David D. Sworder · John E. Boyd

Locating, Classifying and Countering Agile Land Vehicles

With Applications to Command Architectures

Springer

Locating, Classifying and Countering Agile Land Vehicles

David D. Sworder • John E. Boyd

Locating, Classifying and Countering Agile Land Vehicles

With Applications to Command Architectures

Springer

David D. Sworder
Department of ECE, UCSD
University of California, San Diego
La Jolla, CA, USA

John E. Boyd
Cubic Defense Applications
San Diego, CA, USA

ISBN 978-3-319-19430-1 ISBN 978-3-319-19431-8 (eBook)
DOI 10.1007/978-3-319-19431-8

Library of Congress Control Number: 2015944259

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

In many modern tracking applications, target detection and placement are performed to support some specific decision or action. Estimates of target position and motion will be used for some command purpose, perhaps beginning with determining target intent and then selecting an appropriate course of action like intercepting or otherwise countering the target. The challenge is to construct an architecture that uses noisy data sequences of indicated location not only to derive estimated position, velocity, and attitude of the target but also to provide the information required to support effective command functions. This book focuses on command architectures—estimation and information structures that can effectively inform useful command decisions. The target engagements we have in view almost always demand that a command architecture function in real time. Frequently, there are strict limits on the interrogation and countermeasure resources expended. Command architectures considerably expand the character and complexity of target trackers, whether relatively simple, like alpha-beta algorithms, or more sophisticated, like extended Kalman filter and multiple-model estimators.

We contrast the performance of command architectures based upon a single-model representation of motion with those based on multiple-model approaches to the same problem. The latter start with the same raw measurement data sets as the former but use more sophisticated situation assessment tools to facilitate the simultaneous tasks of locating, classifying, and countering. Our purpose is to highlight the advantages a hybrid problem formulation provides for the decision-making command algorithms.

We explore several prototypical engagements. In each, we determine the quality of a single-model algorithm and explore its sensitivities when the target is uncooperative and given to abrupt changes in kinematic state. We contrast the estimator's actual performance with that predicted on the basis of the model. We then turn to a hybrid algorithm that integrates an element of situation awareness with the same kinematic data set found deficient in the single-model algorithm. Again the sensitivities are presented. The comparison is done with a view to the requirements of a command architecture and the effectiveness of indicated decisions.

The book is intended for engineers who are tasked with developing command algorithms for applications that must operate in highly variable and ambiguous environments. The cases presented and the architectures considered will also serve to illustrate the crucial dependence of effective command on model complexity, model realism, and estimation quality for real-world plants and systems.

We have included, for the interested reader, detailed analytical developments in an appendix. The appendix also provides an example of a step-by-step implementation of the hybrid estimation algorithm. To function in a wildly changing environment, the hybrid algorithm is necessarily quite intricate. For the architectures in this book, we have chosen to work in 2D (motion in the plane) and, to make the engagements concrete, have referred to the target as a ground object. These constraints can be easily relaxed.

The authors' earlier book, *Estimation Problems in Hybrid Systems* (Cambridge, 1999), explored the mathematical structure of one form of hybrid estimation. This book utilizes that approach in an assortment of different engagement applications. The strict mathematical basis of tracking algorithms is often violated in applications. This book explains and illustrates the engineering judgments that must be employed when the developmental hypotheses are violated.

This work contrasts the performance of command architectures in several particular engagements. In each case, the algorithms are presented the same kinematic data stream. The idiosyncratic engagements are chosen to illustrate the distinctive sensitivities of alternative modeling approaches rather than to provide a detailed performance comparison for a specific application.

The authors wish to thank Cubic Defense Applications for their support in publishing this book and, in particular, for the graphics and editing support of their technical publications department.

La Jolla, CA, USA David D. Sworder
San Diego, CA, USA John E. Boyd
February 2015

Contents

Symbols and Notational Conventions xi

1 A Model for Tracking and Classification 1
- 1.1 Introduction to the Problem 1
- 1.2 A Hybrid State Model of a Maneuvering Target 8
  - 1.2.1 The Modal State 8
  - 1.2.2 Kinematic State 13
- 1.3 The Measurement Process 19
  - 1.3.1 Regime-State Measurement 19
  - 1.3.2 Kinematic-State Measurement 21
- 1.4 State Estimates Using the Gaussian-Sum Distribution 22
- 1.5 Assurance Regions 25
- 1.6 Tracker Architecture 30
  - 1.6.1 The Conditional Distribution of the Target State 30
  - 1.6.2 The Projected Distribution 32
- 1.7 Organization of the Book 33
- 1.8 Conclusion 34

2 Target Location Using the Extended Kalman Filter 37
- 2.1 Introduction 37
- 2.2 The Extended Kalman Filter 40
- 2.3 Examples of the EKF-Tracker 42
  - 2.3.1 The Cooperative Target 42
  - 2.3.2 The Uncooperative Target 45
- 2.4 Range-Only Tracking 47
  - 2.4.1 The EKF-Tracker on a Roadway 47
  - 2.4.2 GDOP and Tracker Accuracy 51
  - 2.4.3 The EKF-Tracker Including the North Node 52
  - 2.4.4 Map-Enhanced Tracking 54
- 2.5 Tracking in the Presence of Spoofing 57
  - 2.5.1 Registration Errors 57

2.5.2 Flagging Registration Errors Using Assurance Regions .. 58
2.5.3 Map-Enhanced Flagging 63
2.6 Conclusions 64

3 Tracking an Agile Target 67
3.1 Introduction 67
3.2 Multi-regime Engagement Model 69
3.2.1 Kinematic Model 69
3.2.2 Modal Model 69
3.2.3 Kinematic Measurement 71
3.2.4 Regime Measurement 71
3.3 The Uni-Model Tracker 73
3.3.1 The Engagement 73
3.3.2 The EKF-Tracker 74
3.4 The Multi-model Tracker 78
3.4.1 The Engagement 78
3.4.2 The GWE Algorithm 80
3.4.3 The GWE-Tracker 82
3.5 Conclusion 88

4 Intercept and Target Prediction 91
4.1 Introduction 91
4.2 The EKF 93
4.2.1 EKF-Tracker 93
4.2.2 EKF-Predictor 94
4.3 The $\mathscr{Y}[k]$-GWE 98
4.3.1 $\mathscr{Y}[k]$-GWE-Tracker 98
4.3.2 $\mathscr{Y}[k]$-GWE-Predictor 100
4.4 The $\mathscr{G}[k]$-GWE 104
4.4.1 $\mathscr{G}[k]$-GWE-Tracker 104
4.4.2 $\mathscr{G}[k]$-GWE-Predictor 107
4.5 Conclusion 109

5 Classification and Target Tempo 113
5.1 Introduction 113
5.2 The Engagement 118
5.3 $\mathscr{G}[k]$-Tracking: Markov or Gamma 120
5.4 Target Classification 124
5.4.1 Friendly or Hostile 124
5.4.2 Common Lifetimes and Regimes 126
5.4.3 Distinctive Kinematics, Range-Bearing Sensors 134
5.5 Conclusion 136

6 Assurance Regions on a Road Grid 141
6.1 Introduction 141
6.2 The Engagement 143
6.2.1 Kinematic Model 143
6.2.2 Measurements 146

6.3 The EKF ... 148
6.3.1 Nominal EKF ... 148
6.3.2 Pseudo-Noise Augmentation ... 150
6.3.3 Tracker Contrasts ... 152
6.4 The Map-Enhanced GWE ... 157
6.4.1 Algorithm Overview ... 157
6.4.2 The GWE-Tracker ... 163
6.5 Conclusions ... 173

7 Action Windows with Resource Limits ... 177
7.1 Introduction ... 177
7.2 Action Windows with a Grid Map ... 180
7.2.1 The Engagement ... 180
7.2.2 EKF-Illuminator ... 182
7.2.3 GWE-Illuminator ... 185
7.3 Ballistic Fire Control ... 190
7.3.1 The Engagement ... 190
7.3.2 EKF Fire Control ... 191
7.3.3 GWE Fire Control ... 193
7.4 Conclusions ... 198

8 Serendipitous Events in Tracking and Classification ... 201
8.1 System Specifications ... 201
8.2 The Visible Target ... 203
8.2.1 The Engagement Model ... 203
8.2.2 Omnipresent EKF ... 204
8.3 The Veiled Target ... 208
8.3.1 The Engagement ... 208
8.3.2 Blinded EKF ... 209
8.3.3 GWE ... 211
8.3.4 Algorithm Contrasts ... 217
8.4 The Veiled Target with More Ambiguous Para-Measurements ... 219
8.4.1 The Engagement ... 219
8.4.2 Blinded EKF ... 220
8.4.3 GWE ... 222
8.5 Conclusions ... 233

9 Deceiving a Classifier ... 235
9.1 Introduction ... 235
9.2 The Engagement ... 237
9.3 An EKF Classifier ... 238
9.3.1 No Spoofing ... 238
9.3.2 Spoofing ... 240
9.3.3 A Beacon Assist ... 241
9.4 A Hybrid Classifier ... 242
9.4.1 Spoofing ... 242
9.4.2 A Beacon Assist ... 247

9.5 Countermeasure Effectiveness 249
9.6 Conclusions 251

10 The Gaussian Wavelet Estimator 253
10.1 GWE Details 253
10.1.1 Engagement Structure: Kinematics 253
10.1.2 Engagement Structure: Sensor Architecture 255
10.1.3 Hybrid Estimation 255
10.2 A Single-Step Example of the GWE 266
10.2.1 Introduction 266
10.2.2 Kinematic Extrapolation 267
10.2.3 The Measurements 268
10.2.4 The GWE Update 269
10.2.5 The Location Estimate 273
10.2.6 Reinitialization 276
10.3 The Map-Enhanced GWE 277
10.3.1 The (i44)-Local Estimator 278

References 289

Symbols and Notational Conventions

Precise description, the defining and comparison of models, estimators, and command architectures requires rather complex notation. The authors have attempted to enhance accessibility of this book by following particular notational conventions. This section briefly discusses these principles and spells out the usage or meaning of many of the symbols used in the text. Some pronunciation hints are provided.

The work, and the definitions given here, assume familiarity with mathematical analysis of probability and stochastic processes and with estimation of dynamical systems. A search of the Internet may yield useful reviews and quick definitions where the authors' assumptions are optimistic.

Notational Conventions

- Font styles are used to distinguish related concepts. For example, x is used for the kinematic state 4-vector of a target moving in the plane. The Greek letter chi, χ, is used to denote the location coordinate 2-vector of such a target.
- Math calligraphic style is used for collections, especially evolving time series collections, of information such as filtrations.
- Diacritical and accent marks name a quantity derived from and related to a variable. For example, "hat" of x, $\hat{x}$, indicates the estimated value (mean value) of the variable x.

Symbology

Operators

$\hat{x}$ A carat set above the symbol for a variable, e.g., $\hat{x}$, is used to denote the estimated or mean value of the variable, in this case x. [“x hat”]

$\tilde{x}$ A tilde set above the symbol for a variable, e.g., $\tilde{x}$, is used to denote the error in an estimate of the variable. In this case $\tilde{x} = x - \hat{x}$. [“x tilde”]

$\vee$ The filtration generated by the union of two filtrations, e.g., $\mathscr{Y}[k] \vee \mathscr{Z}[k]$. [“vee”]

$\boldsymbol{E}$ Mathematical expectation.

$|$ In an expression for a random variable, the vertical bar, $|$, denotes a conditional. For example, $\mathbb{P}(x\,|\,y)$ is the probability of x given the condition y. [read as “given”]

diag The main diagonal of a square matrix or, if D is an n-vector, $\mathrm{diag}(D)$ is the n by n diagonal matrix of D.

cov Covariance.

$\triangleq$ Defined. For example, $\mathrm{cov}(x) \triangleq E[\tilde{x}\tilde{x}']$ means $\mathrm{cov}(x)$ is defined as the covariance of the error in x.

$\mathbf{H}^{+}$ The mapping by which the tracker’s kinematic state estimate is adjusted to accommodate the regime for the next time step.

$\mathbf{C}^{\mathscr{M}}$ The mapping by which a kinematic state estimate is returned to the closest point on the map. Putting the target position onto the nearest road.

Estimators

KF Kalman filter

EKF Extended Kalman filter. Any version of the Kalman optimal linear estimator that is applied to a problem that does not meet the linearity, independence, and Gaussian assumptions required by the Kalman filter. A common example in this work arises from the conversion of range-bearing measurements of target position to cartesian coordinates using as the conversion reference point the estimated target position, violating the independence assumptions required by the KF.

GWE Gaussian Wavelet Estimator.

Filtrations and Information Sets

$\mathscr{F}$ A filtration, an increasing sequence of σ-algebras. ["F"]

$\mathscr{Y}$ The filtration generated by a time sequence of kinematic measurements, i.e., position and velocity. ["Y"]

$\mathscr{Z}$ The filtration generated by a time sequence of modal, or regime, observations. ["Z"]

$\mathscr{G}$ The composite filtration of $\mathscr{Y}$ and $\mathscr{Z}$: $\mathscr{G} \triangleq \mathscr{Y} \vee \mathscr{Z}$. ["G"]

$\mathscr{O}[k]$ Filtration generated by the exogenous process of system modes or regimes. ["O"]

$\mathscr{M}$ The set of all information contained in a map: road locations, nominal local speeds, sensor obstructions, intersections, and the like. $\mathscr{M}$ contains all the information known to the estimator but is not a filtration. ["M"]

Symbols

$\mathbf{1}$ Vector of 1s.

$\mathbf{I}$ The identity matrix.

$\boldsymbol{\alpha}$, $\boldsymbol{\alpha}_t$ Often used in a Gaussian sum as the normalized or unnormalized modal distribution.

$\mathbf{D}$ Discernibility matrix: the inverse of a covariance matrix.

$\mathbf{e}_i$ The ith unit vector.

$\mathbf{E_i}$ The ith unit matrix. If $i \leq n$, $\mathbf{E_i} = \text{diag}(\mathbf{e}_i)$.

$\boldsymbol{\iota}$ A regime sequence of length L.

$\boldsymbol{\iota}^+$ A regime sequence of length $L + 1$.

$\boldsymbol{\kappa}$ The modal set. The set of possible regime sequences of length L.

$\boldsymbol{\phi}$, $\boldsymbol{\phi}_t$, $\boldsymbol{\phi}[k]$ Our convention is to use ϕ to represent the mode or regime of a hybrid process.

$\mathbb{P}$ Probability. Used for the probability distribution function of a random variable.

$\mathbb{P}^\sim$ The projected distribution. A Gaussian approximation to $\mathbb{P}$ matching $\mathbb{P}$'s first two moments. ["P tilde"]

$\boldsymbol{P}_{xx}$ Covariance of the kinematic state, x.

$\boldsymbol{R}_x$ Covariance of the sensor noise process.

$\boldsymbol{\varrho}$ The distance between two points. For example, the distance between an expected position measurement and its actual value. Also used for Mahalanobis distance, the distance of a point from a distribution, typically measured in units of standard deviation. [“rho”]

S The regime index. The total number of regimes is S.

$\boldsymbol{\nu}$ The velocity components of a kinematic state.

$\boldsymbol{x_t}$ The value at time t of the continuous-time kinematic state process x.

$\boldsymbol{x[k]}$ The value at time kT of the discrete time kinematic state process x at $t = k$T.

$\boldsymbol{\chi}$ The location coordinates of a target in a 2D plane. Thus the target kinematic state x is the “stack” of χ and ν: $x_t = \begin{bmatrix} \chi_t \\ \nu_t \end{bmatrix}$.

Chapter 1
A Model for Tracking and Classification

Abstract In modern system architectures, the command algorithm must do more than merely locate a point equivalent of an agile target. The algorithm must place the target, classify it, and, if need be, counter any threat. The command algorithm designer must produce a causal transformation that not only tracks the motion but also places an assurance window about the target for interrogation and perhaps neutralization. When the target is uncooperative, this requires a sophisticated multi-model approach. This chapter frames the problem and introduces the notation used in what follows.

1.1 Introduction to the Problem

System integration requires an understanding of the system's basic capabilities along with a sense of the interrelationships of the system states and the influences of the operational environment. The system architect begins with an analytical representation of the internal and external excitations and infers the system response based upon these processes. In a practical problem, the system is required to accomplish some specific task with limits on available resources. For example, the system must direct countermeasures from a limited inventory toward an unwelcome object passing toward a secured location. In another example, the system must locate and visually classify an object of indeterminate intent. In both examples, the command system must locate a dynamic object and then make a decision based on the quality of its location estimate. This book presents a study of command architectures for estimation and resource allocation. In the context of several rather unadorned engagements, we will discuss the sensitivities of command algorithms to simplified environmental models.

Over at least the past century, model-based approaches have proven useful in system integration. Such approaches broadly require:

- An analytical model of the system that includes all interfaces with the outside world. This system model relates objective inputs and outputs to an internal group of dynamic states that produce the distinctive system behaviors. In the target tracking problems we consider in this book, these behaviors are primarily the target kinematics. In this context, *state* has a specific meaning: a set of internal

D.D. Sworder, J.E. Boyd, *Locating, Classifying and Countering Agile Land Vehicles*, DOI 10.1007/978-3-319-19431-8_1

variables that allow the external valuables to be extrapolated forward in time. For simplicity in architectural design, however, the model is a reductionist abstraction that creates a notional future that may differ significantly from the actualized future.

- An observational model that delineates the situational measurements available to the command system. This model not only lists the elements of this data stream, but it also provides the timing and the quality of the individual components. Some architectures are able to avail themselves of more information than others. We want to quantify the utility of auxiliary measurements to insure the cost of acquiring them is reflected in their utility.
- A causal algorithm selected to map the observations into an estimate of the relevant states of the system. This, along with a high quality measure of the reliability of the estimate, permits rational command decisions to be made. Based upon the algorithm, countermeasures can be deployed autonomously, action agents can be enabled.

Command architectures require that we must first locate the object of inquiry (the target). This part of the problem is frequently referred to as target tracking. A vehicle is moving within some fixed motion space and we must find it. The commander may want to classify the target as friendly or hostile, predict the future position of the target, counter the target at an opportune time, etc. For example, the target vehicle may be a car following a road path toward a designated area, and the engineer seeks an algorithm that provides current position and classifies the vehicle as friend or foe. In another example, the vehicle is an airplane moving toward a protected area, and the engineer seeks an algorithm that classifies the airplane's intent and selects appropriate countermeasures. In both cases, we wish to determine where the vehicle is, and where it is likely to go. In the target tracking problem, the system is the moving vehicle, its model is a mathematical representation of how its observable outputs (position, velocity, perhaps orientation or current maneuver mode) derive from its inputs like operator control (if known), random influences, and constraints (e.g., roads). We assume the system outputs are measured with quantifiable accuracy and frequency. We require an implementable algorithm (the *estimator* or *tracker*) that maps a measurement sequence into an estimate of the current (or future) state of the vehicle.

For the tracking sub-architecture (the system estimator), recognition that not all target classes move and maneuver in the same way can greatly improve tracker accuracy and command effectiveness. There may be different kinds of targets in view, and any one target might have multiple maneuver modes. Trucks and motorcycles turn and accelerate differently. An aircraft may spend much of its time in constant velocity flight, but in turns its motion is better modeled as constant lateral acceleration. A tracker that can account for these differences is greatly aided by a system model that acknowledges the different classes and modes. In a comprehensive model of the engagement, the *kinematic* states of the system are augmented by *regime* states that correspond to target type and motion mode.

Fig. 1.1 A tank is moving on an irregular terrain

A carefully constructed tracker will take the regime process into account, and a classifier, if available, can greatly aid estimation in this hybrid environment.

Tracking is enhanced by the fact that target motion occurs within a context. For example, consider a tank moving in the open area as shown in Fig. 1.1. The basic information set available to the tracker is a sequence of such images. From the camera gimbal angle, the bearing to a point object is measured. But there is far more information implicit in the image process. For example, we might classify the target by comparing the measured image with a list of stored images of like type; e.g., Fig. 1.2. A fit metric quantifies a class distribution. Further processing could then yield the longitudinal orientation of the vehicle. Range could be inferred from the size of the image. All of these target properties flow from careful image processing. From these rather ambiguous image interpretations, the command algorithm must generate a region within which the target will be found and an action appropriate to the encounter. Commonly, we split this problem into sub-problems: identify the target type and intent, neutralize the target if appropriate.

Figure 1.1 is an optimistic case in which the target is contrasted with an undifferentiated backdrop. The target bearing can be measured. Its range can be estimated from its size. Direction can be estimated from its aspect angle. Of course, all of these attributes are subject to uncertainty.

Even the existence of a target in the field of view may not be a sure thing. Figure 1.3 shows a situational image in which there are no targets of interest. But simple edge or contrast detection algorithms might well identify many targets. Hence, in a complex engagement, the classifier must be aware of the possibility of false detections.

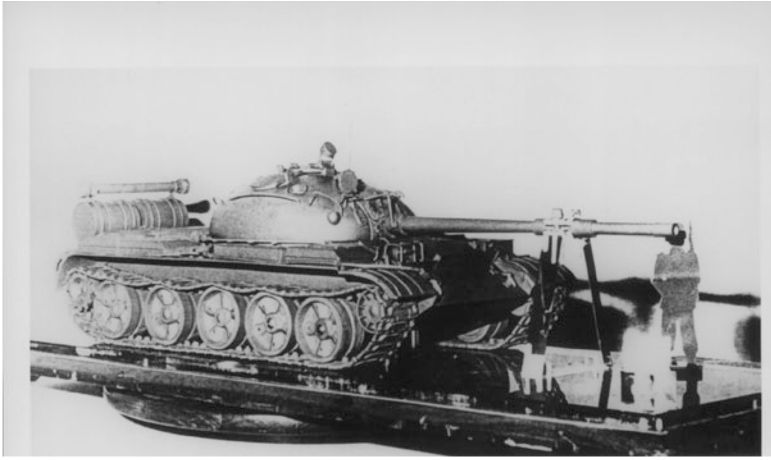

Fig. 1.2 A single template in an archive of possible targets

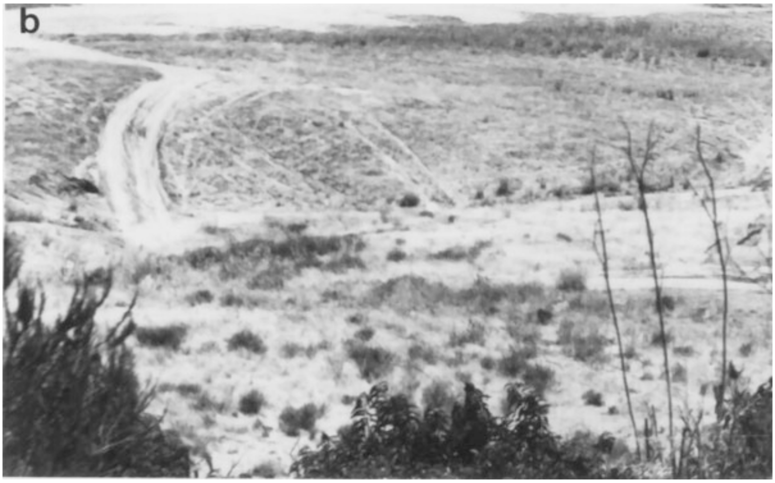

Fig. 1.3 An image in which there are no targets of interest

Figure 1.1 provides a clear visual image of the target. But other sensors are not nearly as sharp. Figure 1.4 is a FLIR-image of a tank. The engine is the differentiated sub-region of the figure A sequence of FLIR images is more difficult to interpret because of the low contrast.

The kinds of measurements used in tracking and prediction are of various sorts: range-only (GPS), range-bearing (radar), optical target image (TV), infrared target image (FLIR), etc. The measurements may yield information on a spatially distributed object (an optical silhouette), or they may be reduced to information about a point-equivalent target (range to the center-of-reflection).

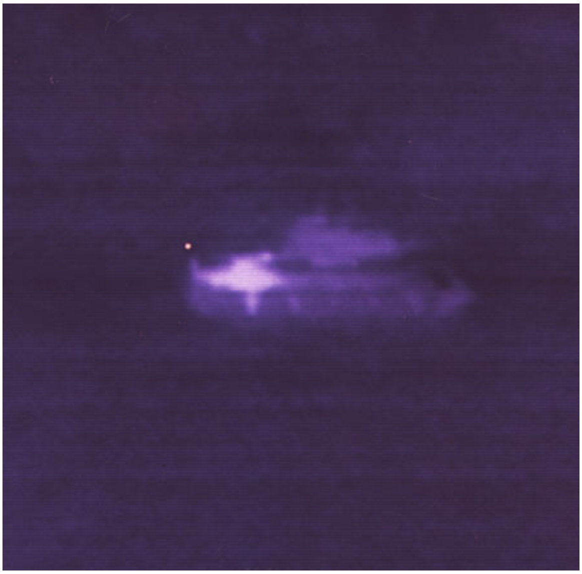

Fig. 1.4 A FLIR image of a tank. The engine in the main region of contrast. At some aspect angles, the engine is obscured

Other relevant tracking information is not explicitly temporal: a topographic map may display regions of possible motion along with prohibited regions. Or a road profile could be accessible by the tracker with the intent of correlating the measured motion with the profile [11]. Broadly, we will refer to such non-temporal resources as an inventory of para-measurements. Para-measurements will be integrated with the conventional kinematic measurements to produce the final actions in an engagement.

To illustrate some of the algorithm development issues, consider ballistic missile defense. A tracker/classifier identifies an approaching ballistic target (BT) within a specified category list. Impact point prediction (IPP) is accomplished by extrapolating the estimated kinematic state of the BT forward to impact. Accurate IPP requires rapid target classification, a high quality motion model, and good missile state estimates during flight.

A sophisticated tracker will do more than simply provide IPP. It will compute complementary performance metrics including bias (in this example, the mean deviation of impact point from a missile aim-point), and the *circular error probable* (CEP). The CEP is a circle of such radius that it captures 50 % of the impacts. The use of CEP assumes that the actual impact points are distributed over a roughly circular area. Thus, a comprehensive figure of merit for a BT tracker would involve both the magnitude of the point-placement bias and the size of the dispersal region (the CEP).

In the applications we will investigate, the uncertainty regions in tracking and prediction are not well modeled as circular. The target may have preferred directions as it is constrained by internal and external factors. For this reason, we will generalize CEP, and talk about *elliptical error probable* (EEP). The EEP ellipse is such that it captures the target with probability 50 %.

In a command architecture, the estimator's inferred accuracy influences the actions that follow. We may deploy countermeasures to intercept the target with a probability of, say, 90 %. A flawed calculation of EEP may lead either to a too little cover area and a failed intercept or to a too big cover area with a waste of limited resources.

For reasons developed above, the self-reported estimate of tracker accuracy, whether CEP, EEP, error-covariance matrix, or other metric, assumes considerable importance in the command architectures. Even in Kalman filters and their kin, the update of target location estimate uses a filter gain (a multiplier of the difference between the extrapolated observation and that measured) to adjust target placement. The gain depends upon the measurement noise statistics and the uncertainty in the filter's extrapolated state. If the latter is overly large, the filter gain is large and the location estimate volatile. If the latter is overly small, the filter gain is small and the location estimate has not the power to follow rapid changes in direction.

In the engagements we will explore in the sequel, the system uncertainties cannot be expressed in terms of the second order statistics of the tracking error. We will encounter multi-mode location densities that will test our engineering judgements. We will use the EEP as a broad confidence measure. But precise asset allocation will require a more nuanced view of situational awareness.

Model-based trackers integrate the measurement aggregate (including para-measurements) into an estimate of the state of a target using an engagement model as an intermediary. The first part of the model, the motion model, describes the temporal evolution of the target's kinematic state.

Within the motion model, the engineer distinguishes the idiosyncrasies of various target types: nominal speed and turn rate, location constraints, number and coordination of vehicles, etc. For example, the target shown in Fig. 1.5 is distinguishable from the tank in Fig. 1.1 both in its maneuverability and terrain capability. A kinematic model for the former would not be appropriate for the latter, and if used, estimator accuracy is likely to be degraded.

The second part of the model—the measurement model—describes the type and quality of the measurements. In addition to the temporal measurements; e.g., GPS pseudo-range, this sub-model contains a description of the para-measurement aggregate.

The tracker architecture depends on the cooperation, or lack thereof, of the target itself. A cooperative vehicle will declare its type and intent to the tracker; e.g., an aircraft in an air-traffic-control application is cooperative insofar as it gives the tracker its identification number and flight plan. Type and intent information is useful for reducing the uncertainties in the kinematic state estimate. For example, the tracker is cued to look for particular flight patterns when intent is signaled. The cooperative target is the most commonly studied because classificational ambiguities are avoided in the engagement model.

Alternatively, tracking an uncooperative target is more difficult because important global identifiers of the engagement are missing. If such qualities are required,

Fig. 1.5 Shape analysis allows the tracker to identify the specific target type and match the kinematic model with the mobility of the target

they must be inferred concurrently with state estimation. Indeed, some uncooperative targets make location prediction difficult by deliberately masking relevant identifiers; e.g., the hostile target masks intent. In this circumstance, the tracker/classifier must predict position by inferring intent from the motion patterns of the target.

In this book, we will look at the problem of tracking and classifying an agile ground target. We will see that the cooperative–uncooperative dichotomy oversimplifies many engagements; e.g., a cooperative target may wish to signal intent to the tracker but the data link is not adequate for the task. We will, therefore, consider tracking algorithms in a mixed environment in which the classificational attributes of the target may be only partially known. The objective is to present high quality, implementable (recursive) estimators of both the kinematic state and also such category variables as are relevant. The form that the estimator takes depends upon the sensor architecture used and the para-measurements available.

The performance of the proposed algorithms will usually be illustrated with an example of planar motion. A 2D motion space is rich enough to display important issues in tracker synthesis without becoming so complex as to become tedious. A space of higher dimension has the same broad character as 2D motion, but the trackers become more complex in appearance. Unfortunately, even in 2D, the notation required to describe the tracking environment is more convoluted than is typical in estimation theory. The next sections introduce some conventions used in the sequel and explain their need.

1.2 A Hybrid State Model of a Maneuvering Target

The concept of the target state conveys the notion of predictability. For example, the target state at time t_0 is the information required to uniquely determine the output at $t > t_0$ given the forward excitation [2]. The extrapolation requires an analytical model that relates the relevant processes. We will call this the kinematic model.

In this book we will consider vehicles that move in an n-dimensional Euclidean state space with kinematic state vector x_t. The state has position components, χ_t, and velocity components, ν_t along with such other dynamic variables as are appropriate. If the target is moving in 2D, χ_t is two-dimensional as is the velocity. Necessarily, n is at least 4. But n will be larger than 4 when the target model moves in a higher dimensional space or when the target has other relevant states like acceleration, actuator dynamics, and so on. The totality of these vector components is called the *global kinematic state* of the target. For convenience, we will stack position over velocity at the top of x_t.

Extrapolation is based upon a model of the dynamic properties of the target. But such properties depend upon the type of target we are following. A large truck has not the agility of a motorcycle even though both may traverse the same road course. Either could be described by a simple Newtonian state model, but the coefficients in the equations would be different. So when we develop a command architecture, we must first settle on the class of targets we can expect to see in a particular engagement.

To impose a structure on the exercises that follow, we will need a notational convention flexible enough to delineate uncertainty in both situational and kinematic conditions. The former we call the regime state; the latter, the kinematic state.

1.2.1 The Modal State

1.2.1.1 The Operating Regime

We will have more to say about x_t in the next section. But for now we observe that sophisticated engagement models acknowledge that the target vehicle (or vehicles) will confront different macro-conditions as the engagement evolves. We use a discrete variable, the regime state, ϕ_t, to point to the current situational status of the engagement [19].

For example, suppose the engagement involves tracking a moving vehicle. The target could be an M1 Abrams tank or it could be a motorcycle. The tracker must distinguish between two macro-conditions: regime #1 ($\phi_t = 1$) if the target is an M1 Abrams tank; regime #2 ($\phi_t = 2$) if the target is a motorcycle. (We will frequently use a right arrow to indicate such mappings, and later, transitions and implications. Here, $\phi_t = 1 \mapsto$ the target is a tank.) The agility and directional restrictions on these two vehicle classes are very different, and they must be distinguished in the kinematic model.

In contrast to the kinematic state vector, the regime index ϕ_t is a pointer. We call ϕ_t the regime state (though it may not have the predictive properties we want from a state variable), and assume it has a finite range. As an example, there might be a list of possible targets, each distinguished by its own kinematic model. ϕ_t points to the particular kind of target in view. More generally, for any kind of regime, we can list the S alternatives. Let $\mathbf{S}$ denote the sequence of the first S natural numbers: $\mathbf{S} = \{1, 2, \ldots, S\}$. Then the ith regime is identified by $\phi_t = i \in \mathbf{S}$. The regime state is identified with a number in a counting system with radix S. In this framework, $\mathbf{S}$ is called the regime alphabet, and it represents an ordered list of possible regimes. In the (tank, motorcycle) scenario, we make the identification: $\phi_t = 1$ means the target is a tank; $\phi_t = 2$ means the target is a motorcycle. The kinematic extrapolation equation utilizes $\phi_t = i$ where i is chosen appropriately.

The regime structure can be described in another way. Let $\mathbf{e}_i$ be the ith canonical unit vector in $\mathbb{R}^S$; e.g., $\mathbf{e}_1 = (1, 0, \ldots, 0)'$. We can associate the ith unit vector with the regime, the target is of type i: $\mathbf{e}_i \mapsto$ target is type i. The regime being $\mathbf{e}_1$ means the target is a tank. For expositional simplicity, we will use the vector and numerical designations interchangeably and without comment: $\phi_t = \mathbf{e}_1$ and $\phi_t = 1$ are alternative labels for the statement that the target is a tank. The modal primitive is classificational and has none of the common analog properties of the conventional kinematic state; e.g., $\mathbf{e}_1 + \mathbf{e}_2$ is not a regime because it is not a unit vector. So the regime space is not a vector space in the usual mathematical sense.

When the regime variable is target type, the interpretation of ϕ_t is unambiguous; the target is a tank or it isn't. In many applications, however, the separation of local- and macro-conditions is not so clear cut. Suppose the target is a motorcycle constrained to follow a rectangular road grid aligned to the four cardinal directions. With the target type known, the regime space covers travel north (N or ϕ_t=1), travel south (S or ϕ_t=2), travel east (E or ϕ_t=3), travel west (W or ϕ_t=4) with perhaps *stop* appended. So if $\phi_t = \mathbf{e}_1$, the north-specific motion model should be used with strong longitudinal accelerations and weak lateral accelerations. In this example, ϕ_t is a pointer to a velocity within four directional bins. But *north* is an analog variable in that north motion includes a range of velocities that are only close to true north. The ambiguity concerning direction of motion within the $\phi_t = \mathbf{e}_1$ bin must be transferred to the kinematic state which contains velocity as a component. In this case, we see that the vagary of direction is masked in our simplified definition of the regime.

The regime can point to composite events. For example, $\{\mathbf{e}_1, \ldots, \mathbf{e}_4\}$ may correspond to the tank moving on the (north, south, east, west) grid; e.g., $\mathbf{e}_1$ implies a tank is headed north. Suppose $\{\mathbf{e}_5, \ldots, \mathbf{e}_8\}$ corresponds to a motorcycle moving on the same motion space. In this case, the regime space with two target types and four directions is partitioned into eight bins.

If the regime is target type, ϕ_t is unchanging. In other applications the macro-conditions change during the engagement; e.g., the northbound motorcycle may come to a road junction and turn east. The range of the regime state is still $\mathbf{S}$, but ϕ_t may vary in time: $\phi_t = \mathbf{e}_5$ transitions to $\phi_t = \mathbf{e}_7$ at the turn. We will suppose that the regime process, $\{\phi_t\}$, is constant with isolated discontinuities. When $\{\phi_t\}$ changes from the jth regime state to the ith state, $\Delta\phi_t = \phi_t - \phi_{t^-} = \mathbf{e}_i - \mathbf{e}_j$. We will say that ϕ_t experienced a transition event at time t. The regime process is a right-continuous temporal process with range space $\mathbf{S}$.

The global state of the target evolves on some time interval $[0, \mathsf{T}]$. But in the work that follows, the tracker/classifier will be designed on the basis of a time-discrete approximation to the state process. Let T be a sample interval. The time-continuous regime process is reduced to a sequence, $\{\phi[k] = \phi_{kT}; k \in \{0, 1, 2, \ldots\}\}$. If the time interval of the engagement is not specified, the complete regime process becomes a string of arbitrary length—or a radix S number with an arbitrary number of digits.

We will call $\{\phi[k]\}$ the regime event process. If $\phi[k+1] \neq \phi[k]$, there has been a regime transition in the interval $(kT, (k+1)T]$; if $\phi[k+1] = \phi[k]$, the regime continues unchanged—at least at the sample times.

There are some important modeling ambiguities that arise when we approximate a time-continuous process with a time-discrete representation. The discontinuities in the regime process can occur at any time in $[0, \mathsf{T}]$. When we place these discontinuities at the sample times, we ignore the influence on system of the precise time of a change in regime. We also ignore multiple transition events within an interval; e.g., $\phi[k] = \phi[k+1]$ does not imply $\phi_t \equiv \phi[k] \in (kT, (k+1)T]$. In what follows, the consequences of these temporal uncertainties is implicit, but T will be assumed to be small enough that we can aggregate all of these effects within the basket of exogenous uncertainties.

An issue of more subtlety is the choice of the principal modal condition over a sample interval. If there is a regime transition in the interval, when did it occur? Using our discrete time process model and notation, the time-continuous process $\{(x_t, \phi_t)\}$ is mapped to $\{(x[k], \phi[k])\}$. The vector $(x[k], \phi[k])$ appears as an initial condition in a kinematic extrapolation formula for the forward step. However, the actual mapping $x[k] \mapsto x[k+1]$ is an explicit function of the regime (or regimes) of operation over $t \in [kT, (k+1)T)$. The initial value $\phi[k]$ does not capture the portion of the kth-time interval during which the regime is actually $\phi[k]$. Within this interval, perhaps $\phi_t = \phi[k]$ for a long time. Or perhaps $\phi_t = \phi[k+1]$ is dominant—or, in the case of multiple modal transitions in a single interval, even a regime different from either. The temporal partition precludes a nuanced description of the intra-interval variation in the regime. With this is mind, we will suppose that multiple intra-sample transitions occur infrequently, and further, that the initial regime will serve as a proxy over the whole of the interval at least as concerning the kinematic extrapolation to $x[k+1]$. Of course, this adds to our basket of uncertainty.

1.2.1.2 Regime Sequences and Languages

Extrapolation in the command architecture depends upon both where you are and where you have been—the future depends upon the past and not just the present. Our regime model requires concepts and notation for the past–present–future of the regime state. Focusing on the kth intra-sample interval, the regime proxy is $\phi_{kT} = \phi[k] = \mathbf{e}_i$. The kinematic extrapolation uses the ith primitive over the full interval, $[kT, (k+1)T)$. Label the principal regime in the preceding interval $\mathbf{e}_j$, and the successor regime by $\mathbf{e}_p$. Then $[\mathbf{e}_p, \mathbf{e}_i, \mathbf{e}_j]$, also represented as pij, is a regime string centered on $[kT, (k+1)T)$, representing a segment of state history in reverse time. We can view pij as a word with alphabet $\mathbf{S}$, or a three-digit number with radix S. In this context, we will use "word," "number," and "string" interchangeably. If the target is a motorcycle moving on the usual road grid, the event 144 points to the motorcycle making a north turn from the west at $t = (k+1)T$.

Our notation differs from that used to describe languages in automata [2]. In that convention, the state sequence is listed from left to right with the oldest event to the left; e.g., in the above illustration, the motorcycle motion would be labeled 441 or (west, west, turn north). We wish to emphasize the current and future regimes, and we will do so by placing them at the left of the regime sequence. Our strings are retrograde when contrasted with the usual automata sequence labels.

We will have frequent occasion to consider these sequences of regimes. Let us define three sets of interest in this application. Let L be an integer memory length in the application. Let $\boldsymbol{\kappa}_L$ (which we shall usually abbreviate as $\boldsymbol{\kappa}$) be the set of all S-radix, L-digit numbers with elements $\boldsymbol{\iota}$. Define $\boldsymbol{\kappa}^+ = \boldsymbol{\kappa}_{(L+1)}$ to be the set of all S-radix, $(L+1)$-digit numbers with elements $\boldsymbol{\iota}^+$, and define $\boldsymbol{\kappa}^- = \boldsymbol{\kappa}_{L-1}$ to be the set of all S-radix, $(L-1)$-digit numbers with elements $\boldsymbol{\iota}^- \in \boldsymbol{\kappa}^-$. For example if $L=3$, $\boldsymbol{\iota} = 144 \in \boldsymbol{\kappa}$ and $\boldsymbol{\iota}^+ = 1144 \in \boldsymbol{\kappa}^+$ (west, west, turn north, north), reading right to left as is our convention. In this case $\boldsymbol{\iota}^- = 44 \in \boldsymbol{\kappa}^-$. Here $\boldsymbol{\iota}$ is a word in $\boldsymbol{\kappa}$, $\boldsymbol{\iota}^-$ is a word in $\boldsymbol{\kappa}^-$, and $\boldsymbol{\iota}^+$ is a word in $\boldsymbol{\kappa}^+$.

As we model the evolving system behavior, we will refocus on the next sample time, $t = (K+1)T$, and we will consider a sequence of regimes of length $L+1$ moving back from $t = (k+2)T^-$. We can indicate this string by $\boldsymbol{\iota}^+ = [\mathbf{e}_p, \mathbf{e}_i, \mathbf{e}_j, \ldots, \mathbf{e}_r, \mathbf{e}_l] = pij\ldots rl \in \boldsymbol{\kappa}^+$ where $\boldsymbol{\kappa}^+$ is the set of all words of length $L+1$ from the alphabet $\mathbf{S}$. Again, $\boldsymbol{\iota}^+$ is a word in $\boldsymbol{\kappa}^+$.

The augmented regime strings can be designated with numbers or unit vector arrays. If $\boldsymbol{\iota}^+$ is an $S \times (L+1)$ array of unit vectors, it can also be represented as an $(L+1)$-digit, S-radix number, or a word of length $(L+1)$, or a string of regimes of length $(L+1)$. The set $\boldsymbol{\kappa}^+$ is the collection of all such regime strings however they are indicated. For example, if $L = 4$, then $\boldsymbol{\iota}^+ = [\mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_3] = (11113)$ identifies the event sequence $[\phi[k+1] = \mathbf{e}_1, \phi[k] = \mathbf{e}_1, \phi[k-1] = \mathbf{e}_1, \phi[k-2] = \mathbf{e}_1, \phi[k-3] = \mathbf{e}_3]$. In the illustration in which the regimes are direction events, $\boldsymbol{\iota}^+ = [\mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_3]$ describes a north-bound motion coming from the east: the vehicle was E-bound four samples ago but then turned and continued north. In this illustration, $\boldsymbol{\kappa}^+$ contains all possible directional sequences over an interval of length $5T$.

To limit the complexity of the command algorithms we shall consider, we limit the memory length L. At each sample time, before proceeding, we will reduce the augmented regime sequence by "forgetting" the oldest regime. To illustrate, suppose that at the current time the vehicle is moving north after a eastbound interval. With an $L = 4$ memory, $\boldsymbol{\iota} = [\mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_3]$. Now suppose the vehicle continues north: $\boldsymbol{\iota}^+ = [\mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_3]$. As we move forward in time, our event memory must truncate ι^+. We do so by dropping the oldest (rightmost) regime from ι^+ to obtain the updated $L = 4$ regime history, $[\mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_1] \mapsto \iota$. In this example, command algorithm is no longer aware of the previous east motion.

1.2.1.3 Regime Dynamics

Change in system mode or regime is unpredictable. All of the random processes[1] that follow are defined on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and time interval $t \in [0, \mathsf{T}]$. These random processes are adapted to a common right-continuous filtration $\{\mathscr{F}_t; 0 \leq t \leq \mathsf{T}\}$. Specifically, the regime process, $\{\phi_t\}$, is constant in the main, but isolated events cause abrupt changes.

To first order, the regime state will be thought to be exogenous—though much more will be said of this later. Specifically, we will suppose $\{\phi_t\}$ is a right continuous, piecewise constant, $\mathscr{F}_t$-adapted Markov process taking on values in the set of canonical unit vectors in $\mathbb{R}^S$ ($\phi_t \in \{\mathbf{e}_1, \ldots, \mathbf{e}_S\}$).

We will represent the incremental behavior of the regime state with an $S \times S$-matrix Q: if $i \neq p$, $\mathbb{P}(\phi_{t+dt} = \mathbf{e}_p \mid \phi_t = \mathbf{e}_i) = Q_{ip}\, dt$ with $Q_{ii} = -\sum_{p \neq i} Q_{ip} > 0$. It is known that the mean sojourn time in state $\phi_t = \mathbf{e}_i$ is $-Q_{ii}^{-1}$, and if $\phi_t = \mathbf{e}_i$, the probability that the next modal transition will be $\mathbf{e}_i \mapsto \mathbf{e}_p$ is $-Q_{ip}/Q_{ii}$. Consequently, Q can be evaluated from observations of the regime process.

The regime state process is described by the stochastic equation

$$d\phi_t = Q'\phi_t dt + dm_t \tag{1.1}$$

with initial condition ϕ_0. The second term in (1.1) is a purely discontinuous $\mathscr{F}_t$-martingale increment: $E\{dm_t \mid \mathscr{F}_t\} = 0$ [4].

For now, we will suppose that the transition rates of the regime process are constant. This restriction will be considerably modified in the context of specific applications. But more of that later. The model of the time-discrete regime state process derives from (1.1). If we sample $\{\phi_t\}$ every T seconds, we have

$$\phi[k+1] = \Pi\phi[k] + m[k+1], \tag{1.2}$$

where $\mathscr{F}[k] = \mathscr{F}_{kT}$, $\{m[k]\}$ is a time-discrete, $\mathscr{F}[k]$-martingale difference sequence, $(E\{m[k+1] \mid \mathscr{F}[k]\} = 0)$, and $\Pi_{pi} = \mathbb{P}(\phi[k+1] = \mathbf{e}_p \mid \phi[k] = \mathbf{e}_i)$ is the modal transition matrix. With a fixed sample interval T, $\Pi = \exp(Q'T)$. The time-discrete regime model may allow events prohibited in (1.1). For example, a transition from $\mathbf{e}_1$ to $\mathbf{e}_2$ may be prohibited—a sudden U-turn from north-to-south is not allowed

[1] Readers seeking an introduction to stochastic processes and terms like martingale, adapted, and filtration will find an abundance of material in the published literature. Introductions and overviews abound on the worldwide web. See, for example, Wikipedia articles on stochastic processes and probability space.

and $Q_{12} = 0$. However a transition from $\mathbf{e}_1$ to $\mathbf{e}_2$ through $\mathbf{e}_4$ may be permissible. Since a north-west-south segment is allowed over a time interval of length T, the north-south transition possibility will appear in Π, ($\Pi_{21} > 0$), albeit with small probability if T is small.

Equation (1.2) gives the time-discrete regime dynamics. The evolution of a modal state $\boldsymbol{\iota} \in \boldsymbol{\kappa}$ derives from this. The regime sequence, $\{\phi[k]\}$, is a random process with initial value $\phi[0]$. The $\{\phi[k]\}$ process is adapted to the $\{\mathscr{F}[k]\}$-filtration. Clearly, the realization of $\boldsymbol{\iota}$ is adapted to $\mathscr{F}[k]$, and $\boldsymbol{\iota}^{+}$ is adapted to $\mathscr{F}[k+1]$ though neither gives the full regime history. To refer to this latter, we will use $\boldsymbol{\iota}^{\oplus} \in \mathscr{F}[k]$. Then $\boldsymbol{\iota}$ is a prefix of $\boldsymbol{\iota}^{\oplus}$—the most recent part of the modal string.

We have used the notation $\phi[k] = \mathbf{e}_i$ to express the fact that the current regime is the ith. Let us extend this notation to say that $\phi[k] = \mathbf{e}_\iota$ if the L-prefix of $\boldsymbol{\iota}^{\oplus}$ is $\boldsymbol{\iota}$. Or more broadly, ϕ_ι points to the L-string $\boldsymbol{\iota}$: $\phi_\iota[k] = 1$ if $\boldsymbol{\iota}$ is true and $\phi_\iota[k] = 0$ otherwise. Indeed, $\phi_{\text{string}} = 1$ if the string condition is satisfied and zero otherwise. Or we might even say $\phi[k] = \mathbf{e}_{\text{string}}$. This notation is useful in sums; e.g., $\sum_{\iota^- \in \kappa_{(i,\iota^-)}} (M_\iota)\phi_\iota$is a function of $i \in \mathbf{S}$.

1.2.2 Kinematic State

1.2.2.1 Fundamental Models

At first order, we have supposed that the modal process is exogenous. But the kinematic state, $\{\boldsymbol{X}_t\}$, is tightly linked to the modal state. The range space of the kinematic state is $\mathbb{R}^n$. The evolution of the kinematic state depends upon the operating regime, past regime intervals, endogenous commands, and an agglomeration of exogenous disturbances.

At some inclusive level, target motion would naturally be represented with a non-linear stochastic differential equation. During intervals in which the motion mode is constant, $\phi_t \equiv \mathbf{e}_i$, the evolution of the kinematic state equation would be delineated by

$$d\boldsymbol{X}_t = \mathbf{f}(\boldsymbol{X}_t, u_t, \mathbf{e}_i)\, dt + \mathbf{g}(\boldsymbol{X}_t, \mathbf{e}_i)\, dw_t, \tag{1.3}$$

where $\boldsymbol{X}_t$ is this global kinematic state, $\{u_t\}$ is an endogenous command, and the random process $\{w_t\}$ represents the unstructured and unpredictable exogenous influences on the system.

The kinematic model is an intermediary for synthesizing an implementable estimator. Except in special cases, (1.3) is far too complex to be used directly. Such equations are difficult to use to create an implementable tracker/classifier.

In some applications, there is an $(n \times S)$ array of stasis conditions, one for each regime: $\boldsymbol{X}_t \approx \boldsymbol{X}^S_{.i}$ when $\phi_t = \mathbf{e}_i$. The local base-state is the deviation of the global state from the current reference, $x_t = \boldsymbol{X}_t - \boldsymbol{X}^S \phi_t$, and the ith local model provides the forward dynamics. If the global kinematic state process is continuous and if $\mathbf{e}_i \mapsto \mathbf{e}_p$

at time t, the reference level changes from $\boldsymbol{X}^S_{.i}$ to $\boldsymbol{X}^S_{.p}$, and $x_t = x_{t^-} + \boldsymbol{X}^S_{.i} - \boldsymbol{X}^S_{.p}$. In the time-discrete case, $x[k+1]^- \mapsto x[k+1]^- + \boldsymbol{X}^S_{.i} - \boldsymbol{X}^S_{.p}$. Thus, there are several simultaneous values of the deviation variables for the same global kinematic state.

On the other hand, if the common reference state is the origin in the state space, the global state and the local state are identical. In algorithm development, we will suppose that $\{\boldsymbol{X}_t\}$ evolves in a regime-constrained mode of operation in which the nonlinear kinematic model can be replaced locally by a linear dynamic equation. Each such localization is indexed by a regime state (or more broadly, the mode). The global state, $\boldsymbol{X}_t$, is replaced by the pair (x_t, ϕ_t) where x_t is deviation from the nominal kinematic state associated with index ϕ_t. The deviation variable, x_t, is called the base or kinematic state and carries the same labels as did $\boldsymbol{X}_t$; e.g., the first component of x_t is horizontal position referenced to a nominal condition associated with ϕ_t, the second component is vertical position, and so on. This will be the common situation in what follows.

To illustrate this, consider a target moving in the (east, north)-plane. The motion equation in the X-direction is

$$\frac{d^2}{dt^2}X = a_X, \tag{1.4}$$

where $\{(a_X)_t\}$ combines the endogenous acceleration with exogenous disturbances in the X-direction. Motion in the Y-direction is described similarly.

The planar model integrates both motions:

$$\frac{d}{dt}\begin{bmatrix}\chi \\ \nu\end{bmatrix} = \begin{bmatrix}0_{2\times 2} & I_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2}\end{bmatrix}\begin{bmatrix}\chi \\ \nu\end{bmatrix} + \begin{bmatrix}0_{2\times 2} \\ I_{2\times 2}\end{bmatrix}\begin{bmatrix}a_X \\ a_Y\end{bmatrix}, \tag{1.5}$$

where χ_t is position with respect to the (X, Y) coordinate system and $\nu_t = (V_X, V_Y)'$ the velocity. Equation (1.5) is a global description of motion in the plane.

1.2.2.2 Constant Velocity and Constant Turn Models: Continuous and Discrete

In many tracking applications, the endogenous acceleration is integrated into the reference path. The residual acceleration is a wide-band random process which creates the ensemble of possible deviations about the nominal. Commonly, this residual is represented with a vector white-noise process.

Complicated motions are achieved by selecting the endogenous acceleration in keeping with some objective; e.g., evasion [14]. The endogenous accelerations create primal motion templates, and a composite motion is generated by joining these templates at random times. For example, suppose an agile target is capable of a coordinated turn with angular frequency $\omega_i; i \in \mathbf{S}$. An endogenous acceleration can be chosen to execute this motion. Indeed, the acceleration is a linear function of the kinematic state, and (1.5) can be adjusted to yield a linear differential equation that delineates a *constant turn* (CT) motion:

$$\frac{d}{dt}\begin{bmatrix} X \\ Y \\ V_X \\ V_Y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_i \\ 0 & 0 & \omega_i & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ V_X \\ V_Y \end{bmatrix}. \tag{1.6}$$

When the turn rate is zero, we have a *constant velocity* (CV) motion.

A *jinking* motion is created by mixing periods of turning with nearly CV flight. For example, the target can intersperse intervals in which the endogenous acceleration is zero with intervals in which the turn rate is $\pm\omega_1$. In this instance, the kinematic states in the local turn-specific models are all referenced to the same point in motion space, $\chi = 0$; $\nu = 0$. But the different motion regimes have different kinematic descriptions.

The process model given in (1.6) describes a point target moving without exogenous forcing. The target is actually an extended object. Further, the vehicle is subject to various exogenous accelerations that will be viewed as wide-band in the context of the tracking problem. A more useful motion model would be

$$\frac{d}{dt}\begin{bmatrix} X \\ Y \\ V_X \\ V_Y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_i \\ 0 & 0 & \omega_i & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ V_X \\ V_Y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \sqrt{P_x^w} & 0 \\ 0 & \sqrt{P_y^w} \end{bmatrix} \begin{bmatrix} \dot{w}_X \\ \dot{w}_Y \end{bmatrix}, \tag{1.7}$$

where $\dot{w}_X$ is Gaussian white noise and similarly $\dot{w}_Y$. Equation (1.7) aggregates all of the disturbances and modeling errors into the white-noise process $\{\dot{w}\}$.

In what follows, when operating in the ith regime, we will use the local model

$$dx_t = A_i x_t\, dt + \sqrt{P_i^w}\, dw_t, \tag{1.8}$$

where $\{w_t\}$ is an $\mathscr{F}_t$-unit Brownian motion (an $\mathscr{F}_t$-martingale) independent of other exogenous excitations.

In each regime, there is a specific local model what manifests the idiosyncrasies of the kinematic state in that operating condition. We will assume that the state labels are the same for each regime; e.g., the first components of x_t are χ. But the deviation variables are referenced to the common nominal; i.e., the kinematic state is *centered* at the origin in the state space. In this way, the single nonlinear motion model in (1.3) is replaced with the family of S localizations in (1.8).

Equation (1.8) is a plausible approximation when regime state is constant. However, the kinematic state may be discontinuous at regime transition times: $\Delta\phi_t \neq 0$ might imply $\Delta x_t \neq 0$. If the reference point for a kinematic state changes, the perturbation state will experience a corresponding increment. For example, when a target moves from CV-motion to a coordinated turn, the position and velocity