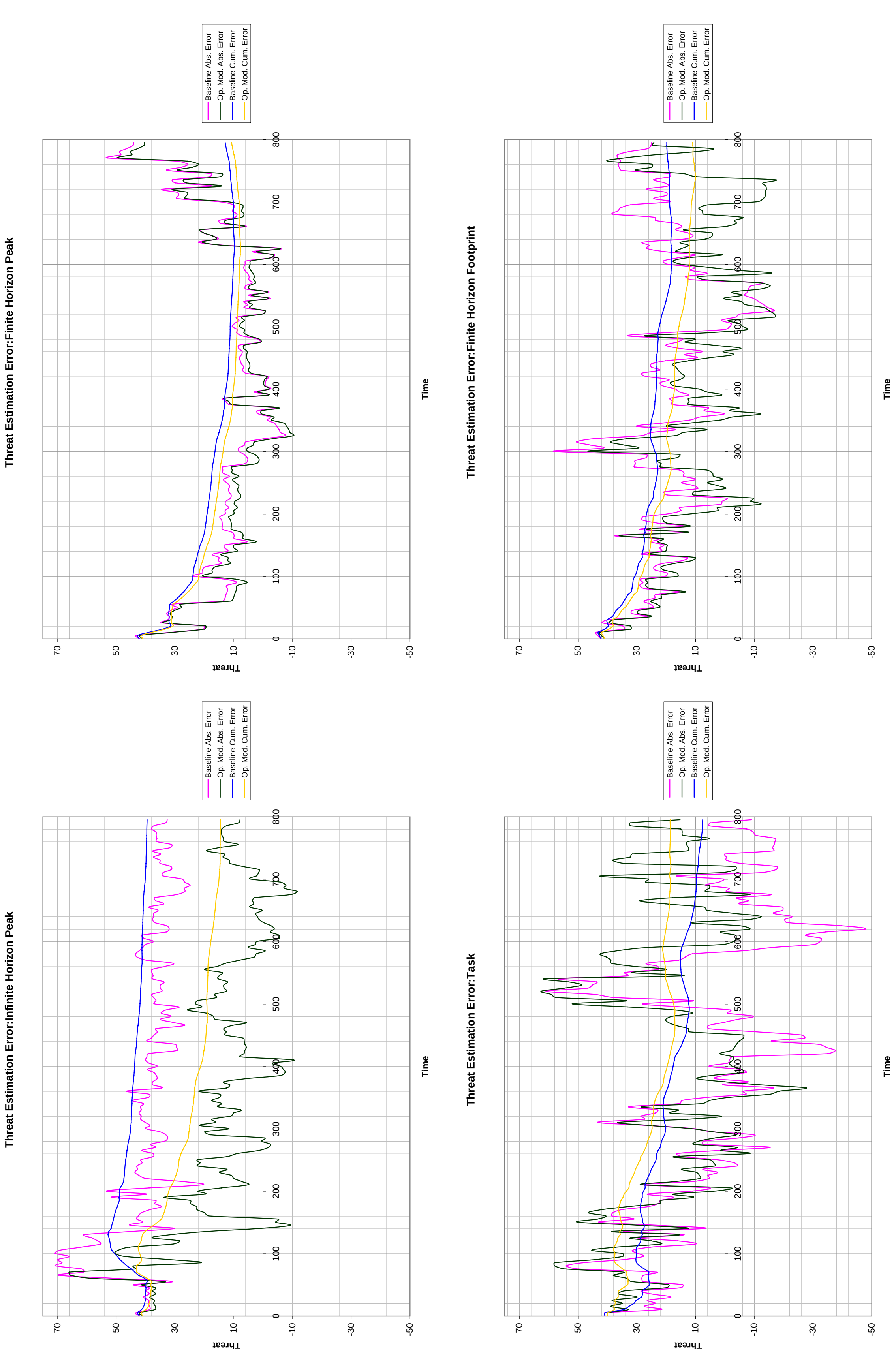
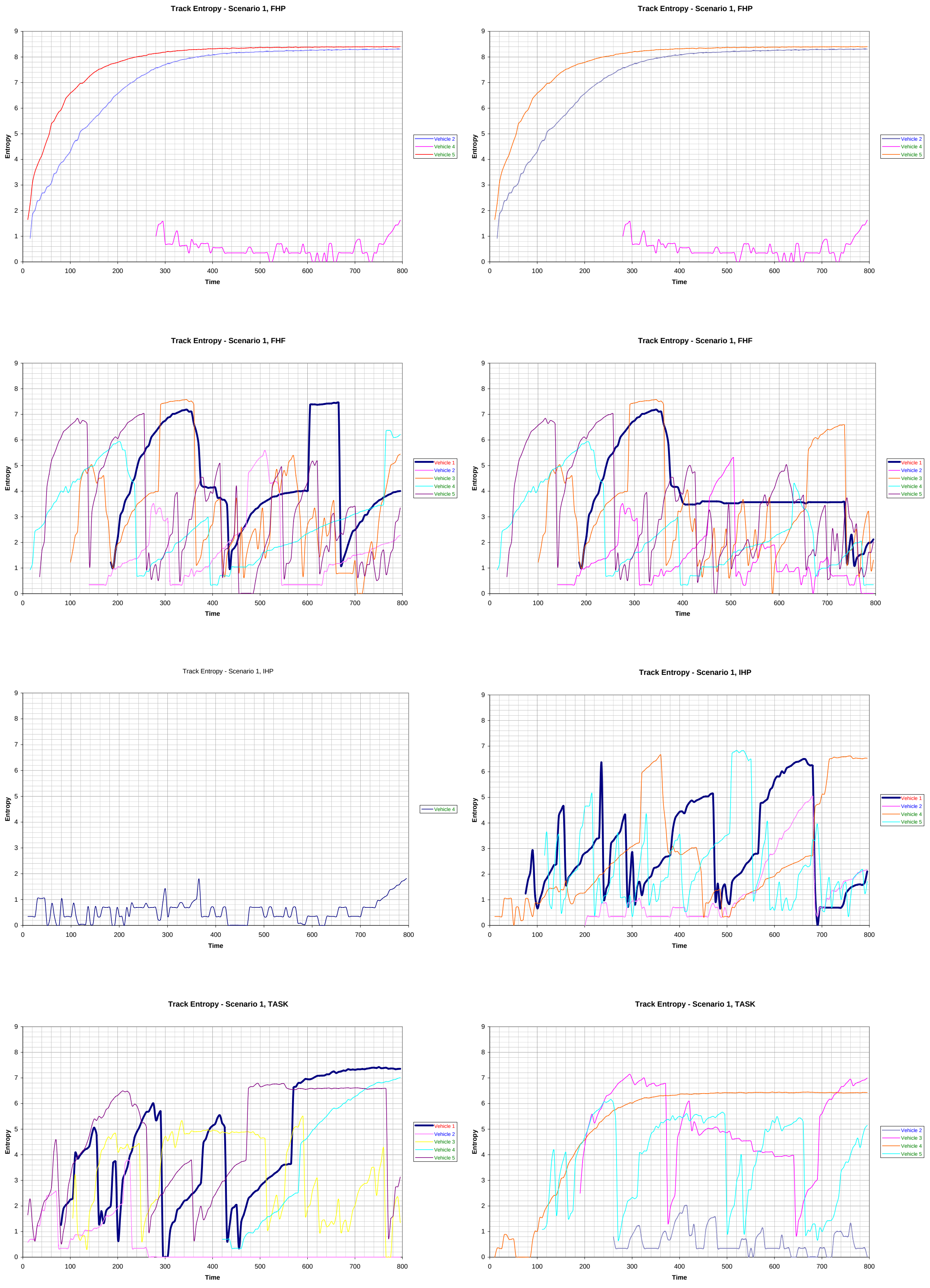


Figure 6.8: The threat estimate error and cumulative average error for each control scheme in Scenario 1.

The Track Entropy graph in figure 6.9 shows the same track behaviour in both the preference and non-preference ordered scenarios. This indicates that the small improvements in performance due to ordering are the result of reductions in the search state entropy. As neither IHP or Task controllers detected Target 1 in both preferenced an preferenced cases, these graphs provide little to inform the change of performance with respect to threat estimation, however from the entropy of the targets detected it can be seen that preference ordering does have a significant effect on the observations taken.

The effect of preference ordering is most clearly seen in the FHF graph with the entropy of the Target 1 track distribution. The relatively constant track entropy between 400 and 750s indicates constant observation of track particles without actually observing the target itself. The Baseline plot shows 2 addition observations of Target 1 at approximately 450s and 660s, however by the subsequent peaks it can be seen that the system has not allocated resources to maintain a good track estimate, despite potentially being a high threat target. Conversely the preference ordered plot indicates resources being allocated to tracking Target 1 while allowing the entropy to increase on other less threatening tracks, most clearly seen in the plot of Target 3.


Track Entropy - Scenario 1, FHP

Track Entropy - Scenario 1, FHP

Track Entropy - Scenario 1, FHF

Track Entropy - Scenario 1, FHF

Track Entropy - Scenario 1, IHP

Track Entropy - Scenario 1, IHP

Track Entropy - Scenario 1, TASK

Track Entropy - Scenario 1, TASK

Figure 6.9: Track entropy for each each control scheme in Scenario 1. Baseline left and Operator modified right.

Scenario 2

The target detection times for FHP and FHF 6.11 with and without preference ordering in Scenario 2, closely mirror Scenario 1. As in Scenario 1, preference ordering when using the Task based controller resulted in fewer detected targets and those detected often being detected later. The most significant change in behaviour from Scenario 1 to 2 is in the IHP sets. In Scenario 2 detection times are very similar in the Baseline and Operator Modified simulations with only a small change for Targets 3 and 5.

Table 6.11: Targets Tracked - Baseline vs Operator Modified for Scenario 2

	FHP	**FHP**	**FHF**	**FHF**	**IHP**	**IHP**	**Task**	**Task**
	BSL	**OPM**	**BSL**	**OPM**	**BSL**	**OPM**	**BSL**	**OPM**
Targets Found	6	6	10	10	10	10	10	9
Time to Find (s)								
1	25.5	25.5	25.5	25.5	5.5	5.5	5.5	95.5
2	-	-	10.5	10.5	5.5	5.5	30.5	65.5
3	30.5	30.5	100.5	100.5	180.5	240.5	50.5	-
4	-	-	45.5	45.5	5.5	5.5	5.5	10.5
5	130.5	130.5	380.5	375.5	115.5	250.5	5.5	115.5
6	5.5	5.5	35.5	35.5	5.5	5.5	5.5	25.5
7	240.5	240.5	15.5	15.5	10.5	10.5	10.5	15.5
8	-	-	10.5	10.5	5.5	5.5	5.5	25.5
9	-	-	350.5	230.5	5.5	5.5	5.5	40.5
10	135.5	135.5	10.5	10.5	10.5	10.5	5.5	15.5
Targets Classified	2	2	9	10	10	10	10	8
Time to Classify (s)								
1	-	-	120.5	120.5	500.5	115.5	210.5	255.5
2	-	-	350.5	490.5	130.5	125.5	75.5	80.5
3	55.5	55.5	780.5	580.5	250.5	295.5	350.5	-
4	-	-	195.5	220.5	440.5	80.5	55.5	75.5
5	-	-	-	720.5	380.5	605.5	525.5	-
6	-	-	115.5	120.5	285.5	660.5	315.5	310.5
7	445.5	445.5	45.5	45.5	80.5	210.5	200.5	80.5
8	-	-	130.5	130.5	135.5	85.5	45.5	400.5
9	-	-	450.5	535.5	105.5	110.5	75.5	125.5
10	-	-	120.5	120.5	460.5	195.5	80.5	80.5

As for Scenario 1 the threat estimate plots for FHP, shown in Figure 6.10, with and without preference ordering are similar. Unlike Scenario 1, the FHF threat estimate with preference ordering is considerably closer to the true threat for most of the simulation. For the IHP controller, the clear performance improvement with respect to threat estimate observed in Scenario 1 is not present in Scenario 2.

As for Scenario 1, for FHP, FHF and IHP controllers there is an improvement in threat estimation performance with preference ordering, while the Task controller exhibits worse performance with ordering.

The Track Entropy graphs in figure 6.12 for FHP show the same behaviour for Scenario 2 as Scenario 1 and thus the conclusions regarding the performance improvements due to search entropy also apply. For the FHF and IHP controllers, there is no evidence of any performance improvement due to preference ordering and the resulting track of the Red Force ground vehicle, Target 1. For the Task based controller however, the Baseline does not display the high track entropy plateau from 500 to 800s seen in the Operator Modified scenario.

Baseline vs Operator Modified Summary

From the results of Scenarios 1 and 2, FHP, FHF and IHP control schemes showed improved estimation of true threat with the use of preference ordering to assign value to assets and targets. For these 3 control schemes the overall target detection rate, detection times, classification rate and classification times either stayed the same or showed improvement. By the same performance measures, the application of preference ordering for the Task based control scheme showed poorer performance.

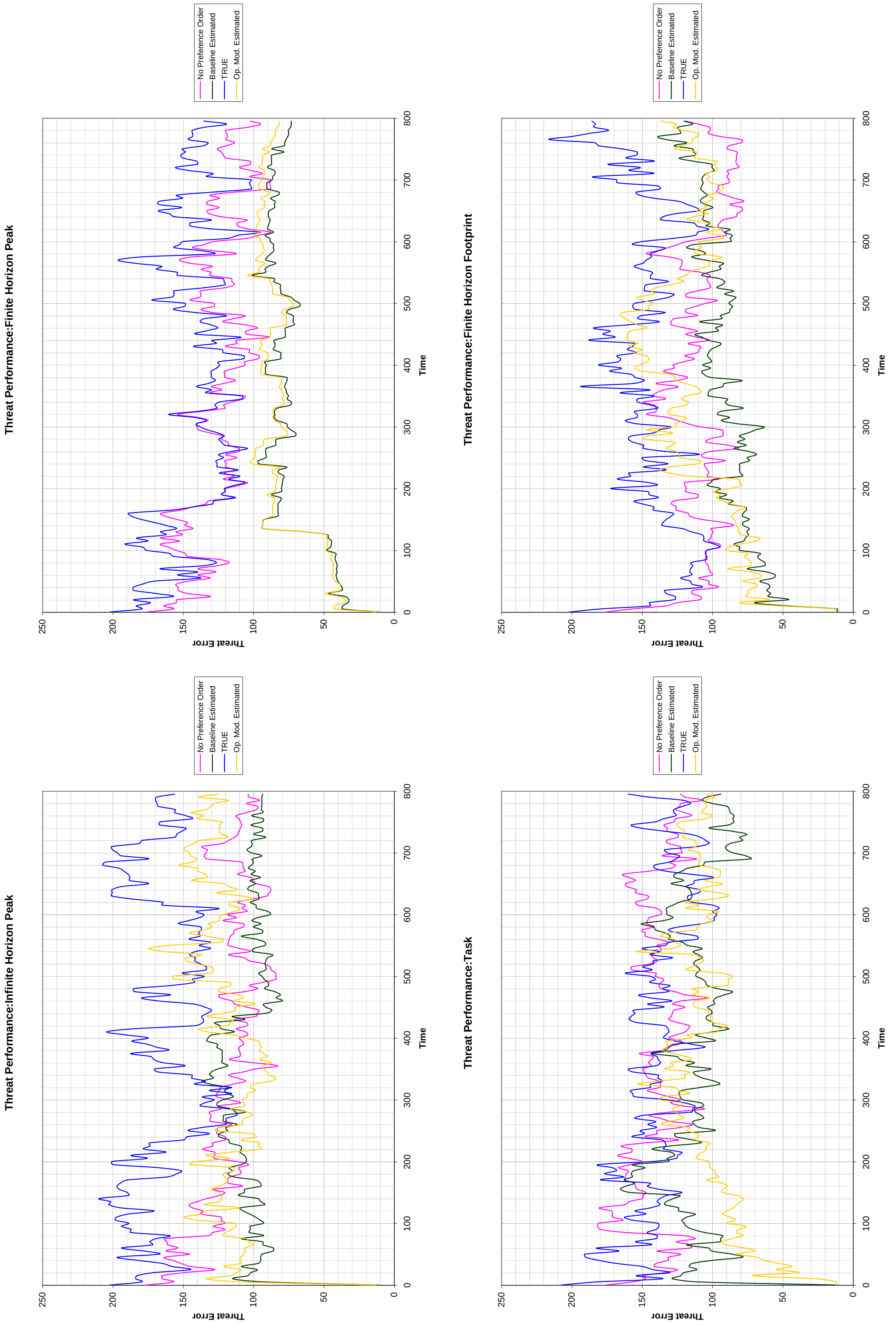


Figure 6.10: True and Estimated Threat for Scenario 2 for each control scheme.

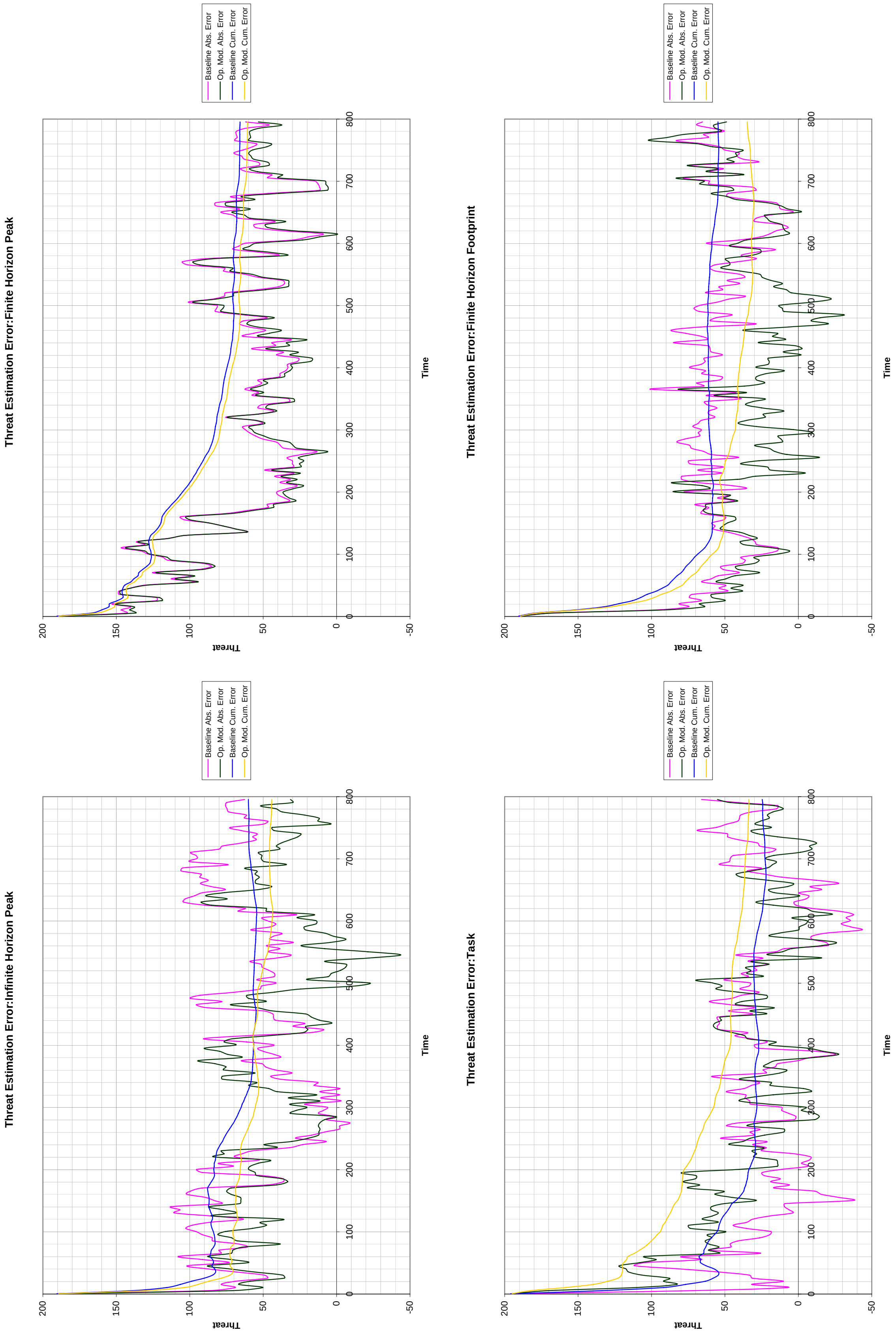


Figure 6.11: The difference between True and Estimated Threat for Scenario 2 with Baseline left and Operator modified right.

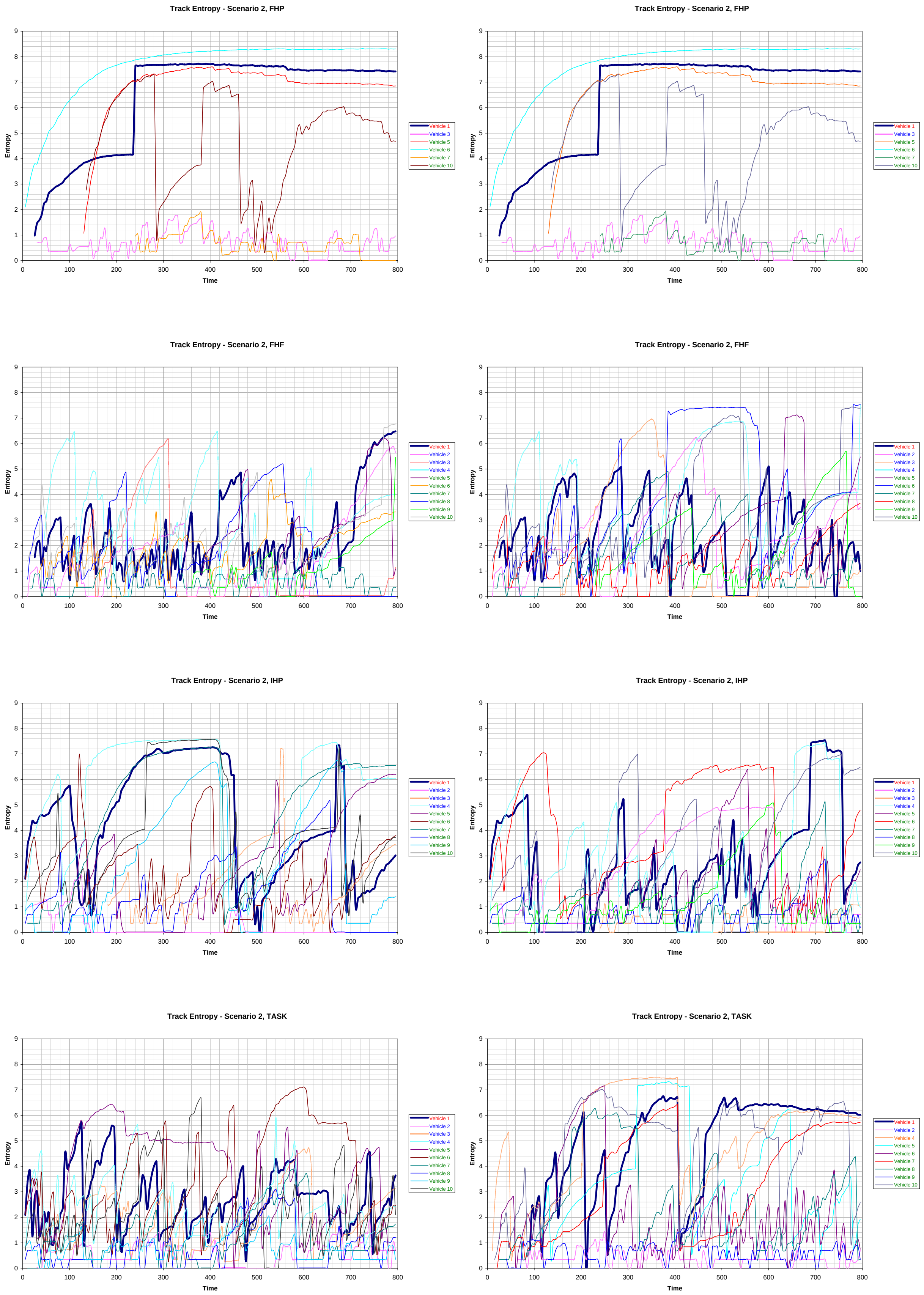


Figure 6.12: Track entropy for each each control scheme in Scenario 2. Baseline left and Operator modified right.

6.4.2 Control Type Comparison

The Operator Modified system used in Section 6.4.1 is examined here in more detail to compare the different control types and to highlight some behaviours of the system resulting from utilising information value to make control decision.

Figures 6.13 and 6.14 show the entropy of each state observed by the system for each control scheme in Scenarios 1 and 2 respectively. As one of the states observed the target track entropy graphs presented in Section 6.4.1 are contained within these graphs.

From Figures 6.13 and 6.14 it can be seen that the search entropy remains largely constant. This is due to the area of the map continuously observed by the sensor platforms being small and relatively constant compared to the total search area. A more detail comparison of the search component of the observed states for each control scheme is shown in Figures 6.21 and 6.22.

Target observations can be clearly seen as steps in the classification entropy plots and troughs in the track entropy plots. Troughs in the track entropy plots that do not have a corresponding step in classification indicate observation of track particles without observation of the target. An example of this discussed in Section 6.4.1 can be seen in the FHF graph. No observations of Target 1 are made during the period 200 to 700s, however the track entropy is maintained through observation of the track.

The entropy limit can be seen for both the search and target track states. For the track entropy plots this limit is the point at which there is equal probability on each track particle and each particle is centred on a different node. This is seen in both FHP graphs. Where this entropy limit is reached, the particle filter tracker is likely to become ‘starved’ unless an observation is made quickly. In the starved state, the filter does not have enough particles to represent the true state of the target track and subsequent observations cannot be associated to the track without additional classification information. With a dense road network, particle starvation of implemented tracker can occur quickly. Starvation can be observed in the Scenario 1 FHP track plot for Vehicles 2 and 5 and in the Scenario 2 FHP track plot for Vehicle 6. Other tracks approach starvation, but in each case an observation is made before this occurs.

Step increases in track entropy can be caused both by characteristics of the map network and characteristics of the particle filter tracker. Each road junction will cause a step change in track entropy and a large number of intersections will very rapidly decay the track estimate. However these steps are likely to be small. A null target observation of the track can also result in increased entropy in the track distribution if the null observation of the target is associated with a number of particles which have a high probability prior. The redistribution of these particles and their associated probability across the track distribution can cause a step increase in track entropy.

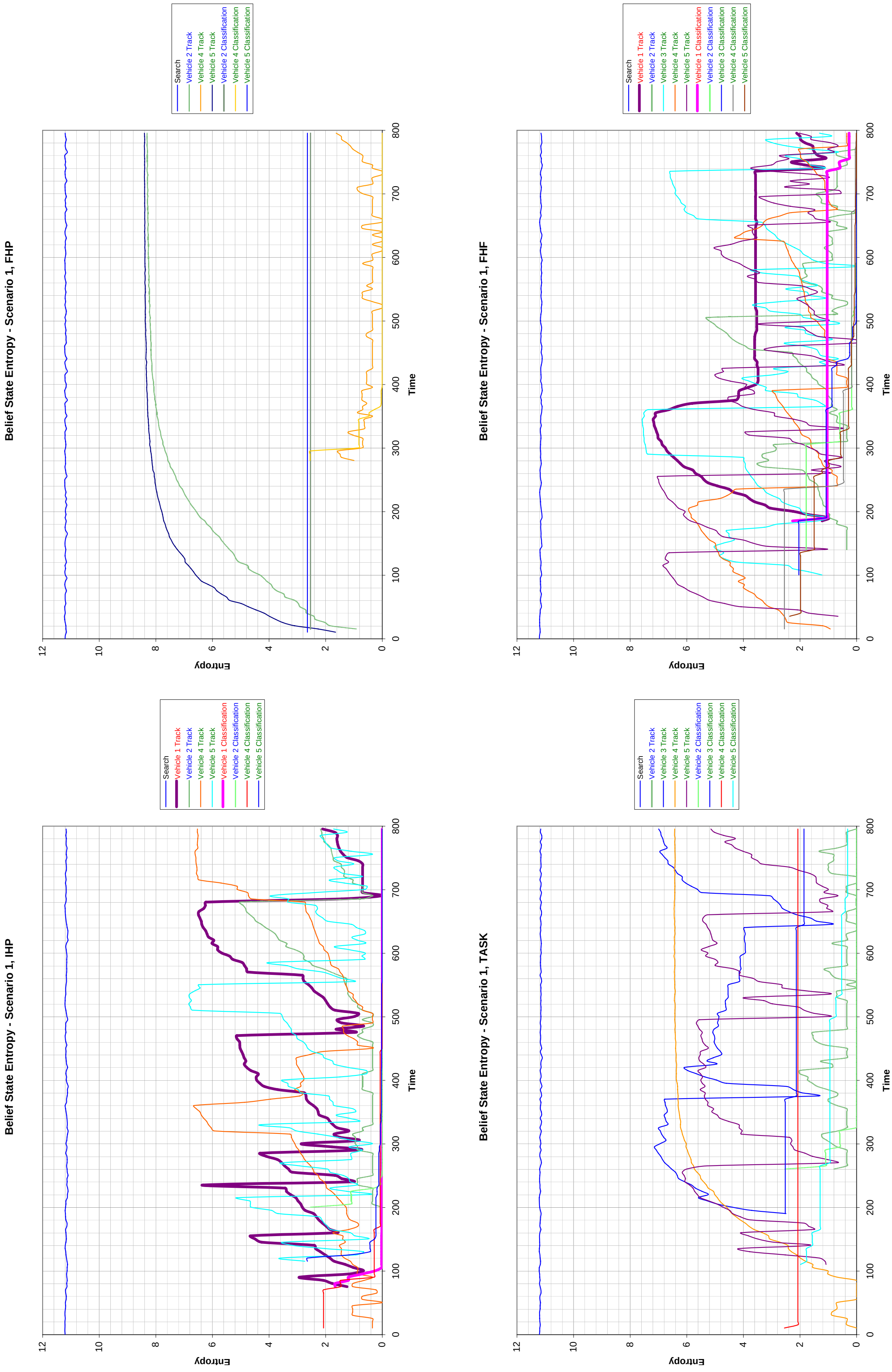


Figure 6.13: Entropy of each belief state observed for Scenario 1.

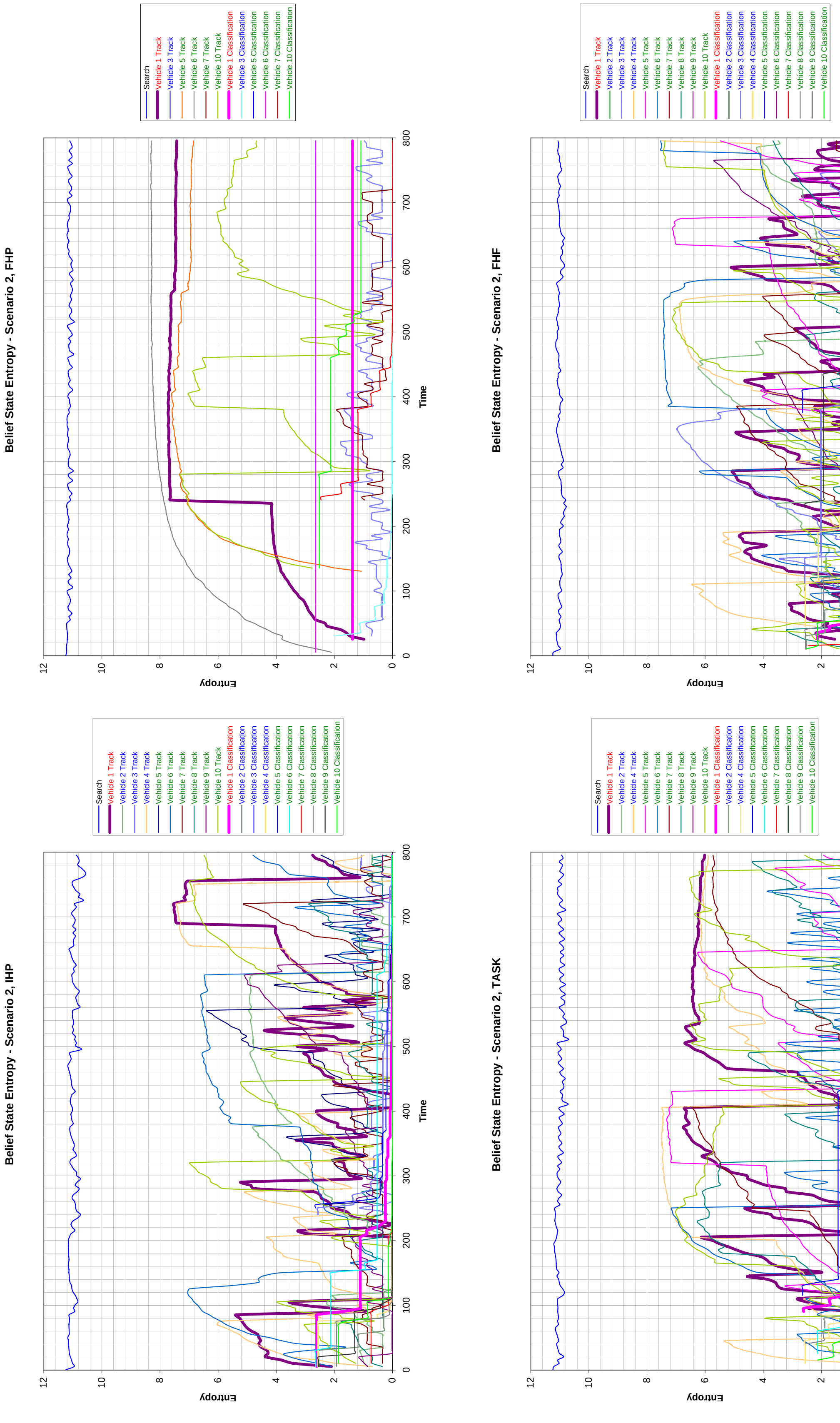


Figure 6.14: Entropy of the belief states for Scenario 2.

The track Hellinger-Bhattacharya distance in Figures 6.15 and 6.18 is a measure of the distance of the current track estimate from a naive or flat distribution. When a vehicle is detected particles are redistributed by the tracker such that they cluster around the point on the map associated with detection. Fusing a target observation results in minimum track distribution entropy and maximum Hellinger-Bhattacharya distance from the naive distribution. These observations can be seen as steps in the plot and correspond with the points of minimum entropy in Figures 6.13 and 6.14. The Hellinger-Bhattacharya distance graphs mirror the associated entropy graphs, however the decay in the target position estimate as the target ground vehicle moves around the road network can be seen more clearly.

It is the weighted Hellinger-Bhattacharya distance shown in Figures 6.17 and 6.20 that is used by all control schemes as the basis for assigning information value. This weighted distance considers the track distribution with respect to the assets being protected, the target classification and the classification entropy. The later shown in Figures 6.16 and 6.19.

An example of the insight into the system behaviour provided by the weighted Hellinger-Bhattacharya distance is the plot for Target 1 with FHF control during Scenario 2. In Figure 6.14 the track entropy for Target 1 shows an oscillating tendency with peaks and troughs at similar values. This is reflected in the track Hellinger-Bhattacharya distance in Figure 6.18. However in the weighted Hellinger-Bhattacharya plot in Figure 6.20 there are 3 distinct segments for the Target 1 peak, from 0 to 150s, 150 to 650s and 650 to 800s. At approximately 150s it can be seen from Figure 6.19 that Target 1 is classified, and being a Red Force vehicle means that the value for observing the Target 1 track will increase. This is reflected in the higher peaks in the weighted Hellinger-Bhattacharya distance after this time. The better track estimate resulting from a number of track observations from 650s, as seen in Figure 6.14, combined with the Target beginning to move away from a high priority asset results in reduced peaks in the weighted Hellinger-Bhattacharya plot.

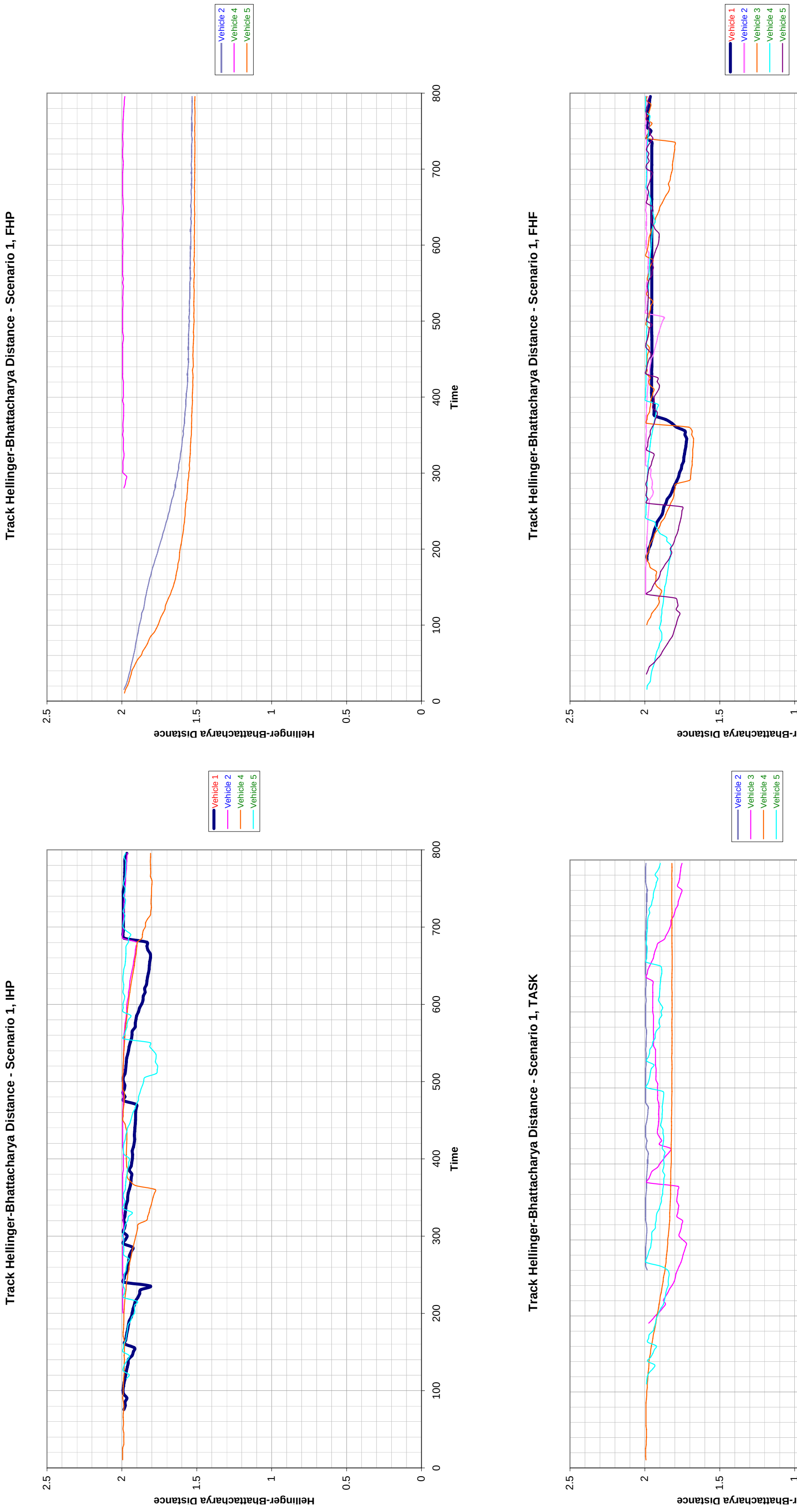


Figure 6.15: The track Hellinger-Bhattacharya distance for Scenario 1 for each control scheme.

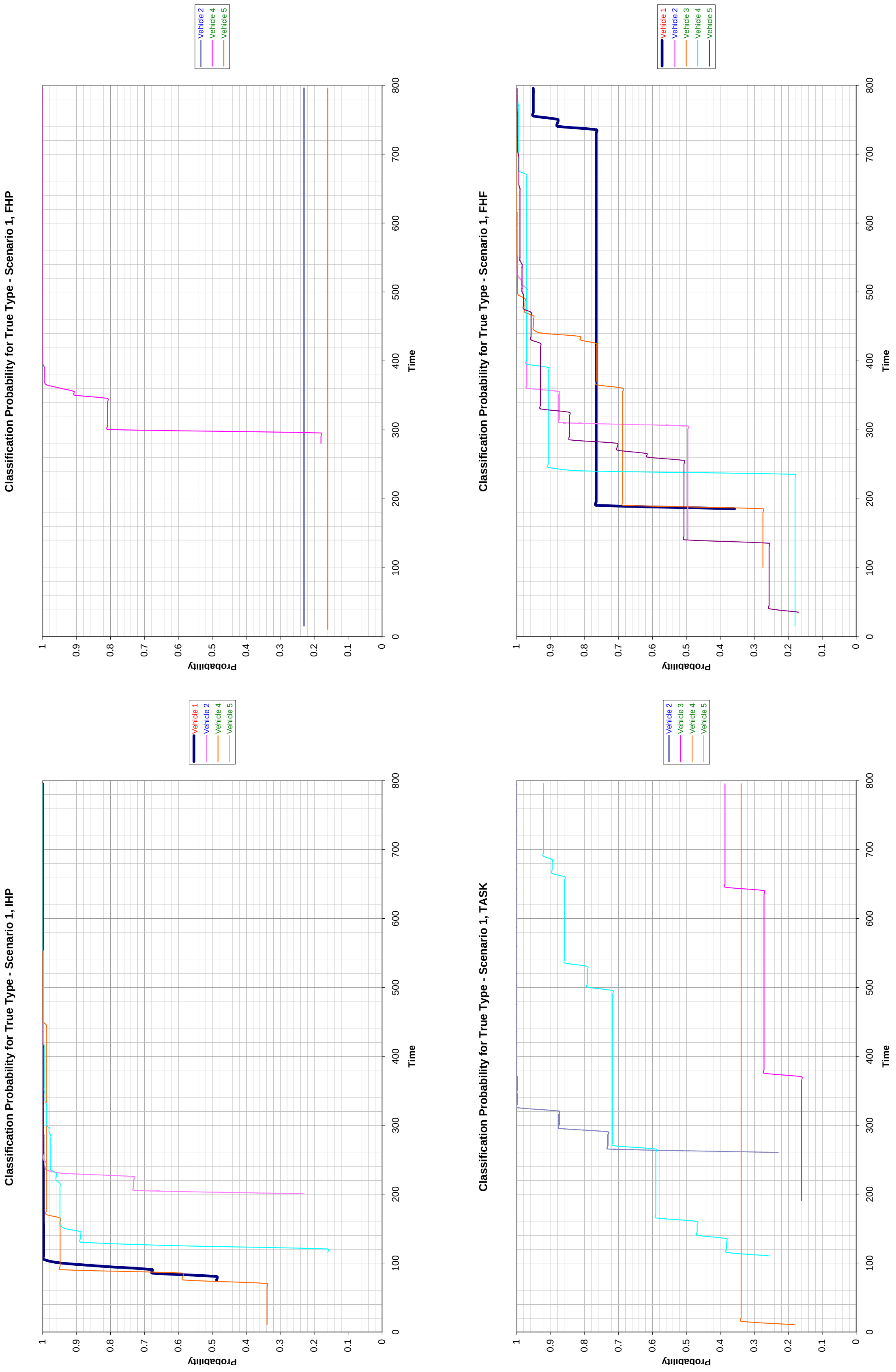


Figure 6.16: Classification probability conditioned on true type for targets detected using each control scheme for Scenario 1.

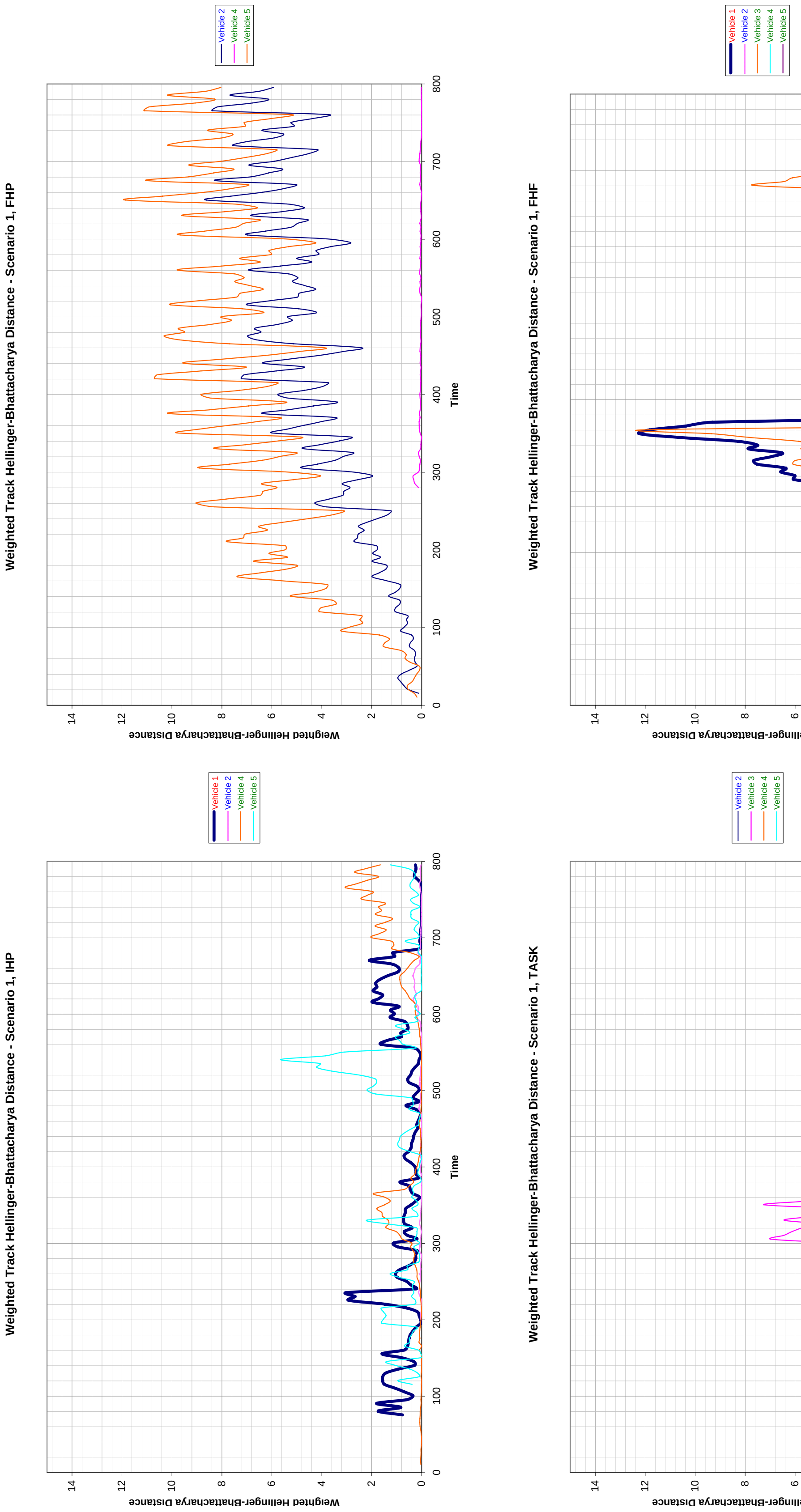


Figure 6.17: Weighted Hellinger-Bhattacharya distance for Scenario 1 for each control scheme.

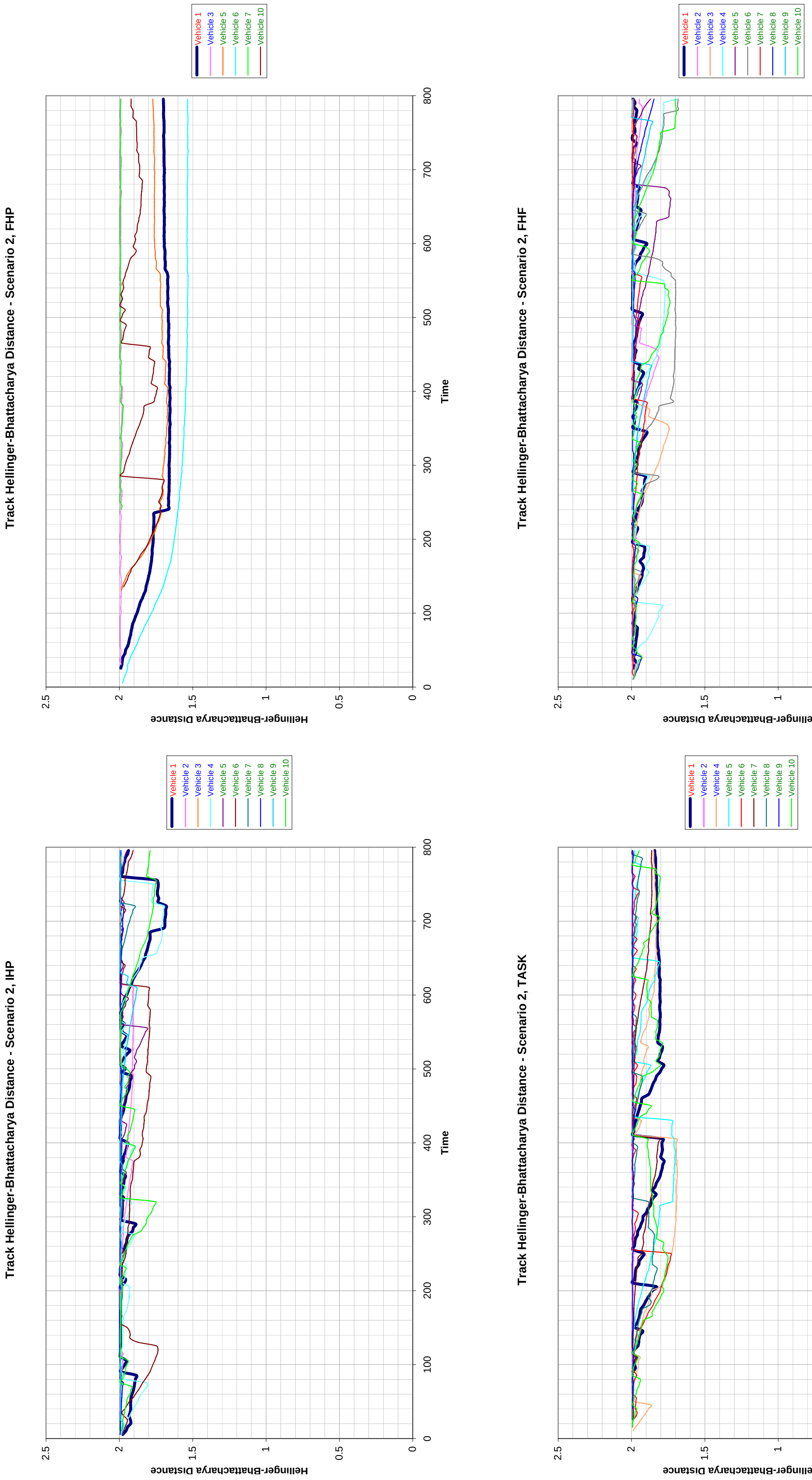
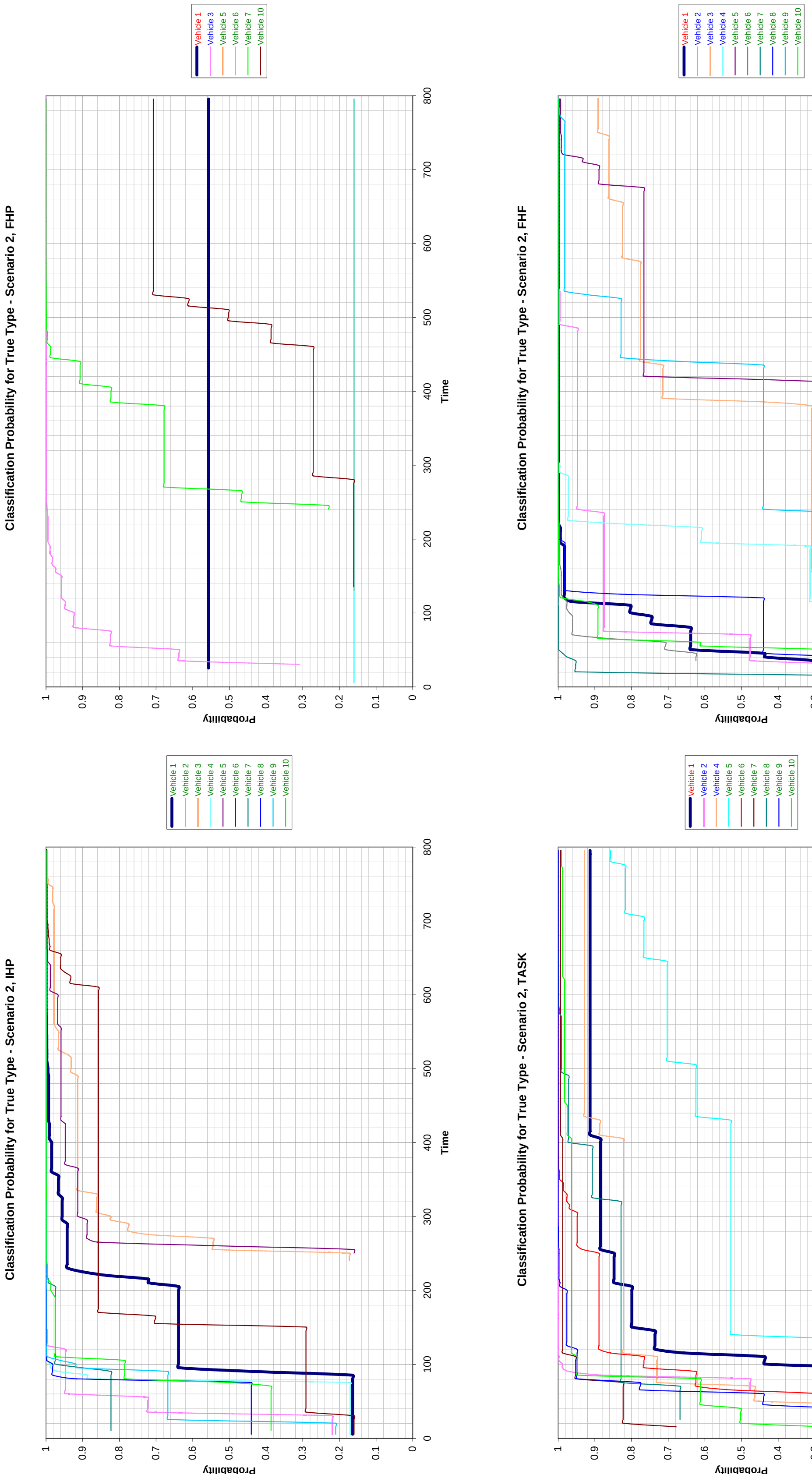


Figure 6.18: The track Hellinger-Bhattacharya distance for Scenario 2 for each control scheme.


Classification Probability for True Type - Scenario 2, FHP

Classification Probability for True Type - Scenario 2, FHF

Classification Probability for True Type - Scenario 2, IHP

Classification Probability for True Type - Scenario 2, TASK

Figure 6.19: Classification probability conditioned on true type for targets detected using each control scheme for Scenario 2.

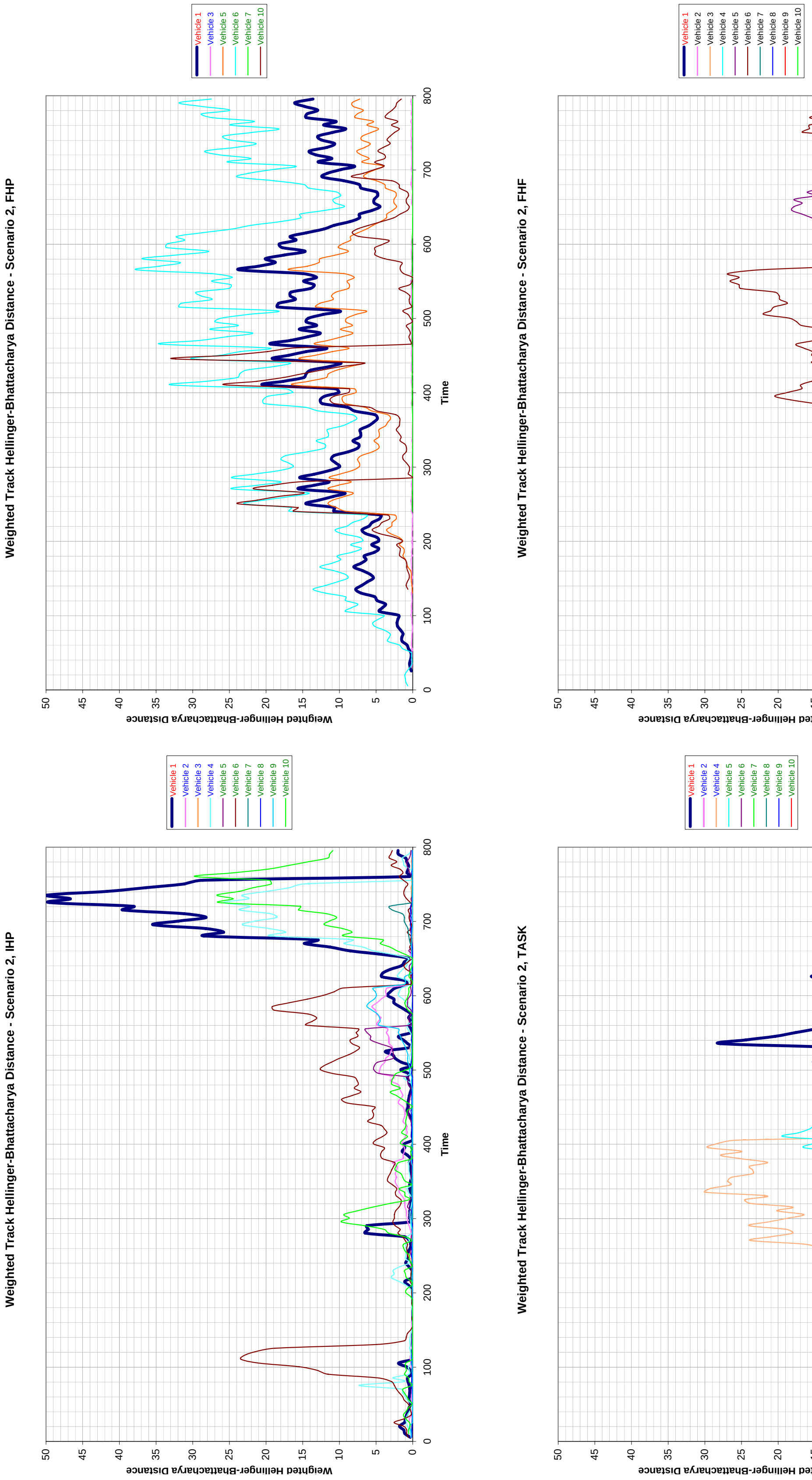


Figure 6.20: Weighted Hellinger-Bhattacharya distance for Scenario 2 for each control scheme.

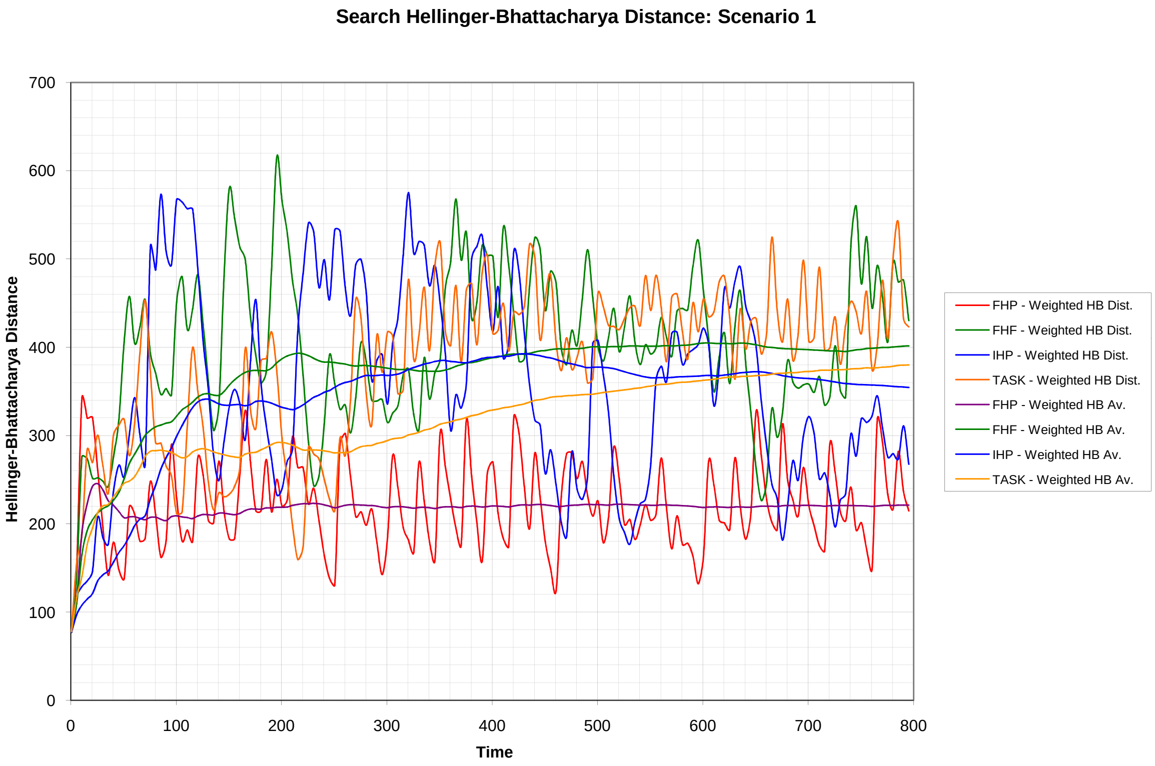


Figure 6.21: Weighted Hellinger-Bhattacharya distance for the search distribution for each control scheme in Scenario 1.

Figures 6.21 and 6.22 show the weighted Hellinger-Bhattacharya distance of the search distribution from the naive or flat distribution along with the cumulative average. Lower values indicate less probability of undetected targets around valuable assets and/or a more compact search distribution in these areas. For Scenario 1, there is a clear performance ranking between the different control schemes:

- FHP
- Task
- IHP
- FHF

For Scenario 2 the difference in performance between FHF, IHP and Task is minimal, however FHP shows significantly better performance.

From the detection/classification Tables 6.10 and 6.11 for Scenario 1 using the Operator Modified preference ordering the performance ranking for the control types is:

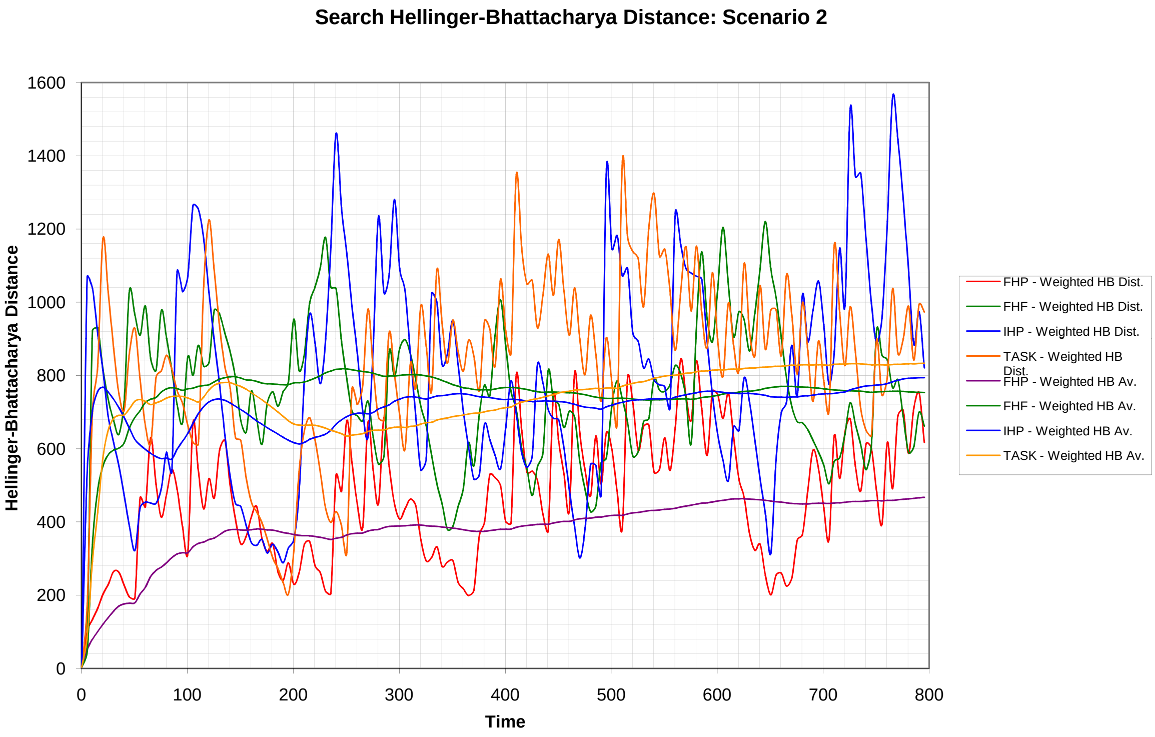


Figure 6.22: Weighted Hellinger-Bhattacharya distance for the search distribution for each control scheme in Scenario 2.

- FHF with 5 detections, 5 classifications
- IHP with 4 detections, 4 classifications
- Task with 4 detections, 1 classification
- FHP with 3 detections, 1 classification

For Scenario 2 the performance ranking is:

- IHP with 10 detections, 10 classifications
- FHF with 10 detections, 10 classifications
- Task with 9 detections, 8 classifications
- FHP with 6 detections, 2 classification

Note that for both scenarios, the detection and classification performance of the Task based controller with no preference ordering is similar to the top ranked control schemes.

Considering the cumulative average threat error in Figure 6.8, the performance ranking for Scenario 1 is:

- FHF
- FHP
- IHP
- Task

From Figure 6.11 the ranking for Scenario 2 is:

- FHF
- Task
- IHP
- FHP

Note that using Task based control with no preference ordering, the cumulative average threat error is below all other control schemes using preference ordering. However the speed at which the average threat error drops to the lower value is slower, this indicates that the other control schemes provide a better threat estimate during the initial states of simulation.

In Figure 6.20 the weighted Hellinger-Bhattacharya plots for Target 1 have consistently low peaks for FHF and IHP control. This indicates that these control systems are providing good track information on Target 1, particularly when the track propagates into areas where Target 1 may pose a threat to assets. While there is a very large peak at around 700s in the IHP plot, a number of observations are made relatively quickly bringing it back down. This is in contrast to the FHP plot which remains consistently high and the system does not bring the value back down. The Task based controller shows characteristics present in both FHF and FHP plots at different time. Up to approximately 300s a low weighted Hellinger-Bhattacharya distance for Target 1 is maintained, while after 500s this value increases significantly and the system does not react as aggressively as with IHP control to bring it back down.

Control Type Comparison Summary

Despite using the same preference ordering, the relative performance of the different control schemes implemented varies significantly with the best performing depending on the metric of performance used. Discounting the performance of the Task based controller without preference ordering, the FHF and IHP control schemes are the best performers with respect to detecting, classifying and tracking targets. The FHP control in general performs worst with respect to these metrics, with the Task based controller performing somewhere in the middle. However with respect to search performance, the FHP controller shows a significant performance advantage over the other control types.

6.4.3 Mission: Online Operator Modified

In the Online Operator Modified scenario, mission parameters are modified during the simulation as described by Table 6.8. Detection and classification times along with the threat estimate are used for comparison with the Operator Modified scenario in which the preference ordering is static.

In Figure 6.23 a small step increase in threat can be seen at 250s, which corresponds to the operator upgrading the value of Assets 14 and 15 to ‘Highly Valuable’. This change results in both undetected and tracked ground vehicles in the vicinity of these assets being considered a higher threat and thus there is an increase in the overall true threat. A second larger step is seen at 450s. This corresponds to the operator receiving information that the Red Force is commandeering civilian vehicles and as a result upgrades the priority for detecting and tracking civilian ground vehicles. As there are 6 civilian vehicles in the simulation, this change results in a large increase in the true threat. As can be seen in Table 6.12, all control schemes have a number of civilian vehicles being tracked, thus the estimated threat also increases when this priority is changed. Note that for FHP control only 3 civilian vehicles are in track at 450s and thus the estimated threat does not increase to the same extent as the true threat. For all cases it can be seen that the operator’s actions have approximately doubled the true threat.

In Figure 6.24 both the threat error and cumulative threat error plots can be seen to diverge at 250s, which corresponds to the first online modification of the mission. In all 4 control cases the average threat estimation error is significantly reduced compared to the unmodified scenario.

Despite the significantly better performance with respect to threat estimation, the detection and classification times for all control schemes is little changed. In Table 6.12 only FHF reduces in classification performance with one less civilian target classified while FHF and IHP take longer to detect civilian Targets 4 and 5 with online modification of the mission.

6.4.4 Online Operator Modified Summary

The threat estimate plots show that the online changes to the mission have an immediate effect on the system’s threat estimate and in some cases the change in the threat estimate

Table 6.12: Targets Tracked - Online Operator Modified Scenario 2

	FHP BSL	**FHP OOPM**	**FHF BSL**	**FHF OOPM**	**IHP BSL**	**IHP OOPM**	**Task BSL**	**Task OOPM**
Targets Found	6	6	10	10	10	10	10	10
Time to Find (s)								
1	25.5	25.5	25.5	25.5	5.5	5.5	5.5	25.5
2	-	-	10.5	10.5	5.5	5.5	30.5	70.5
3	30.5	30.5	100.5	100.5	180.5	250.5	50.5	40.5
4	-	-	45.5	45.5	5.5	5.5	5.5	10.5
5	130.5	130.5	380.5	340.5	115.5	250.5	5.5	570.5
6	5.5	5.5	35.5	35.5	5.5	5.5	5.5	85.5
7	240.5	240.5	15.5	15.5	10.5	10.5	10.5	15.5
8	-	-	10.5	10.5	5.5	5.5	5.5	65.5
9	-	-	350.5	175.5	5.5	5.5	5.5	40.5
10	135.5	135.5	10.5	10.5	10.5	10.5	5.5	15.5
Targets Classified	2	1	9	8	10	10	10	7
Time to Classify (s)								
1	-	-	120.5	120.5	500.5	360.5	210.5	-
2	-	-	350.5	285.5	130.5	125.5	75.5	275.5
3	55.5	55.5	780.5	-	250.5	295.5	350.5	-
4	-	-	195.5	270.5	440.5	80.5	55.5	60.5
5	-	-	-	-	380.5	645.5	525.5	-
6	-	-	115.5	120.5	285.5	720.5	315.5	360.5
7	445.5	-	45.5	45.5	80.5	210.5	200.5	65.5
8	-	-	130.5	130.5	135.5	85.5	45.5	200.5
9	-	-	450.5	365.5	105.5	110.5	75.5	220.5
10	-	-	120.5	120.5	460.5	195.5	80.5	65.5

will depend on the system's current belief state. The threat error plots show the changes quickly becoming part of the decision making process for all control schemes and by updating the mission with external information, provide a significantly better estimate of the true threat.

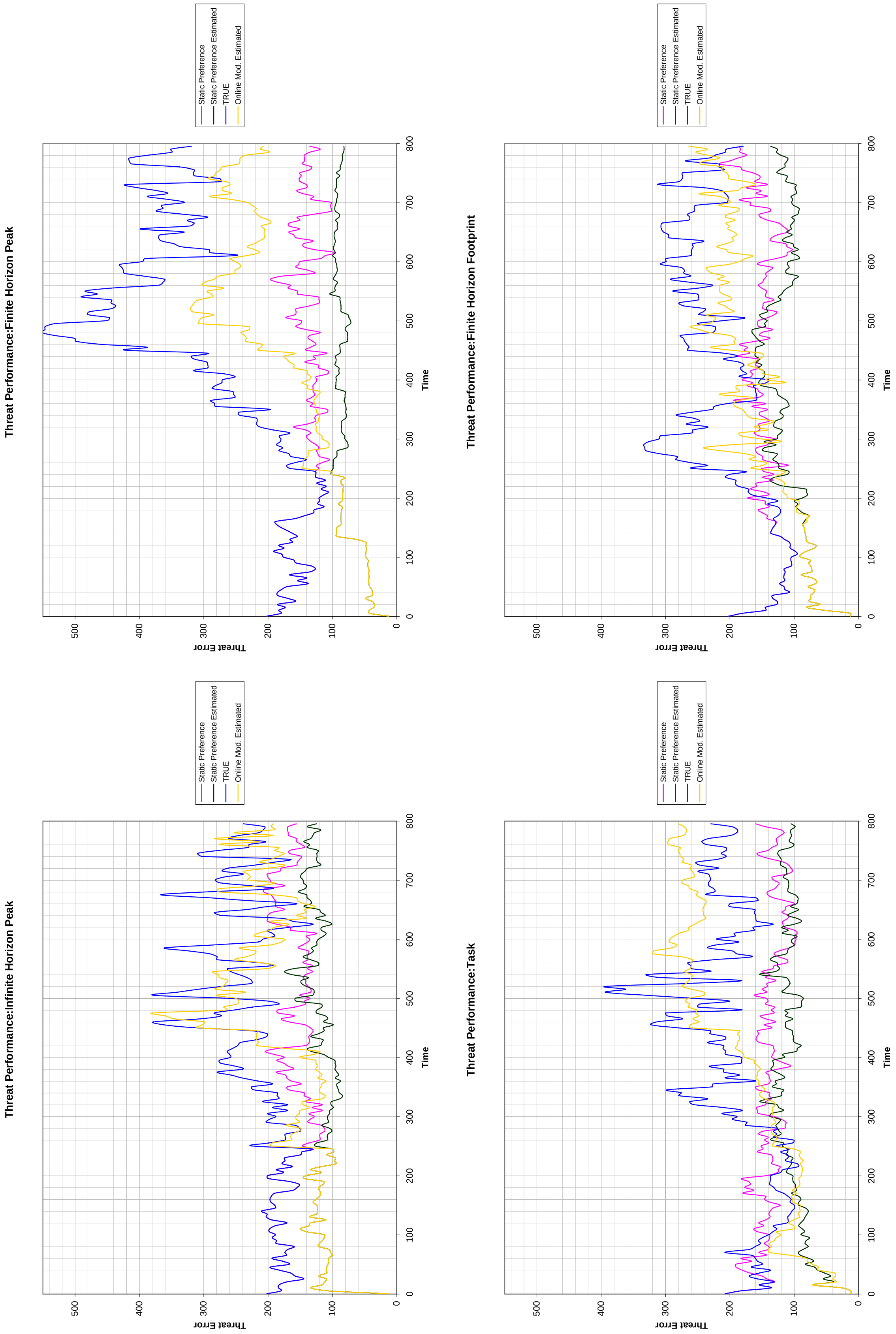
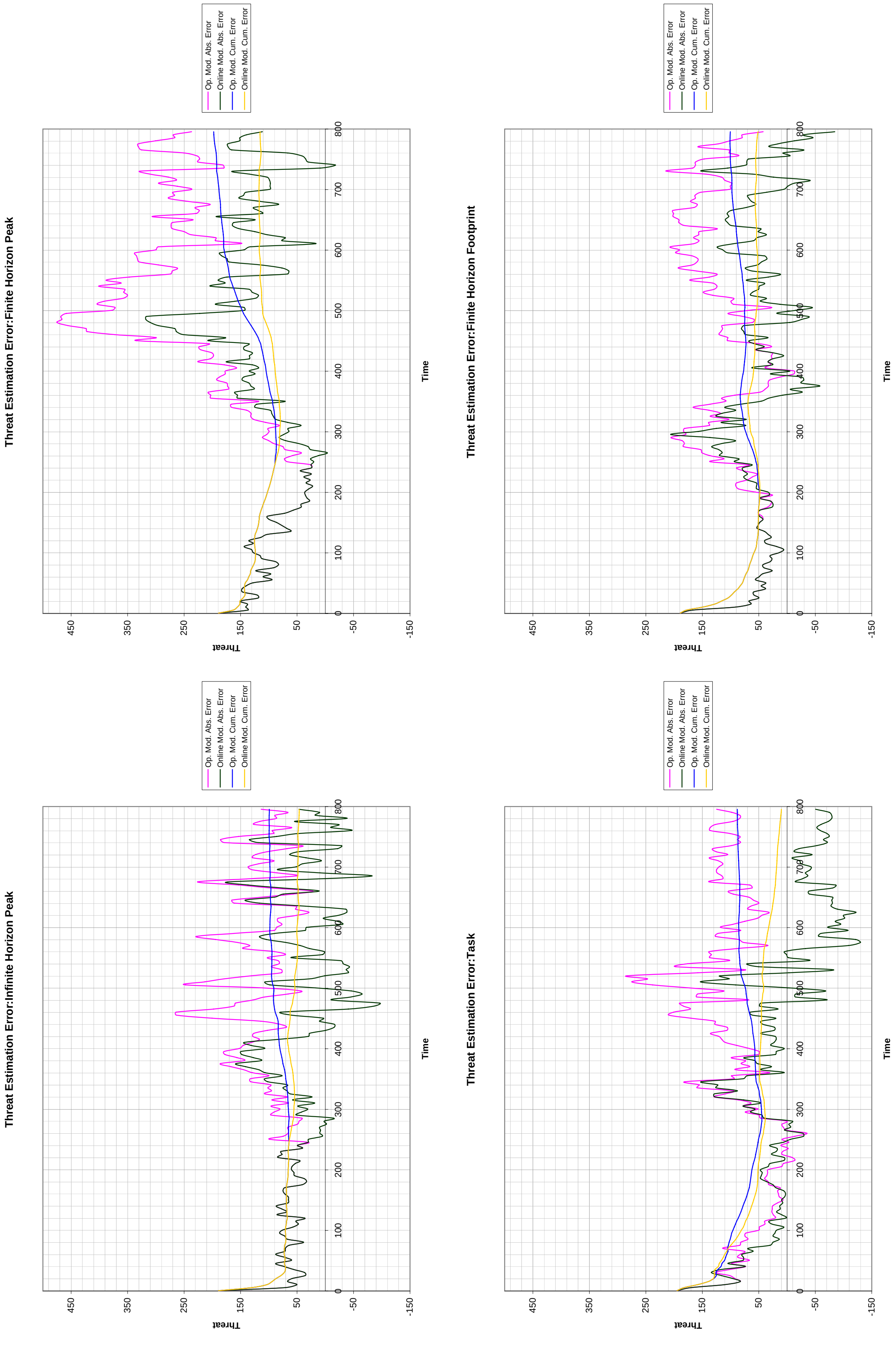


Figure 6.23: True and Estimated Threat for the Online Operator Modified situation.



Threat Estimation Error:Finite Horizon Peak

Threat Estimation Error:Infinite Horizon Peak

Threat Estimation Error:Finite Horizon Footprint

Threat Estimation Error:Task

Figure 6.24: The threat estimate error and cumulative average error for each control scheme.

6.4.5 Mission: Observation Points

In the observation points scenario a portion of the map is continuously observed, either through surveillance cameras or an *Observation Post* (OP). The system treats this as another sensor in the network and updates the belief accordingly. The location and area covered by these additional sensors is shown in Figure 6.6.

The scale and shape of the system estimated threat graphs shown in Figure 6.25 indicate that the additional sensors have little effect on the overall threat estimate. The reason for this is the small area of the map covered by the additional sensors, and as static OPs, only observations of targets and tracks passing through the field of view are made.

The cumulative threat error plots shown in Figure 6.26 show the addition of static sensors results in only a small change in threat estimation performance. However this change in performance is for the worse with the IHP controller and for the FHF controller in the later stages of the simulation.

Table 6.13: Targets Tracked - Observation Points Scenario 2

	FHP BSL	**FHP OBPS**	**FHF BSL**	**FHF OBPS**	**IHP BSL**	**IHP OBPS**	**Task BSL**	**Task OBPS**
Targets Found	6	10	10	10	10	10	10	10
Time to Find (s)								
1	25.5	25.5	25.5	25.5	5.5	55.5	5.5	20.5
2	-	10.5	10.5	10.5	5.5	10.5	30.5	10.5
3	30.5	30.5	100.5	250.5	180.5	295.5	50.5	10.5
4	-	450.5	45.5	90.5	5.5	80.5	5.5	45.5
5	130.5	80.5	380.5	40.5	115.5	285.5	5.5	30.5
6	5.5	5.5	35.5	35.5	5.5	5.5	5.5	40.5
7	240.5	445.5	15.5	15.5	10.5	20.5	10.5	20.5
8	-	480.5	10.5	10.5	5.5	65.5	5.5	190.5
9	-	715.5	350.5	35.5	5.5	45.5	5.5	30.5
10	135.5	65.5	10.5	10.5	10.5	65.5	5.5	165.5
Targets Classified	2	7	9	10	10	9	10	10
Time to Classify (s)								
1	-	-	120.5	130.5	500.5	210.5	210.5	190.5
2	-	15.5	350.5	15.5	130.5	15.5	75.5	15.5
3	55.5	35.5	780.5	500.5	250.5	390.5	350.5	190.5
4	-	455.5	195.5	130.5	440.5	85.5	55.5	50.5
5	-	125.5	-	215.5	380.5	-	525.5	85.5
6	-	70.5	115.5	85.5	285.5	125.5	315.5	330.5
7	445.5	-	45.5	85.5	80.5	65.5	200.5	205.5
8	-	485.5	130.5	225.5	135.5	80.5	45.5	225.5
9	-	720.5	450.5	130.5	105.5	130.5	75.5	75.5
10	-	-	120.5	185.5	460.5	225.5	80.5	185.5

The changes in decision making due to the addition of the static sensors is more clearly seen in the detection/ classification times shown in Table 6.13. In the case of IHP, Target 2 is detected at 5.5s due to directing a UAV to observe a high threat area. With the addition off an OP in the vicinity there is reduced utility in assigning a UAV to observe the area and thus this target is not detected by a UAV. The table shows that Target 2 is first detected by an OP at 10.5s and this sensor then classifies the target at 15.5s. As the observation is fused into the common belief state, subsequent target observation to update the track estimate are performed by UAV observation. The addition of static sensors is also responsible for the additional detections seen in the FHP results. Targets 4, 8 and 9 are first detected by a static sensor and these observations are also responsible for enabling classification of Target 4,5,6,8 and 9. Despite the additional tracks and classifications with FHP, from Figure 6.26 the overall estimate is poorer. This is as a result of the additional targets being civilian vehicles which pose a comparatively small threat to assets.

6.4.6 Observation Points Summary

In the scenario presented, the addition of observation points or static sensors did not have a large impact on the system's performance with respect to threat estimation. The main change in the system's performance will be seen in the search belief state, however the impact will be largely dependant on the ratio of the area covered by the static sensors to the overall area. The probability of high threat targets passing through the static sensor's field of view may also contribute to some extent. As shown by threat estimate error plots, the additional static sensors do not necessarily result in better threat estimation performance.

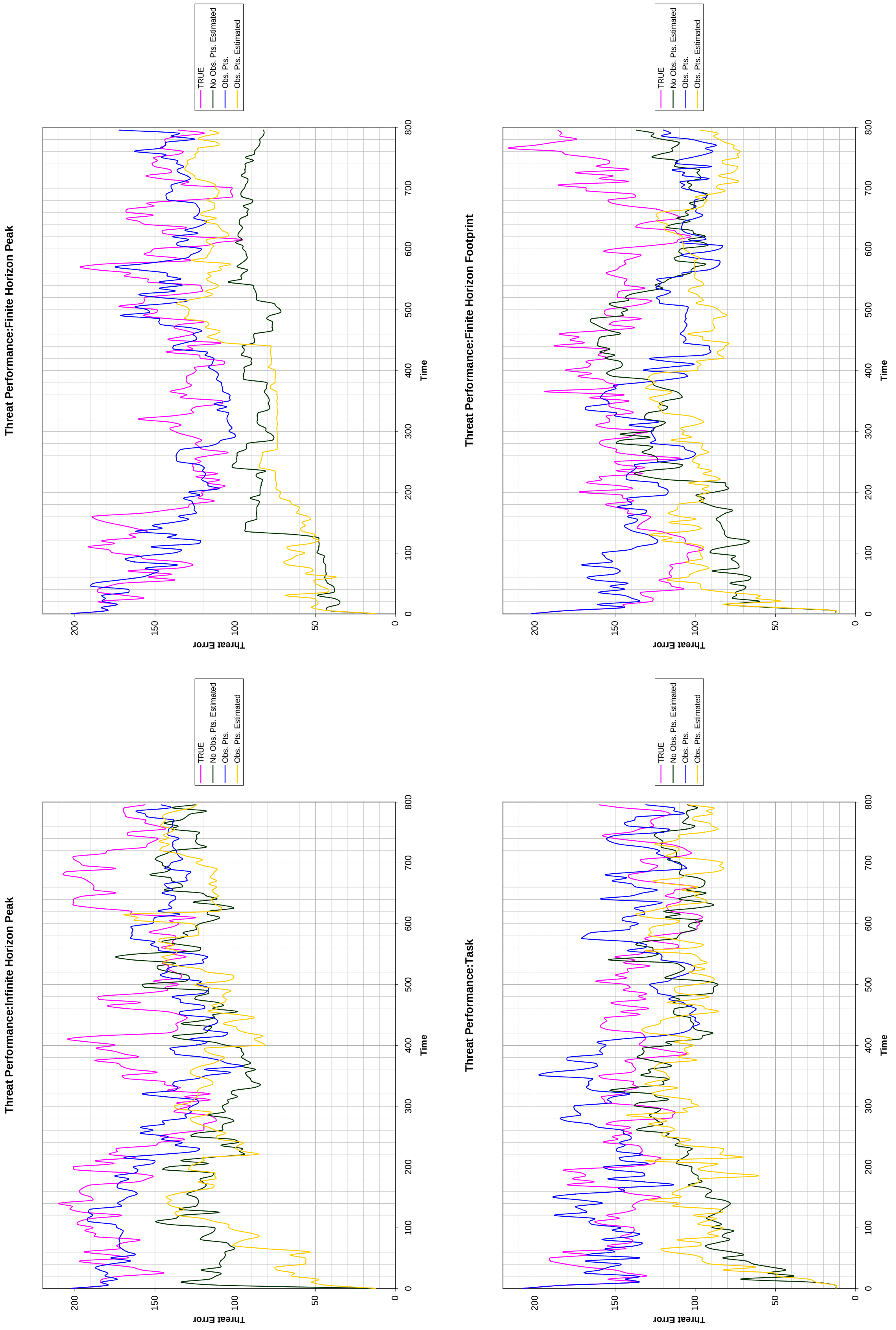


Figure 6.25: True and Estimated Threat for the Observation Points situation.

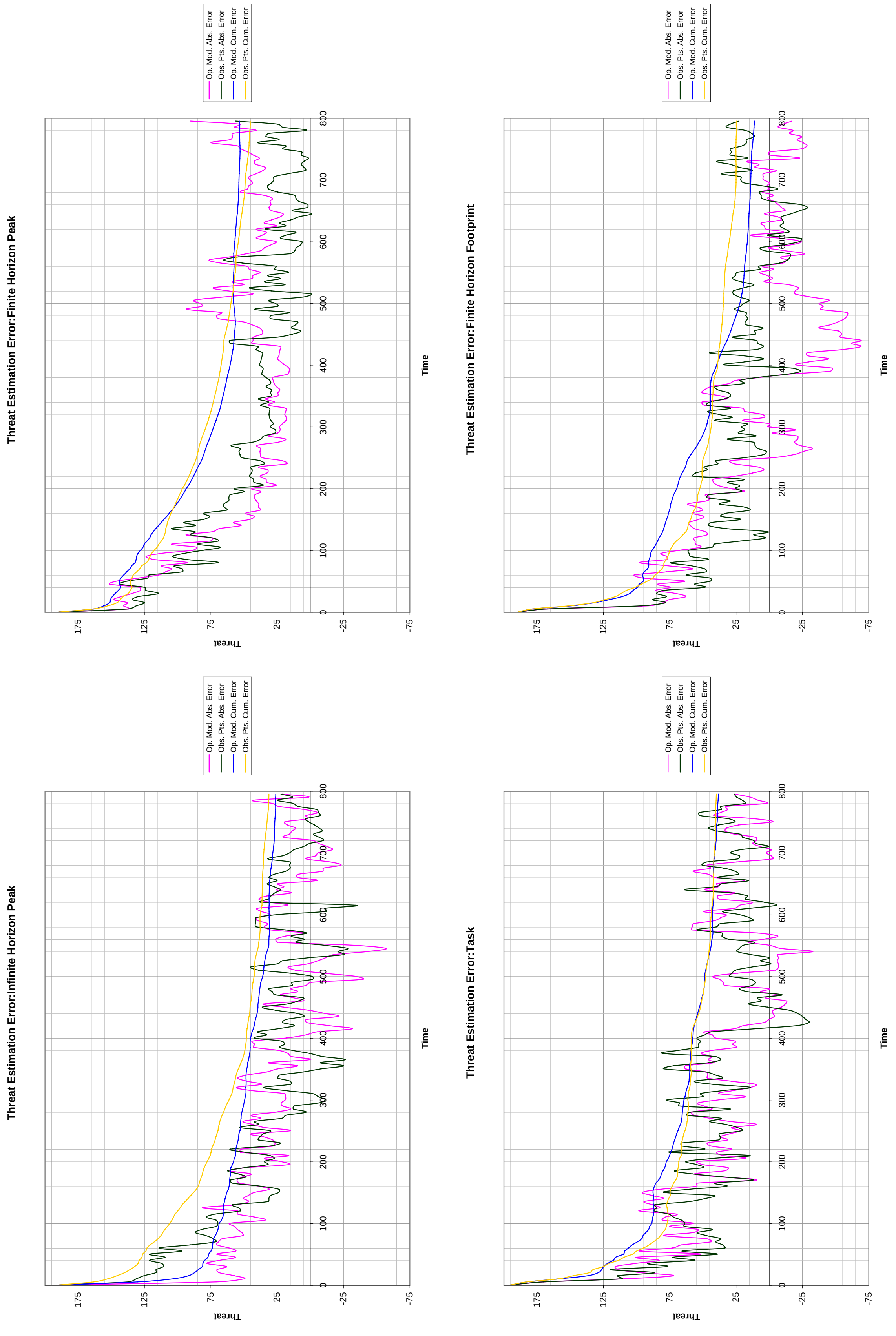


Figure 6.26: True and Estimated Threat for the Observation Points situation for each control scheme.

6.4.7 Mission: External Observations

In the External Observations scenario a number of target observations are provided to the system from an external source as outlined in Table 6.9. The observations take the form of vehicle type and location at a given time and are fused into the belief state in the same update cycle as UAV observations.

As in the Observation Points Scenario in Section 6.4.5, it can be seen from Figure 6.27 that changes in the system's estimated threat due to additional observations is minimal. Two of the external observations are of targets that contribute little to the overall threat while the observation of the Red Force vehicle, Target 1, occurs after it is already being tracked by all control schemes.

The threat error plots in Figure 6.28, show that the threat estimation performance for all control schemes, except Task, result on poorer performance compared to the Operator Modified scenario. This result is result is counter intuitive, as the external observations add information to the system and would be expected to aid threat estimation. With the addition of information in the Observation Points scenario also causing poorer performance in 2 cases, this indicates that information addition may result in poorer performance when the metric used for utility is not solely based on information gain. This is taken up further in Chapter 7.

As for Observation Points, the four external observations fused into the belief state have relatively little impact on the difference between the True Threat and Estimated Threat.

All 4 control schemes detect Target 1 prior to the external observation. As a result from Table 6.14, the most significant change in detection/classification times is in classification time. The external observation of Target 1 at 120s does however result in target classification by 125.5s for FHP and IHP.

6.4.8 External Observations Summary

It has been shown that external observations of the correct format can be seamlessly integrated into the system, with the information adding to the belief state. However it was shown that these changes to not necessarily result in a performance improvement.

Table 6.14: Targets Tracked - External Observations Scenario 2

	FHP BSL	**FHP EXTO**	**FHF BSL**	**FHF EXTO**	**IHP BSL**	**IHP EXTO**	**Task BSL**	**Task EXTO**
Targets Found	6	8	10	10	10	10	10	10
Time to Find (s)								
1	25.5	25.5	25.5	25.5	5.5	5.5	5.5	5.5
2	-	500.5	10.5	10.5	5.5	5.5	30.5	35.5
3	30.5	30.5	100.5	100.5	180.5	320.5	50.5	50.5
4	-	515.5	45.5	45.5	5.5	5.5	5.5	5.5
5	130.5	240.5	380.5	385.5	115.5	385.5	5.5	5.5
6	5.5	5.5	35.5	35.5	5.5	5.5	5.5	5.5
7	240.5	270.5	15.5	15.5	10.5	10.5	10.5	10.5
8	-	-	10.5	10.5	5.5	5.5	5.5	5.5
9	-	410.5	350.5	345.5	5.5	5.5	5.5	5.5
10	135.5	-	10.5	10.5	10.5	10.5	5.5	5.5
Targets Classified	2	2	9	9	10	10	10	10
Time to Classify (s)								
1	-	125.5	120.5	120.5	500.5	125.5	210.5	100.5
2	-	-	350.5	500.5	130.5	125.5	75.5	60.5
3	55.5	55.5	780.5	420.5	250.5	395.5	350.5	100.5
4	-	-	195.5	205.5	440.5	80.5	55.5	55.5
5	-	720.5	-	-	380.5	485.5	525.5	420.5
6	-	-	115.5	120.5	285.5	240.5	315.5	375.5
7	445.5	-	45.5	45.5	80.5	190.5	200.5	180.5
8	-	-	130.5	130.5	135.5	85.5	45.5	55.5
9	-	-	450.5	675.5	105.5	110.5	75.5	50.5
10	-	-	120.5	120.5	460.5	170.5	80.5	130.5

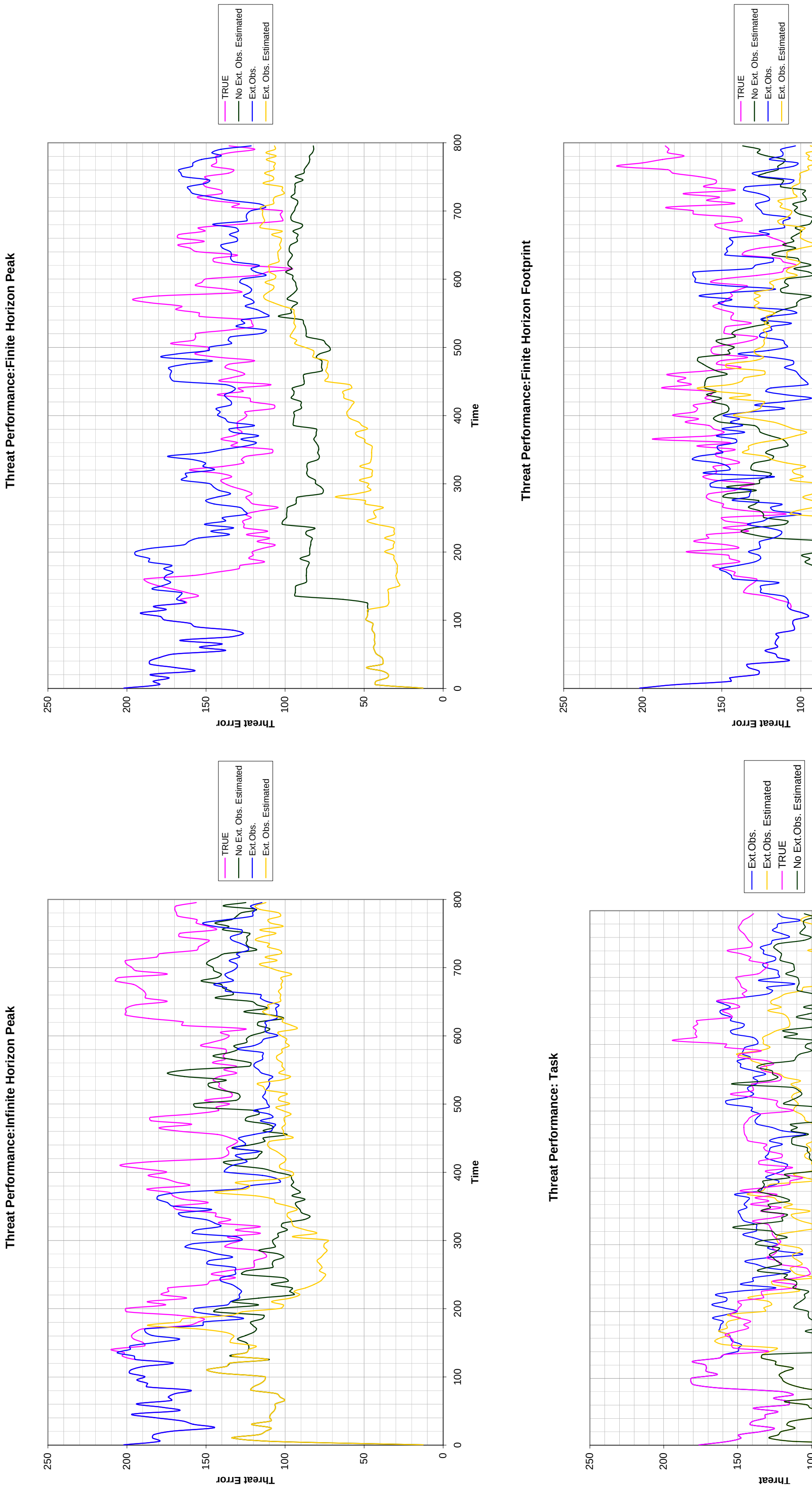


Figure 6.27: True and Estimated Threat for the External Observations situation.

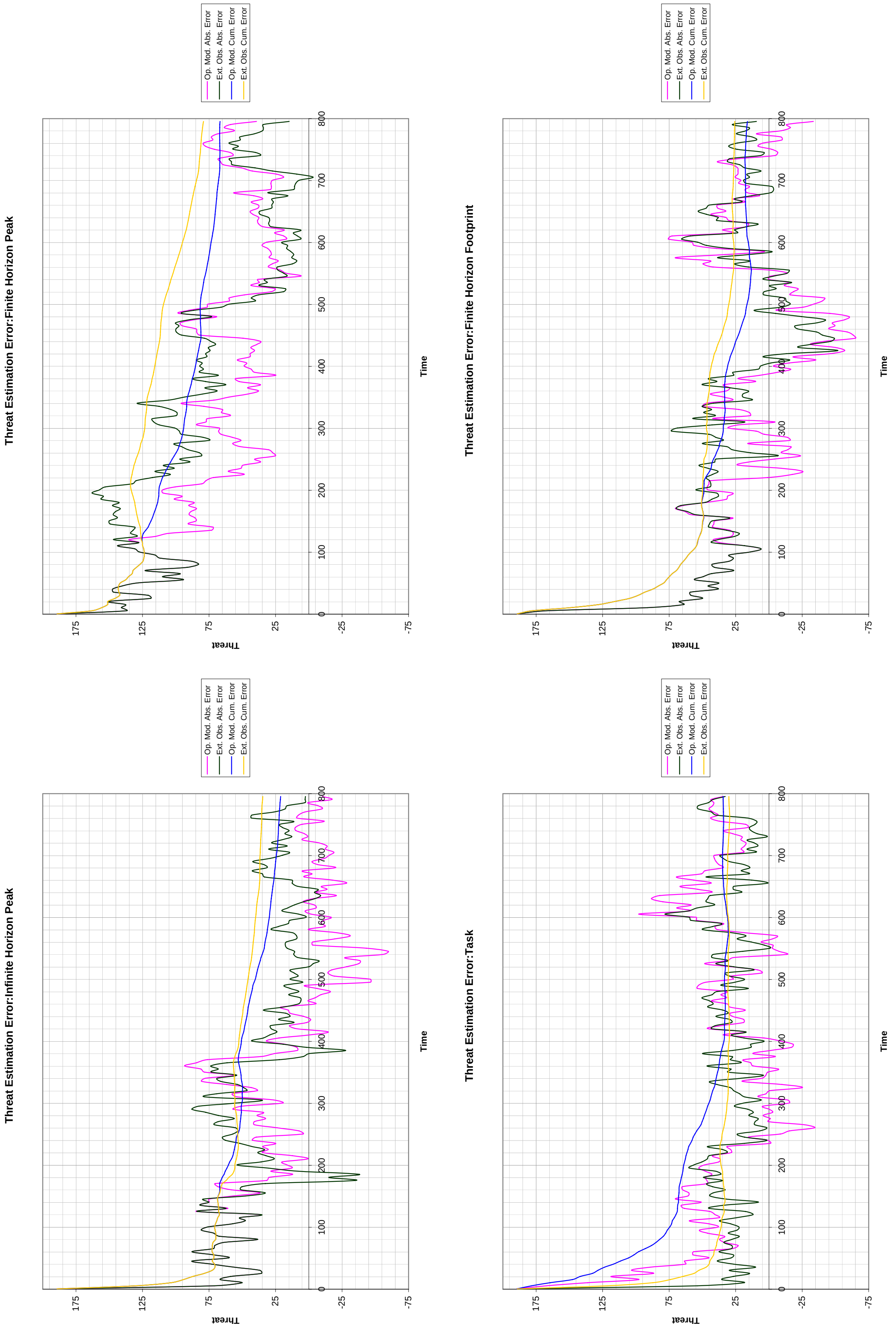


Figure 6.28: True and Estimated Threat for the External Observations situation for each control scheme.

Chapter 7

Conclusions and Future Work

7.1 Architecture

A flexible framework for human-robot cooperation has been implemented and demonstrated in a simulation environment. The basic architecture has been shown to be adaptable, via module substitution, to a variety of control concepts. Two different control schemes, in information value and task based control, have been demonstrated using the same basic architecture.

The implementation of the cooperative architecture has been shown to allow human judgement of the wider mission to be captured and fed into the system. This capturing of human situational evaluation has been demonstrated both as an *a priori* input and as an online dynamic process.

The architecture allows for the use of different sensor types and platforms and and has been demonstrated with sensor input from sources external to the system.

7.2 Information Value Based Control

A consistent formulation of information value, based on information theoretic concepts that takes into account wider mission objectives and the current information state, has been presented. This has been demonstrated as a valid basis for controlling a number of heterogeneous sensor platforms for an information gathering task.

It has also been shown that this concept of control can be used to capture some of the wider situational context and operator knowledge. Further, any changes to the situation are quickly assimilated into the belief state and the behaviour of the system modified.

7.3 Control Schemes

Four control schemes were demonstrated in simulation, with 3 using the information value based control discussed in Chapter 3. The fourth scheme utilised an identical valuation system but used a platform to task allocation method of control.

A number of different metrics were used to evaluate the performance of each control type:

- Overall estimation of the true threat state
- Maintenance of Red Force vehicle tracks
- Detection and Classification times
- Search performance

With respect to these metrics the performance ranking of the control schemes was significantly different depending on the metric used. FHF and IHP controllers performed best against the detection and classification metric, Task in threat estimation, Task and FHP for tracking and FHP for searching.

The Task controller explicitly assigns a ‘track’ tasking to a UAV. As a result, high threat targets are more likely to be reacquired and observed once in track. In the case of FHP, the limited area considered in the path planning loop results in a tendency to remain in the vicinity of a high threat target once detected. This same characteristic is responsible for the FHP search performance. As a weighted measure, once the UAV is in an area considered to be vulnerable the control loop will tend to continues searching the local area. However this becomes a liability when looking at detection/classification performance as the controller is reluctant to move the UAV to new lower threat areas.

The increased area of the map considered by the FHF controller overcomes the deficiencies in the FHP controller for detection and classification. However the weighted search performance suffers as a result.

It has been shown that small changes to the control scheme can significantly affect the performance of the system and that the resulting changes may be either desirable or undesirable depending on the performance metric used. Thus when this system is run, both the mission and the control scheme used need consideration.

7.4 Scenarios

Three scenarios were used to demonstrate the ability of the architecture adapt to changes in situation and accept external information. Where the operator was able to modify mission parameters on the fly to reflect a new situation, performance for all control schemes improved significantly.

In the scenarios where external information was added to the system, via fixed sensors and target observations, the performance benefit varied depending on the control scheme used. Additional information did not necessarily improve performance. This indicates a controller-belief interaction that cannot be described as ‘more information equals better performance’ and the requires further investigation.

7.5 Future Work

The framework for human-robot cooperation has been shown to be a useful approach to tackling the problem of single operator, multiple platform control for an information gathering mission. However within the scope of this thesis it has been impossible to fully characterise the system developed. Application of the framework to different environments, tasks where information is gathered in support of other activities and tasks where the objective is not defined in terms of information states is left as future work.

Running the entire simulation on a single CPU lead to severe limitations in computational power. The nature of the road network and tracker makes the system complex even with 10 targets. When combined with the computational resources required to run multiple autopilots, ground vehicle models and motion models, the simulation quickly bogs down on a single processor. By distributing modules to different CPUs as would be done in a real system, such as with a CPU per autopilot, larger more complex scenarios may be investigated. This is again left to future work.

The results for the 2 scenarios in which external information is added to the system pose a number of questions for future work. The beahviour observed indicates that additional information may harm the system's performance, depending on the control scheme adopted. This runs counter to the principle of using information gain as utility, though may reinforce the importance or valuing information. To properly examine the behaviour observed, it needs to be determined whether it is specific to the scenario used as an example or whether it is a trend more widely observed. Changes should include static sensor placement and observation area, additional targets and sensors, longer simulation time, UAVs flying fixed patterns and additional external target observations. This too is left to future work.

Appendix A

Common Simulation Components and Modules

A.1 Sensors

The sensor component contains all the functions and data required to generate observations for a single simulated sensor. Each sensor object in the simulation is associated to a single UAV platform and is updated with new position and pose information each time the platform updates its own position. On being triggered to generate an observation, the area currently visible to the sensor, or footprint, is found from the current state of the platform X and the fixed sensor *Field Of View* (FOV) parameters. The bounds of the footprint are passed to Search state component for fusion of the null observation. If the true position of a ground vehicle is within the current footprint a probabilistic target observation is created and passed to the Tracker and Classifier components for fusion.

A.1.1 Sensor Footprint

The sensor footprint is determined by a coordinate transformation from the sensor frame to the world frame. The sensor frame is defined relative to the UAV frame, which in turn is known relative to the world frame. Sensors are fixed to the platform, simplifying the transformation.

The body frame of the UAV is defined relative to the fixed world frame via origin x,y,z nominally attached to the platform *Centre of Gravity* (CG). The orientation of the UAV body, b, relative to the world frame, w, is given by the transformation matrix shown in Equation A.1 where $s_{(.)}$ is sin(.) and $c_{(.)}$ is cos(.).

$$C_b^w = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi s_\phi \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix} \tag{A.1}$$

- Subscript b represents the body frame.
- Superscript w the world frame.
- ψ yaw angle of the platform.
- θ pitch angle of the platform.
- ϕ roll, or bank angle of the platform.

The transformation matrix for world frame to the UAV frame is the matrix transpose shown in Equation A.2.

$$C_w^b = C_b^{w^T} \tag{A.2}$$

A coordinate in the body frame can be transformed to world frame coordinates via the transformation shown in Equation A.3.

$$P_t^w = C_b^w P_t^b \tag{A.3}$$

- Subscript t represents the target frame, which is the same as the world frame.
- P is the 3-dimensional coordinate vector.

Target observations are made in the sensor frame. As the sensor frame may have a different orientation to the UAV body frame, a transformation to the body frame may be required. The origin of both the sensor frame and body are both considered as nominally the platform CG or $P_t^b = P_t^s$. All sensors are fixed relative to platform and are constrained to look in a single direction. Thus the transformation matrix C_s^b is the same as C_b^w shown in equation

A.1 with fixed substitutions as shown in Equation A.4. The *forward*, *down*, *half left/right* and *left/right* sensor orientations are shown visually in Appendix C with the respective vectors shown in Equation A.4.

$$\begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{\pi}{2} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -\frac{\pi}{4} \\ 0 \end{bmatrix}, \begin{bmatrix} \pm\frac{\pi}{4} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \pm\frac{\pi}{2} \\ 0 \\ 0 \end{bmatrix} \tag{A.4}$$

The complete transformation of observed coordinates to world coordinates is shown in Equation A.5 where subscript s refers to the sensor frame.

$$P_t^w = C_b^w C_s^b P_t^b \tag{A.5}$$

In addition to dependance on C_b^w and P_b^w, the sensor footprint is dependent on the sensor FOV and location of the ground plane. The FOV is defined by a horizontal and vertical angle designated as α_h and α_v respectively. The ground plane is designated by Z_g.

The footprint is determined by projecting a unit vector from the sensor origin to each corner of the sensor's FOV. The azimuth and elevation of each unit vector can be determined as shown below.

$$azimuth = \pm\frac{\alpha_h}{2} \quad elevation = \pm\arctan(\tan\frac{\alpha_v}{2}\cos\left(azimuth\right)) \tag{A.6}$$

The FOV transformation matrix, or unit vector transformation matrix C_f^s, for each corner can be determined by substituting the relevant azimuth and elevation into Equation A.1. The subscript f identifies the transformation matrix as referring to a single corner of the sensor's FOV. The unit vector, n_{fi}^w, for each corner can be determined in the world frame as shown in Equations A.7 and A.8.

$$n_{fi}^w = C_w^b C_b^s C_s^{fi} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{A.7}$$

– i is the index of the unit vector, for a rectangular FOV $i = 1 \ldots 4$.

- $\psi = azimuth$
- $\theta = elevation$
- $\phi = 0$

To transform a coordinate or unit vector defined in the sensor frame to the world frame the transformation below is used.

$$P_s^w = P_b^w + C_w^b P_s^b \tag{A.8}$$

The sensor footprint can then be mapped onto the ground by projecting each unit vector and determining the intersection with the ground plane. Solving the parametric equations shown in Equation A.9 results in the *x,y* coordinates of the footprint corners in the world frame.

$$\begin{aligned} x &= x_0 + at \\ y &= y_0 + bt \end{aligned} \tag{A.9}$$

Given the ground plane $z = Z_g$, the substitution terms in Equation A.9 can be obtained using Equations A.10 to A.12.

$$P_s^w = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{A.10}$$

$$n_{fi}^w = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{A.11}$$

$$t = \frac{Z_g - z_0}{c} \tag{A.12}$$

The target detection ability of a sensor and the probability is dynamically affected by the pixel density $pixel/m^2$. The instantaneous pixel density at a point within the footprint can be determined using the method outlined in Equations A.9 to A.12 and dividing the projected FOV edges into n equidistant sections according to the resolution. If the sensor pixels are nominally square then the pixel density, μ can be found from Equation A.13.

$$\mu = \frac{\bar{\alpha}_h}{n_h} = \frac{\bar{\alpha}_v}{n_v} \tag{A.13}$$

Taking into account the sensor height and pose, both of which are coupled to the UAV height and pose, the angle between a unit vector anywhere in the footprint, α can be calculated. Using the same notation from Equations A.10 and A.11 the pixel density κ can be found via the Equation A.15.

$$\alpha = \arctan \frac{\sqrt{a^2 + b^2}}{c} \tag{A.14}$$

$$\kappa = \frac{1}{((Z_g - z_0)\sec^2(\alpha)\mu)^2} \tag{A.15}$$

The computational load of determining κ and a subsequent $P(z^t|\mathbf{x}^t)$ for each point in the sensor's footprint resulted in the simplification for κ shown in Equation A.16 being adopted. In order to speed the computational process this approximation is applied over the whole footprint.

$$\kappa = \frac{n_p}{\left(z_0^2 \frac{\tan(\alpha_h)\tan(\alpha_v)}{2}\right)} \tag{A.16}$$

– n_p is the number of pixels.

– h is the height of the UAV/Sensor above the ground plane Z_g

A.1.2 Detection

The ability of the sensor to detect a particular ground vehicle type is a function of both the sensor's ability to 'see' the ground vehicle type and the number of resolved pixels across the target. This later component is encapsulated in the Johnson criteria, J, which represents the number of cycles or line pairs across the target.

$$J = \frac{\kappa\sqrt{\beta}}{2} \tag{A.17}$$

- J Johnson Criteria
- κ is the pixel density (see Equation A.16).
- β is the projected target area in m^2

Empirical studies show that for a detection confidence of 0.9 in a cluttered environment, a J value of between 4 and 17 is required [6]. These values relate to *conspicuous* and *non-conspicuous* targets respectively.

Due to the computational load imposed during simulation, t_a is kept constant and a detectability function determined *a priori* for each sensor type. This is stored as a look up table, relating J to $P(z^t|\mathbf{x}^t)$ values, which are then used to determine if the ground vehicle within the sensor's footprint is detected. The table is shown in Appendix C.

A.1.3 Localisation

The sensor's ability to estimate a detected ground vehicles location is represented by a two dimensional Gaussian as shown in Equation A.18.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{A.18}$$

To determine the location of the detected target, the true ground vehicle x and y are used along with the σ defined in the model for the sensor type. An x and y value for the detection is then selected with probability G from the distribution centered on the true target location.

A.1.4 Classification

The sensor's ability to estimate a detected ground vehicle's type is represented by the distribution $P(z^c|\mathbf{x}^c)$. Each sensor type has a unique discrimination function and these are shown in Appendix C. To determine the classification of a detected ground vehicle, a classification is selected with probability $P(z^c|\mathbf{x}^c)$ conditioned on the true vehicle state.

A.1.5 No Detect

Each time an observation is made the footprint as calculated in A.1.1 is passed to the Search State and Tracker for fusion as the ‘null observation’. As observations of ground vehicles that fall within the footprint are handled separately, the null observation is the sensor estimate of where no ground vehicles are present.

A.2 Platforms

The UAV platform component contains all the data and functions to simulate a single UAV sensor platform. Each platform is associated with at least one sensor for which it provides state updates. The UAV contains a number of subcomponents:

- The Guidance component takes waypoints and generates a desired height, bank angle and speed command. The guidance loop is performed in simulation at 10Hz.
- The Autopilot component takes the guidance command and generates control inputs for angular rates and throttle. This Autopilot loop is performed in simulation at 20Hz
- The Dynamic Model component takes the control inputs, and based on the platform dynamic model generates a new state X for the UAV. The Dynamic Model loop is performed in simulation at 20Hz with the new state at the end of each loop passed back to the Autopilot component.

Two different UAV platform types are available in the simulation, as outlined in Appendix C. Both types use the same subcomponents and functions, however limits and gains will be unique to the type.

A.2.1 Guidance

The Guidance component takes either a single waypoint or list of waypoints, defined as a three dimensional point, and the current UAV state X and generates a guidance command shown below.

$$guidance = \begin{bmatrix} v_d \\ z_d \\ \phi_d \end{bmatrix} \tag{A.19}$$

- v_d is the desired airspeed, which is supplied as part of the waypoint.
- z_d is the desired height, which is the z component of the waypoint.
- ϕ_d is the desired bank angle.

Each time the guidance loop is called, the distance to the current way point is checked. If the waypoint has been reached a check is made for a new waypoint. If none is available the guidance mode is switched to 'orbit' and a constant $\dot{v}$, z_d and ϕ_d applied until a new waypoint becomes available. If the the current waypoint has not been reached, a *Proportional Integral* (PI) control and ϕ limit is applied to the heading error to determine ϕ_d.

$$\phi_d(s) = k_\phi(\psi_w - \psi_c) + \frac{2(\psi_w - \psi_c)}{s} \tag{A.20}$$

- k_ϕ is the proportional controller gain determined from a lookup table based on the UAV current speed.
- ψ_w is the heading to the active waypoint from the current UAV position.
- ψ_c is the current UAV heading.

A.2.2 Autopilot

The Autopilot component takes guidance commands as described in Section A.2.1 and generates an autopilot command which defines the control rates to be applied to the platform.

$$autopilot = \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{v} \end{bmatrix} \tag{A.21}$$

The guidance command is passed through height, speed, bank and pitch control functions in order to generate the output in Equation A.21. For simulation it is assumed that all turns will be coordinated, hence the Dynamic Model omits side slip effects and therefore no rudder control, $\dot{\psi}$, value is calculated.

Height Control

The height control function is in two stages, first a demanded *Rate Of Climb* (ROC) is determined using a proportional controller via Equation A.22, then this is converted to a demanded pitch or climb angle, θ using Equation A.25.

$$\dot{z}_d = k_h(z_c - z_d) \tag{A.22}$$

- $\dot{z}_d$ is the desired ROC
- k_h is the proportional controller gain determined from a lookup table based on UAV type and current speed.
- z_c current height above the ground.
- z_d desired height above the ground.

The desired flight path angle at the current speed is then calculated as shown below.

$$\gamma_d = \left(\frac{\dot{z}_c - \dot{z}_d}{v}\right) \tag{A.23}$$

$$\gamma_e = \gamma_d - \gamma_c \tag{A.24}$$

- γ_d is the flight path angle need to achieve the desired ROC from Equation A.22
- v is the current airspeed.
- z_c current height above the ground.
- z_d desired height above the ground.
- γ_e is the flight path angle error.

As the simplified Dynamic Model does not account for effects such as side slip or skidding the current pitch angle is the current flight path angle, $\theta_c = \gamma_c$. The desired pitch angle, θ_d is then calculated using the γ values, before being passed through a limiting function as shown in Equation A.25.

$$\theta_d(s) = \gamma_d + \gamma_e + \frac{0.1\gamma_e}{s} \tag{A.25}$$

Pitch Control

The pitch controller uses the demanded pitch angle from the height controller to generate a $\dot{\theta}$ command using the PI controller in Equation A.26.

$$\dot{\theta}(s) = k_\theta(\theta_d - \theta_c) + \frac{\theta_d - \theta_c}{s} \tag{A.26}$$

- $\dot{\theta}$ is the applied pitch rate.
- k_θ is the pitch gain determined from a lookup table based on UAV type and current speed.
- $\theta_d - \theta_c$ is the pitch angle error.

Speed Control

The speed control control function is a lead-lag compensated error feed back loop. The $\dot{v}$ error is passed through the lead-lag filter in Equation A.27.

$$G(s) = \frac{s + 0.1}{s + 0.01} \tag{A.27}$$

The controller gain k_v is determined from a lookup table based on UAV type and current speed allowing the desired $\dot{v}$ to be found via Equation A.28.

$$\dot{v} = k_v G(s) \tag{A.28}$$

Bank Angle Control

The bank angle controller is a gain scheduled proportional controller with gain k_p determined from a lookup table based on UAV type and current speed. The desired roll rate is found:

$$\dot{\phi} = k_p(\phi_d - \phi_c) \tag{A.29}$$

- $\dot{\phi}$ is the applied roll rate.
- k_p is the controller gain.
- $\phi_d - \phi_c$ is the bank angle error.

A.2.3 Dynamic Model

The Dynamic Model takes the applied control inputs from the Autopilot and the current UAV state and applies a fourth order Runge-Kutta solution to the 6-DoF *Ordinary Differential Equation* (ODE) of state. The general Runge-Kutta solution for the initial UAV state X_n is found using Equation A.30.

$$X_{n+1} = X_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)dt \tag{A.30}$$

$$\begin{aligned} k_1 &= X_n dt \\ k_2 &= X_n + 0.5k_1 dt \\ k_3 &= X_n + 0.5k_2 dt \\ k_4 &= X_n + k_3 dt \end{aligned} \tag{A.31}$$

Where dt is the incremental time step, for simulation purposed a constant value of 0.1 was used.

The control input from the Autopilot in Equation A.21 is assumed constant over the time step dt. As the applied $\dot{v}$, $\dot{\theta}$ and $\dot{\phi}$ values are supplied the remaining velocities can be calculated and the k values determined via:

$$\begin{aligned} \dot{x} &= v_c \cos\theta_c \cos\psi_c \\ \dot{y} &= v_c \cos\theta_c \sin\psi_c \\ \dot{z} &= v_c \tan\theta_c \\ \dot{\psi} &= \tfrac{\tan\phi_c\left(v_c \tan\dot{\theta} + 9.81\right)}{v_c} \end{aligned} \tag{A.32}$$

The output state $X_n + 1$ is then applied as the current state X_n of the platform for the next Autopilot/Dynamic Model loop.

A.3 Ground Vehicles

The Ground Vehicle component contains all the functions and data required to simulate a single ground vehicle. Each Ground Vehicle contains Guidance and Motion Model components which control position, along with static characteristics which are unique to each type. The various vehicle type are outlined in Appendix C.

The target type, size and conspicuousness may be used in conjunction with Equations A.17 and the table in Appendix C to determine the detection probability. However due to the computational load for running the entire simulation on a single processor, the properties size and conspicuousness properties were not used for the simulation results shown in Section 6.4.

A.3.1 Motion Model

Ground vehicles are restricted to moving along the urban road network shown in Appendix C. The network is discretised into nodes, where each node is nominally placed at a $1m$ interval along the road. Each node is given a unique identifier with a connectivity map indicating which nodes are connected and in what direction. These directions are shown in Appendix C.

During each Motion Model loop the ground vehicle centroid is moved forward along the network, by the number of nodes specified by the time interval and the vehicle velocity. The velocity for each vehicle type is give in Appendix C and the Motion Model loop is run at $20Hz$.

A.3.2 Guidance

The Guidance subcomponent has two different modes of operation, *random* and *path.* If the current mode is set to random, on reaching an intersection node a new direction is chosen randomly subject to the propagation filter shown in Appendix C.

The Guidance subcomponent can hold a list of waypoints. A path from the current position to the next waypoint in the list can be found using the A^* algorithm. The output of the $A*$ algorithm is a set of intersection nodes and the subsequent path nodes, which are stored. If the current Guidance mode is set to path, then on reaching an intersection the new direction is defined by the stored next path node. If the waypoint is reached and there are waypoints yet to be visited a new path to the next waypoint is found, however if no unvisited waypoints remain, the Guidance mode is forced to random.

The A^* algorithm is the recursive application of the function in Equation A.33

$$f'(n) = g(n) + h'(n) \tag{A.33}$$

- $g(n)$ is the total distance travelled from the path start point to the current location.
- $h'(n)$ is the estimated distance from the current position to the waypoint determined through a heuristic function.
- $f'(n)$ is the current estimate of the shortest path.

The final path $f(n)$ is equal to $f'(n)$ once the algorithm has finished. The heuristic used for estimating the distance to the waypoint is shown in Equation A.34.

$$\begin{aligned} h^{'}_{\bar{x}>\bar{y}}(n) &= |\bar{x}| + 0.4\,|\bar{y}| \\ h^{'}_{\bar{x}<\bar{y}}(n) &= 1.4\,|\bar{y}| \end{aligned} \tag{A.34}$$

- $\bar{x}$ is the distance from the current position to the waypoint along the x axis.
- $\bar{y}$ is the distance from the current position to the waypoint along the y axis.

A.4 Tracker

The Target Tracking Module is part of the Level 2 Data Fusion functionality and is responsible for providing the current b_c and future belief b_f of the location of all ground vehicles detected. The complete track component of the belief B^t is a vector of q probability distributions, where q is the number of ground vehicles that have been observed. If a tracked ground vehicle's position is given by the variable $\mathbf{x^t}$ the belief state is:

$$B^t = [P(\mathbf{x}_1^t) \ldots P(\mathbf{x}_q^t)] \tag{A.35}$$

The distribution $P(\mathbf{x}^t)$ represents the probability of the tracked target being at any given node in the map. The belief of the target's position given all observations made up to time k can be found using Equation A.36.

$$b_k(\mathbf{x}^t) = P(\mathbf{x}^t | z_k^t, z_{k-1}^t, \ldots, z_1) \tag{A.36}$$

Assuming the ground vehicle's motion is Markovian, Equation A.36 can be calculated recursively using Bayes rule as in Equation A.37 .

$$\begin{aligned} b_k(\mathbf{x}^t) &= P(\mathbf{x}_k^t|z_{1:k}^t) \\ &= KP(z_k^t|\mathbf{x}_k^t)P(\mathbf{x}_k^t|x_{k-1}, z_{1:k-1}^t) \\ &= Kp(z_k^t|\mathbf{x}_k^t)P(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)P(\mathbf{x}_{k-1}^t|z_{1:k-1}^t) \end{aligned} \tag{A.37}$$

Where the prior $P(\mathbf{x}_{k-1}^t|z_{1:k-1}^t)$ is the posterior from the previous recursion and K is a normalising constant.

A.4.1 Prediction

As the ground vehicle will not necessarily be observed at each k, a prediction of the state $\mathbf{x}^t$ provides the current belief estimate when no observation is available and may be used to generate the future belief estimate. The propagation of $P(\mathbf{x}^t)$ is governed by the Chapman-Kolmogorov and Fokker-Planck-Kolmogorov equations, for the continuous and discrete cases respectively. The predicted belief state can be found:

$$\begin{aligned} b_{k+1}(\mathbf{x}^t) &= P(\mathbf{x}_{k+1}^t|z_{1:k}^t) \\ &= K\int P(\mathbf{x}_{k+1}^t|\mathbf{x}_k^t)P(\mathbf{x}_k^t|z_{1:k}^t)d\mathbf{x}_k \end{aligned} \tag{A.38}$$

Where $P(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is the target's motion model. The setup of the module allows prediction of $P(\mathbf{x}^t)$ to the UAV planning horizon.

A.4.2 Update

When an observation is available, the belief is updated using Bayes theorem in Equation A.39.

$$\begin{aligned} b_{k+1}(\mathbf{x}^t) &= P(\mathbf{x}^t_{k+1}|z^t_{1:k+1}) \\ &= KP(z^t_{k+1}|\mathbf{x}^t_{k+1})P(\mathbf{x}^t_{k+1}|z^t_{1:k}) \end{aligned} \tag{A.39}$$

If observations are available from multiple sensors and each is conditionally independent given the true state of $\mathbf{x}^t$, then $z^t_{k+1} = \left[z^1_{k+1} \dots z^r_{k+1}\right]$. The update, Equation A.39 can then be rewritten as in Equation A.40.

$$\begin{aligned} b_{k+1}(\mathbf{x}^t) &= P(\mathbf{x}^t_{k+1}|z^t_{1:k+1}) \\ &= KP(\mathbf{x}^t_{k+1}|z^t_{1:k})\prod_{i=1}^{r} P(z^{t_i}_{k+1}|\mathbf{x}^t_{k+1}) \end{aligned} \tag{A.40}$$

A.4.3 The Particle Filter

Initialisation

The Track prediction and update are implemented in a multi-hypothesis framework, using a particle filter approach. On observation of a previously unknown ground vehicle a new track is created to represent the distribution $P(\mathbf{x}^t_q)$. The track consists of a fixed number of particles where each particle represents a hypothesis of the true target location $\mathbf{x}^t_q$ and type $\mathbf{x}^c_q$. The initial particle locations are assigned according to the sensor's localisation model as discussed in Section A.1.3 and the classification model discussed in Section A.1.4. The number of particles assigned to a point x is found from Equation A.41.

$$|h_{x,m}| = G(x_x, x_y)P(\mathbf{x}^c_m)|\mathbb{H}|\forall m \in \mathbb{C} \tag{A.41}$$

Propagation

The belief of target classification will be constant with respect to time, hence each particle is propagated according to Equation A.38, where the motion model used is model for the type assigned to the particle. At road junctions, the direction for propagation is chosen randomly, subject to the direction filter shown in Appendix C.

Update

When an observation becomes available of a ground vehicle in track, a new set of particles is created as for initialisation where $|h| \ll |\mathbb{H}|$. Then each original particle is updated according to Equation A.39 using the joint probabilities $P(z_{k+1}|\mathbf{x}^t_{k+1}, \mathbf{x}^c)$ and $P(\mathbf{x}^t_{k+1}, \mathbf{x}^c|z_{1:k})$.

The new particle set is normalised and the $|h|$ particles with the lowest $P(\mathbf{x}^t, \mathbf{x}^c)$ are removed from the set $|H|$ before renormalisation.

When the null observation corresponding to a particles position becomes available the new probability for the hypothesis is determined by Equation A.39 where the sensor model applied is $P(z^s|\mathbf{x}^s) = 1 - P(z^t|\mathbf{x}^t)$. If the probability falls below a threshold it is set to zero and removed from the set to be propagated in the net time step. The particle probabilities are then renormalised.

A.5 Classifier

The Target Classifier Module is part of the Level 2 Data Fusion functionality and is responsible for providing the classification belief of all ground vehicles detected. The classify component of B^c is a vector of q probability distributions, where q is the number of targets currently being tracked by the Target Tracking Module.

$$B^c = [P(\mathbf{x}^c_1) \ldots P(\mathbf{x}^c_q)] \tag{A.42}$$

The distribution $P(\mathbf{x}^c)$ is the probability of the true ground vehicle type taken over all possible types. The classification update is performed via a recursive application of the Bayes update equation:

$$P(\mathbf{x}^c|z^c) = \frac{P(z^c|\mathbf{x}^c)P(x)}{P(z^c)} \tag{A.43}$$

Given the observation z^c is received by the Classifier the classification estimate can be obtained from Equation A.50.

$$P(\mathbf{x}^c|z^c_k) = KP(z^c_k|\mathbf{x}^c)P(\mathbf{x}^c) \tag{A.44}$$

Where K is a normalising factor and $P(\mathbf{x}^c)$ is the estimate of the classification prior to the update. This is initially the uninformed distribution over all target types, while for all subsequent updates the posterior $P(\mathbf{x}^c|z_k^c)$ from the previous update is used as the new prior. The sensor model $P(z|\mathbf{x}^c)$ is known for each sensor type *a priori* and stored as a likelihood matrix. See Appendix C for details of the different sensor models.

If observations are available from multiple sensors within the same time step k and each is conditionally independent given the true state of $\mathbf{x}^c$, then $z_{k+1}^c = \left[z_{k+1}^1 \ldots z_{k+1}^r\right]$. The update equation can be written:

$$P(\mathbf{x}^c|z_k^c) = [P(z_k^c)^{-1} P(\mathbf{x}^c) \prod_{i=1}^{r} P(z_i^c|\mathbf{x}^c) \tag{A.45}$$

In the implemented Classifier and Tracker, correct observation to track is assumed. Hence the interaction between the Classifier and Track is limited to selection of the number of particles to assign to a particular motion model based on the classification probability.

A.5.1 Classification Sets

The ground vehicle types are grouped into predefined classification sets. These sets are used in constructing the mission for the task based control and for threat definition in value based control. Instead of assigning priority or threat to an individual type the operator may assign it to one of the sets and deal with the set as if it were a single type.

The probability of membership of set $\mathbb{C}$ is the sum of probabilities for each member of the set as shown in Equation A.46.

$$P(\mathbf{x}^{\mathbb{C}}) = \sum_{m} P(\mathbf{x}_m^c) \forall m \in \mathbb{C} \tag{A.46}$$

The predefined classification sets are outlined in Appendix C.

A.6 Search State

The Search component is part of the Level 2 Data Fusion functionality and is responsible for providing the current b_c and future belief b_f of the location of ground vehicles that have yet to be observed. The complete search component B^s is a probability map over all points within the area of operation.

$$B^s = \sum_{n=1}^{\mathbb{S}} P(\mathbf{x}_n^s) \tag{A.47}$$

In the urban security mission defined in Section 3.4 the number, or indeed existence, of ground vehicles in the area of operation is unknown *a priori*. This makes setting up the search distribution and assessing the initial of $P(\mathbf{x}^s)$ difficult, particularly in relation to finding appropriate priors. As a result the state of search task is represented as the distribution $P(\mathbf{x}^s)$ over the entire area of operation conditioned on the existence of a single previously unobserved ground vehicle. When a ground vehicle is detected it is immediately handed off to the Track and Classification components and a null observation passed to the Search State component.

A.6.1 Predict

As only a small section of the area of operation is observed at any time, the prediction of the state $\mathbf{x}^s$ provides the current belief estimate for the unobserved areas and may be used to generate the future belief estimate. The propagation of $P(\mathbf{x}^s)$ is nominally governed by the Chapman-Kolmogorov equation as for the track state in Equation A.38. However given no *a priori* information on location, number or type of vehicles, then each type and possible location needs to be propagated to time k. This is computationally intractable for the node network map, therefore a time constant information decay is applied according to Equation A.48 where τ is the decay constant.

$$P(\mathbf{x}_{k-1}^s | \mathbf{x}_k^s) = P(\mathbf{x}_k^s) \tau dt \tag{A.48}$$

A.6.2 Update

The classification update is performed via a recursive application of the Bayes update equation:

$$P(\mathbf{x}^s|z^s) = \frac{P(z^s|\mathbf{x}^s)P(x)}{P(z^s)} \tag{A.49}$$

Given the null target observation z^s is received the updated estimate can be obtained from Equation A.50.

$$P(\mathbf{x}^s|z^s_k) = KP(z^s_k|\mathbf{x}^s)P(\mathbf{x}^s) \tag{A.50}$$

Where K is a normalising factor and $P(\mathbf{x}^s)$ is the estimate of the classification prior to the update. This is initially the uninformed distribution over the area of operation, while for all subsequent updates the posterior $P(\mathbf{x}^s|z^s_k)$ from the previous update is used as the new prior. The value of $P(z^s|\mathbf{x}^c)$ is related to the detection probability outlined in Section A.1.2 by Equation A.51.

$$P(z^s|\mathbf{x}^s) = 1 - P(z^t|\mathbf{x}^t) \tag{A.51}$$

Appendix B

Task Based Control

B.1 Framework Components

B.1.1 Organization

The decision process performed at the Organization level can be summarized as the mapping:

$$\mathcal{O} : \mathcal{M}_c \times \mathcal{M}_d \times \mathbb{P} \times \mathbb{A}_h \times \mathbb{P}_h \rightarrow (\mathbb{A}, \mathbb{W}) \tag{B.1}$$

– $\mathcal{M}_c$ is the current state of the mission derived from the the world belief state B.

– $\mathcal{M}_d$ is the desired mission state derived from the operator input and the the structure of the mission M.

– $\mathbb{P}$ is the set of priorities associated with task types.

$$\mathbb{P} = \{p_q^t, p_q^c, p^s\} \forall q = 1 \ldots \mathrm{Q} \tag{B.2}$$

– p_q^t and p_q^c are the priority for tracking and classifying the q^{th} target type and p^s is the priority of search for the whole area of operation.

– $\mathbb{A}_h$ is set of additional tasks as a result of Level 3 data fusion.

– $\mathbb{P}_h$ is the set of priorities associated to tasks specified by the operator which for this implementation is the search of areas around protected assets.

$$\mathbb{P}_h = \{p_{\mathrm{a}}^{\mathbb{S}}, \ldots\} \tag{B.3}$$

- $p_{\mathrm{a}}^{\mathbb{S}}$ is priority for searching the sub area associated with the a$^{\mathrm{th}}$ asset.
- $\mathbb{A}$ is the unified set of tasks.
- $\mathbb{W}$ is the set of weights associated with the set $\mathbb{A}$.

B.1.2 Coordination

In line with the concept of IPDI, the *Coordination* level deals with a lower level of task abstraction than the *Organiser*. The first stage of the *Coordination* process is allocation of the task set, $\mathbb{A}$, to the available resources. This results in the allocated task set Υ

$$\mathcal{C} : \mathbb{W} \times \mathbb{A} \rightarrow \Upsilon \tag{B.4}$$

Certain tasks may be assigned to multiple platforms or sensors and may need to be coordinated in time and/or space. This is the second stage of the coordination process. For tasks which include parameters such as simultaneous arrival or observing a target from different angles, the time and space coordination may require multiple iterations of the path plan for each platform.

B.2 Modules

The task based control scheme is implemented in code as the modules shown in Figure B.1. These modules are discussed in the following sections.

B.3 Organiser

The Organiser performs the mapping shown in Equation B.1 in stages, with responsibility for the different stages divided between the Mission and Mission Planner modules.

- Interface with the operator to capture and store the mission parameters.
- Evaluate the current and desired mission states.
- Create the current task pool from the current belief and mission, adding any other tasks specified by the operator.
- Calculate a weighting for each task in the current task pool.

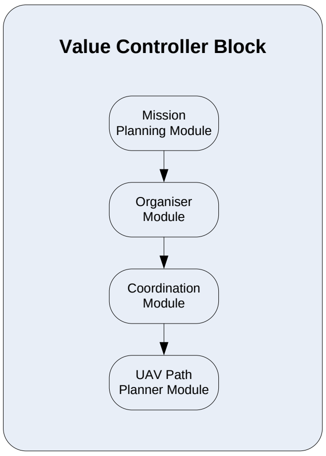


Figure B.1: Software modules for tasked based control.

B.4 Mission Module

The *Mission Module* provides the interface for the operator to enter and modify mission parameters. These parameters are maintained as the mission object M, which consists of three components corresponding to the search, track and classify objectives.

$$M = \left[\mathrm{M}^s, \mathrm{M}^t, \mathrm{M}^c\right] \tag{B.5}$$

Search M^s

The *Search* component of the mission consists of the operator defined parameters 'Priority' and 'Likelihood' for the *General Search* task and any *Area Search* tasks.

When entering the priority and likelihood parameters, the operator is able to specify whether they apply to the entire area of operation, $\mathbb{S}$ or a specific area S. In the security scenario, S will normally be an area surrounding a protected asset.

The priority term is assigned according to one of the descriptions shown in Equation B.6. A numerical value corresponding to the human level description is then automatically applied and stored in the mission object.

$$p^s = |\text{No Impact, Some Impact, Desirable, Highly Desirable, Mission Critical}| \tag{B.6}$$

It is desirable that the values corresponding to each descriptive level are assigned by a human with high level knowledge of the situation, or extracted from historical data. The corresponding numerical values are applied as either the linear set $1 \ldots 5$ or the logarithmic set $2^1 \ldots 2^5$.

The 'Likelihood' describes the desired maximum likelihood of the presence of an undetected target being present at any point in either S or $\mathbb{S}$.

Track M^t

The *Track* component consists of the operator defined parameters 'Priority' and 'Track Variance' for each ground vehicle type. The 'Priority' term is assigned, as for search, according to selected priority level shown in Equation B.6. The desired track variance in m, for the target type is entered directly by the operator

Classify M^c

The *Classification* component consists of the operator defined parameters 'Priority' and 'Classification Threshold' for each target type. The priority term is again assigned according to the levels in Equation B.6.

The classification threshold is entered directly by the operator. If the vehicle type probability of of a tracked target reaches this threshold the target is considered to have been classified.

B.5 Mission Planning Module

The *Mission Planning Module* generates a current mission state $\mathcal{M}_c$ based on the system belief B and a desired mission state $\mathcal{M}_d$ based on the parameters in the mission object M. These mission states are then mapped to a set of tasks and associated priorities as per equation B.1. From section B.4, the mission M consists of *search*, *track* and *classify* components.

$$M = f[\underbrace{(\sigma^2_{trk}|\mathbf{x}^c)}_{Track}, \underbrace{P_{\min}(\mathbf{x}^c)}_{Classification}, \underbrace{P_{\max}(\mathbf{x}^s|\mathbb{S}), P_{\max}(\mathbf{x}^s|S)}_{Search}] \tag{B.7}$$

Where the priority for each search, track or classify task is determined from the priority p applied by the operator from Equation B.6.

B.5.1 Current Mission State $\mathcal{M}_c$

The current mission state is defined by the Hellinger-Bhattacharya distance between the current distribution $P(\mathbf{x})$ and the naive or flat distribution. For two discrete distributions $P(\mathbf{x}^a_i)$ and $P(\mathbf{x}^b_i)$ of length i, the distance δ is fund using Equation B.8.

$$\delta\left(P(\mathbf{x}^a_i), P(\mathbf{x}^b_i)\right) = \left[\sum_i \left(P(\mathbf{x}^a_i)^{0.5} - P(\mathbf{x}^b_i)^{0.5}\right)^2\right]^{0.5} \tag{B.8}$$

The naive distribution is constant, hence δ is determined via the *Hellinger-Bhattacharya* (HB) affinity to allow the distributions to be handled separately. The HB affinity α for a discrete distribution of n possible states is shown in Equation B.9.

$$\alpha = 2\log\sum_{i=1}^{n}\sqrt{\frac{P(\mathbf{x}_i)}{n}} \tag{B.9}$$

The affinity in Equation B.9 has an associated HB distance shown below.

$$\delta_\alpha = 2(1 - e^{0.5\alpha}) \tag{B.10}$$

The distance between the distributions is then found using equation B.11.

$$\delta = \delta^a_\alpha - \delta^b_\alpha \tag{B.11}$$

Search

The state of general search component of the mission can be obtained by finding the HB distance of the $P(\mathbf{x}^s)$ from a uniform distribution of the same length using the Equations B.9 to B.11.

$$\mathcal{M}_c^s = \delta_\alpha^s - \delta_\alpha^U \tag{B.12}$$

Where δ^U is the HB distance of an uniform distribution with length $|\mathbb{S}|$.

The area search component of the current mission state is the set of the individual search states of all areas S specified by the operator.

$$\mathcal{M}_c^s = \left|\mathcal{M}_c^s(S)^i\right| \forall i = 1 \dots A \tag{B.13}$$

Where A is the set of search areas specified by the operator. The distribution associated with the points in S can be extracted from the main search map $P(\mathbf{x}^s)$ as $S \subset \mathbb{S}$.

$$\begin{aligned} \mathcal{M}_c^s(S) &= \delta_\alpha^s - \delta_\alpha^U \\ &= 2\left(1 - e^{\log \sum_{i=1}^{|S|} \sqrt{\frac{P(\mathbf{x}^{s \in S})}{|S|}}}\right) - \delta_\alpha^U \end{aligned} \tag{B.14}$$

Track

The track component of the current mission state is the set of individual track states of all ground vehicles being tracked by the system.

$$\mathcal{M}_c^t = \left|\mathcal{M}_c^t(q)\right| \forall q \in \mathrm{Q} \tag{B.15}$$

Where Q is the set of all ground vehicles in track. The track distribution for target q is obtained from $P(\mathbf{x}^t)$ with the distribution length $|\mathbb{S}|$. The uniform distribution for the tracker implemented consists of g points with $P(\mathbf{x}^t) = \frac{1}{g}$ where g is equal to the number of particles used by the tracker and $|\mathbb{S}| - g$ points where $P(\mathbf{x}^t) = 0$.

Classify

The classification component of the current mission state is the set of individual classification states of all ground vehicles being tracked.

$$\mathcal{M}_c^c = |\mathcal{M}_c^c(q)|\, \forall q \in \mathrm{Q} \tag{B.16}$$

Where Q is the set of all ground vehicles in track. The classification distribution for target q is obtained from $P(\mathbf{x}^c)$ and δ_α^U is the uniform distribution over all possible ground vehicle types.

B.5.2 Desired Mission State $\mathcal{M}_d$

The desired mission state is based on the operator preferences captured by the mission module. A desired distribution for the search, track and classify components is created according to the parameters in M. The HB distance of the resulting distribution from the naive equivalent is stored as the desired mission state.

Search

The desired state of the general search component is defined by the operators specified maximum likelihood for detecting a previously unknown target at any point the area of operation or $P_{\max}(\mathbf{x}^s)$. The desired distribution for the assumption of one unknown target is found from Equation B.17.

$$P(\mathbf{x}^s) = \begin{cases} \frac{P_{\max}(\mathbf{x}^s)}{|\mathbb{S}|-1} & \text{for } |\mathbb{S}|-1 \text{ points} \\ 1 - P_{\max}(\mathbf{x}^s) & \text{for 1 point} \end{cases} \tag{B.17}$$

For the uniform distribution $P(\mathbf{x}^s) = \frac{1}{|\mathbb{S}|}$ for all points.

The area search component of the desired mission state is the set of the individual desired search states of all areas S specified by the operator.

$$\mathcal{M}_d^s = \left|\mathcal{M}_d^s(S)^i\right| \forall i = 1 \ldots A \tag{B.18}$$

Where A is the set of search areas specified by the operator. The distribution associated with each area in A is found in a similar manner to Equation B.17, where $S \subset \mathbb{S}$ and the total distribution length is $|S|$. The desired distribution can then be construction from the operator assigned value for $P^A_{\max}(\mathbf{x}^s)$.

$$P^A(\mathbf{x}^s) = \begin{cases} \frac{P_{\max}(\mathbf{x}^s)}{|S|-1} & \text{for } |S|-1 \text{ points} \\ 1 - P_{\max}(\mathbf{x}^s) & \text{for 1 point} \end{cases} \tag{B.19}$$

For the uniform distribution $P^A(\mathbf{x}^s) = \frac{1}{|S|}$. The desired mission state for the area S can then be found.

$$\begin{aligned} \mathcal{M}^s_d(S) &= \delta^s_\alpha - \delta^U_\alpha \\ &= 2\left(1 - e^{\log \sum_{i=1}^{|S|} \sqrt{\frac{P^A(\mathbf{x}^{s \in S})}{|S|}}}\right) - \delta^U_\alpha \end{aligned} \tag{B.20}$$

Track

The desired state for the track component of the mission is the set of individual track states for each ground vehicle in track.

$$\mathcal{M}^t_d = \left|\mathcal{M}^t_d(q)\right| \forall q \in \mathrm{Q} \tag{B.21}$$

The desired track distribution is defined by the track variance $\sigma^2_{\max}$ specified for the target q. A Gaussian distribution over g points with $\sigma^2 = \sigma^2_{\max}$ is constructed where g is the number of particles used by the tracker to estimate the location of a single target.

$$P(\mathbf{x}^t) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} e^{-0.5\left(\frac{x-\mu}{\sigma}\right)^2} & \text{for } x \in (-\frac{g}{2}, \frac{g}{2}) \text{ points} \\ 0 & \text{for all } x \notin g \end{cases} \tag{B.22}$$

The length of the distribution is $|\mathbb{S}|$ where $x \in \mathbb{S}$. The uniform distribution is that same as used for comparison of the current mission track state.

Classification

The desired state for the classification component of the mission is the set of individual classification states for each ground vehicle in track.

$$\mathcal{M}_d^c = |\mathcal{M}_d^c(q)| \, \forall q \in \mathrm{Q} \tag{B.23}$$

The desired classification distribution for target q is obtained described by the classification threshold $P_{\min}(\mathbf{x})$ specified for q by the operator.

$$P(\mathbf{x}^c) = \begin{cases} P_{\min}^m(\mathbf{x}_i^c) & \text{for } i = 1 \ldots m-1 \\ \frac{1-P_{\min}^m(\mathbf{x}_i^c)}{|m|-1} & \text{for } i = m \end{cases} \tag{B.24}$$

Where m is the set of all possible ground vehicle types. The uniform distribution over all possible ground vehicle types or $P(\mathbf{x}^c) = \frac{1}{|m|}$ is used to find δ_α^U.

B.5.3 Task Generation

The purpose of the task generation component of the *Mission Planning Module* is to generate the task pool $\mathbb{A}$ from the belief state B. A task or set of tasks is generated for each of the mission components of search track and classify as appropriate, considering also the ground vehicles being tracked and what is known about each.

Any tasks contained in the set $\mathbb{A}_h$, are added or removed from the task pool at the discretion of the operator. If a human generated task a_h conflicts with an automatically generated task, the system generated task is removed from the pool until reinstated by the operator's removal of the conflicting task.

Search

The probability of a previously undetected target being at any point within the area of operation is maintained as the search state $P(\mathbf{x}^s)$ as detailed in Appendix A. This gives rise to the general search task a^s which can be defined as *maintain the probability for detecting a previously unseen target, at any point in the area of operation, below the threshold set by the operator.* The parameters of the general search task are given by m^s

$$m^s = f[p^s, P_{\max}(\mathbf{x}^s|\mathbb{S})] \tag{B.25}$$

- $\mathbf{x}^s$ is the state variable representing the existence of a previously undetected target at node s.
- $P_{\max}(\mathbf{x}^s|\mathbb{S})$ is the operator specified desired likelihood of detecting a previously undetected target in the area of operation.
- p^s is the operator assigned priority assigned to the general search objective.

Area Search

The general search task, which relates to the entire area of operation, has a single operator applied priority and threshold. Searching the area around any protected asset is treated as a separate task to general search and is included as part of the set A_h.

The area search task a^S is dealt with in the same way as general search with the associated mission components taking the same form as Equation B.25. However the priority and threshold values applied by the operator only apply to the area S where $S \in \mathbb{S}$.

$$m^s = f[p^s_S, P_{\max}(\mathbf{x}^s|S)] \tag{B.26}$$

The general search task a^s is generated when the system is initialised along with an area search task a^s_S for each asset in the area of operation.

Track

When an observation is made of a ground vehicle, the position estimate is passed to the tracking component which maintains the track estimate $P(\mathbf{x}^t_q)$ for all q detected targets. This is detailed in Appendix A. The track task can then be defined as *reduce the uncertainty regarding the location of tracked target q.* Hence for each of q observed targets a track task a^t_q defined by m^t is generated.

$$m^t = f[p^t_c, (\sigma^2_c|\mathbf{x}^c_q), P(\mathbf{x}^c_q)]\forall \mathrm{c} = 1\ldots\mathrm{C} \tag{B.27}$$

- p_{c}^t is the assigned priority for tracking a given target type or set of types c.
- $\mathbf{x}_q^c$ is the state variable indicating the classification of the q^{th} target.
- σ_{c}^2 is the desired maximum variance for tracks belonging to the classification set c.

The desired track variance indicates the desired upper bound of uncertainty in a detected target's position. Both desired track variance and priority can be assigned on target type or classification set as in Section B.5.3 and Appendix A.

The area to be observed for each a_q^t is found by using the future belief state b_f for the track of target q at the planning horizon.

Target Track

The track task a_q^t defines the track priority and desired track state with respect to the target type. If different parameters define the desired track state for a specific target q, then the track task generated for the target q will be treated as part of the additional task set A_h.

$$m_q^t = f[p_q^t, (\sigma_q^2|\mathbf{x}_q^c), P(\mathbf{x}_q^c)] \tag{B.28}$$

- p_q^t is the operator assigned priority for tracking target q.
- $\mathbf{x}_q^c$ is the state variable indicating the classification of target q.
- σ_q^2 is the desired maximum variance for the track of target q.

Classify

When an observation of a ground vehicle is made, the target type estimate is passed to the classification component which maintains the classification for each track. This is detailed in Appendix A. The classify task a_q^c can then be defined as *reduce the uncertainty regarding the classification of target q given it is of the type m.* Hence for each of q observed targets a classification task a_q^c defined by m^c is generated. However as the task parameters are a function of the type of ground vehicle, the true task parameters often cannot be determined prior to the task being completed. As a result the classification supersets described in Appendix A are used to find m^{c}. The classification task for target q inherits the priority and threshold values for the superset c for which it has the highest probability of being a member.

$$m^{\mathrm{c}} = f[p_{\mathrm{c}}^{c}, P_{\min}(\mathbf{x}_{\mathrm{c}}^{c}|\mathbf{x}_{q}^{c})]\forall \mathrm{c} = 1 \dots C \tag{B.29}$$

- p_{c}^{c} is the operator assigned priority for the set of types c
- $\mathbf{x}_{q}^{c}$ is the state variable representing the classification of the q^{th} target.
- $\mathbf{x}_{\mathrm{c}}^{c}$ is the state variable representing the classification of the q^{th} target with respect to the classification group on which the threshold is defined.

If the classification probability exceeds the specified $P_{min}(\mathbf{x}_{q}^{c})$ value for the target then the associated classify task is removed.

Classify Target

The track task a_{q}^{c} defines the classification priority and threshold with respect to the probability of membership to a classification superset. If different parameters define the desired classification state for a specific target q, then the classification task generated for the target q will be treated as part of the additional task set A_{h}.

$$m_{q}^{c} = f[p_{q}^{c}, P_{\min}(\mathbf{x}_{q}^{c}] \tag{B.30}$$

- p_{q}^{c} is the operator assigned priority for the target q
- $\mathbf{x}_{q}^{c}$ is the state variable representing the classification of the q^{th} target.

B.5.4 Task Weights

The task weight is a function of three components reflecting the priority/criticality of the information, the desired uncertainty of the information and the preference between the different information types.

$$w_{a} = \begin{cases} p_{a}R\bar{\mathcal{M}}^{a} & \text{for } \mathcal{M}_{d}^{a} > \mathcal{M}_{c}^{a} \\ 0 & \text{for } \mathcal{M}_{d}^{a} \leq \mathcal{M}_{c}^{a} \end{cases} \tag{B.31}$$

Where p_a is the operator assigned priority relevant to the task a, R is the information type preference and $\bar{\mathcal{M}}^a$ is difference between the current and desired information distributions as described in Sections B.5.1 and B.5.2. The term $\bar{\mathcal{M}}^a$ is thus the marginalised HB distance between the desired and current distributions of the $\mathbf{x}$ relevant to the task, as shown in Equation B.32

$$\bar{\mathcal{M}}^a = \frac{\mathcal{M}_d^a - \mathcal{M}_c^a}{\mathcal{M}_d^a} \tag{B.32}$$

By using the same priority structure for search, track and classify the issue of information type preference is partially addressed. However due to the influence of the physical area covered by $P(\mathbf{x})$ in determining the $\bar{\mathcal{M}}^a$ value, tasks associated with a small physical area will typically have very high weighting compared to those such as the general search task which cover a much larger areas. Thus in order to make the information types preferentially independent [21] the preference term R is added to the task weighting function. The value of R is then the ratio of the physical area associated with the $P(\mathbf{x})$ relevant to the task, to the area covered by the general search task.

B.6 Coordination Module

The role of the Coordination Module is to efficiently allocate the tasks generated by the Organiser Module to the available resources and provided coordination mechanism between platforms, where necessary to complete the task. In addition to the task list $\mathbb{A}$ and weights $\mathbb{W}$ provided by the Organiser, the Coordinator uses information such as sensor accuracy, modalities, coverage and platform characteristics to determine the optimal task allocation. At this level, the task pool $\mathbb{A}$ is considered as a whole and thus does not make any distinction between human generated tasks or system generated task.

B.6.1 Task Allocation

Given the the task pool $\mathbb{A}$ and the associated set of weights $\mathbb{W}$, the task allocation process seeks to optimise the mapping in Equation B.33.

$$\mathcal{C} : \mathbb{W} \times \mathbb{A} \rightarrow \Upsilon \tag{B.33}$$

Where Υ is an ordered set of task to resource assignments, with the first element corresponding to the first agent. The optimization process seeks to maximize the objective function $\mathcal{J}(\Upsilon)$, where Υ contains all tasks assigned in the current planning cycle. The objective function for a single task is described by Equation B.34.

$$\mathcal{J}(a) = f(\eta(\Gamma, a), \gamma(\Pi, a), \kappa(X, a)) \tag{B.34}$$

- a is an individual task to be allocated.
- Γ is the suitability of a given sensor for the task a.
- Π is the suitability of the platform for the task a.
- X is the current state of the sensor/platform being tested for assignment.

1. $\eta(\Gamma, a)$ The suitability of the sensor for the task.
   - For tracking tasks the suitability score is defined by the sensor's localisation ability. The upper bound is given by the distribution of the true ground vehicle location given that it is observed at point x or $P(z^x|\mathbf{x}^x)$. The score for tracking a target the is determined by the HB distance between this distribution and a uniform distribution.
   - For classification tasks the suitability score is defined by the sensor's ability to discriminate between target types. The upper bound is given by the sensor model $P(z^c|\mathbf{x}^c_m)$, where m is the true ground vehicle type. The score for classifying a target of type m is determined by the HB distance between this distribution and a uniform distribution.
   - For search tasks the suitability score is the probability of detecting a ground vehicle given that it is present in the observation or $P(z^t|\mathbf{x}^t)$. This probability of detection is discussed in detail in section A.1.2.
2. $\gamma(\Pi, a)$ The platform suitability describes the relative ability of the sensor platform to place the sensor in a position to make the observations required by the task. For example traversability, size, speed or noise signature limitations may reduce the effectiveness or even prevent the agent from making the observations required to accomplish a. For the airborne sensors implemented in the simulation the various platform's

suitability for all tasks is assumed to be comparable and thus this component is not used in the objective function.

3. $\kappa(X, a)$ The current state of the platform will affect when a task can be started and the expected time to completion. Further, certain platform states may prevent the task from being performed. In the system implemented, only the agent's position is considered, with platforms closest to the area associated with the task receiving the highest suitability score.

For the task list $\mathbb{A}$ the utility for each possible agent j assigned to a task is found via Equation B.35.

$$\mathcal{U}(a_i^j) = \eta(S_j, a_i) \times \gamma(X_j, a_i) \times w_i \tag{B.35}$$

If an agent is limited to a single task assignment for the current planning cycle then for i tasks assigned to j resources, where $i <= j$, the best utility for Υ is obtained by Equation B.36.

$$\max \mathcal{U}_\Upsilon = \sum_j (\max u_a | a_{ij}) \tag{B.36}$$

A person-by-person optimization [33] process is used to obtain a pareto optimal allocation for Equation B.36. This process is outlined in B.6.1.

```
1. Assign task with highest utility gain to each agent

2. If multiple agents are assigned to the same task, the new individual
   agent utility is the maximum utility for the task divided by the
   number of agents assigned

3. Calculate the group utility

4. For $1 \dots j$ agents
       For $1 \dots i$ tasks
           Assign $a_i$ to agent $j$
           Calculate new group utility
```

```
        If new group utility $>$ old group utility
            Update allocated task list with $a_{ij}$
            Old group utility = new group utility
5. Return final allocated task list $\Upsilon$
```

B.6.2 Path Planner

Once the set Υ has been filled, the Path Planner determines the best waypoint/s for each agent in order to accomplish the assigned task ,taking into consideration platform and sensor capability. This is done in a number of stages with the resulting waypoint/s then passed to the agent controllers for execution.

1. The bounding agent states reachable at time k are found from the platform manoeuvre envelope, corresponding to maximum and minimum control inputs applied to the current state X.
2. Five intervening states are selected at even intervals between the bounding states and the sensor footprint for each of the resulting seven possible future states found.
3. The $P(\mathbf{x})$ of interest is projected to time k.
4. For each sensor footprint the probability of observing the true state of $\mathbf{x}_k$ is then found.
5. The pose corresponding to the footprint with the best $P(z|\mathbf{x}_k)$ is then used to define a waypoint which is communicated to the agents guidance module.
6. If the footprints yield a zero probability of detecting the true state, then the control input resulting in the agent moving closest to the area associated with a is chosen. For both the track and classify tasks this is the $P(\mathbf{x}^t)$ weighted centroid of the track estimate. For the area search task this is the centre of the area.

Appendix C

Models and Parameters

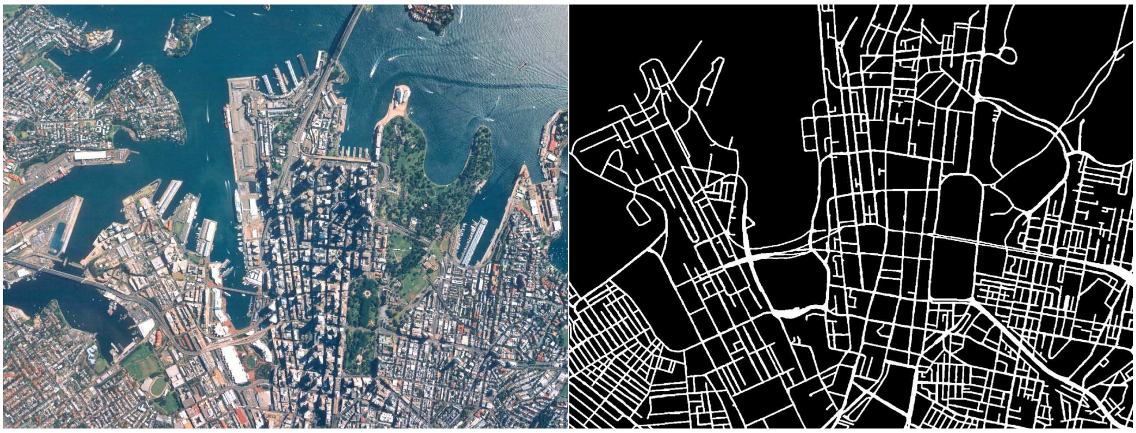


Figure C.1: Aerial photo of the Sydney CBD left and the road network used by the simulator right.

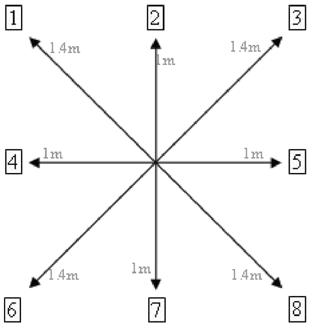


Figure C.2: Template showing the distances and the 8 possible direction nodes can be connected. The numbers in the square boxes are used as a column index in a matrix used to store the map.

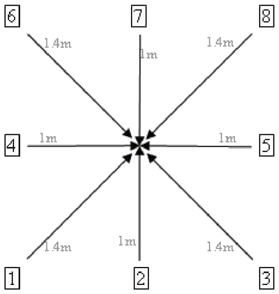


Figure C.3: Template showing the orientation assigned to a ground vehicle based on the direction it enters the current node.

C.1 Ground Vehicles

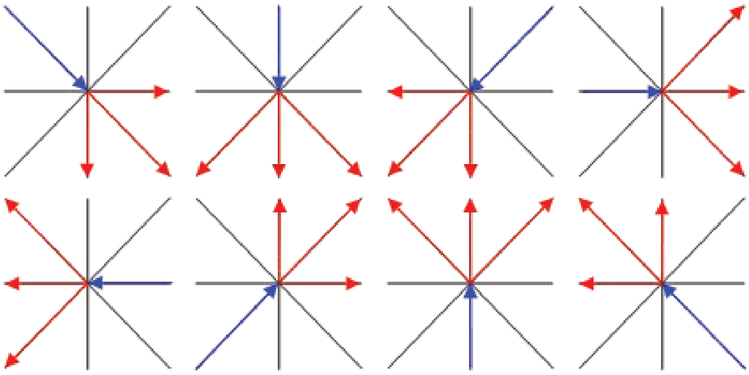
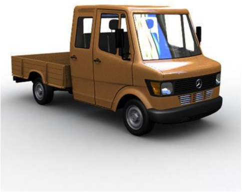
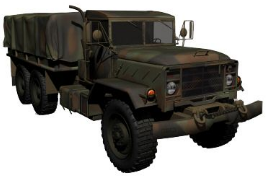


Figure C.4: Direction filter for propagation of Ground Vehicles at intersections. Direction of node arriving from is shown in blue, the permitted directions of travel are shown in red.

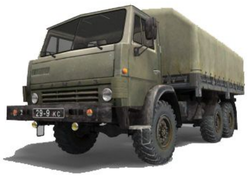
C.1.1 Ground Vehicle Types



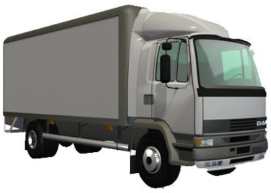
Truck, Light, Civilian	
Average Speed	$6.5m/s$
Projected Area	$11.0m^2$
System Enum	0
Conspicuous	



Truck, Light, Blue Forces	
Average Speed	$6.0m/s$
Projected Area	$12.0m^2$
System Enum	1
Inconspicuous	



Truck, Light, Red Forces	
Average Speed	$6.0m/s$
Projected Area	$12.0m^2$
System Enum	2
Inconspicuous	



Truck, Medium, Civilian	
Average Speed	$5.5m/s$
Projected Area	$18.0m^2$
System Enum	3
Conspicuous	

Truck, Medium, Blue Forces	
Average Speed	$5.0m/s$
Projected Area	$22.0m^2$
System Enum	4
Conspicuous	

Truck, Medium, Red Forces	
Average Speed	$5.0m/s$
Projected Area	$24.0m^2$
System Enum	5
Conspicuous	

Truck, Heavy, Civilian	
Average Speed	$4.5m/s$
Projected Area	$36.0m^2$
System Enum	6
Conspicuous	

Truck, Heavy, Blue Forces	
Average Speed	$4.0m/s$
Projected Area	$42.0m^2$
System Enum	7
Inconspicuous	

Truck, Heavy, Red Forces	
Average Speed	$4.0m/s$
Projected Area	$42.0m^2$
System Enum	8
Inconspicuous	

4WD, Civilian	
Average Speed	$8.5m/s$
Projected Area	$9.0m^2$
System Enum	9
Conspicuous	

4WD, Blue Forces	
Average Speed	$8.0m/s$
Projected Area	$8.5m^2$
System Enum	10
Inconspicuous	

4WD, Red	
Average Speed	$8.0m/s$
Projected Area	$7.5m^2$
System Enum	11
Inconspicuous	

4WD, Militia	
Average Speed	$8.5m/s$
Projected Area	$7.5m^2$
System Enum	12
Conspicuous	

Motorbike, Civilian	
Average Speed	$10.0m/s$
Projected Area	$2.0m^2$
System Enum	13
Conspicuous	

Motorbike, Militia	
Average Speed	$10.0m/s$
Projected Area	$2.0m^2$
System Enum	14
Conspicuous	

Car, Civilian	
Average Speed	$9.0m/s$
Projected Area	$7.0m^2$
System Enum	15
Conspicuous	

Car, Blue Forces	
Average Speed	$8.5m/s$
Projected Area	$7.5m^2$
System Enum	16
Conspicuous	

Car, Red Forces	
Average Speed	$8.5m/s$
Projected Area	$7.5m^2$
System Enum	17
Conspicuous	

Car, Militia	
Average Speed	$9.0m/s$
Projected Area	$6.5m^2$
System Enum	18
Conspicuous	

Armour, Light, Blue	
Average Speed	$6.5m/s$
Projected Area	$19.0m^2$
System Enum	19
Inconspicuous	

Armour, Light, Red	
Average Speed	$6.5m/s$
Projected Area	$21.0m^2$
System Enum	20
Inconspicuous	

Armour, Heavy, Blue	
Average Speed	$5.5m/s$
Projected Area	$36.0m^2$
System Enum	21
Inconspicuous	

Armour, IFV, Blue	
Average Speed	$6.0m/s$
Projected Area	$21.0m^2$
System Enum	22
Inconspicuous	

C.1.2 Classification Groups and Membership

Unarmoured	***Armed***	***Friendly***	***Light Vehicle***
Truck, Light, Civilian	Truck, Heavy, Blue Forces	Truck, Light, Civilian	4WD, Civilian
Truck, Light, Blue Forces	Truck, Heavy, Red Forces	Truck, Light, Blue Forces	4WD, Blue Forces
Truck, Light, Red Forces	4WD, Blue Forces	Truck, Medium, Civilian	4WD, Red Forces
Truck, Medium, Civilian	4WD, Red Forces	Truck, Medium, Blue Forces	4WD, Militia
Truck, Medium, Blue Forces	4WD, Militia	Truck, Heavy, Civilian	Motorbike, Civilian
Truck, Medium, Red Forces	Motorbike, Militia	Truck, Heavy, Blue Forces	Motorbike, Militia
Truck, Heavy, Civilian	Armour, Light, Blue Forces	4WD, Civilian	Car, Civilian
Truck, Heavy, Blue Forces	Armour, Light, Red Forces	4WD, Blue Forces	Car, Blue Forces
Truck, Heavy, Red Forces	Armour, Heavy, Blue Forces	Motorbike, Civilian	Car, Red Forces
4WD, Civilian	IFV, Blue Forces	Car, Civilian	Car, Militia
4WD, Blue Forces		Car, Blue Forces	
4WD, Red Forces		Armour, Light, Blue Forces	
4WD, Militia		Armour, Heavy, Blue Forces	
Motorbike, Civilian		IFV, Blue Forces	
Motorbike, Militia			
Car, Civilian			
Car, Blue Forces			
Car, Red Forces			
Car, Militia			

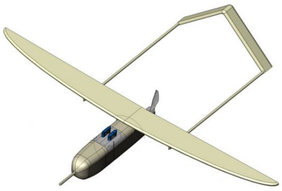
Red Forces	***Militia***	***Red Forces Armed***	***Red Forces Unarmed***
Truck, Light, Red Forces	4WD, Militia	4WD, Red Forces	Truck, Light, Red Forces
Truck, Medium, Red Forces	Motorbike, Militia	4WD, Red Forces	Truck, Medium, Red Forces
Truck, Heavy, Red Forces	Car, Militia	Armour, Light, Red Forces	Car, Red Forces
4WD, Red Forces			
Car, Red Forces			
Armour, Light, Red Forces			

Blue Armed	***Blue Unarmed***	***Blue Armour***	***Civilian***
Truck, Heavy, Blue Forces	Truck, Light, Blue Forces	Armour, Light, Blue Forces	Truck, Light, Civilian
4WD, Blue Forces	Truck, Medium, Blue Force	Armour, Heavy, Blue Forces	Truck, Medium, Civilian
Armour, Light, Blue Forces	Car, Blue Forces	IFV, Blue Forces	Truck, Heavy, Civilian
Armour, Heavy, Blue Forces			4WD, Civilian
IFV, Blue Forces			Motorbike, Civilian
			Car, Civilian

Unfriendly	***Unfriendly Armed***	***Friendly Unarmed***
Truck, Light, Red Forces	Truck, Heavy, Red Forces	Truck, Light, Civilian
Truck, Medium, Red Forces	4WD, Red Forces	Truck, Light, Blue Forces
Truck, Heavy, Red Forces	4WD, Militia	Truck, Medium, Civilian
4WD, Red Forces	Motorbike, Militia	Truck, Medium, Blue Forces
4WD, Militia	Car, Militia	Truck, Heavy, Civilian
Motorbike, Militia	Armour, Light, Red Forces	4WD, Civilian
Car, Red Forces		Motorbike, Civilian
Car, Militia		Car, Civilian
Armour, Light, Red Forces		Car, Blue Forces

Truck	***Blue Forces***	***Armoured***
Truck, Light, Civilian	Truck, Heavy, Blue Forces	Armour, Light, Blue Forces
Truck, Light, Blue Forces	4WD, Blue Forces	Armour, Light, Red Forces
Truck, Light, Red Forces	Armour, Light, Blue Forces	Armour, Heavy, Blue Forces
Truck, Medium, Civilian	Armour, Heavy, Blue Forces	IFV, Blue Forces
Truck, Medium, Blue Forces	IFV, Blue Forces	
Truck, Medium, Red Forces		
Truck, Heavy, Civilian		
Truck, Heavy, Blue Forces		
Truck, Heavy, Red Forces		

C.2 UAVs



Aerosonde	
Cruise Speed	21.0m/s
Max Speed	41.0m/s
Min Speed	15.0.0m/s
Max Bank	0.8rad 14^o
Sensors	1 x IR or 1 x Vision

Fast Delta	
Cruise Speed	30.0m/s
Max Speed	46.0m/s
Min Speed	22.0.0m/s
Max Bank	1.1rad 65^o
Sensors	2 x Vision or 1 x Vision + 1 x IR or 1 x Radar

C.3 Sensor Models

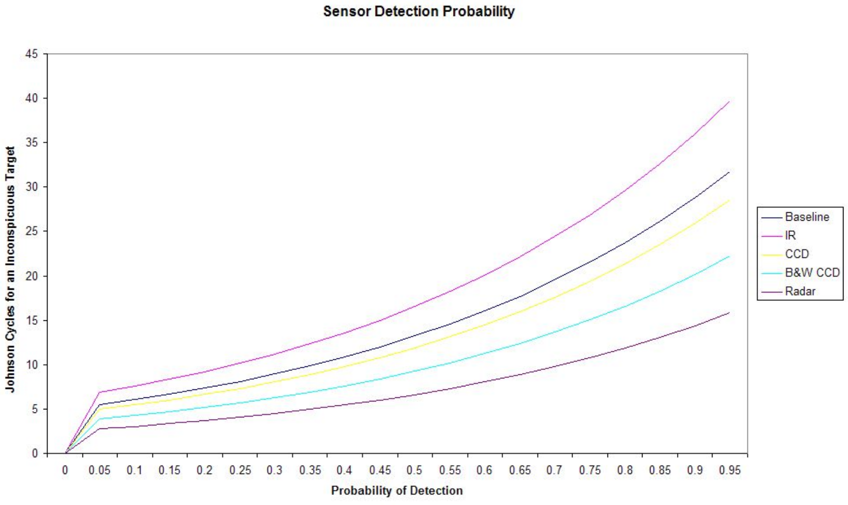


Figure C.5: Probability of detection as a function of Johnson cycles

C.3.1 Sensor Orientations

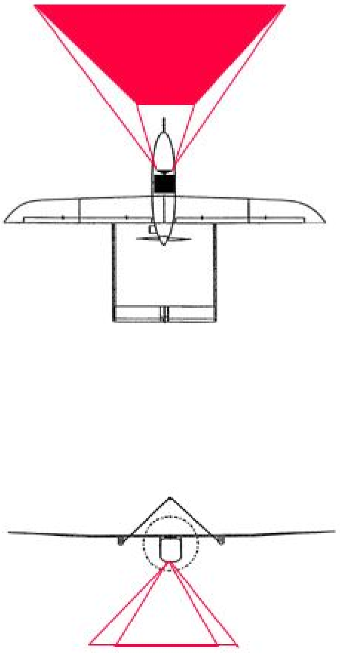


Figure C.6: Look Forward Orientation - Enum 0

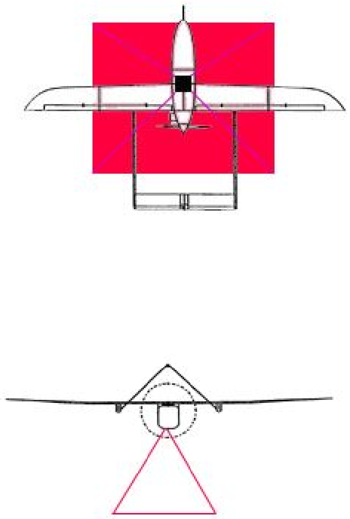


Figure C.7: Look Down Orientation - Enum 1

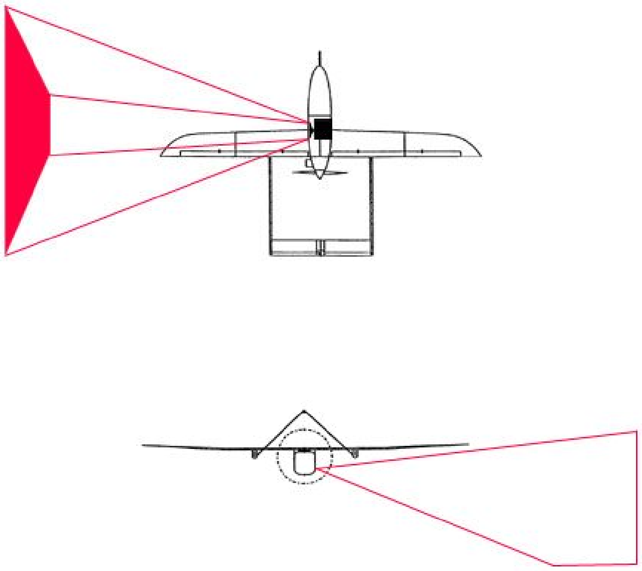


Figure C.8: Look Left Orientation - Enum 2

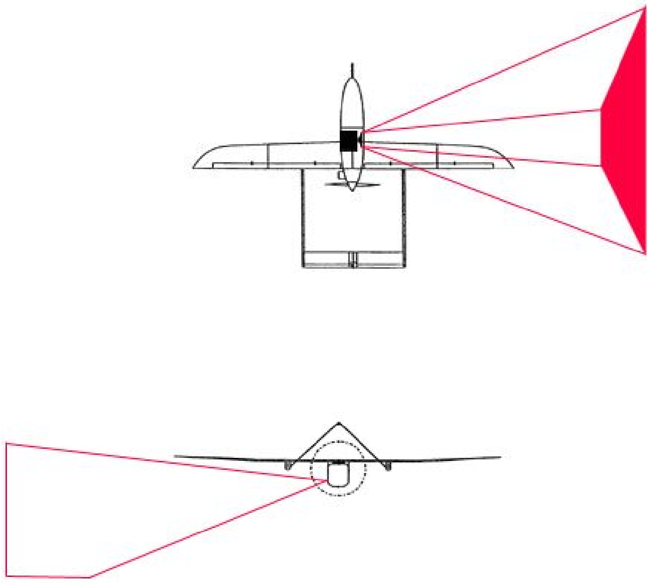


Figure C.9: Look Right Orientation - Enum 3

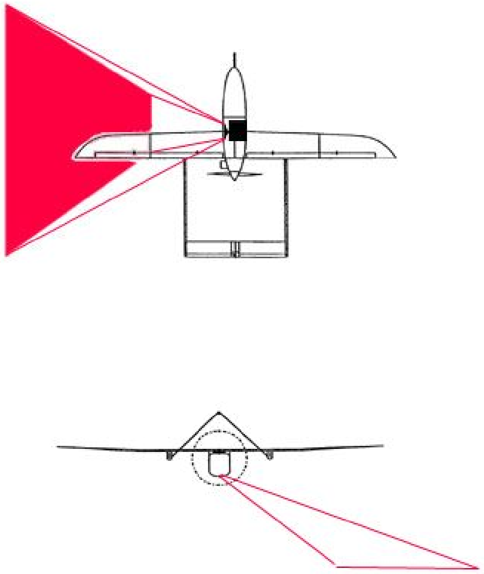


Figure C.10: Look Left and Down Orientation - Enum 4

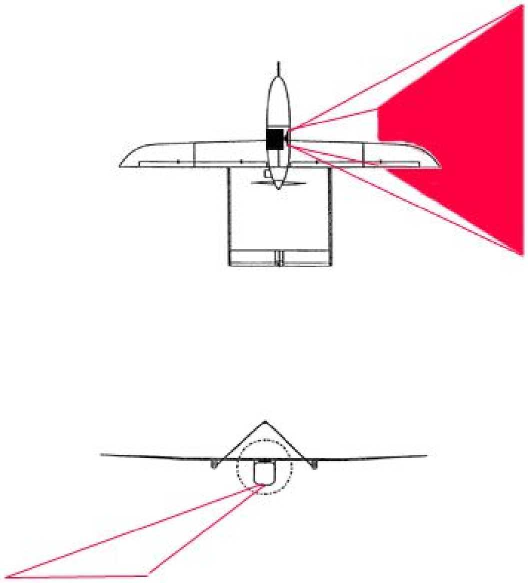
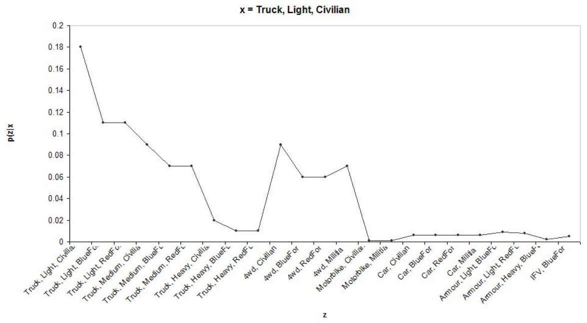
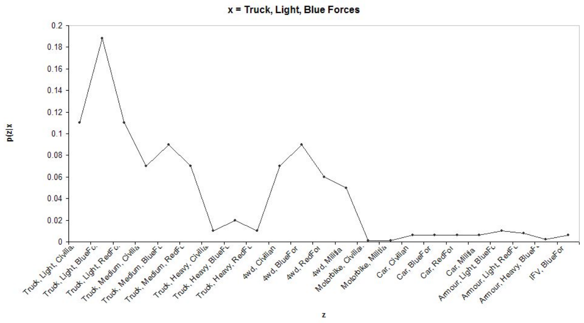
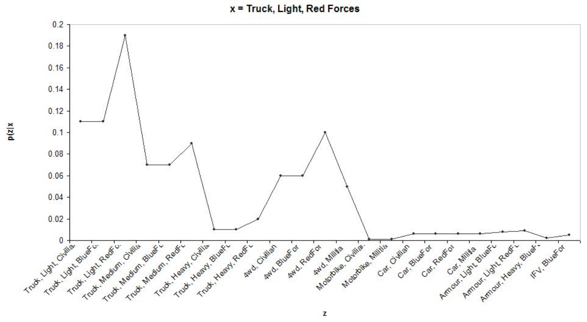


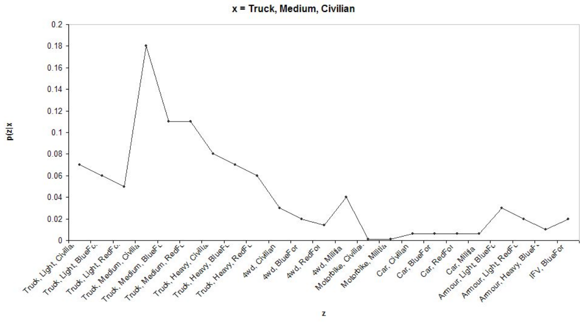
Figure C.11: Look Right and Down Orientation - Enum 5

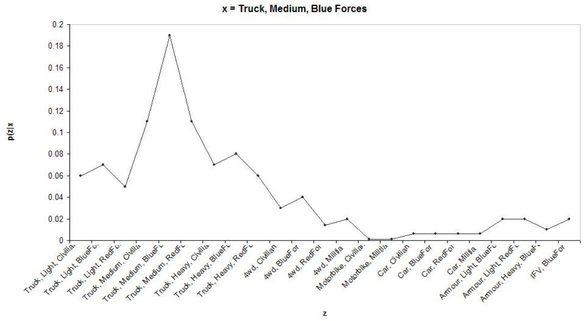
C.3.2 Simulated Infra-Red Sensor

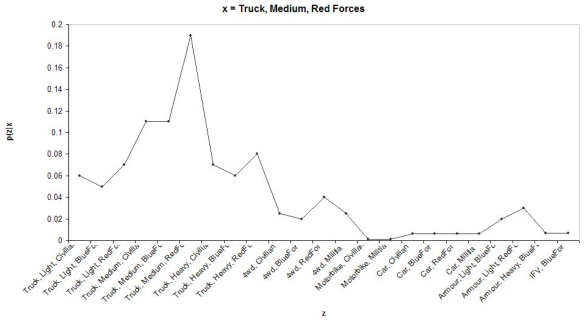


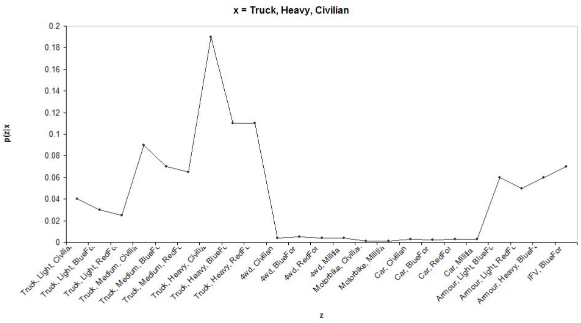


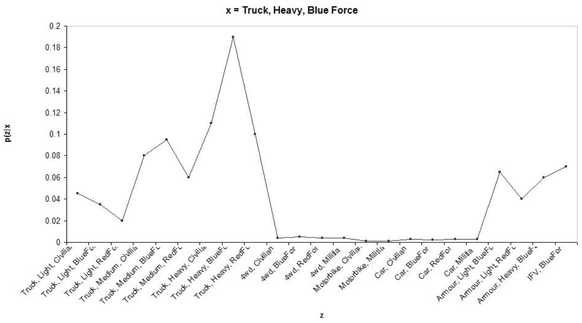


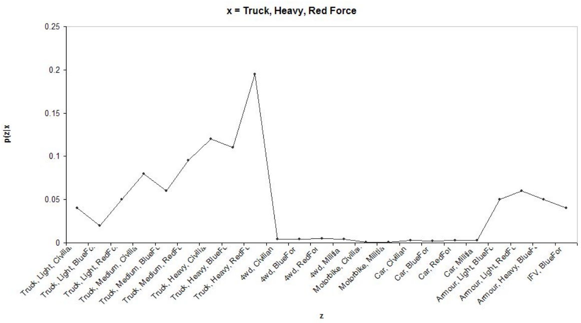


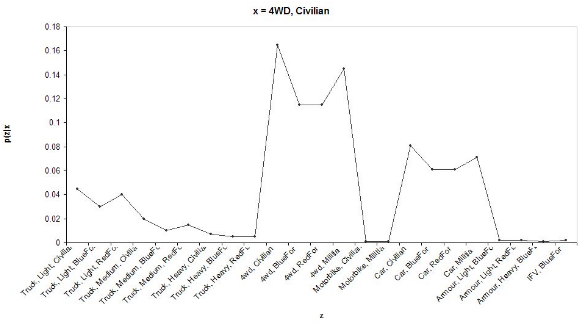


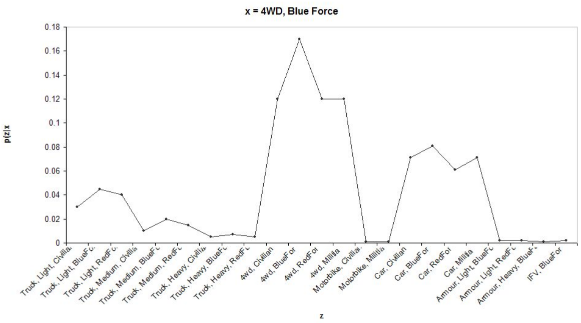


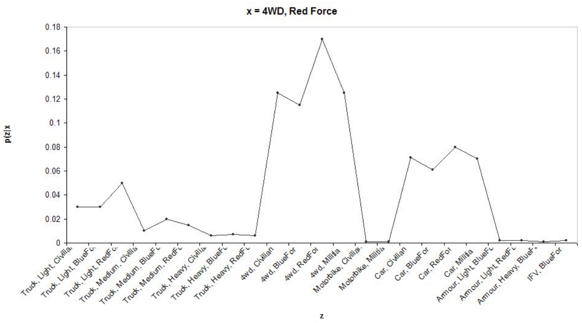

x = Truck, Heavy, Civilian
p(z|x
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

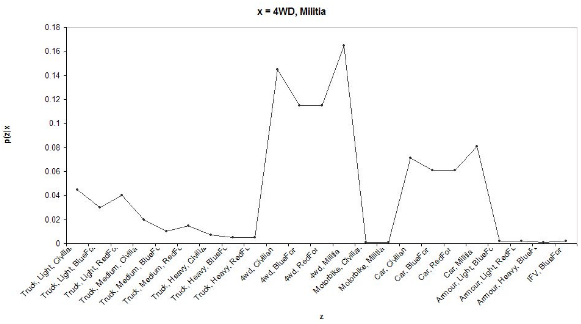

x = Truck, Heavy, Blue Force
p(z|x
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

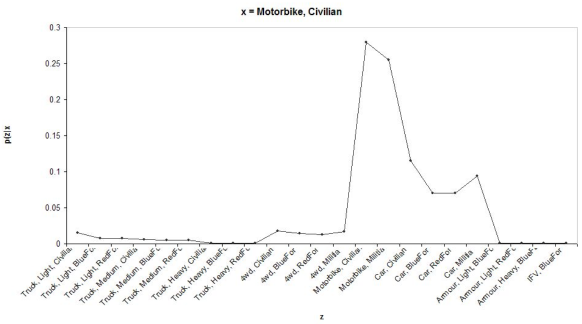

x = Truck, Heavy, Red Force
p(z|x
0.25
0.2
0.15
0.1
0.05
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

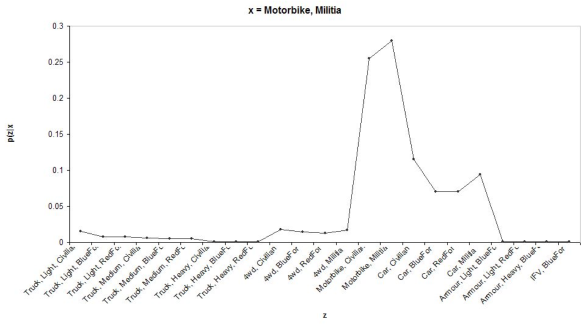

x = 4WD, Civilian
p(z|x
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

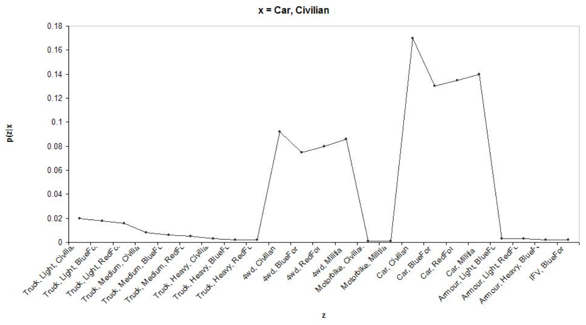

x = 4WD, Blue Force
p(z|x
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

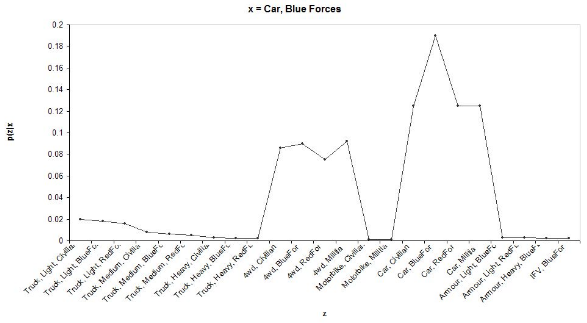

x = 4WD, Red Force
p(z|x
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

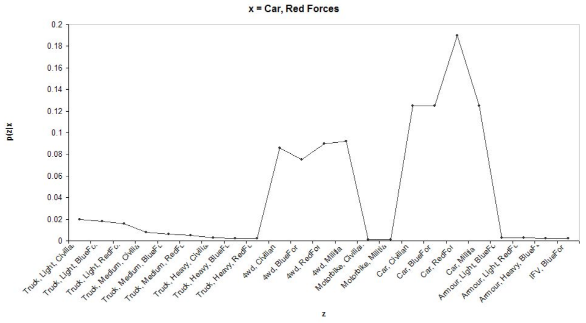

x = 4WD, Militia
p(z|x
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

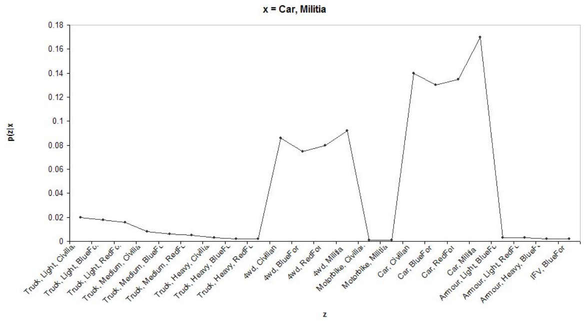

x = Motorbike, Civilian
p(z|x
0.3
0.25
0.2
0.15
0.1
0.05
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

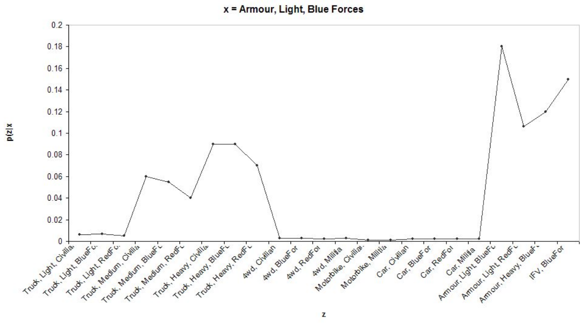

x = Motorbike, Militia
p(z|x)
0.3
0.25
0.2
0.15
0.1
0.05
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

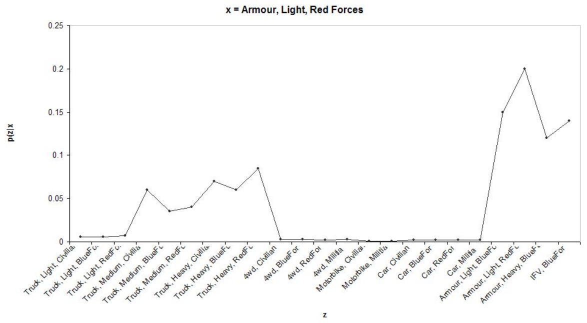

x = Car, Civilian
p(z|x)
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

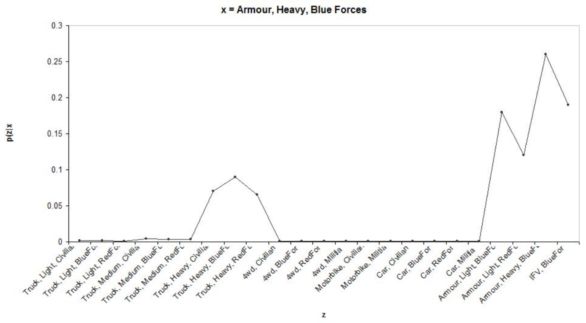

x = Car, Blue Forces
p(z|x)
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

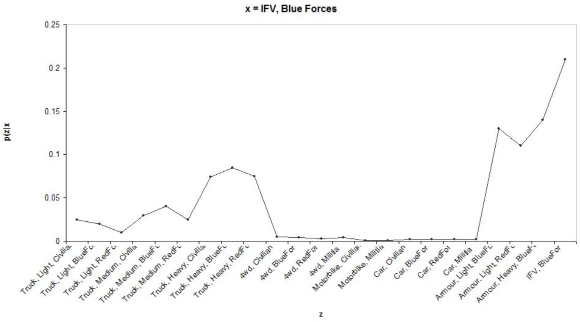

x = Car, Red Forces
p(z|x)
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z


x = Car, Militia
p(z|x)
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

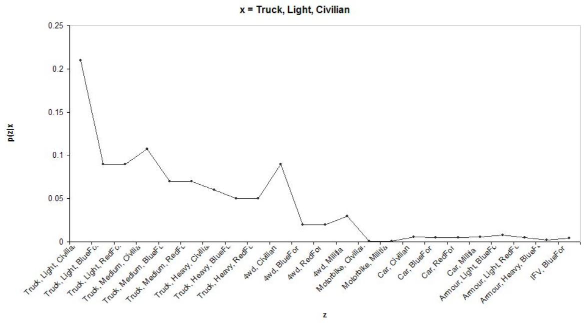

x = Armour, Light, Blue Forces
p(z|x)
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

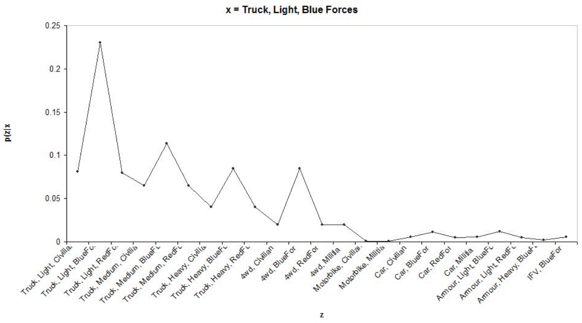

x = Armour, Light, Red Forces
p(z|x)
0.25
0.2
0.15
0.1
0.05
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

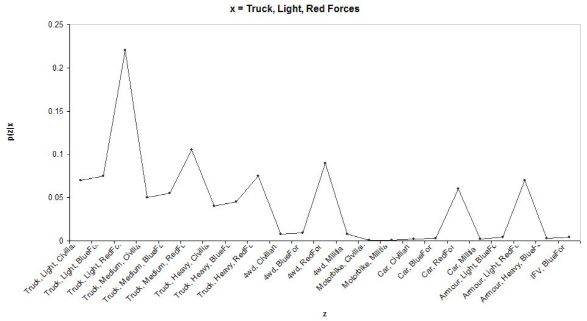

x = Armour, Heavy, Blue Forces
p(z|x)
0.3
0.25
0.2
0.15
0.1
0.05
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

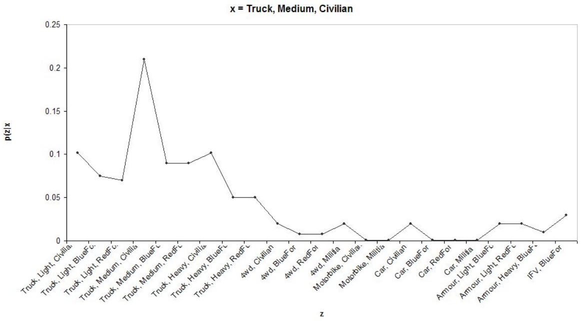

x = IFV, Blue Forces
p(z|x
0.25
0.2
0.15
0.1
0.05
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

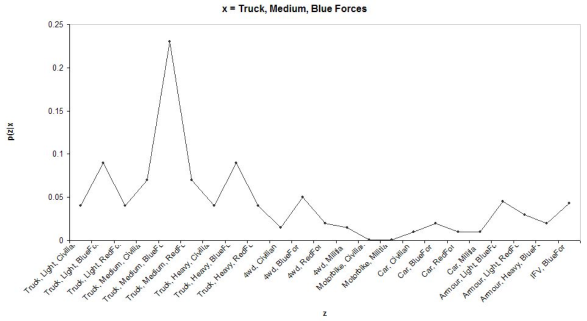
C.3.3 Simulated Colour Vision Sensor

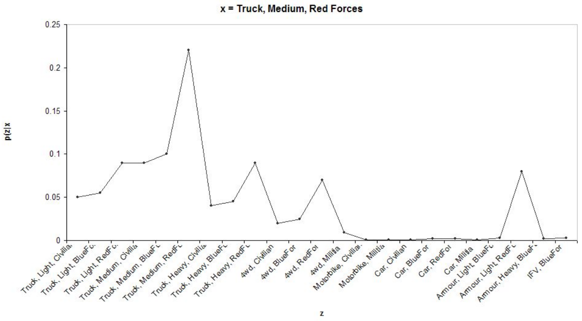


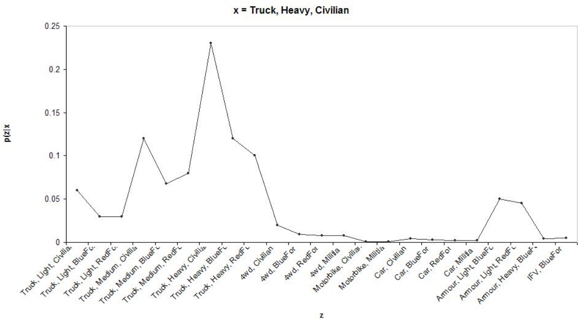


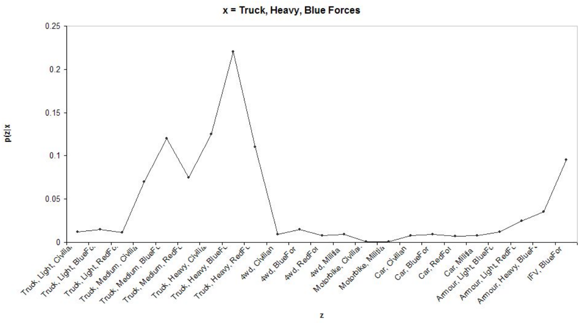


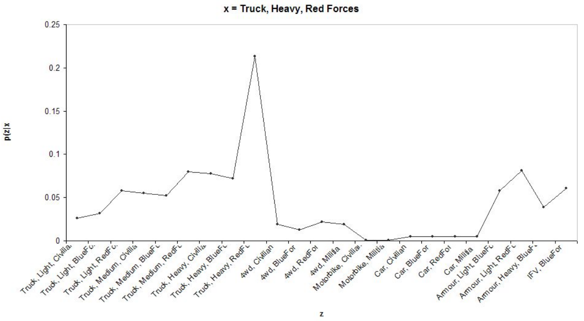


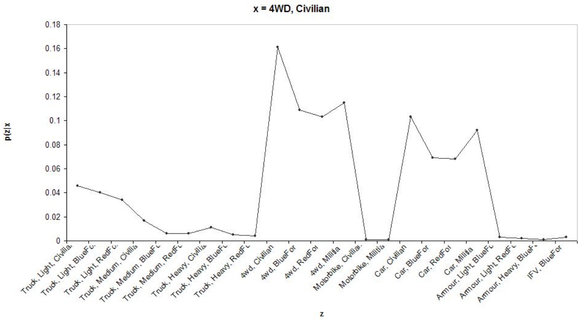


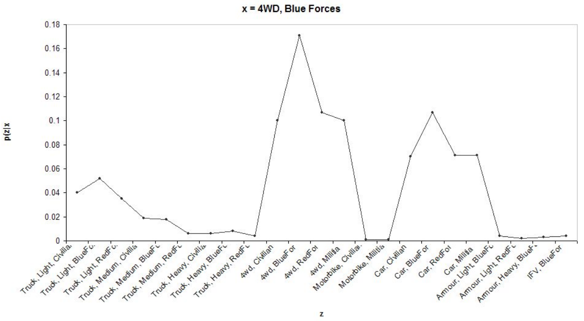


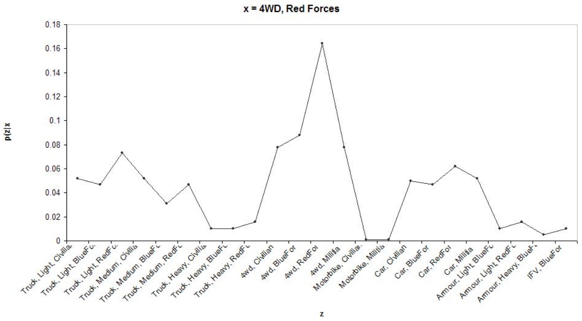

x = Truck, Heavy, Civilian
p(z|x)
0
0.05
0.1
0.15
0.2
0.25
z
Truck, Light, Civilia
Truck, Light, BlueFc
Truck, Light, RedFc
Truck, Medium, Civilia
Truck, Medium, BlueFc
Truck, Medium, RedFc
Truck, Heavy, Civilia
Truck, Heavy, BlueFc
Truck, Heavy, RedFc
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Amour, Light, BlueFc
Amour, Light, RedFc
Amour, Heavy, BlueFc
IFV, BlueFor

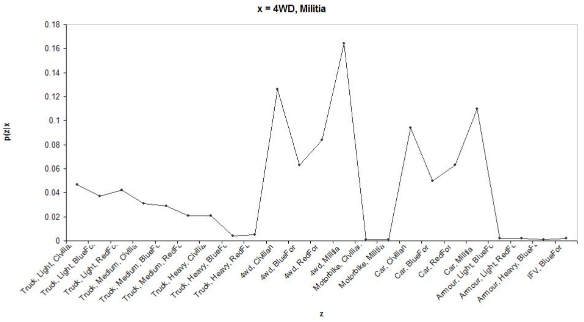

x = Truck, Heavy, Blue Forces
p(z|x)
0
0.05
0.1
0.15
0.2
0.25
z
Truck, Light, Civilia
Truck, Light, BlueFc
Truck, Light, RedFc
Truck, Medium, Civilia
Truck, Medium, BlueFc
Truck, Medium, RedFc
Truck, Heavy, Civilia
Truck, Heavy, BlueFc
Truck, Heavy, RedFc
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Amour, Light, BlueFc
Amour, Light, RedFc
Amour, Heavy, BlueFc
IFV, BlueFor

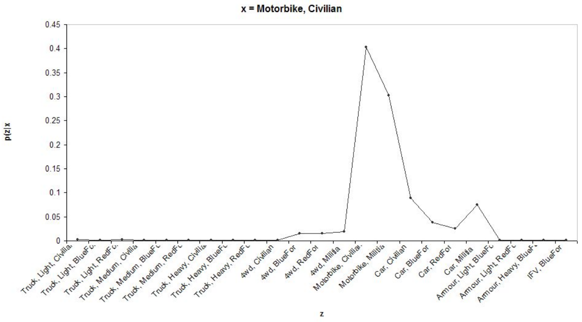

x = Truck, Heavy, Red Forces
p(z|x)
0
0.05
0.1
0.15
0.2
0.25
z
Truck, Light, Civilia
Truck, Light, BlueFc
Truck, Light, RedFc
Truck, Medium, Civilia
Truck, Medium, BlueFc
Truck, Medium, RedFc
Truck, Heavy, Civilia
Truck, Heavy, BlueFc
Truck, Heavy, RedFc
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Amour, Light, BlueFc
Amour, Light, RedFc
Amour, Heavy, BlueFc
IFV, BlueFor

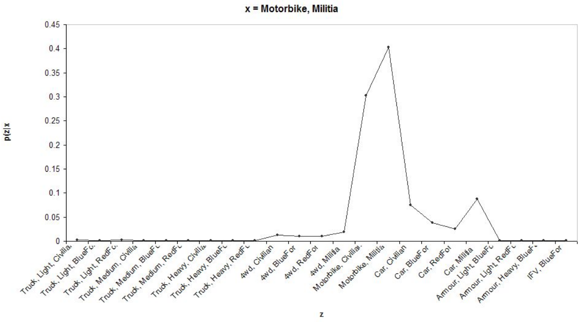

x = 4WD, Civilian
p(z|x)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
z
Truck, Light, Civilia
Truck, Light, BlueFc
Truck, Light, RedFc
Truck, Medium, Civilia
Truck, Medium, BlueFc
Truck, Medium, RedFc
Truck, Heavy, Civilia
Truck, Heavy, BlueFc
Truck, Heavy, RedFc
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Amour, Light, BlueFc
Amour, Light, RedFc
Amour, Heavy, BlueFc
IFV, BlueFor

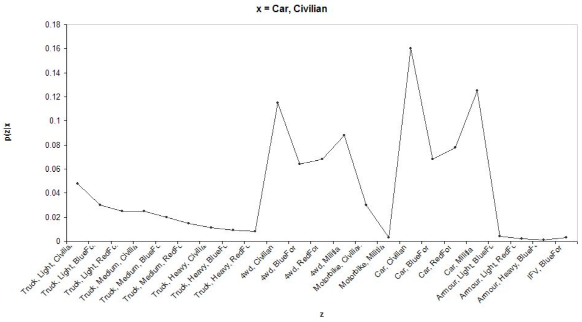

x = 4WD, Blue Forces
p(z|x)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
z
Truck, Light, Civilia
Truck, Light, BlueFc
Truck, Light, RedFc
Truck, Medium, Civilia
Truck, Medium, BlueFc
Truck, Medium, RedFc
Truck, Heavy, Civilia
Truck, Heavy, BlueFc
Truck, Heavy, RedFc
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Amour, Light, BlueFc
Amour, Light, RedFc
Amour, Heavy, BlueFc
IFV, BlueFor

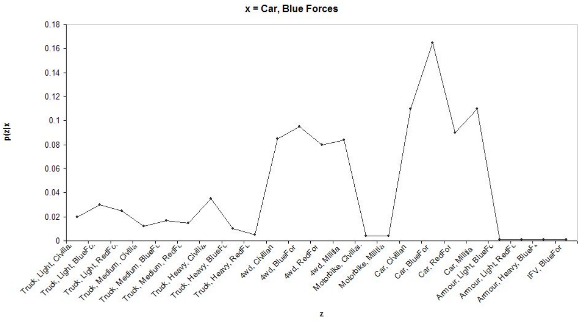

x = 4WD, Red Forces
p(z|x)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
z
Truck, Light, Civilia
Truck, Light, BlueFc
Truck, Light, RedFc
Truck, Medium, Civilia
Truck, Medium, BlueFc
Truck, Medium, RedFc
Truck, Heavy, Civilia
Truck, Heavy, BlueFc
Truck, Heavy, RedFc
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Amour, Light, BlueFc
Amour, Light, RedFc
Amour, Heavy, BlueFc
IFV, BlueFor

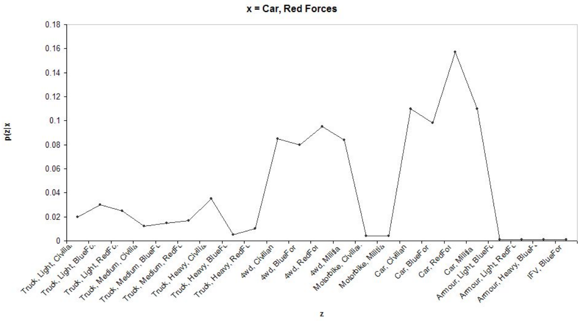

x = 4WD, Militia
p(z|x)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
z
Truck, Light, Civilia
Truck, Light, BlueFc
Truck, Light, RedFc
Truck, Medium, Civilia
Truck, Medium, BlueFc
Truck, Medium, RedFc
Truck, Heavy, Civilia
Truck, Heavy, BlueFc
Truck, Heavy, RedFc
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Amour, Light, BlueFc
Amour, Light, RedFc
Amour, Heavy, BlueFc
IFV, BlueFor

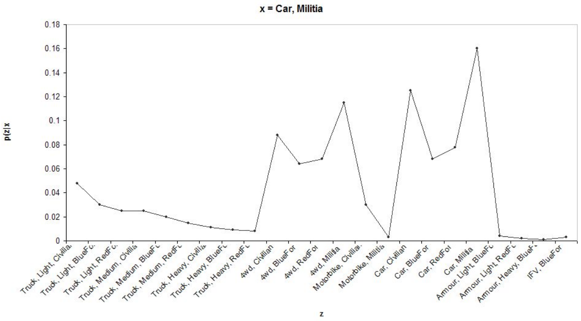

x = Motorbike, Civilian
p(z|x)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
z
Truck, Light, Civilia
Truck, Light, BlueFc
Truck, Light, RedFc
Truck, Medium, Civilia
Truck, Medium, BlueFc
Truck, Medium, RedFc
Truck, Heavy, Civilia
Truck, Heavy, BlueFc
Truck, Heavy, RedFc
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Amour, Light, BlueFc
Amour, Light, RedFc
Amour, Heavy, BlueFc
IFV, BlueFor

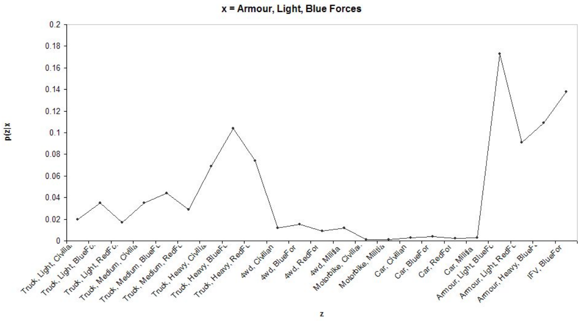

x = Motorbike, Militia
p(z|x
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

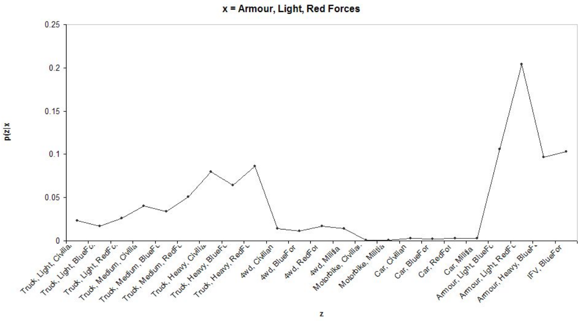

x = Car, Civilian
p(z|x
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

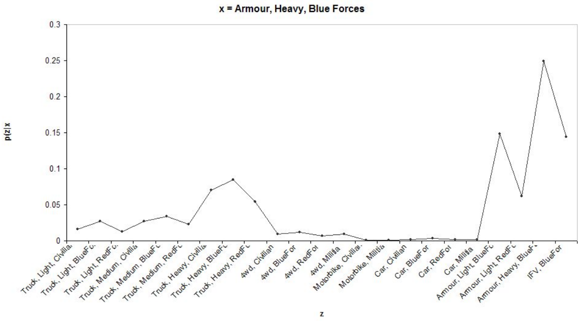

x = Car, Blue Forces
p(z|x
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

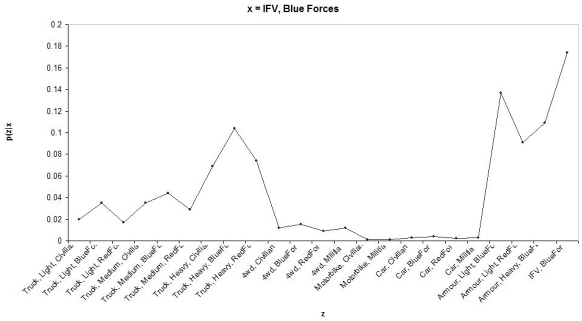

x = Car, Red Forces
p(z|x
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z


x = Car, Militia
p(z|x
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

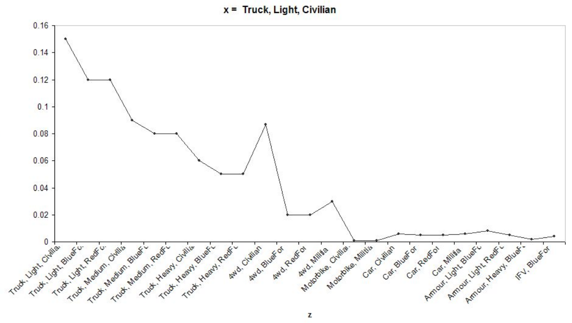

x = Armour, Light, Blue Forces
p(z|x
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

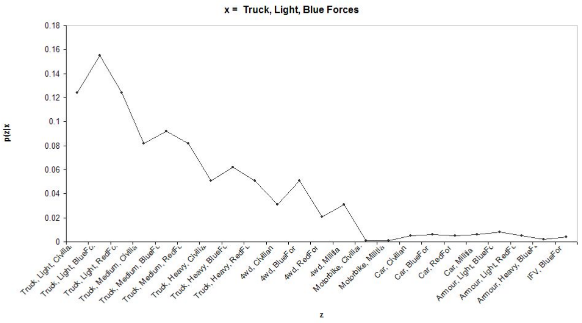

x = Armour, Light, Red Forces
p(z|x
0
0.05
0.1
0.15
0.2
0.25
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

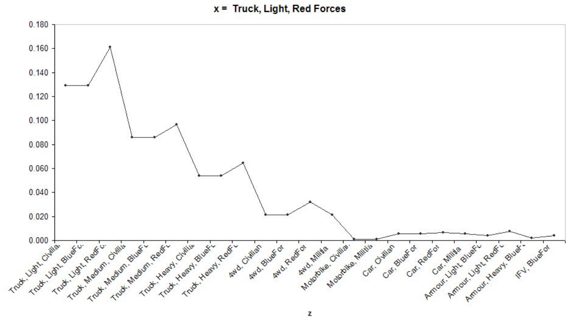

x = Armour, Heavy, Blue Forces
p(z|x
0
0.05
0.1
0.15
0.2
0.25
0.3
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

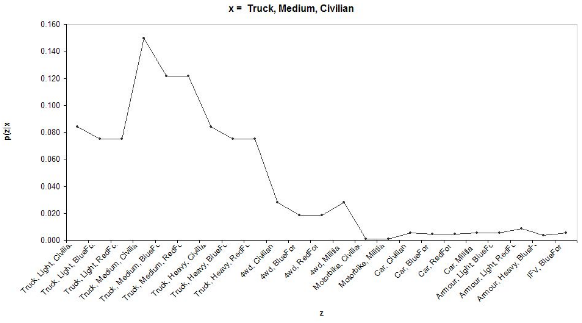

x = IFV, Blue Forces
p(z|x
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

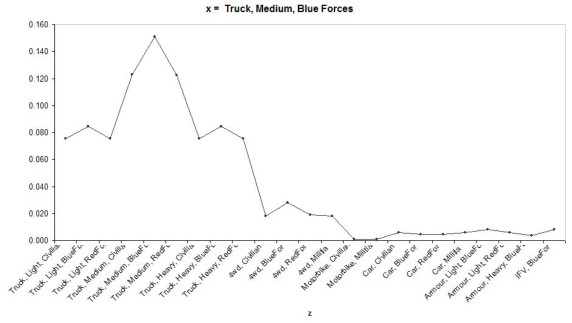
C.3.4 Simulated Black and White Vision Sensor

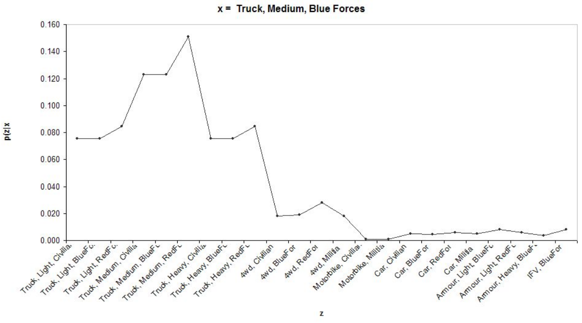


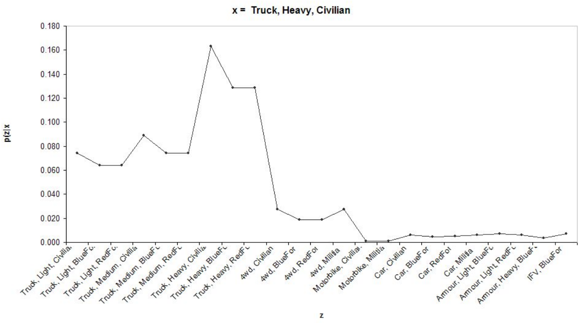


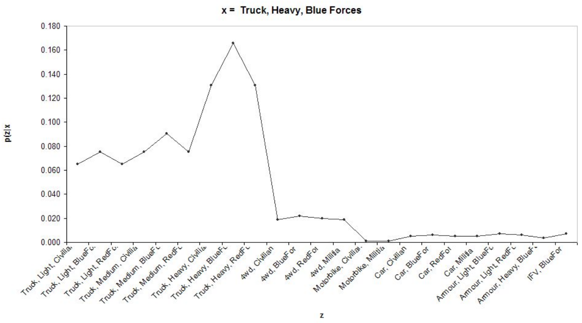


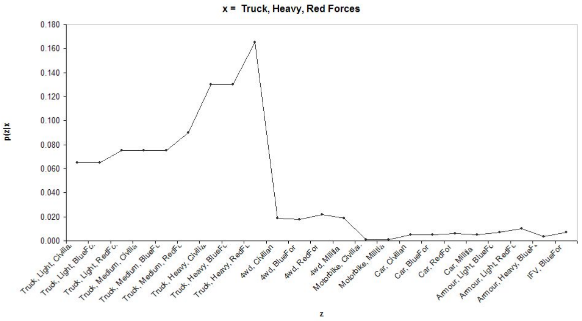


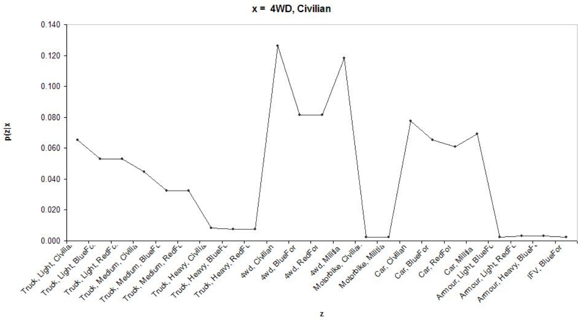


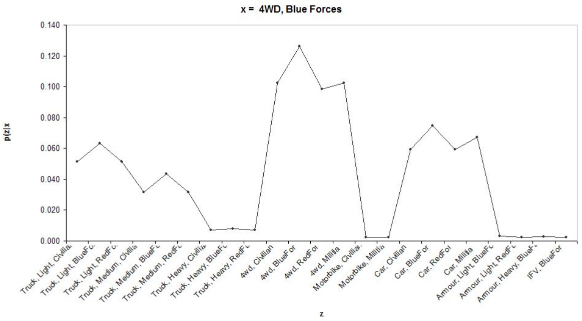


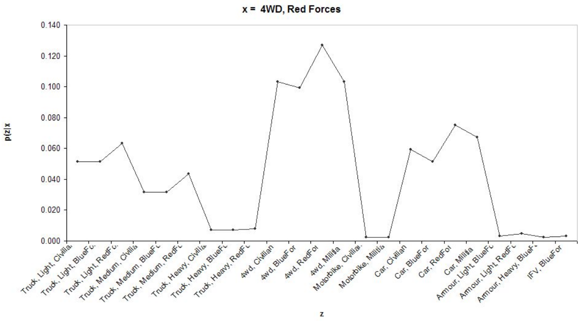

x = Truck, Heavy, Civilian
p(z|x
0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

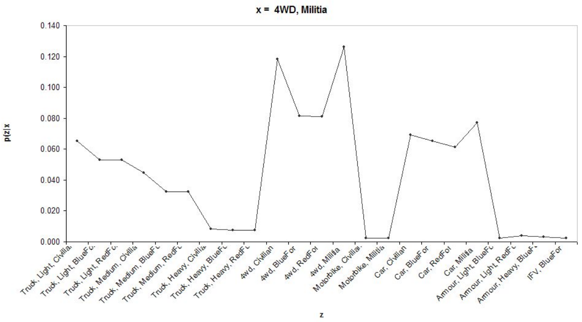

x = Truck, Heavy, Blue Forces
p(z|x
0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

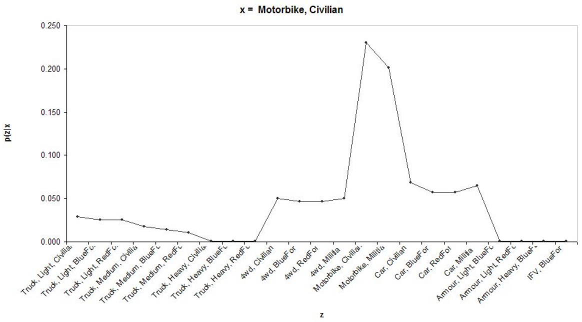

x = Truck, Heavy, Red Forces
p(z|x
0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

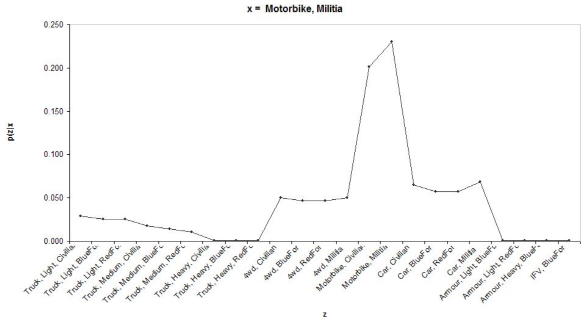

x = 4WD, Civilian
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

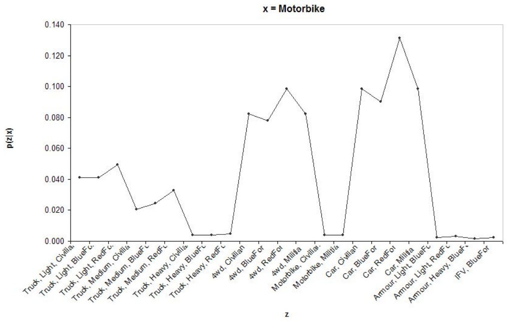

x = 4WD, Blue Forces
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

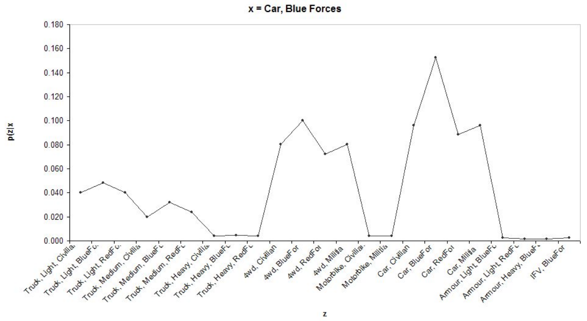

x = 4WD, Red Forces
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

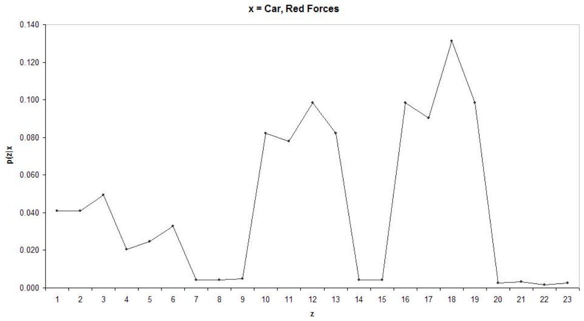

x = 4WD, Militia
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

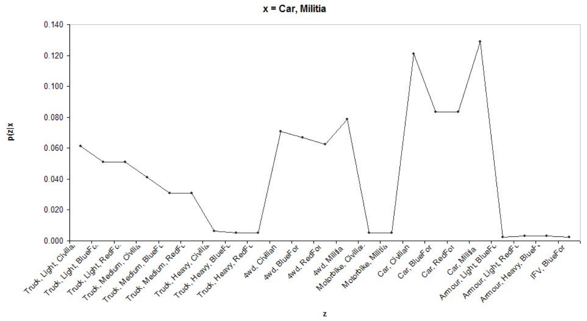

x = Motorbike, Civilian
p(z|x
0.250
0.200
0.150
0.100
0.050
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueF
IFV, BlueFor
z

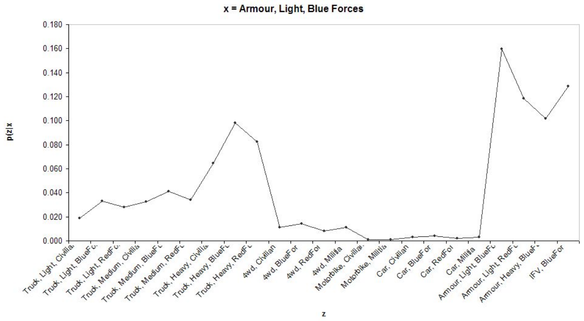

x = Motorbike, Militia
p(z|x
0.250
0.200
0.150
0.100
0.050
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

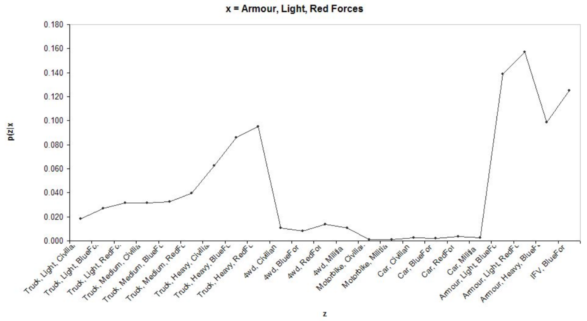

x = Motorbike
p(z|x)
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

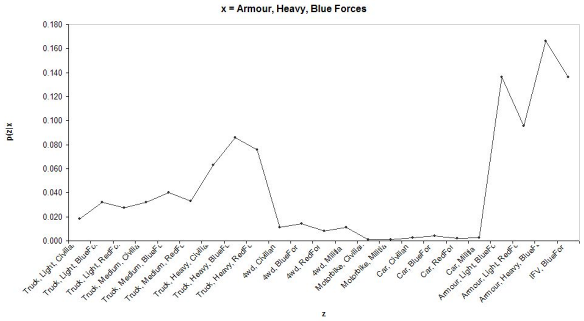

x = Car, Blue Forces
p(z|x
0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

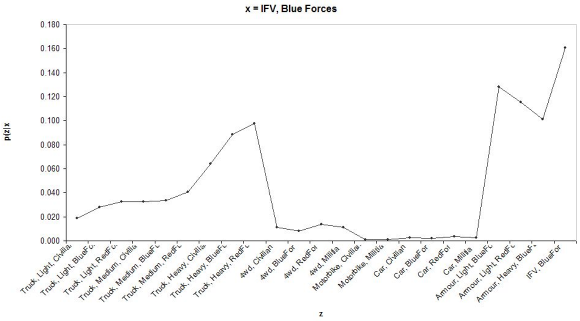

x = Car, Red Forces
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
z


x = Car, Militia
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

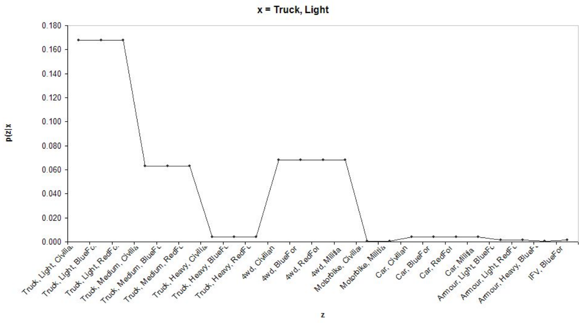

x = Armour, Light, Blue Forces
p(z|x
0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

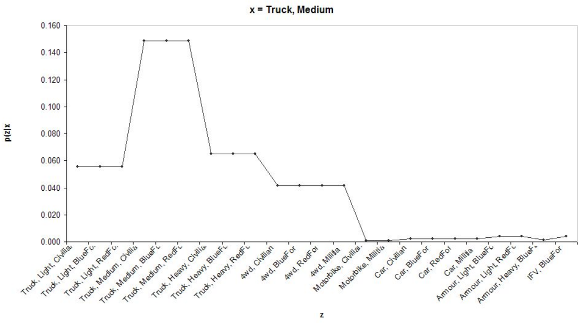

x = Armour, Light, Red Forces
p(z|x
0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

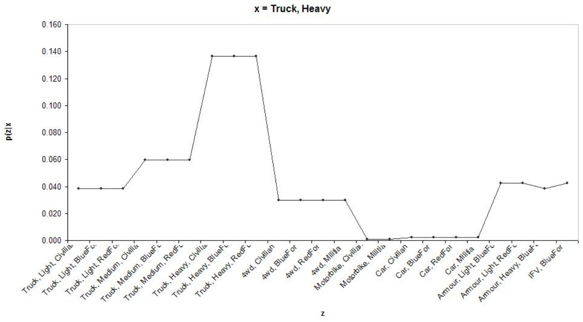

x = Armour, Heavy, Blue Forces
p(z|x
0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

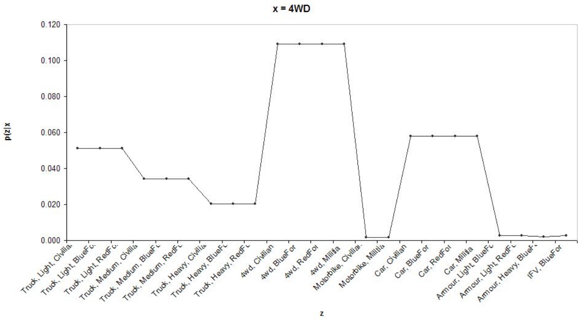

x = IFV, Blue Forces
p(z|x
0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

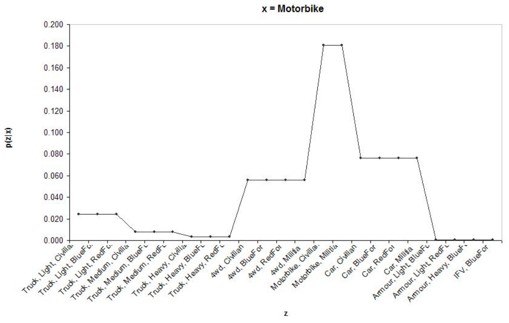
C.3.5 Simulated Radar Sensor

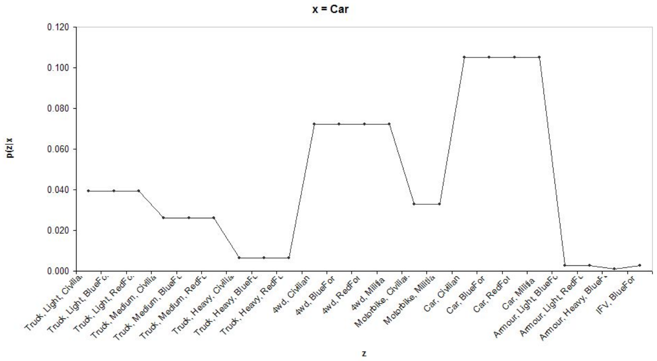


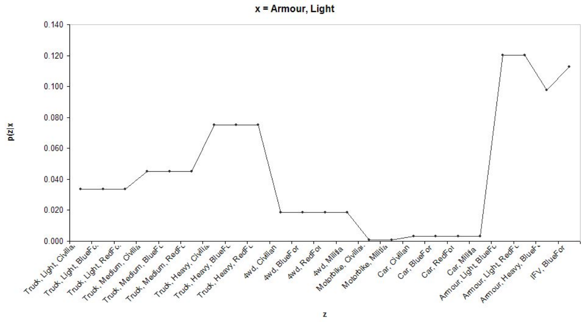


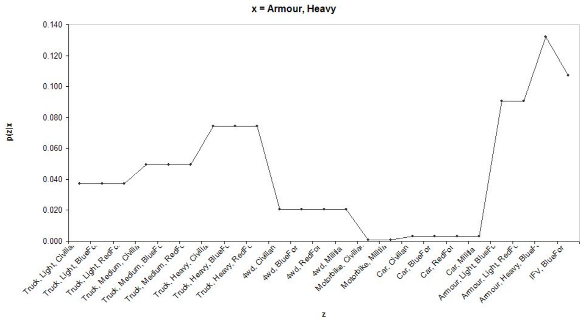


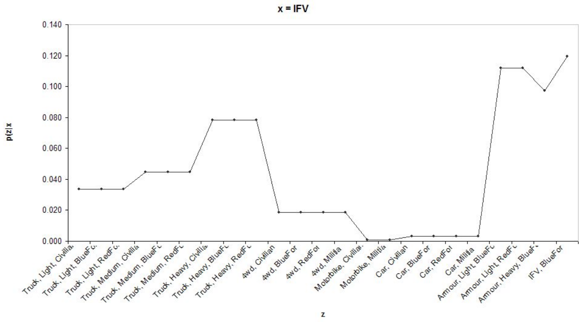







x = Armour, Light
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z


x = Armour, Heavy
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z


x = IFV
p(z|x
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.000
Truck, Light, Civilia
Truck, Light, BlueFo
Truck, Light, RedFo
Truck, Medium, Civilia
Truck, Medium, BlueFo
Truck, Medium, RedFo
Truck, Heavy, Civilia
Truck, Heavy, BlueFo
Truck, Heavy, RedFo
4wd, Civilian
4wd, BlueFor
4wd, RedFor
4wd, Militia
Motorbike, Civilia
Motorbike, Militia
Car, Civilian
Car, BlueFor
Car, RedFor
Car, Militia
Armour, Light, BlueFo
Armour, Light, RedFo
Armour, Heavy, BlueFo
IFV, BlueFor
z

Bibliography

[1] S. Agichtein, E. Lawrence and L. Gravano. Learning search engine specific query transformations for question answering. In *Proc. of the 10th Int. Conf. on World Wide Web*, 2001.

[2] G. Bererton, C. Gordon and S Thrun. Auction mechanism design for multi-robot coordination. In *Proc. of the Int. Conf. on Neural Information Processing Systems*, 2003.

[3] W. Brandt, B. Brauer and Gerhard Weiss. Task assignment in multiagent systems based on vickrey-type auctioning and leveled commitment contracting. *Lecture Notes in Artificial Intelligence*, 1860:95–106, 2000.

[4] D. Chajewska, U. Koller and Ormoneit D. Learning an agents utility function by observing behaviour. In *Proc. of the Eighteenth International Conf. on Machine Learning*, 2001.

[5] D. Chajewska, U. Koller and Parr R. Making rational decisions using adaptive utility elicitation. In *Proc. of the 17th International Conf. on Artifical Intelligence*, 2000.

[6] J. Considine. Methodologies for predicting detection and recognition rages of imaging systems against military targets. Technical report, Australian Defence Science and Technology Organisation, 2003.

[7] J. Cox. *The Concise Guide to Economics.* Savannah-Pikeville Press, 1997, 2nd Ed.

[8] J. Dastani, M. Hulstijn and Leendert van der Torre. How to decide what to do? *European Journal of Operational Research*, 160:762–784, 2005.

[9] M. Dias and A. Stentz. Opportunistic optimization for market-based multirobot control. In *Proc. of the 2002 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems.*, 2002.

[10] A. Madhavan J. Nemes E Dong, X. Halvey and J. Zhang. Similarity search for web services. In *Proc. of the 30th Very Large Database Conf.*, 2004.

[11] R Farber, S. Costanza and M. Wilson. Economic and ecological concepts for valuing ecosystem services. *Ecological Economics*, 41:375–392, 2002.

[12] K. Fullam and K. Barber. Using policies for information valuation to justify beliefs. In *Proc. of the 3rd International Conf. on Autonomous Agents and Multi-Agent Systems AAMAS'02*, July 2002.

[13] B. Gerkey and M. Mataric. Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.

[14] J. Grass and S. Zilberstein. A value-driven system for autonomous information gathering. *Journal of Intelligent Information Systems*, 14:5–27, 2000.

[15] A. Grocholsky, B. Makarenko and H. Durrant-Whyte. Information-theoretic coordinated control of multiple sensor platforms. In *Proc. of the 2003 Int. Conf. on Robotics and Automation*, 2003.

[16] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms.* PhD thesis, The University of Sydney, 2002.

[17] J. Guerrero and G. Oliver. Physical interference impact in multi-robot task allocation auction methods. In *Proc. of the IEEE Workshop on Distributed Intelligence Systems*, 2006.

[18] J. Hicks. *Value and Capital.* Oxford University Press, 1975 2nd Ed.

[19] J Hicks and R Allen. A reconsideration of the theory of value. part 1. *Econometrica*, 1(1):52–76, 1934.

[20] D. Kalra, N. Ferguson and Stentz. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proc. of the 2005 Int. Conf. on Robotics and Automation*, 2005.

[21] R.L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* Cambridge University Press, 1993.

[22] O Koifman, G. Shehory and A. Gal. Negotiation-based price discrimination for information goods. In *Proc. of the Int. Conf. on Autonomous Agents and Multi-Agent Systems*, 2004.

[23] C.L. Rogova G.L. Steinberg A.N. Waltz E.L. Llinas, J. Bowman and F.E. White. Revisions and extensions to the jdl data fusion model ii. In *Proceedings of the Seventh International Conference on Information Fusion*, volume 2. International Society of Information Fusion, June 2004.

[24] M. Loder, T. Van Alstyne and W. Wash. An economic answer to unsolicited communication. In *Proc. of the rth ACM Conf. on Electronic Commerce*, 2004.

[25] P. Markopoulos and J. Kephart. How valuable are shopbots? In *Proc. of the 1st International Conf. on Autonomous Agents and Multi-Agent Systems AAMAS'02*, 2002.

[26] P. M. Markopoulos and L. H. Ungar. Pricing price information in e-commerce. In *Proc. of the ACM Conference on Electronic Commerce*, 2001.

[27] S. Matsubara. Adaptive pricing that can withstand buyer collusion of false-type declaration. In *Proc. of the 3rd ACM Conf. on Electronic Commerce*, 2001.

[28] C. Menger. *Principles of Economics.* New York University Press, 1976.

[29] M. Mostaghimi. Bayesian estimation of a decision using information theory. *IEEE Transactions on Systems, Man and Cybernetics*, 27(4):506–517, 1997.

[30] B Ollman. *Economics as a Social Science: Readings in Political Economoy*, chapter What is Marxism?, pages 106–111. Pluto Press, 2003 2nd Ed.

[31] C. Papadimitriou and M. Yannakakis. On the value of information in distributed decision-making. In *Proc. of the 10th ACM Symposium on Principles of Distributed Computing*, 1991.

[32] T. Peterson and M. Sundareshan. Information value mapping for fusion architectures. In *Proc. of the 3rd International Conf. on Information Fusion*, 2000.

[33] R. Radner. Team decision problems. *Annals of Mathematical Statistics*, 33:857–881, 1962.

[34] S. Rafaeli and Raban D. R. The subjective value of information: The endowment effect. In *Proc. of the IADIS Int. Conf. WWW/Internet*, 2003.

[35] D. Ricardo. *On the Principles of Political Economy and Taxation.* John Murray, London, 1821, 3rd Ed.

[36] G. N. Saradis. *Entropy in Control Engineering.* World Scientific, Singapore, 2001.

[37] D. Bagnell D. Schneider, J. Apfelbaun and R. Simmons. Learning opportunity costs in multi-robot market based planners. In *Proc. of the Int. Conf. on Robotics and Automation*, 2005.

[38] T. Sheridan. Reflections on information and information value. *IEEE Transactions on Man and Cybernetics*, 25(1):194–196, 1995.

[39] A. Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations.* Methuen & Co. 5th Ed., 1904.

[40] T. Wahnfried U. Meissen, S. Pfennigschmidt and K. Sandkuhl. Situation-based message rating in information logistics and its applicability in collaboration scenarios. In *Proc. of the 30th EUROMICRO Conf.*, 2004.

[41] M. W. Van Alstyne. A proposal for valuing information and instrumental goods. In *Proc. of the 20th Int. Conf. on Information Systems*, 1999.

[42] R. Zlot and R. Stentz. Market-based multirobot coordination using task abstraction. In *Proc. of the 4th Int. Conf. on Field and Service Robotics*, 2003.