

APPLIED ESTIMATION FOR HYBRID DYNAMICAL SYSTEMS USING
PERCEPTIONAL INFORMATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND
ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Aaron M. Plotnik

March 2007

© Copyright by Aaron M. Plotnik 2007
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Stephen M. Rock)
Department of Aeronautics and Astronautics
Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Claire J. Tomlin)
Department of Aeronautics and Astronautics

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Günter Niemeyer)
Department of Mechanical Engineering

Approved for the University Committee on Graduate Studies.

Abstract

This dissertation uses the motivating example of robotic tracking of mobile deep ocean animals to present innovations in robotic perception and estimation for hybrid dynamical systems. An approach to estimation for hybrid systems is presented that utilizes uncertain perceptual information about the system's mode to improve tracking of its mode and continuous states. This results in significant improvements in situations where previously reported methods of estimation for hybrid systems perform poorly due to poor distinguishability of the modes. Even in applications where the modes are more easily distinguished, the approach presented can improve performance by decreasing the mode estimation delay of the estimator.

The specific application that motivates this research is an automatic underwater robotic observation system that follows and films individual deep ocean animals. A first version of such a system has been developed jointly by the Stanford Aerospace Robotics Laboratory and Monterey Bay Aquarium Research Institute (MBARI). This robotic observation system is successfully fielded on MBARI's ROVs, but agile specimens often evade the system. When a human ROV pilot performs this task, one advantage that he has over the robotic observation system in these situations is the ability to use visual perceptual information about the target. The human pilot can immediately recognize any changes in the specimen's behavior such as whether it is actively swimming or not and react to those mode changes with the necessary forceful measures required to prevent losing sight of the specimen.

With the approach of the human pilot in mind, a new version of the robotic observation system is proposed which is extended to (a) derive perceptual information (visual cues) about the behavior mode of the tracked specimen, and (b) merge this dissimilar, discrete and uncertain information with more traditional continuous noisy sensor data by extending existing algorithms for hybrid estimation. These performance enhancements are enabled by integrating a wide range of techniques in hybrid estimation, computer

vision and machine learning. First, real-time computer vision is paired with a supervised classifier and other pattern recognition algorithms to extract visually an observation of the target's behavior mode from the video stream. Existing hybrid estimation algorithms are extended to admit this uncertain but discrete observation in addition to the information available from more traditional sensors. State tracking is achieved using a new form of Rao-Blackwellized particle filter called the mode-observed Gaussian Particle Filter. Performance is demonstrated using data from simulation and data collected on actual specimens in the ocean. The framework for estimation using both traditional and perceptual information is easily extensible to other stochastic hybrid systems with mode-related perceptual observations available.

Acknowledgments

This dissertation would not have been possible without the support and efforts of a great many people at Stanford University, Monterey Bay Aquarium Research Institute (MBARI), and my support network of family and friends.

First, I wish to thank my thesis adviser, Stephen Rock, for providing me the opportunity to be a researcher at the Aerospace Robotics Laboratory. Steve has had a profound influence on my development as a researcher over my years at the ARL. I have particularly appreciated Steve's balance between maintaining a very hands-off management environment that lets his students learn for themselves how to recognize and achieve quality graduate research while exerting the right amount of steering influence necessary for each individual student. His insights have been invaluable to this research.

I wish to thank my dissertation reading committee, Claire Tomlin and Günter Niemeyer, for their insightful comments in reviewing this work. I also thank the remainder of my oral defense committee, Matt West and Chris Gerdes, for also providing very helpful and perceptive insights into this research during and after my defense. Claire Tomlin's influence on this research was also passed on indirectly through many enlightening discussions with the students in her lab. In particular, I wish to thank Hamsa Balakrishnan for spending a great deal of time discussing my research with me and helping me find my way into the field of estimation for hybrid systems.

I wish to thank the administrators and staff at the Aeronautics and Astronautics department, especially Lynn Kaiser, Sherann Ellsworth, Dana Parga, and Jane Lintott for always helping facilitate the administrative details at the department and University, and Godwin Zhang and Aldo Rossi for invaluable help in computing, hardware design and assembly for experiments.

Like all of the ARL students whose research involves underwater robotics projects with MBARI, I am heavily indebted to the engineers, scientists, ROV pilot teams and ship crews there. Without the extreme dedication of the MBARI staff, none of the experiments done at sea would be possible. MBARI biologist Bruce Robison provided the initial challenge

to the lab to design a robotic tracker of gelatinous animals and was extremely supportive throughout my tenure on that program, allowing me to tag along on several science dives, observing the action and even commandeering the ROV for extra experimental research time. I particularly wish to thank the *Ventana* ROV pilot team of Knute Brekke, Craig Dawe, D.J. Osborne and Mark Talkovic. On countless occasions, this remarkable team, usually with little advance notice, provided the help necessary to transform our ideas for experiments into reality. They also provided the inspiration for much of the specific research in this dissertation by consistently outperforming the baseline automatic robotic observation system when tracking the more active animals encountered. The simple question of ‘what are they doing that our system is not?’ along their insightful answers pushed me toward the research contained here.

The students of the ARL and its founder, Robert Cannon, provided a wonderful support network for research and life as a graduate student. I wish to thank Professor Cannon for establishing the lab and for his great enthusiasm about my research. I wish to thank the underwater research group members Kristof Richmond, Jason Rife, Jack Langelaan (actually a near-miss on underwater research), Andreas Huster, Jeff Ota and Peter Kimball for all their advice and help in getting experiments together out on Monterey Bay, collaborative software coding and general commiserating about the seasickness we shared in doing most of our experiments from the *Pt. Lobos*. Without Kristof’s help in building the shared software components used for robotic tracking of ocean animals (as well as his own research), I don’t know how the software could have come together. I learned a great deal from Kristof throughout the process. I also must single out Jason Rife, the “godfather” of the robotic observation system (or jelly-tracker as he preferred to call it), who has been unbelievably supportive of my research, providing encouragement and insights from my first day out on Monterey Bay all the way through preparing for my thesis defense.

I wish to thank the sources that funded my time at Stanford as a graduate researcher. My first three years were funded through the Stanford Graduate Fellowship program. The remainder of my funding was received from the Packard Foundation and MBARI.

I wish to thank my parents, Linda and Jim Plotnik, and my sisters, Lisa and Jenny, for a lifetime of love and support. And finally, my greatest thanks are due to my wife Susan and our young son Max, born just short of a year before turning in this dissertation. I am lucky to have been blessed with such a wonderful family.

Contents

| | |
|--|-----------|
| Abstract | iv |
| Acknowledgments | vi |
| 1 Introduction | 1 |
| 1.1 Motivation and Background | 3 |
| 1.1.1 Robotic Observation in the Midwater | 3 |
| 1.1.2 Comparison with Human Pilot in Loop | 5 |
| 1.2 Towards Tracking of Agile Targets by Recognition of Motion Modes | 8 |
| 1.2.1 Estimation of Mode from Velocities in the Water Frame | 8 |
| 1.2.2 Estimation of Mode by Perception | 9 |
| 1.2.3 Estimation for Hybrid Systems Augmented by Perceptual Infor- mation | 10 |
| 1.2.4 Overview of a New Architecture | 11 |
| 1.3 Summary of Contributions | 12 |
| 1.4 Reader’s Guide | 14 |
| 2 Robotic Tracking of Midwater Animals | 16 |
| 2.1 Architecture and Relative Positioning Strategy | 17 |
| 2.2 Improved Initialization and More Sustainable Tracking | 19 |
| 2.2.1 Constraints Imposed on Control Performance by the Vision Sensor . | 20 |
| 2.2.2 A Redesigned Stereo Camera Sensor | 24 |
| 2.3 Mitigating Sustained Circumnavigation of the Specimen | 25 |
| 2.4 Results | 27 |
| 2.4.1 Tracking a Wider Variety of Specimens | 27 |
| 2.4.2 Heading Hold Loop | 28 |
| 2.5 Tracking of More Agile Targets | 28 |

| | | |
|----------|--|-----------|
| 3 | Estimation for Hybrid Systems | 30 |
| 3.1 | Motivations for Hybrid Estimation | 31 |
| 3.2 | Hybrid Estimation Algorithms | 32 |
| 3.2.1 | Overview | 32 |
| 3.2.2 | Multiple Model Estimators | 35 |
| 3.2.3 | Particle Filters for Hybrid Systems | 35 |
| 3.3 | Observability of Mode Transitions in Stochastic Hybrid Systems | 38 |
| 3.3.1 | Necessary Conditions for Mode Transition Detection | 40 |
| 3.3.2 | A Target Tracking Example with One-Dimensional Motion | 42 |
| 3.3.3 | Application to the Robotic Observation System | 45 |
| 4 | Estimation of Water-Relative Motion | 48 |
| 4.1 | Overview | 49 |
| 4.2 | Estimation of Relative States Only | 50 |
| 4.3 | Estimation of Water-Relative Motions of the Target and Vehicle | 52 |
| 4.3.1 | State Vector and Process Model Equations | 52 |
| 4.3.2 | Sensor Models | 54 |
| 4.3.3 | Estimator Design | 56 |
| 4.4 | Results from the Sigma Point Kalman Filter | 58 |
| 5 | Visual Classification of Body Motion Modes | 59 |
| 5.1 | Motion Behavior of Tracked Marine Animals | 60 |
| 5.1.1 | Characteristics of the Motion Behaviors Encountered | 60 |
| 5.1.2 | A Mode Model for Real-Time Monitoring of Animal Motions | 62 |
| 5.2 | Rationale for Algorithms | 63 |
| 5.2.1 | Algorithms Required for Mode Determination | 63 |
| 5.2.2 | Properties of Body Deformations in Sequences of Images | 64 |
| 5.2.3 | Self-Similarity Metrics for Quantification of Body Motion | 65 |
| 5.3 | Algorithms and Example Results | 66 |
| 5.3.1 | Object Tracking and Stabilization | 66 |
| 5.3.2 | Computation of Self-Similarity | 68 |
| 5.3.3 | Presence or Absence of Body Motion using a SVM | 71 |
| 5.3.4 | Detection of Cyclic Motion and Calculation of Period | 74 |
| 5.3.5 | Deviation from Expected Body Pose | 77 |
| 5.3.6 | Real-Time Considerations | 79 |
| 5.4 | Summary | 80 |

| | | |
|----------|---|------------|
| 6 | Estimation Using Perceptual Information | 81 |
| 6.1 | System Model Equations | 83 |
| 6.1.1 | Process and Sensor Models | 84 |
| 6.1.2 | A Dynamic Bayesian Network System Representation | 86 |
| 6.1.3 | A Conditional Probabilistic Model for σ | 89 |
| 6.2 | Estimation Assisted by Mode-Related Observations | 90 |
| 6.2.1 | The Mode-Observed Rao-Blackwellized Particle Filter | 90 |
| 6.2.2 | The Mode-Observed Gaussian Particle Filter | 92 |
| 6.2.3 | Applying Mode Observations to the Multiple Model Bootstrap Filter | 94 |
| 6.3 | Example Results | 95 |
| 6.3.1 | Parameterization for the Estimator Equations | 95 |
| 6.3.2 | Simulated Case | 98 |
| 6.3.3 | Application to Field Data | 100 |
| 6.3.4 | Mode Estimation Bandwidth Comparison | 102 |
| 6.3.5 | Real-Time Considerations | 103 |
| 7 | Conclusions and Future Research | 106 |
| 7.1 | Extensions of the Underwater Robotic Observation System | 107 |
| 7.2 | Perception-Augmented Estimation for Hybrid Systems | 110 |
| 7.3 | Additional Applications of Perception-Augmented Estimation | 111 |

List of Tables

| | | |
|-----|---|-----|
| 2.1 | Calculated maximum standoff distances (relative to the cameras) and control errors for different sizes of observation specimen. | 23 |
| 3.1 | The IMM algorithm for stochastic linear hybrid systems. | 36 |
| 3.2 | The multiple model bootstrap (particle filter) algorithm. | 37 |
| 3.3 | The RBPF algorithm for linear hybrid systems with Gaussian noise and stochastic mode transitions. | 39 |
| 3.4 | The GPF algorithm for nonlinear hybrid systems with Gaussian noise and stochastic mode transitions. | 40 |
| 3.5 | Summary of model parameters assumed for one-dimensional example. | 44 |
| 3.6 | Mode observability test quantities for one dimensional example. | 46 |
| 3.7 | Mode observability test quantities for underwater tracking system. | 46 |
| 5.1 | Example statistics on motion behavior. | 62 |
| 5.2 | Comparison of potential vision approaches. | 66 |
| 5.3 | Summary of results from applying SVM to several clips. | 75 |
| 6.1 | CPT relating σ to m | 89 |
| 6.2 | The mode-observed RBPF algorithm. | 92 |
| 6.3 | The mode-observed GPF algorithm. | 92 |
| 6.4 | The mode-observed multiple model bootstrap filter algorithm. | 95 |
| 6.5 | CPT relating σ to m , tuned to reduce the modeled certainty of the mode-related measurement, σ | 97 |
| 6.6 | Summary of model parameters assumed for estimation. | 97 |
| 6.7 | Average of 2-norm of error in target velocity estimates (cm/s), by target motion phase. | 100 |

List of Figures

| | | |
|------|--|----|
| 1.1 | Remotely operated vehicles used for marine science by direct observations. | 4 |
| 1.2 | Side View of Robotic Tracking of Squid on ROV <i>Tiburón</i> | 5 |
| 1.3 | Baseline processing architecture of robotic observation system (a), and (b) a proposed new processing architecture updated to incorporate the algorithms presented in this dissertation. | 6 |
| 1.4 | Examples of the robotic observation being evaded by agile animals. | 7 |
| 2.1 | Examples of tracked animals. | 18 |
| 2.2 | Block diagram of robotic observation system hardware and its interactions with the human ROV pilot and scientist. | 19 |
| 2.3 | Coordinate systems and frame definitions. | 20 |
| 2.4 | The ROV <i>Ventana</i> | 21 |
| 2.5 | Camera-related geometric parameters. | 22 |
| 2.6 | Photographs of stereo camera sets. | 23 |
| 2.7 | Stereo vision cone overlap for various camera configurations. | 24 |
| 2.8 | Effect of unmodeled yaw/lateral coupling. | 26 |
| 2.9 | Block diagram of robotic observation system with heading hold included. | 27 |
| 2.10 | Range control while tracking an <i>Aegina</i> jellyfish. | 28 |
| 2.11 | Heading hold control while tracking a <i>Solmissus</i> jellyfish. | 29 |
| 3.1 | Velocity trajectories for two simulated cases. | 44 |
| 3.2 | Mode probability estimates for 1D example. | 45 |
| 3.3 | Measurement likelihood functions for 1D example. | 47 |
| 4.1 | Estimator architecture for relative states only. | 50 |
| 4.2 | The ROV <i>Ventana</i> configured for midwater activities. | 55 |
| 4.3 | Architecture of UKF estimator for water-relative states. | 57 |

| | | |
|------|--|-----|
| 5.1 | Body motion behavior examples. | 61 |
| 5.2 | Finite automata representing motion behaviors of tracked specimens. | 63 |
| 5.3 | Body motion classification block diagram. | 67 |
| 5.4 | Example images for calculating similarity. | 69 |
| 5.5 | Similarity matrices for two example clips. | 71 |
| 5.6 | Motion detection results using SVM. | 76 |
| 5.7 | Example of lattice-matching for periodicity detection. | 77 |
| 5.8 | Period and pattern deviation for <i>Ptychogena</i> footage. | 78 |
| 5.9 | Mode indication for <i>Ptychogena</i> footage. | 79 |
| 6.1 | A set of dynamic Bayesian network (DBN) models relating system's variables. | 87 |
| 6.2 | Graphical depiction of Rao-Blackwellized (or Gaussian) Particle Filter for hybrid linear (nonlinear) systems. | 93 |
| 6.3 | Graphical depiction of Mode-observed Rao-Blackwellized (or Gaussian) Particle Filter for hybrid (nonlinear) systems. | 94 |
| 6.4 | σ and m (true mode) for simulated case. | 98 |
| 6.5 | Mode probability results on simulated data. | 99 |
| 6.6 | Vertical velocity of target, estimated for simulated case. | 100 |
| 6.7 | 2-norm of velocity estimation error for simulated case. | 101 |
| 6.8 | Visual mode observation, σ , as applied to field data. | 102 |
| 6.9 | Sensor data during field test. | 103 |
| 6.10 | Mode probability results on field data. | 104 |
| 6.11 | Vertical velocity of target, estimated, for field data. | 105 |
| 6.12 | Velocity and mode probability estimates on field data (deceleration phase). | 105 |

Chapter 1

Introduction

This dissertation uses the motivating example of robotic tracking of mobile deep ocean animals to present innovations in robotic perception and estimation for hybrid systems. Its most significant contribution is an algorithm extending estimation for hybrid systems to augment traditional sensor information with perceptual information about the system's mode to improve tracking of the system's mode and continuous states. This results in substantial improvements in situations where previously reported methods in estimation for hybrid systems for discerning the most likely mode of the system fail due to poor distinguishability of the modes. With the additional perceptual information related to the mode of the system, the estimator is able to favor the correct mode more quickly and accurately than when using measurements of the continuous states alone.

The motivating application for this work is a robotic observation system for following and filming individual deep ocean animals in their natural habitat. Robotic technologies, such as the Remotely Operated Vehicle (ROV), have made it possible to observe directly marine species and geological phenomena that previously were unreachable. The task of tightly following individual ocean animals for scientific observation is normally performed by human ROV pilots, but observation durations are limited by pilot fatigue due to the precise vehicle control required. Thus, the desire to achieve longer duration observations of ocean animals has motivated an automatic robotic solution to this problem.

A first version of such a robotic observation system has been developed and fielded. This observation system successfully integrates solutions to many difficult component problems including robust visual sensing of the relative position of the tracked specimen in the presence of clutter and inconsistent lighting, and precise control of a hovering underwater vehicle in the presence of large unknown disturbances. This automatic robotic observation system has achieved much tracking success, but agile specimens often are able

to elude it. Tracked marine animals display a rich variety of distinct behavior modes, often switching between them very suddenly. With switches between behavior modes can come substantial and rapid changes in velocity by the specimen, thereby greatly increasing the risk of the robot's losing sight of it. Recognition of these mode changes is therefore a key requirement for enabling swift, decisive control actions to mitigate that risk. Hence, this dissertation is largely motivated by the desire to achieve robust and fast recognition of behavior mode changes by marine animals in order to extend the tracking capabilities of the robotic system.

The problem of estimating the mode and velocities of active marine animals is a type of maneuvering target tracking problem, similar in many ways to those encountered in fields such as Air Traffic Control and missile homing guidance. As in these maneuvering target tracking applications, mode can be inferred from measurements of quantities such as the position and velocity of the target. The use of these state measurements to estimate simultaneously a system's discrete mode and its continuous-valued states is a subfield of estimation called hybrid or multiple model estimation. As will be shown later in this dissertation, applying existing hybrid estimation techniques to the problem of tracking these marine animals cannot achieve the fast and robust mode recognition that is sought. At best, the estimates are very limited in bandwidth, and therefore do not enable the fast reactions required by an automatic system.

In addition to traditional position and velocity sensor readings, a second source of information is available to recognize mode changes by the tracked specimen – the visible changes in activity patterns of its body. The human pilot when operating the tracking vehicle takes advantage of this visual perceptual information about the target to handle mode changes by agile animals very successfully. This information is interpreted naturally and immediately by a human pilot, enabling nearly instant recognition of mode changes and ensuing fast and intelligent control reactions.

Inspired by the perceptual approach of the human pilot, the first key contribution of this dissertation is a set of real-time algorithms in computer vision and machine learning to extract this perceptual information about the tracked target from the video stream already being used for relative position sensing. The perceptual information is structured as a classification of the body activity patterns. This classification provides indications of mode changes before they manifest clearly in the noisy, more traditional, sensor data.

The visual classification is noisy (error-prone) enough it cannot be used alone to trigger drastic control actions by a robotic observation system. Hence, the second key contribution of this dissertation is a new extension to existing hybrid estimation techniques which

permit the estimator to augment its traditional measurements of continuous quantities (such as position and velocity) with discrete-valued, mode-related perceptual information. With this fusion, better estimates of mode and target motion states are realized than those which can be achieved with either source of information alone.

In this dissertation, the perception algorithms and the example hybrid systems used to demonstrate estimation results are all placed in the context of tracking marine animals with an ROV. However, the framework developed for both perception and the fusion of perception with more traditional sensors is generally applicable and extensible to any hybrid system with perceptual information available.

1.1 Motivation and Background

1.1.1 Robotic Observation in the Midwater

Deep ocean biological science lags behind the state of its counterparts in the terrestrial sciences, in substantial part due to the difficulty and expense of making direct observations in the ocean on any large scale. In recent decades, robotic technologies have become available to help remedy this situation. Remotely Operated Vehicles (ROVs) such as MBARI's *Ventana* and *Tiburon*, and WHOI's *Jason 2* (see Figure 1.1) are allowing scientists to observe directly the previously inaccessible regions of the ocean and their inhabitants and phenomena. The midwater region of the ocean has historically been particularly underexplored. Here, ROVs have enabled scientists to observe the behaviors and anatomies of numerous species that were previously poorly understood or undiscovered, e.g., [22, 60–62].

While much has been learned about life in the deep ocean by using human-piloted ROVs to track individual animals, the precise control required when tracking for long durations at the close ranges required for quality science film is extremely fatiguing for the pilot. Hence, an automated capability offers the potential for greatly extending the periods over which an animal can be observed.

To this end, a first version of an automatic tracking and observation system has been developed and tested using MBARI's ROVs as the tracking vehicle platforms, successfully tracking midwater ocean animals such as jellyfish, squid, fish and larvaceans in Monterey Bay, California. This system uses stereo vision to localize the tracking vehicle with respect to the target of interest and closes control loops to maintain the target in the views of the cameras [54–57, 59]. Figure 1.2 is a still picture taken from an auxiliary camera on the ROV *Tiburon* while tracking a small squid under automatic positioning control, illustrating some key components of the system. The specimen being tracked is shown in the bottom

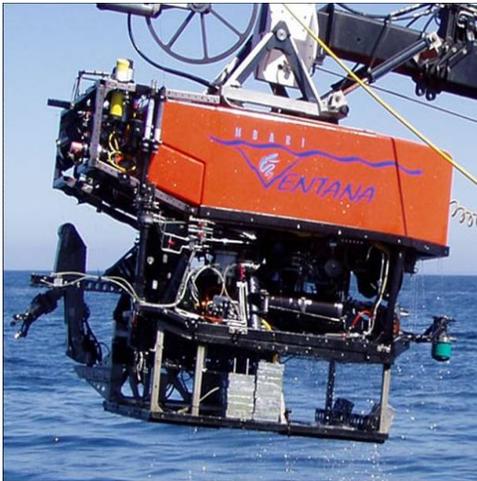
(a) The MBARI ROV *Ventana*.(b) The MBARI ROV *Tiburon*.(c) The WHOI ROV *Jason 2*.

Figure 1.1: Remotely operated vehicles used for marine science by direct observations.

left corner of the photograph. At the top right is the stereo camera sensor, from whose video streams the relative position of the specimen is computed in software. A portion of the science camera is visible in the bottom right of the photo. This camera is entirely decoupled from the sensing and control loops and is therefore free for the scientist to operate at will, panning and zooming to collect high-quality film of the specimen.

Figure 1.3(a) provides a high-level view of the processing architecture of this baseline system. The video streams from the stereo camera set are processed by the “Visual Tracking” block, algorithms which extract bearings to the tracked target. The next block in the architecture is “Estimation”, where those bearings are used to derive relative position and relative velocities of the target with respect to the vehicle. Finally, the “Control” block represents the control loops which use that information to command the vehicle’s thrusters such that the target is maintained within the view of the cameras at all times.

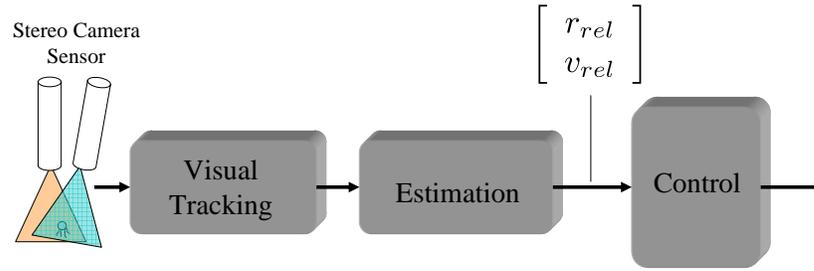


Figure 1.2: Side View of Robotic Tracking of Squid on ROV *Tiburon*

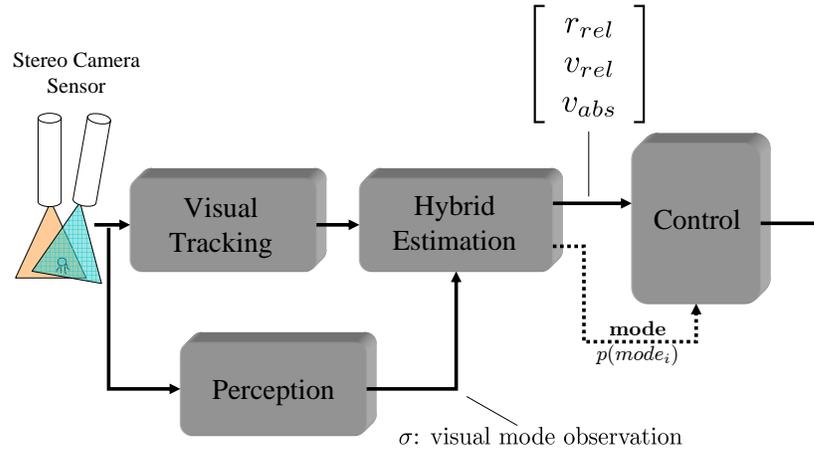
1.1.2 Comparison with Human Pilot in Loop

The baseline robotic observation system, as depicted in Figures 1.2 and 1.3(a), has been demonstrated with much success when deployed upon specimens that fall within its original design range, i.e., those which exhibit only low levels of acceleration. It has achieved the objective of enabling long duration tracking of such animals, including an 89 minute tracking run following a *Ptychogena* jellyfish (which was terminated by the operators, not because of failure [59]). For reasonably passive targets, it achieves steadier control of the vehicle than the ROV pilots, enabling higher quality science camera footage to be garnered. This is particularly true for small targets (on the order of 2.5cm in diameter, such as several in the *Aegina* genus) which the system has recently been proven capable of tracking well after several engineering design improvements [51]. This version of the system is described in detail in Chapter 2, and example still shots of animals tracked by the system are provided in Figure 2.1.

When tracking more agile animals whose changes in behavior mode result in sudden changes in velocity, the baseline system can be evaded. Some examples of such breakdowns of tracking continuity are shown in Figure 1.4. When a human ROV pilot performs this task, however, his performance does not degrade nearly as much as that of the baseline robotic observation system. In these situations, the pilot is able to use significantly different information than does the robotic observation system. These include visual recognition of the current behavior mode of the specimen, awareness of the orientation of the animal (which typically have a thrust axis, i.e., a single direction in which it can swim), *a priori* knowledge of the motion capabilities of the specific type of specimen being tracked, and qualitative information about the motion of the vehicle and target with respect to the



(a) Baseline architecture.



(b) Proposed new architecture.

Figure 1.3: Baseline processing architecture of robotic observation system (a), and (b) a proposed new processing architecture updated to incorporate the algorithms presented in this dissertation.

water column from watching the patterns of relative motion of the marine snow field. These sources of information are all available to the human ROV pilot to make intelligent decisions about how best to track the animal to avoid losing sight of it. The visual perceptual information about the target proves to be particularly valuable, as the human pilot can immediately recognize any changes in the specimen's behavior such as whether it is actively swimming or not, and anticipate its resulting velocity changes naturally and quickly.

The human pilot in the loop is able to react in a nonlinear way to perceived changes in behavior mode by the tracked specimen. For instance, when seeing a swimming target stop actively swimming, he can rapidly decelerate the tracking vehicle. The pilot can also

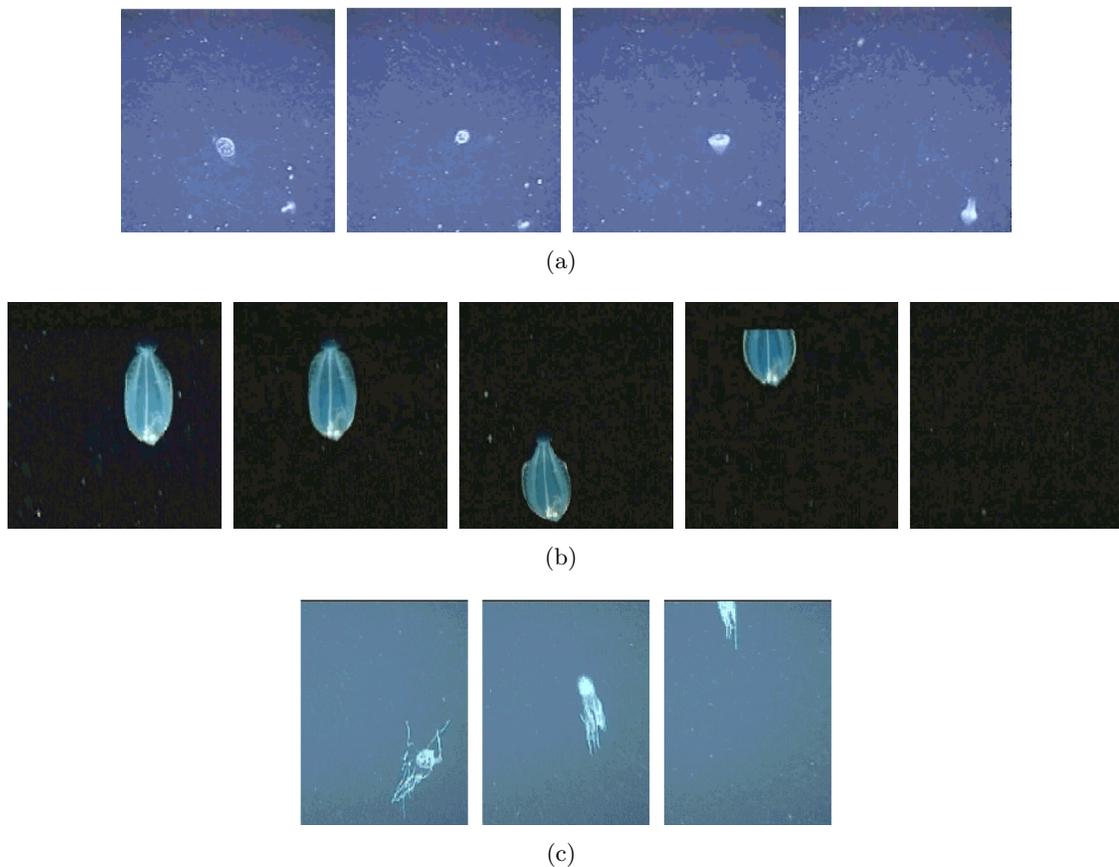


Figure 1.4: (a) A medusa jellyfish begins swimming and evades the robotic observation system, (b) a larval flatfish stops swimming suddenly, decelerating and causing the observation system to overshoot the specimen, (c) a *Colobonema* jellyfish makes a sudden swimming motion while being tracked, evading the observation system. All are time-lapsed at 1 sec intervals. (Images courtesy of MBARI.)

quickly zoom out (and/or increase range in some circumstances) to expand the volume of the water column which can be viewed by the cameras, thus allowing much more tracking error during these excursions at maneuvering mode changes. Furthermore, the orientation and expected swimming velocities for a given specimen are usually known to the pilot. Thus, at the onset of active swimming motions, the pilot can very naturally anticipate the resulting motion of the target with respect to the water column.

The baseline robotic observation system of Figure 1.3(a) uses only information more typical of an automatic control system: measurements of positions and velocities, from which all lead information is acquired by differentiation and filtering. The automatic system has the strengths of tight control on steady targets that one would expect when

comparing to a human in the loop, and leverages those advantages to track targets of low to medium agility steadily for long duration observations. However, when faced with more agile targets, its disadvantages from an informational standpoint are exposed. Where the human operator can quickly and accurately recognize behavior mode changes and the resulting motions, the filtering required by the observation system on its sensors precludes fast recognition of (and therefore reaction to) those motions.

1.2 Towards Tracking of Agile Targets by Recognition of Motion Modes

The motion of the more agile species tracked in the ocean may be decomposed into a small finite set of behavior modes, with a different type of motion through the water column associated with each mode. Some examples are illustrated by the series of photographs in Figure 1.4. The desired responses by the robotic observation system to motion behaviors by these animals depend upon an ability to recognize behavior mode changes reliably and quickly, and upon good estimates of the associated velocities.

Mode recognition for the specimens tracked by a robotic observation system can potentially be achieved using one of two methods, or, after the contributions of this dissertation, through their combination. The first is by inference from estimates of the velocity of the animal in the water column, which is primarily driven by the swimming mode of the animal. The second is from visual information about the body motions of the animal, as swimming is usually achieved by moving or deforming the body of the animal (e.g., through bell compression and expansion or fin motion).

In this section, these two approaches to deriving the mode of marine animals are discussed, including their strengths and weaknesses. As will be demonstrated, neither approach alone can achieve the quality of mode estimation that is sought. Hence, a proposed new architecture which combines them in a complementary fashion will be presented which achieves significantly better mode estimation than either alone. This architecture is depicted in Figure 1.3(b) and incorporates the contributions of this dissertation within the “Perception” and “Hybrid Estimation” blocks.

1.2.1 Estimation of Mode from Velocities in the Water Frame

The swimming mode changes of agile targets result in substantial changes in velocity with respect to the water column. Hence, their mode can potentially be inferred from estimates of water-relative motion. For many such maneuvering target tracking problems,

representing the target as a hybrid system whose mode is a specific maneuvering mode (or one characterized as steady or quiescent) improves the ability to track its motion. This approach to tracking maneuvering targets is summarized in Chapter 3.

In principle, these techniques can be applied to the tracking of marine animals by an ROV. However, existing methods for hybrid estimation will be shown to be inadequate for this application because of the poor distinguishability of the target's modes within the very noisy and relatively slowly sampled water-relative velocity information available. This is effectively a signal-to-noise issue with respect to the detection of mode changes. This issue will be covered in detail in Chapter 3, including a contribution of this dissertation, a condition by which some or all of a hybrid system's mode transitions can be shown to be undetectable by a hybrid estimator given the system's continuous outputs.

Although existing hybrid estimation approaches when applied to this problem will be shown to be unable to detect all mode changes, it will later be demonstrated that an extended hybrid estimator, augmented with additional mode-related information, can achieve this successfully. Therefore, a first prerequisite step toward estimating mode in this framework is to fuse in additional sensor information that makes water-relative motions observable. These motion terms were not necessary and had been neglected by assumption in the design of the baseline robotic observation system. An estimator capable of tracking water-relative motions provides the building block for the hybrid estimation approaches to this problem and therefore receives significant attention in this dissertation in Chapter 4.

1.2.2 Estimation of Mode by Perception

A second source of information about the mode of tracked marine animals is available in the form of visual indications of body motion patterns. Thus, one of the key contributions of this dissertation is a set of algorithms that extract visually apparent information about the specimen's behavior mode by quantifying and classifying patterns of deformation of the body of the target. This classification is inspired by the perception of the human observer when observing the specimen. Most active species tracked by the ROV effect propulsive forces by deforming some part of their bodies (or the entire body) in a pulsing or other motion. For instance, most gelatinous animals such as jellyfish, ctenophores, and siphonophores have bell structures which pump water in and out to generate a net thrust force on the surrounding water, often in periodic patterns. Algorithms are presented in this thesis which derive a fast *visual* indication of the mode of the target.

These visual perceptual algorithms are presented in Chapter 5. There, the motion modes of animals tracked by the robotic observation system are established and characterized in terms of the relationships between body deformation patterns and the resulting propulsion dynamics. A finite automaton mode model characterizing the body motions exhibited by these animals is introduced, including decision criteria that enable the use of this finite automaton to classify the current mode of the animal based on the body motion patterns detected. Recognition of mode changes is achieved by applying techniques in real-time computer vision and supervised machine learning in the form of a support vector machine (SVM) to evaluate the guard conditions (transition criteria) of the proposed finite automaton. Methods are presented to distinguish between active and resting modes, and to detect and measure periodic patterns in the body motions of these animals.

Although the visual mode observation provides a fast indication of mode changes by the specimen, the on-line classification is error-prone enough that it is not possible to base drastic control actions upon it alone. Instead, it is used as one form of observational evidence in a larger estimation framework to derive the current mode of the tracked target, which will be summarized in Subsection 1.2.3 and presented in full in Chapter 6.

1.2.3 Estimation for Hybrid Systems Augmented by Perceptual Information

Two separate sources of information about the mode of maneuvering marine animals tracked by an ROV have been summarized in the preceding subsections: (1) water-relative velocity estimates from which mode can be inferred and (2) visual perceptual information derived from classifying observed body motion activity. Chapter 6 presents an estimation approach that combines these elements in a complementary fashion to estimate the propulsion mode of the target accurately and quickly (enabling potentially drastic mode-based control actions by the robotic observation system) while also improving velocity estimation. The approach consists of a new hybrid estimation framework complemented by the uncertain visual mode observation signal from the classification system that will be presented in Chapter 5. By using the visual mode observation within the framework for hybrid estimation, the resulting estimate of target mode is effectively filtered by balancing the evidence from the body motion classifier with the evidence in the continuous-valued sensors. Furthermore, the estimate of target velocity is improved, using visual cues from the body motion classifier to anticipate velocity changes before they are clear in the noisy sensor data.

Using the visual mode classification in an estimator poses some unique challenges because it is discrete-valued and uncertain, but its uncertainty cannot be represented by a Gaussian distribution as typically assumed in most estimators. State estimation involving distributions that include non-Gaussian models is typically accomplished using particle filters. These can become prohibitively expensive computationally for systems with large or even modestly sized state dimensions, as is the case for the robotic observation system, which is implemented at a 10 Hz sample rate.

A dynamic Bayesian network (DBN) is used to represent the discrete and continuous dynamics of the target and both Gaussian and non-Gaussian observation models simultaneously. Choosing to use a DBN provides a great deal of flexibility in structuring the observation model relating the mode-related visual classification to the true propulsive mode of the target. State tracking is achieved using a new form of Rao-Blackwellized particle filter on this DBN called the mode-observed Gaussian Particle Filter. This type of filter combines aspects of the particle filter for non-Gaussian portions of the assumed probabilistic models with the analytical methods of propagating Gaussian statistics that make up the Kalman filter and its variants. Thus, significant computational efficiency improvements are achieved compared to a classical particle filter. Chapter 6 includes estimation results from both simulation and application to field data collected during experiments in Monterey Bay, California.

1.2.4 Overview of a New Architecture

The proposed new processing architecture of Figure 1.3(b) extends the baseline architecture of Figure 1.3(a) by incorporating the contributions of this dissertation. This new architecture provides the ability to perceive and quantify the behavior mode changes by the tracked targets, thus supplying the necessary inputs for mode-driven control strategies. This is achieved using perceptual computer vision and machine learning algorithms combined with estimation algorithms for hybrid dynamical systems.

The new architecture adds a component called “Perception”, summarized in Subsection 1.2.2, which monitors the incoming video to derive a visual observation, σ , of the current mode of the tracked target, based entirely on the exhibited patterns in body deformations. This observation may be interpreted as an estimate of the target’s mode that is derived purely from visual cues. This higher bandwidth mode observation (or estimate) is accurate enough to be very useful but contains enough error content to be inadequate on its own to achieve the goal of providing a mode estimate on which relatively drastic control actions can be taken.

The “Hybrid Estimation” component of the new architecture contains the augmented hybrid estimation framework described in Subsection 1.2.3, combining water-relative velocity estimation described in Subsection 1.2.1 with the perceptual mode observation, σ . The result is a more reliable, accurate estimate of the target’s mode achieved by fusing the visual mode observation with more traditional sensor information (relative position measurements from stereo vision and water-relative velocities from acoustic instruments on the tracking vehicle) and rudimentary *a priori* knowledge of the dynamics associated with each behavior mode of the target. While none of these sets of information alone is adequate for deriving a good estimate of target mode, their use together in a complementary manner by the “Hybrid Estimation” component of the architecture results in estimates of target mode of a high enough quality and bandwidth to provide the basis for mode-conditioned control reactions by the robotic observation system, as well as improved estimates of velocities.

The remainder of the architecture of Figure 1.3(b) is left unchanged from the baseline of Figure 1.3(a). Extensions to the “Control” block to implement mode-driven strategies based on the outputs of the “Hybrid Estimation” block are enumerated as the first item of future work by this dissertation.

1.3 Summary of Contributions

To achieve high quality estimation of behavior modes and the resulting velocities for deep ocean animals by the robotic observation system, this dissertation presents the following contributions:

Robotic perception

In this dissertation, real-time algorithms are presented which extract perceptual information about the target being tracked by the robotic observation system. This set of algorithms is inspired by the human observer’s ability to recognize visually body motion modes of deep ocean animals. These perception algorithms consist of a combination of techniques in computer vision, machine learning and other pattern recognition techniques for detection of periodic motions. Thus, the contributions made in this area can be naturally placed into the context of real-time vision-based surveillance applications.

The perception portion of the observation system ultimately takes the form of a classifier, one which applies a fairly generic mode model that captures the relevant characteristics of body motions of many ocean animals. The perception algorithms’ output is

characterized as an uncertain, discrete-valued observation related to the mode of a system (the ocean animal for this case) and an uncertainty function is defined relating the true mode of the target to the observation of that mode output by these algorithms.

Observability of mode transitions by a hybrid estimator

Included in this dissertation is the derivation of a necessary, but not sufficient, condition on a stochastic hybrid system that must be met for all mode transitions of the system to be detectable by a hybrid estimator using observations of the system's continuous states. This condition is most useful for proving that, given a stochastic hybrid system and its uncertainty properties, some or all mode transitions cannot be guaranteed to be detected by such an estimator. It is a variation on conditions, derived previously in the literature, for guaranteeing the ability to detect all of a system's mode transitions. The condition here is more straightforward to evaluate but provides only a necessary condition for mode detections. This is used in the context of the robotic observation system to formally corroborate the empirically demonstrated limitations of existing hybrid estimators when applied here.

Estimation for hybrid systems using perceptual information

This dissertation presents methods for achieving estimation for hybrid systems, augmented by uncertain perceptual information related to the mode of the system. The estimator presented here provides the capability to estimate the mode and continuous states of hybrid dynamical systems with nonlinear component models and stochastic mode switching, with both observations of the continuous states (from traditional noisy sensors) and uncertain discrete-valued observations related to the mode of the system (perceptual information). The techniques presented here can enable successful estimation for hybrid systems where previously reported methods fail due to poor distinguishability of the modes due to some combination of high degrees of sensor noise and unknown disturbances to the system. By utilizing additional perceptual information related to the mode of the system, the estimator is able to detect mode transitions more quickly and accurately. Even for applications where existing techniques are applicable, the approach in this dissertation can improve performance by decreasing the mode estimation delay of the estimator when perceptual information is available.

An extension to the Rao-Blackwellized Particle Filter for hybrid linear systems and to the Gaussian Particle Filter for hybrid nonlinear systems is presented which enables the use of uncertain (visual for this application) mode observations in addition to more

traditional sensor information. This new estimator for hybrid systems with nonlinear submodels is called the *mode-observed* Gaussian Particle Filter.

An extensible framework

The specific techniques presented in this dissertation for robotic perception are designed for visual recognition of the motion modes of marine animals, and the estimation results use parameters specific to the robotic observation system and its targets. However, the overall framework developed in this dissertation for both perception and estimation using perceptual observations is very flexible. Both the perceptual and estimation algorithms are well encapsulated such that the framework can be easily extended to other systems than the deep ocean observation system by substituting other forms of robotic perception, sensors and dynamical models. Hence, any of the component algorithms presented for the robotic observation system application could potentially be improved or modified incrementally without changing any fundamental aspect of the overall architecture.

A fielded underwater robotic observation system

An underwater robotic system for observation of individual ocean animals was initially developed by Rife and Rock [54–57, 59] to track specimens that were assumed to be relatively large and passive. This dissertation documents the partial redesign of that first fielded system to improve its robustness and enable it to track much smaller specimens. With these changes, the system has been made sufficiently robust to be fielded as an operational pilot aid on ROVs operated by MBARI. (The integration into that pilot aid of the mode estimation framework presented in this dissertation remains as future work.)

1.4 Reader's Guide

The remainder of this dissertation is organized as follows: Chapter 2 describes the design and field demonstrations of a vision-based robotic observation system for individual mid-water animals. This chapter briefly summarizes the first version of the system developed by Rife and documents the design changes made to improve its robustness and extend it to smaller specimens. Chapter 3 discusses the application of hybrid, or multiple model, estimation approaches to maneuvering targets and other hybrid systems using noisy sensor data, and demonstrates why estimation using the techniques in recent literature can be predicted to perform poorly for when tracking marine animals with an ROV. Chapter 4 presents the design of an estimator for water-relative velocities of these marine animals,

which will provide a building block for hybrid estimation approaches to the motion of these animals. Chapter 5 discusses in detail the motion behaviors typically encountered when tracking animals in the midwater with an ROV and describes a vision-based perceptual algorithm that classifies the body motion patterns of marine animals. Chapter 6 presents algorithms that enable hybrid estimation using both noisy sensors and the perceptual algorithms of Chapter 5 by developing a new form of Rao-Blackwellized particle filter. This chapter presents the new, mode-observed Gaussian Particle Filter as the final estimation framework for this application, and includes the application of this estimator to both simulated data and field data recorded in Monterey Bay, California. Finally, Chapter 7 offers some conclusions and areas of future work.

Chapter 2

Robotic Tracking of Midwater Animals

The primary contributions of this dissertation are in estimation of system modes using both traditional sensors and perceptual information (derived here from computer vision techniques), all in the context of their application to a robotic observation system used for long-term observation of individual ocean animals. This ability to quickly and accurately detect mode changes by the specimen will enable such a system to react quickly and more intelligently to the velocity changes that accompany these mode changes.

This chapter provides the background on the existing robotic observation system that is necessary to understand the issues solved by these contributions. The first version of the observation system was established and field tested previously by Rife and Rock [54–57, 59], but was designed under the assumption that the targets being tracked are fairly large and slow-moving. To improve the robustness of this fielded system and enable it to track much smaller targets, several design changes were made [51]. These enable tracking of smaller animals by increasing the flexibility of the system in terms of operation at varying relative positions to the target, thereby increasing the control system’s margin for error before losing sight of the target.

These design changes have increased the robustness of the observation system, particularly when applied to targets that are much smaller than were possible in earlier experiments. However, even with these improvements the system can still be evaded when faced with a target exhibiting high accelerations and decelerations. Nonetheless, this redesign was a key prerequisite step necessary before any response to target accelerations could be effective and is therefore documented here in this chapter.

This chapter is organized as follows: first the architecture of the robotic observation system, including both hardware and software, is reviewed. Then, the rationale behind the redesign of the sensors to enhance the observation system’s operational flexibility is given, resulting in the updated system’s new physical configuration. Another significant issue encountered during field testing, sustained unwanted circumnavigational motions around the specimen, is also discussed and a solution is introduced. Field results using the improved system are presented. Finally, it is documented that even with these improvements which enable tracking of smaller and somewhat more agile animals than the previous version of the observation system could handle, tracking continuity can still be broken when faced with very agile animals, thus motivating the remaining contributions of this dissertation.

2.1 Architecture and Relative Positioning Strategy

The vision-based robotic observation system for midwater animals has been demonstrated as a pilot assist on the ROVs *Ventana* and *Tiburón* in Monterey Bay, California [51, 54–57, 59]. The close-up photographs of Figure 2.1 were taken during recent experiments using this system. The system’s architecture, including interfaces to the human operator(s), is depicted in Figure 2.2.

The robotic observation system is operated from a surface support vessel, the R/V *Point Lobos*. The ROV is operated by a human pilot who is directed by a scientist to choose a specimen in the midwater to track. To initialize the system, the ROV pilot positions the vehicle such that the specimen is visible in the viewing cones of both control cameras, and at a close enough range that the vision algorithms on the observation system computer can “see” the specimen. With the animal visible in both cameras, the observation system uses the calibrated intrinsic parameters of each camera and the calibrated parameters of the stereo rig to triangulate to the animal. Thus, a relative position for use by the automatic control system is generated. When satisfied that the system is tracking correctly and the vehicle’s relative position to the target is steady, the automatic control system is engaged.

The coordinate frames used by the observation system are illustrated in Figure 2.3. The stereo camera pair measures the position of the tracked animal in the sensor frame (subscript s) defined to be aligned with one camera. Using measurements of the pan and tilt angles of the camera mount, this position vector is transformed into the vehicle frame (subscript v) where control loops are closed using the vehicle’s thrusters. The control

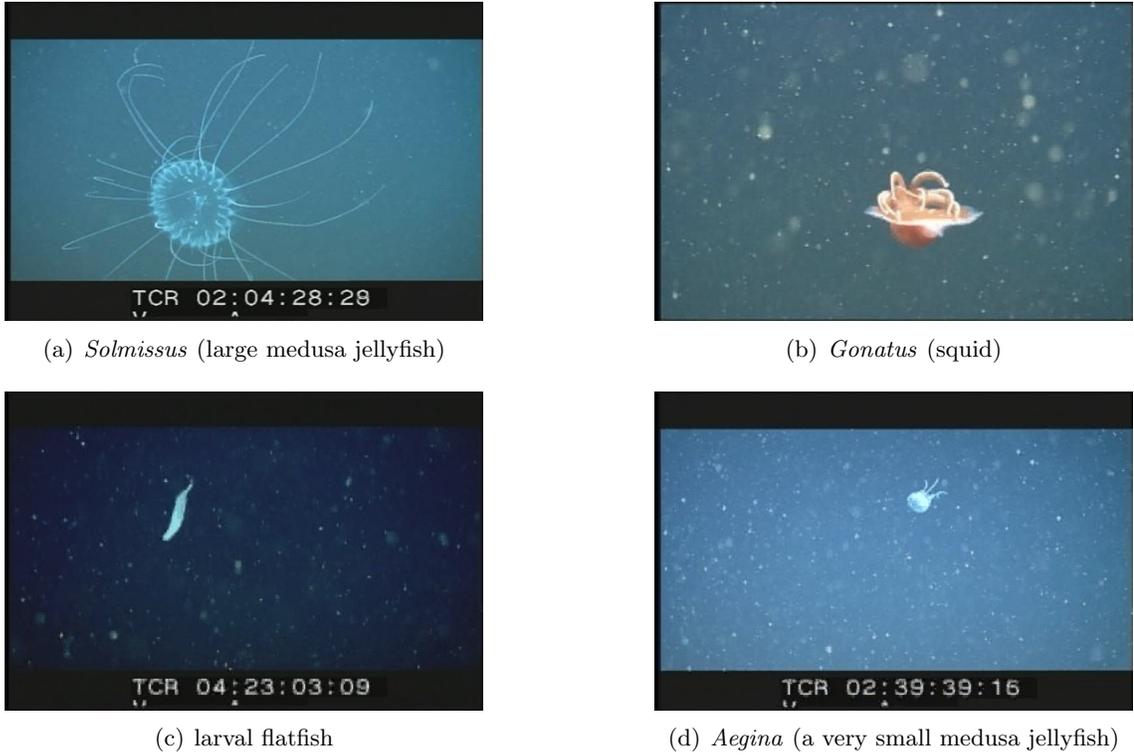


Figure 2.1: Example photographs of tracked animals taken by robotic observation system. The first row are samples tracked before the design improvements documented in this chapter, the second row includes examples that were only enabled after these redesigns. (Images courtesy of MBARI.)

cameras are shown mounted on the pan/tilt servo of the ROV *Ventana* in the upper circled area of Figure 2.4.

The control loops position the vehicle such that the specimen is always in the fields-of-view of both cameras. The control system operates in a cylindrical coordinate system centered on the animal that matches its objective – to keep the vehicle pointed at the animal and at a constant distance from it. Thus, the measurement is converted to the form of (2.1), made up of the distance to the specimen (r), relative heading to the specimen (ψ_{rel}) and relative altitude (z_{rel}):

$$z_{des} = \begin{bmatrix} r & \psi_{rel} & z_{rel} \end{bmatrix}^T \quad (2.1)$$

The closed-loop control system strives to regulate this vector to a constant value that places the object at the center of the views of the cameras.

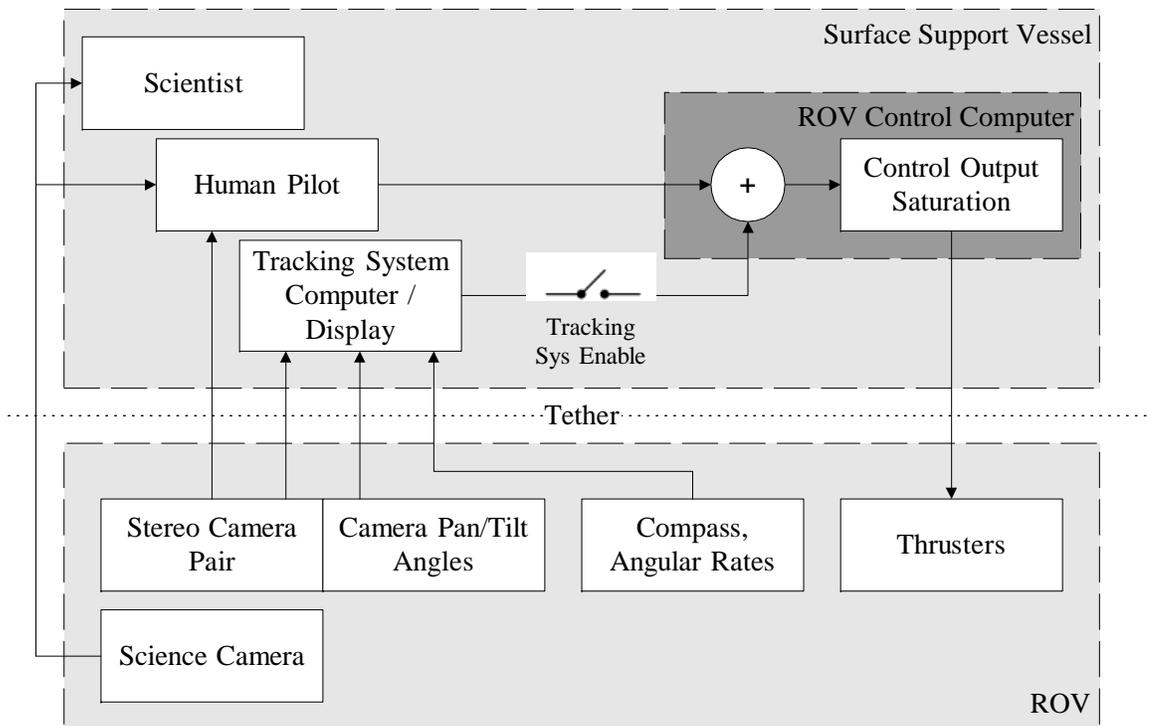


Figure 2.2: Block diagram of robotic observation system hardware and its interactions with the human ROV pilot and scientist.

The control loops are closed using the vehicle's hydraulic thrusters. Vehicle thrusters are set up in three pairs, one to actuate fore/aft and (differentially) yaw, one for lateral motion, and one for vertical. Thrusters do not actively control the vehicle's pitch and roll angles, which are instead stabilized by passive buoyancy moments.

With the specimen maintained in view of the upper cameras, the ROV *Ventana*'s high definition science camera (the large lower camera of Figure 2.4) is free to move without disturbing the control system. Hence, the scientist may freely pan, tilt, shoulder and zoom this camera to capture close-up film and still photographs. For further detail on the design of the control loops and core vision algorithms, see [54, 58, 59].

2.2 Improved Initialization and More Sustainable Tracking

Experience in field testing of the robotic observation system has shown that initialization of the system by the pilot and sustaining tracking for long periods of time can be difficult. This is primarily due to the small volume (with respect to the cameras) in which the

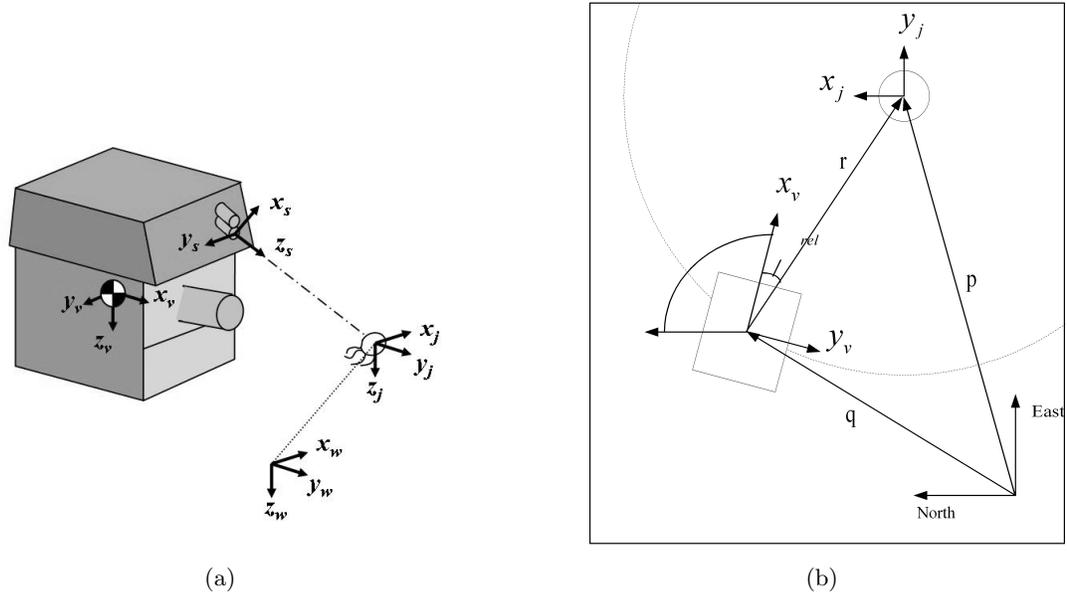


Figure 2.3: Vision-based sensing (a) coupled with cylindrical control system (b) for tracking of midwater ocean animals. In (a), subscripts indicate frames as water (w), sensor (s), vehicle (v) and jelly (j). In (b), the vehicle is indicated by the rectangle, the target jelly by the circle.

target must be maintained. As the system is applied to small specimens, these require closer standoff distances to be adequately visible, resulting in even tighter control error limits on the observation system.

To redesign the system for easier, more robust operation, the constraints imposed on the control system by the sensor must first be explicitly quantified. This section establishes the relationships between control performance constraints, the size and geometry of the common viewing volume of the stereo camera pair defined by the cameras' viewing cones and relative position, the camera rig's position on the vehicle, and the size of the target specimen.

2.2.1 Constraints Imposed on Control Performance by the Vision Sensor

The size of the shared viewable volume of the cameras for a given target and standoff distance imparts constraints on the performance of the control system (or the pilot during initialization) required to maintain the animal within that volume. These constraints, as a function of the camera and specimen parameters, are quantified in this section.

Figure 2.5 illustrates the geometry defined by the camera parameters and position, and the size of the tracked animal. Each camera may be parameterized by its field-of-view

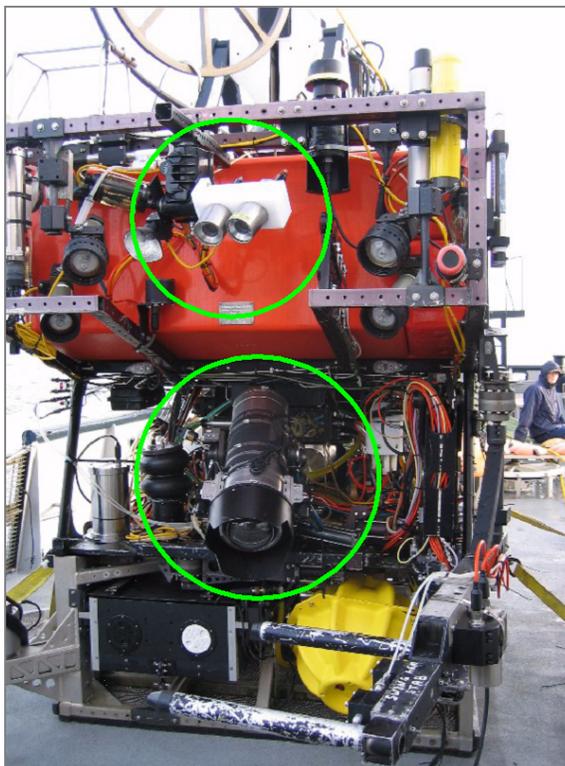


Figure 2.4: The ROV *Ventana*, configured with cameras for midwater tracking (circled, top center). The high definition science camera is indicated in the lower circle.

half-angle, ω , and its distance from the center of the vehicle, L . (This analysis is simplified by assuming that pan and tilt angles are zero, without loss of generality.) The tracked animal is represented by its diameter, D , and its standoff distance from the camera, r .

In addition to the boundaries of the field-of-view of the camera, an effective constraint on distance to the target is imposed by the vision algorithms used to track the animal in each image stream. This constraint takes the form of a minimum percent occupancy of the image that a target must occupy to be discerned consistently from marine snow and other image noise. This occupancy, O , depends on the camera field-of-view, the distance to the target and the size of the target, as defined in (2.2). The lower limit is approximated at 8% of the image dimension (just above the minimum pixel area accepted by the segmentation algorithms, below which any blob is assumed to be marine snow), thus O_{min} is set equal to $.08^2$.

$$O = \left(\frac{D}{2r \sin(\omega)} \right)^2; O_{min} \triangleq .08^2 \quad (2.2)$$

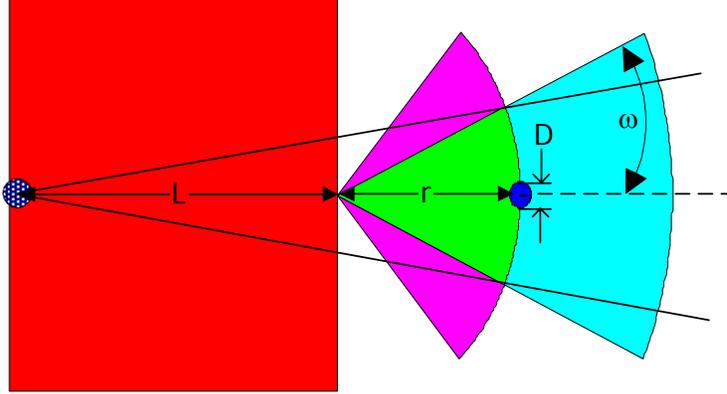


Figure 2.5: Parameterization of camera view of target of diameter D with camera with half-angle field-of-view ω . Camera is located L m from the center of the vehicle. View is from above, looking down at vehicle, and camera as illustrated has no pan or tilt angles. Maximum allowable pointing error is given by (2.4).

For a given O_{min} , D and ω , this can be rearranged to define r_{max} as:

$$r_{max} = \frac{D}{2 \sin(\omega) \sqrt{O_{min}}} \quad (2.3)$$

Thus, r_{max} is the maximum range that the robotic observation system can allow between the camera and the target of diameter D such that it is visible to the vision system using a camera with field-of-view of 2ω .

The constraints on pointing and positioning by the controller (or pilot during initialization) may now be defined. For a given set of parameters r , ω , D , and L , the maximum pointing error, e_{max}^{ψ} , is given by

$$e_{max}^{\psi} = \arctan\left(\frac{r \sin(\omega) - D}{L + r}\right) \quad (2.4)$$

The maximum error in the camera vertical direction is a function of r , the camera's vertical half field-of-view, ${}^c\omega^x$, and D given by

$${}^c e_{max}^x = r \sin({}^c\omega^x) - D \quad (2.5)$$

The stereo camera pair used in earlier field testing consisted of a narrow field-of-view camera, the Insite Orion and a wide angle camera, the Insite Aurora, aligned vertically



Figure 2.6: Camera sets for relative position sensing: (a) vertically aligned parallel wide angle and narrow angle cameras, (b) horizontally aligned and converging narrow angle cameras.

Table 2.1: Calculated maximum standoff distances (relative to the cameras) and control errors for different sizes of observation specimen.

| D (cm) | r_{max} (m) | e_{max}^{ψ} (deg) | ${}^c e_{max}^x$ (cm) |
|----------|---------------|------------------------|-----------------------|
| 10 | 1.54 | 6.3 | 41 |
| 5 | 0.77 | 4.1 | 21 |
| 2.5 | 0.38 | 2.1 | 10 |

(both are products of Insite Pacific, Inc. [31]). When paired with the wide angle Aurora, the entire field-of-view of the Orion is guaranteed to be visible within the view of the wider angle camera. The intersections of the camera fields-of-view are illustrated in Figure 2.7(a). A photograph of this camera set is included in Figure 2.6(a).

With this configuration, maximum standoff distance is limited to the smaller distance allowable by the wide angle camera, and the shared viewing volume of the cameras is defined by that standoff distance and the smaller field-of-view of the narrow camera. In both cases, the control performance constraints are limited unfavorably by the worst case camera. The constraints can be calculated by evaluating e_{max}^{ψ} and ${}^c e_{max}^x$ using (2.4) and (2.5) respectively, at $r = r_{max}^{wide}$ and $\omega = \omega_{narrow}$. The results are tabulated in the left hand entries of Table 2.1 for several diameters of tracking targets.

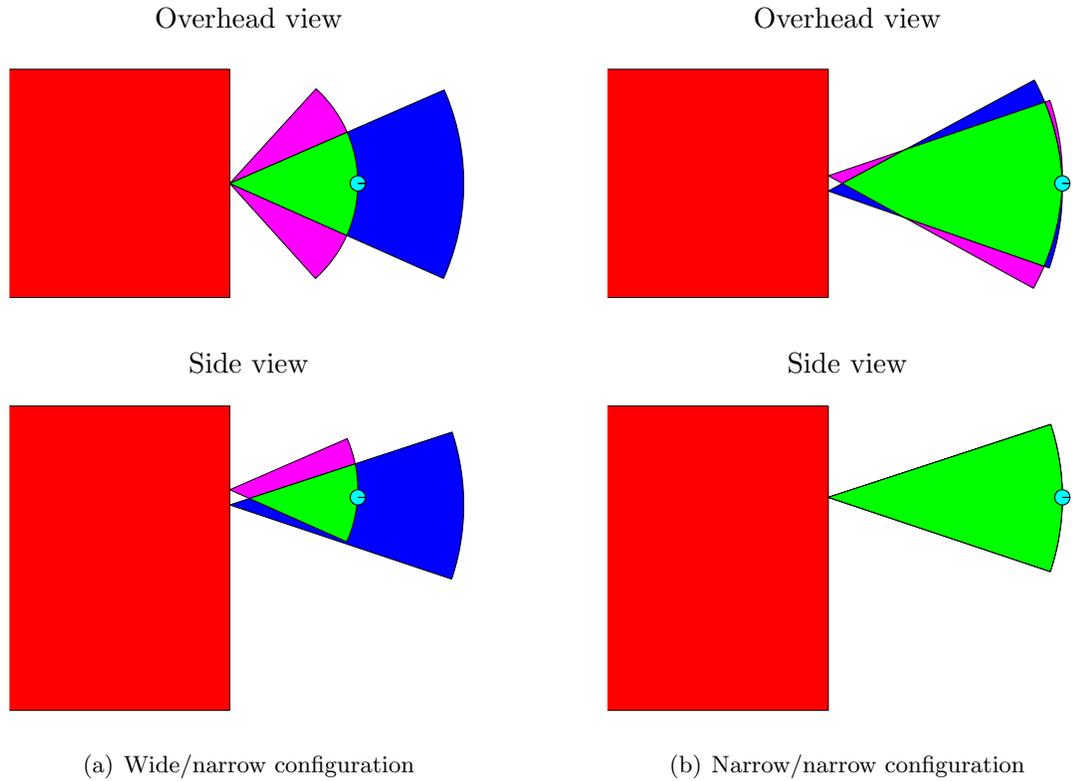


Figure 2.7: Stereo vision cone overlap for (a) a wide/narrow camera pair aligned vertically and parallel and (b) a narrow/narrow camera pair horizontally aligned and converging at a 10 degree angle. Both illustrate the maximum range to target based on the worst case O_{min} constraint for the camera set. The stereo vision sensor was redesigned from the configuration of (a) to that of (b) to increase the volume in which the target can be seen.

2.2.2 A Redesigned Stereo Camera Sensor

Based on the preceding analysis, the camera system was redesigned to enlarge the viewable volume formed by the intersection of the vision cones. Three changes were made: changing to two narrow view cameras, converging the cameras inward toward each other, and aligning them horizontally instead of vertically. A photograph of the redesigned stereo camera pair is shown in Figure 2.6(b).

The choice of two narrow view cameras gives the advantage of being able to maintain a larger standoff distance to the target. This is due to the minimum occupancy condition, (2.2), and its effect on maximum standoff distance, (2.3). The worst case field-of-view used in the previous configuration was the horizontal field-of-view of the wide angle camera, which forced the system to operate very near to the specimen, as shown in Figure 2.7(a).

Now, the worst case field-of-view is the horizontal angle of the narrower camera type, significantly increasing the maximum standoff distance, as illustrated in Figure 2.7(b).

The new configuration also converges the cameras at a small angle (10 degrees). With identical fields-of-view cameras, some portion of the viewing cone of each camera is not visible in the cone of the other. This problem is mitigated by the small convergence angle, which causes the cones to overlap much more, even at close standoff distances.

Because there will always be some part of the view of each camera not visible by the other, the new configuration aligns the cameras horizontally rather than vertically. This takes advantage of the wider field-of-view in the horizontal angle, which when impinged upon by the incomplete overlap, leaves a larger usable volume overall.

To recompute e_{max}^{ψ} for this system, the extrema of the intersected region are calculated based on the diagram shown in Figure 2.7(a). (2.4) cannot be used directly for this case. Instead, the positions of the far extrema of the area shared by both vision cones are calculated, along with an angle with respect to the vehicle center. The pointing error constraint is then half of the difference between the angle made up of the rays from the vehicle's center to those points, and the angle subtended by the target with respect to the vehicle's center. (2.5) for ${}^c e_{max}^x$ is still valid for this configuration, now with the advantage that r_{max} is calculated using the narrow field-of-view.

The right side entries of each column in Table 2.1 tabulate the results from the new camera configuration. With the new camera design, allowable control errors and standoff distances are increased for all cases.

2.3 Mitigating Sustained Circumnavigation of the Specimen

In many field demonstrations of the robotic observation system, unintended and sustained circumnavigation about the specimen at steady-state has occurred. This is due to uncompensated lateral forces, which act in the null space of the control system. These forces likely stem from several sources, such as imperfect alignment of the vehicle's thrusters, non-zero null points on the lateral thrusters, and varying drag properties due to the particular equipment installed on the vehicle.

Any disturbances in the vehicle lateral direction operate in the null space of the cylindrical coordinate control system, which was intended to give the pilot a mechanism to circumnavigate the tracked specimen [54, 59]. The mechanics of this effect are illustrated in Figure 2.8. In that illustration, at time step k (left), the control system commands

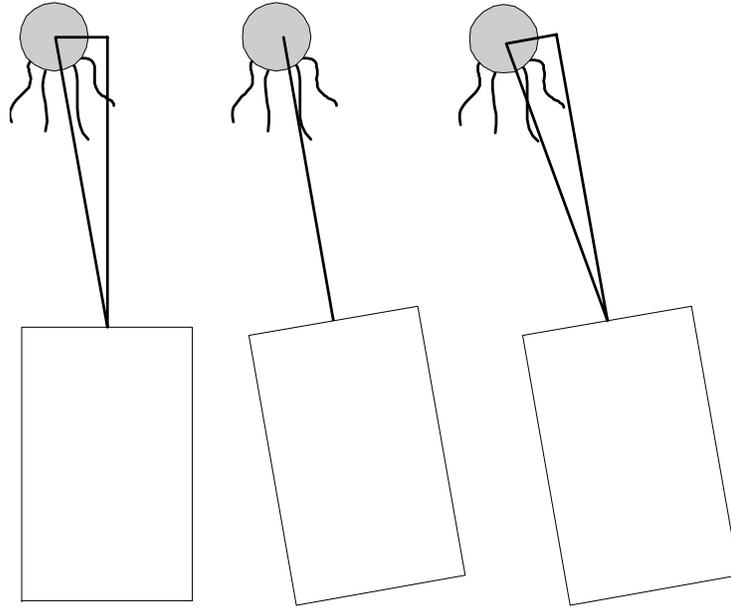


Figure 2.8: The effect of yaw/lateral coupling when tracking a midwater specimen, resulting in sustained circumnavigation about the specimen in the null space of the cylindrical coordinate control system.

a counter-clockwise yaw. The desired result at time step $k + 1$ is a yaw motion to zero relative heading error (middle). With null space disturbances or uncompensated coupling, the actual result is a combined yaw and lateral motion to starboard, resulting in no net reduction in relative heading error at time step $k + 1$ (right). As this process continues over time, the vehicle steadily circumnavigates the specimen. In addition to being an unintended nuisance, when fielded on the ROV *Ventana*, this phenomenon can quickly lead to unacceptable tether management issues.

To mitigate this problem, a secondary (and optional) heading control loop was added to the system, illustrated in Figure 2.9. The loop provides closed-loop compensation for unexpected forces in the null space of the primary tracking loops. This has the effect of placing the vehicle at a fixed position on the cylindrical coordinate system's circumference (see Figure 2.3). The ability to circumnavigate the specimen is preserved in this mode by allowing the system's operator to modify the heading reference.

The new heading loop is secondary to the priority control loops that maintain the tracked specimen in view of the control cameras. Therefore it operates separately from the main loops, utilizing the null space inputs in the lateral direction to position the vehicle

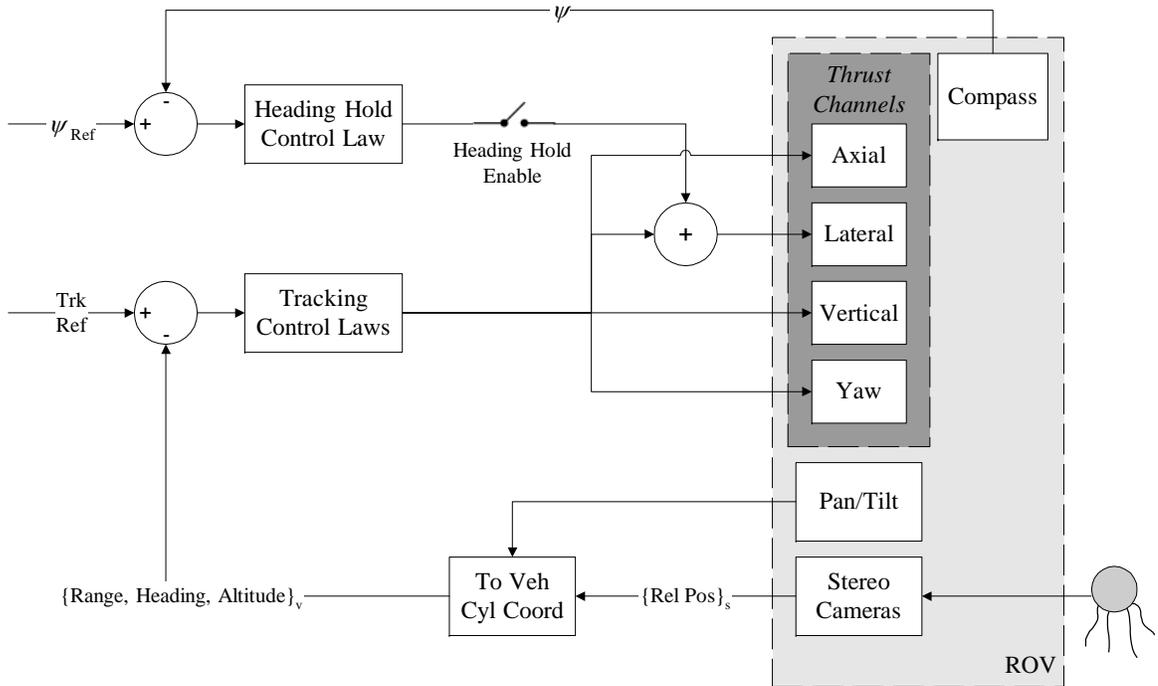


Figure 2.9: Block diagram of robotic observation system with heading hold included.

on the circle. Furthermore, it is designed with a much lower bandwidth than the tracking loops to minimize its effects on their performance.

2.4 Results

2.4.1 Tracking a Wider Variety of Specimens

With the changes in design, a much larger spectrum of animal sizes and types have been tracked by the system, with several examples shown in Figure 2.1. The example photograph in the lower right of Figure 2.1, was taken by the robotic observation system while following an *Aegina* jellyfish. This is a small but mobile jellyfish with diameter on the order of 2cm, too small to be successfully tracked by the system before modification of the camera system. Detailed tracking data from this demonstration is shown in Figure 2.10.

Note that the range maintained in this case was less than 0.5m from the control cameras. The increased viewable volume due to the camera redesign now permits operation at this range. With the previous configuration, a specimen of this size would require operation at a range of approximately 0.2m to see it in the wide angle Aurora model

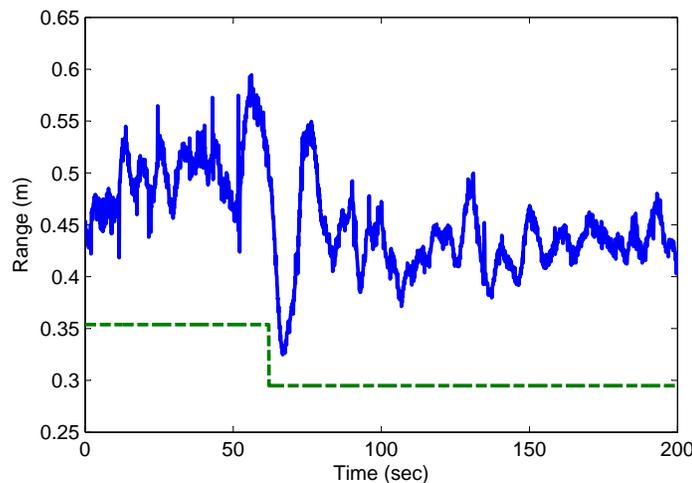


Figure 2.10: Range control while tracking an *Aegina* jellyfish. The lower line (dotted) indicates the range reference value. (The offset is due to tether disturbance forces and the lack of an integral control term during this experiment.)

camera. Operation at that distance is extremely difficult to initialize by the ROV pilot and difficult for the control system to maintain for a significant length of time.

2.4.2 Heading Hold Loop

The heading hold loop has been demonstrated on several tracking demonstrations. An example is shown in Figure 2.11, where heading hold was toggled on and off while tracking a *Solmissus* jellyfish. This example shows the slow action of the heading hold loop to arrest the clocking motion of the vehicle, which was varying between 1 and 2.5 deg/sec counterclockwise before the heading hold loop was enabled at time of about 111 seconds into that dataset.

2.5 Tracking of More Agile Targets

The partial redesign of the robotic observation system documented in this chapter has improved the operability and effectiveness of the system significantly. By giving the controller a much less constraining volume in which the observation targets must be kept, the set of species that is trackable with this system has been expanded to include much smaller and somewhat more agile animals. The improved observation system now can tolerate larger disturbances without breaking tracking continuity, and the larger operating

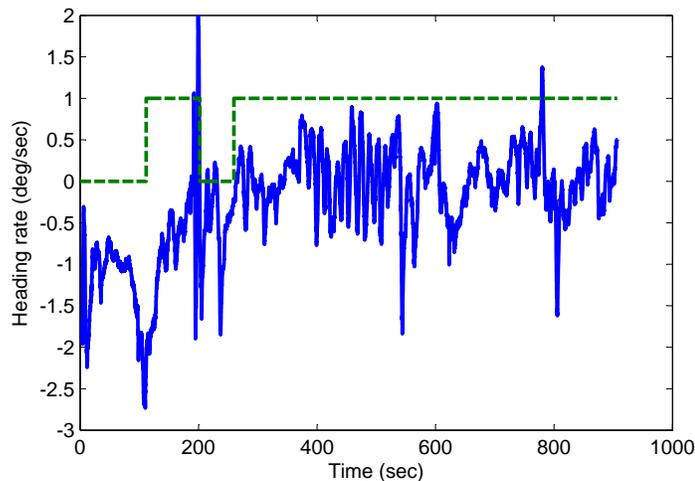


Figure 2.11: Demonstration of heading hold performance while tracking a *Solmissus* jellyfish. The solid line is measured yaw rate of the tracking vehicle. The dotted line indicates heading hold on (1) or off (0).

volume has noticeably improved the ease of initialization by the ROV pilot. With these improvements and the addition of the heading hold loop in the main tracking loops' null space, the system has been made sufficiently robust to be fielded as an operational pilot aid on MBARI's ROVs.

Although the system is fielded and has been demonstrated on many tracking specimens, the redesigned system still often breaks down when tracking animals that exhibit high degrees of acceleration or deceleration. Representative examples of this were given earlier in Figure 1.4. When tracking such marine animals, human pilots of the ROV can recognize changes in the animal's velocity very naturally, using information such as visual indications of changes in swimming modes. The remainder of this dissertation seeks to approximate these capabilities in a proposed enhanced robotic observation system.

Chapter 3

Estimation for Hybrid Systems

The mode estimation framework contributed by this dissertation, combining both traditional sensors and perceptual information, builds upon previously established methods for estimation for hybrid systems. Hence, before presenting that framework, this chapter provides a brief overview of these existing approaches. Perhaps most importantly, this overview includes some required conditions that must be met in order for a hybrid estimator to discern the mode of a system given its mode-based dynamics and available noise-corrupted observations. These conditions may be thought of as requiring a sufficiently high signal-to-noise ratio for each possible mode transition. It will be demonstrated that the dynamics and traditional observations available to infer the mode of the targets of the robotic observation system do not meet those conditions and therefore the additional perceptual information, which will be covered in Chapter 5, is required. Chapter 6 will develop and demonstrate methods which extend hybrid estimation to utilize that kind of information.

For the robotic observation system, both high quality estimates of the mode of the target and improved estimates of the target's velocity will enable tracking of more agile, modal targets. Although both aspects are important, the most powerful improvements are enabled by the mode estimate for the target, and therefore this element will receive the most attention here. Estimation of the target's mode is potentially achievable as an extension of the water-relative velocity estimator of Chapter 4, which uses a Kalman filter-based estimator on a single-mode representation of the joint vehicle/target dynamical system. Once the target's dynamical model from Chapter 4 is replaced by the more expressive hybrid dynamical model introduced in Chapter 5 and fully specified in Chapter 6, the joint dynamical system representing the target and tracking vehicle becomes a

hybrid system. Hence, a number of proven techniques for hybrid estimation could, in principle, be applied to this problem.

This chapter is organized as follows. First, some of the motivating applications where hybrid estimation has been successfully applied are discussed. Then, an overview of the hybrid estimation problem is given, and the main classes of algorithms are summarized. Finally, necessary conditions are derived such that a hybrid estimator can detect all of a system's mode transitions. These are functions of the system's mode-dependent dynamics and its disturbance and observational uncertainty parameters. A simplified one-dimensional version of the marine animal tracking problem is included and parameterized in two different ways to demonstrate these conditions. This example explains in a simpler, intuitive manner the reasons why standard hybrid estimation techniques, given the traditional sensors available to the robotic observation system, are not able to detect the target's mode transitions, motivating the estimation framework that adds perceptual information as covered in Chapter 6.

3.1 Motivations for Hybrid Estimation

Using sensors that observe a system's continuous states and multiple models for the system that correspond to different modes (or other discrete states), estimators for hybrid systems infer the most likely mode of the system in addition to its continuous states. These techniques have been applied in a wide variety of settings. They are useful for tracking switching systems whose dynamics change significantly due to some discrete action (e.g., a valve opening or closing) or large fundamental changes in control inputs (such as those of a maneuvering aircraft) about which the agent doing the tracking estimation does not have direct knowledge. Switching dynamics can also be encountered due to failures of system elements [25, 26, 34–36, 71] or sensors [34] (in which case, the observation model switches rather than the dynamics). For these applications, the primary purpose of the estimator may be to detect fault conditions, modeled as one or more discrete states of a hybrid system. For some systems with modes or maneuvers, in addition to the dynamics changing with mode, the set of sensors utilized also changes. In these cases both the process and observation models can be conditioned on the system's mode.

A particularly important area for the use of hybrid estimation is in tracking of the states of systems whose inputs are unknown to the observer, e.g., maneuvering targets such as aircraft within the Air Traffic Control system [1, 3, 27, 42]. When tracking maneuvering targets without direct knowledge of the system's inputs, it is possible to

model the inputs as state variables using techniques such as the Kalman filter [33] (or nonlinear variants such as the EKF or UKF). However, the large modeling errors in the single model with respect to the maneuvering modes requires the assignment of a large degree of uncertainty to the process model. This reduces the predictive capability of the dynamic models resident in the estimator and increases the undesired response to noise in the observations.

Compared to estimating the unknown input explicitly, better performance can be achieved if the flight regimes into which these inputs place the maneuvering target are modeled separately as modes, each with unique dynamics [3]. The maneuvering mode is recognized by deciding which dynamical model best explains the observations taken on the system, and these more accurate mode-conditioned models permit the estimator to track states more accurately. Hence, tracking of the states of maneuvering targets is improved by utilizing hybrid estimators, enabling capabilities such as enhanced identity management [1] and Air Traffic Control conflict detection [27]. It is in this context of maneuvering target estimation whose maneuver onset times are unknown to the surveillance agent that the tracking of deep ocean animals very naturally fits.

When choosing to apply a hybrid model to a system, an assumption is made that additional knowledge about the modes of that system and the related mode-conditioned dynamics (and/or observations) is available to the design engineer. This additional information may come from domain knowledge of the system, possibly with parameters computed using system identification procedures for hybrid systems, e.g., [2, 5, 72]. The choice to utilize a hybrid model for estimation of a system's states involves a tradeoff between the potential benefits of improved tracking of mode-dependent continuous states versus the additional computational and design complexity incurred by using a hybrid approach.

3.2 Hybrid Estimation Algorithms

3.2.1 Overview

The discussion of estimation for hybrid systems in this chapter pertain to the subset of hybrid systems which are stochastic. That is, the mode transitions are assumed to be stochastic (not based on criteria placed upon the continuous states of the system) and the process dynamics and observations are both corrupted by noise. Such systems are

described as follows:

$$\begin{aligned}\mathbf{x}(k+1) &= f_i(\mathbf{x}(k), \mathbf{u}_i(k), \mathbf{w}_i(k)) \\ \mathbf{y}(k) &= h_i(\mathbf{x}(k), \mathbf{v}_i(k))\end{aligned}\tag{3.1}$$

In (3.1), the continuous state dynamics and observations in state-space form are shown to be parameterized by the current mode index, i , equal to the mode of the system, m , at the current time step, k . Here \mathbf{x} is vector of continuous states, \mathbf{u} are the (deterministic) system inputs, \mathbf{y} are the observations of the continuous states, and \mathbf{w} and \mathbf{v} are normally distributed process and observation noises, respectively.

The mode of the system, m , is a discrete variable from the finite set:

$$m \in \{1, \dots, N_{modes}\}\tag{3.2}$$

with N_{modes} equal to the number of discrete modes of the system¹.

The mode transition model for a stochastic hybrid system is given by that of a Markov process whose properties are specified by a square stochastic transition matrix, H , of dimensions equal to the number of modes of the hybrid system. Thus, each element of H represents a transition probability given by:

$$\begin{aligned}h_{ij} &= p(m_k = i | m_{k-1} = j) \\ i &= 1, \dots, N_{modes}; \quad j = 1, \dots, N_{modes}\end{aligned}\tag{3.3}$$

The hybrid estimation problem is to estimate both the continuous states, $\mathbf{x}(k)$ and discrete mode of the system, $m(k)$, given observations $\mathbf{y}(0), \dots, \mathbf{y}(k)$ and the models of (3.1)–(3.3). The mode estimate usually takes the form of probabilities, $\mu_j(k)$, that $m(k)$ is mode j . If the mode sequence were known, the correct model could be applied at each time step, conditioned on the current mode. However, in general the current mode is not known and must be estimated along with the continuous state. This is a challenging problem, as the estimates of the discrete and continuous states are highly correlated.

Because the mode is not known, the full hybrid estimation problem (often referred to as the *optimal* hybrid estimator) involves consideration of the probabilities of the full set of mode history hypotheses. That is, beginning from N_{modes} current mode hypotheses, at the next time step, there are N_{modes} transition events possible per (3.3), and $(N_{modes})^2$ mode

¹It can be more intuitive to enumerate the modes with names rather than just an index, but those enumerations may always be translated into a numbered set of indices as in (3.2).

trajectory hypotheses to consider. Thus, the number of hypotheses grows exponentially with times considered and this approach cannot be practically implemented.

To address this, hybrid estimation techniques in practice are forced to make approximations. Three classes of approximate hybrid estimation techniques will be summarized here: the multiple model (or bank of filter) techniques, particle filters, and Rao-Blackwellized particle filters.

The most commonly used class of algorithm for approximate hybrid estimation, multiple model algorithms, involve merging the possible mode histories down to a small set, often just one per mode, with state estimates merged into single Gaussian distributions via moment matching. The result is a bank of mode-matched filters, consisting typically of only N_{modes} filters, or in some implementations, $(N_{modes})^2$ filters. Examples of this approach are the Multiple Model Adaptive Estimator (MMAE) [41] and the Interacting Multiple Model estimator (IMM) [3]. These algorithms achieve approximate hybrid estimation at low computational cost and are very popular in practice. They generally assume linearity of the process and observation models, but can handle moderately nonlinear models by replacing the mode-matched Kalman filters with corresponding Extended Kalman filters or Unscented Kalman filters.

On the other end of the approximation spectrum are applications of the sequential Monte Carlo (or particle) filter, to the hybrid estimation problem [36, 42, 63]. These numerically approximate the optimal hybrid estimator by effectively pruning the possibilities among mode histories to only a smaller subset that are not unlikely. This can improve accuracy of the filtering results, and a great deal of flexibility is provided by their abilities to model any uncertainty specifications that can be numerically sampled (not just Gaussian), to model any nonlinear process and observation model, and to estimate in the presence of autonomous mode transitions (those based on criteria on the continuous states). However, their computational cost is significantly higher and can become prohibitive with moderately sized system states.

In between these two classes of algorithm are Rao-Blackwellized forms of particle filtering. When applicable, this form of particle filter partitions the system models into Gaussian and non-Gaussian partitions, computing non-Gaussian variables using the sampling techniques of the particle filter and propagating the Gaussian variables and their uncertainties analytically using Kalman updates. Hence, a great deal of computational cost is avoided compared to classical particle filters. As applied to stochastic hybrid systems, these algorithms act as a larger bank of filters than used in algorithms such as the IMM, and can achieve more accuracy by still pruning unlikely mode histories rather than

merging them together by current mode [14, 18, 25]. This class of hybrid estimation algorithm assumes Gaussian uncertainties in the process and continuous state observations, and the same assumptions about models being linear to moderately nonlinear apply.

3.2.2 Multiple Model Estimators

Multiple model estimators, such as the Interacting Multiple Model estimator (IMM) [3] and Multiple Model Adaptive Estimator (MMAE) [41], are the most commonly used forms of hybrid estimator. These merge the possible mode histories down to a small set, often just one per mode, resulting in a bank of mode-matched filters. For a system whose dynamics in a given mode are linear with Gaussian uncertainty, the Kalman filter computes a Gaussian state estimate. The total posterior density for each mode branching possibility from time steps $k-1$ to k is then a sum of Gaussians. To maintain the ability to use a single Kalman filter for each mode, the sums of Gaussians of each hypothesis ending in that mode are merged into an approximate single Gaussian by matching the first and second moments of the sum. Hence, the overall computational load is kept very low at the expense of this approximation.

The detailed algorithm for the most popular form of multiple model estimator, the IMM, is given in Table 3.1. Note that the MMAE is identical to the IMM except for the omission step 2b, the mixing of the filter initial conditions and covariances of (3.6) and (3.7). Thus, $\hat{\mathbf{x}}_{0j}(k-1|k-1)$ and $\hat{\Sigma}_{0j}(k-1|k-1)$ are substituted with the unmodified $\hat{\mathbf{x}}_j(k-1|k-1)$ and $\hat{\Sigma}_j(k-1|k-1)$ respectively.

3.2.3 Particle Filters for Hybrid Systems

Particle filters in their various forms have enabled a wide range of estimation problems to be solved numerically. For a broad review of these methods, see [17]. Their particular strength lies in their ability to model any type of probability distribution that may be numerically sampled, rather than being limited to Gaussians or mixtures of Gaussians as in Kalman filter variants. Particle filters are most useful when applied to problems that must track multiple hypotheses which result in multi-modal probability distributions on some set of the states.

For hybrid systems, particle filters provide a numerical approximation of the optimal hybrid estimator [42] and enable estimation of hybrid systems with non-Gaussian component distributions [34, 35] or with autonomous mode transitions [36, 63]. With these methods, a more accurate representation of the distribution on continuous states is achieved, particularly during switching events, where the probability density functions

1. **Initialization:** for $i = 1, \dots, N_{modes}$, initialize $\hat{\mathbf{x}}_i(0)$ and $\hat{\Sigma}_i(0)$ from the prior $P(X_0)$ and mode probabilities $\boldsymbol{\mu}$ from $P(M_0)$.

2. **Update:** For each time step, $k > 0$, do:

(a) (Hypothesize/predict mode transitions)
Calculate mixing probabilities:

$$\mu_{ij}(k-1|k-1) = \frac{1}{c_j} h_{ij} \mu_i(k-1) \quad i, j = 1, \dots, N_{modes} \quad (3.4)$$

with normalizing constant c_j given by

$$c_j = \sum_{i=1}^{N_{modes}} h_{ij} \mu_i(k-1) \quad (3.5)$$

(b) Compute the mixed initial states and covariances for each mode-matched filter:

$$\hat{\mathbf{x}}_{0j}(k-1|k-1) = \sum_{i=1}^{N_{modes}} \hat{\mathbf{x}}_i(k-1|k-1) \mu_{ij}(k-1|k-1) \quad (3.6)$$

$$\begin{aligned} \hat{\Sigma}_{0j}(k-1|k-1) = & \sum_{i=1}^{N_{modes}} \mu_{ij}(k-1|k-1) \left\{ \hat{\Sigma}_i(k-1|k-1) \right. \\ & + [\hat{\mathbf{x}}_i(k-1|k-1) - \hat{\mathbf{x}}_{0j}(k-1|k-1)] \\ & \cdot [\hat{\mathbf{x}}_i(k-1|k-1) - \hat{\mathbf{x}}_{0j}(k-1|k-1)]^T \left. \right\} \end{aligned} \quad (3.7)$$

(c) (Mode-matched filtering)

Perform Kalman filter updates for each mode:

$$\begin{aligned} & \left(\hat{\mathbf{y}}_j(k|k-1), \hat{\mathbf{S}}_j(k|k-1), \hat{\mathbf{x}}_j(k|k), \hat{\Sigma}_j(k|k) \right) \\ & \leftarrow KF_j \left(\hat{\mathbf{x}}_{0j}(k-1|k-1), \hat{\Sigma}_{0j}(k-1|k-1), \mathbf{y}(k) \right) \end{aligned} \quad (3.8)$$

and compute likelihoods of the observation $\mathbf{y}(k)$ for each mode:

$$\Lambda_j(k) = \mathcal{N} \left(\mathbf{y}(k); \hat{\mathbf{y}}_j(k|k-1), \hat{\mathbf{S}}_j(k|k-1) \right) \quad (3.9)$$

where $\hat{\mathbf{y}}_j(k|k-1)$ is the predicted observation mean for the j^{th} filter, and $\hat{\mathbf{S}}_j(k|k-1)$ is the predicted residual covariance.

(d) Mode probability update:

$$\mu_j(k) = \frac{1}{c} \Lambda_j(k) c_j, \quad j = 1, \dots, N_{modes}; \quad c = \sum_{j=1}^{N_{modes}} \Lambda_j(k) c_j \quad (3.10)$$

(e) Compute outputs (decoupled from estimation process):

$$\hat{\mathbf{x}}(k|k) = \sum_{j=1}^{N_{modes}} \mu_j(k) \hat{\mathbf{x}}_j(k|k) \quad (3.11)$$

$$\hat{\Sigma}(k|k) = \sum_{j=1}^{N_{modes}} \mu_j(k) \left\{ \hat{\Sigma}_j(k|k) + [\hat{\mathbf{x}}_j(k|k) - \hat{\mathbf{x}}(k|k)] [\hat{\mathbf{x}}_j(k|k) - \hat{\mathbf{x}}(k|k)]^T \right\} \quad (3.12)$$

Table 3.1: The IMM algorithm for stochastic linear hybrid systems.

1. For N particles $p^{(i)}$, $i = 1, \dots, N$, sample discrete modes $m_0^{(i)}$, from the prior $P(M_0)$.
2. For each particle $p^{(i)}$, sample $\mathbf{x}_0^{(i)}$ from $p(\mathbf{x}_0|m_0^{(i)})$.
3. For each time-step k do:
 - (a) For each $p^{(i)} = (m_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)})$ do:
 - i. (Hypothesize/predict mode transition)
Sample a new mode $\hat{m}_k^{(i)} \sim p(m_k^{(i)}|m_{k-1}^{(i)})$
 - ii. (Mode-matched time update)
Sample a new continuous state, $\hat{\mathbf{x}}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, \hat{m}_k^{(i)})$ and a predicted observation, $\hat{\mathbf{y}}_k^{(i)} \sim p(\mathbf{y}_k|\hat{\mathbf{x}}_k^{(i)}, \hat{m}_k^{(i)})$
 - iii. (Likelihood weighting)
Compute the weight of particle $\hat{p}^{(i)}$:
 $w_k^{(i)} \sim p(\mathbf{y}_k|\hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{m}_k^{(i)})$
 - (b) Resample the particle set $p^{(i)}$ such that $P(p^{(i)} = \hat{p}^{(j)}) \propto w_k^{(j)}$
 - (c) Compute outputs (decoupled from estimation process)
 - i. Compute approximate mode probabilities:

$$\mu_j(k) \approx \frac{1}{N} \sum_{i=1}^N I(\hat{m}_k^{(i)} = j), \quad j = 1, \dots, N_{modes} \quad (3.13)$$

with $I(\hat{m}_k^{(i)} = j)$ the indicator function equal to 1 when $\hat{m}_k^{(i)} = j$, 0 otherwise.

- ii. Compute the mean state estimate:

$$\hat{\mathbf{x}}_k = \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{x}}_k^{(i)} \quad (3.14)$$

Table 3.2: The multiple model bootstrap (particle filter) algorithm.

conditioned on each possible value of the current mode of the system are multi-modal - for instance one lobe representing the state distribution if the current mode was also the previous mode, and other lobes representing the density contributions from the hypotheses that the previous mode was different than the current one. Bank of filters approaches merge these contributions into a single Gaussian via moment matching, but that process can result in significant error for some applications during switching events [42]. Particle filters for estimation of hybrid systems do not require this approximation and can therefore track the distribution based on these various hypotheses more accurately. An example of such an algorithm is the multiple model bootstrap filter algorithm [42] for estimation on stochastic hybrid systems, given in Table 3.2.

Although particle filters are extremely flexible with respect to the systems they can model, with fairly modestly-sized state spaces they can quickly become computationally intractable. This is particularly a problem when the filter is designed to run in real-time

at a fast update rate. Therefore, much research in recent years has focused on mixing the sampled numerical approach of particle filters with the analytical uncertainty propagation techniques for normal distributions used in Kalman filters. This class of particle filter is the Rao-Blackwellized particle filter (RBPF) [14, 18] which partitions the states into two sections: one that may be modeled analytically by linear Gaussian conditional distributions and one that is non-Gaussian and is modeled using the sampled techniques of the classical particle filter. This approach can be applied to stochastic linear hybrid systems by modeling the component state evolution dynamics and continuous observations using linear Gaussian relationships, but modeling the discrete mode evolution using the numerical sampling techniques of the particle filter. Thus the continuous states for each sample are updated using Kalman filter updates, conditioned on the mode-history of the particular sample, while the discrete states of each sample are updated numerically by sampling them from the assumed mode transition density function. Table 3.3 summarizes the algorithm. An illustration depicting the process is included later in Figure 6.2.

For systems with nonlinear component dynamics, but where Gaussian uncertainty relationships for state evolution and observations are applicable, the Gaussian Particle Filter [25, 26] utilizes the Unscented Kalman Filter (UKF) for its update of the continuous states within each sample. Table 3.4 summarizes this variant of the RBPF algorithm. Thus, the Rao-Blackwellized particle filter has been extended for nonlinear hybrid systems with partially Gaussian uncertainty. In addition to the GPF, others have applied similar techniques, for instance by utilizing the EKF to model nonlinear component dynamics with Gaussian uncertainty as in the FastSLAM algorithm [44].

3.3 Observability of Mode Transitions in Stochastic Hybrid Systems

Estimation techniques for hybrid systems using measurements related to the systems' continuous states have been applied successfully in many areas, as documented in the previous section. Mode transitions cause divergence of the residuals of filters applying different models – in the presence of noise in the observations and uncertainty within the dynamics of individual models, the task becomes more difficult. Transitions can be detected if their effective signal to noise levels are sufficiently high. In this signal-to-noise analogy, the signal generated by the mode transition is the expected change in residuals of mode-matched filters and the noise is the expected uncertainty of the residuals as a

1. For N particles $p^{(i)}$, $i = 1, \dots, N$, sample discrete modes $m_0^{(i)}$, from the prior $P(M_0)$.
2. For each particle $p^{(i)}$, set $\hat{\mathbf{x}}_0^{(i)}$ and $\hat{\Sigma}_0^{(i)}$ to the prior mean and covariance in state $m_0^{(i)}$.
3. For each time-step k do:
 - (a) For each $p^{(i)} = (m_{k-1}^{(i)}, \hat{\mathbf{x}}_{k-1}^{(i)}, \hat{\Sigma}_{k-1}^{(i)})$ do:
 - i. (Hypothesize/predict mode transitions)
Sample a new mode $\hat{m}_k^{(i)} \sim p(m_k^{(i)} | m_{k-1}^{(i)})$
 - ii. (Mode-matched filtering)
Perform Kalman update using linear Gaussian model parameters $\Theta(\hat{m}_k^{(i)})$ matching mode $\hat{m}_k^{(i)}$:
 $(\hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{\mathbf{S}}_{k|k-1}^{(i)}, \hat{\mathbf{x}}_k^{(i)}, \hat{\Sigma}_k^{(i)}) \leftarrow KF(\hat{\mathbf{x}}_{k-1}^{(i)}, \hat{\Sigma}_{k-1}^{(i)}, \Theta(\hat{m}_k^{(i)}), \mathbf{y}_k)$
 - iii. (Likelihood weighting)
Compute the weight of particle $\hat{p}^{(i)}$:
 $w_k^{(i)} \sim p(\mathbf{y}_k | \hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{\mathbf{S}}_{k|k-1}^{(i)}) = \mathcal{N}(\mathbf{y}_k; \hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{\mathbf{S}}_{k|k-1}^{(i)})$
 - (b) Resample the particle set $p^{(i)}$ such that $P(p^{(i)} = \hat{p}^{(j)}) \propto w_k^{(j)}$
 - (c) Compute outputs (decoupled from estimation process)
 - i. Compute approximate mode probabilities:

$$\mu_j(k) \approx \frac{1}{N} \sum_{i=1}^N I(\hat{m}_k^{(i)} = j), \quad j = 1, \dots, N_{modes} \quad (3.15)$$

with $I(\hat{m}_k^{(i)} = j)$ the indicator function equal to 1 when $\hat{m}_k^{(i)} = j$, 0 otherwise.

- ii. Compute the mean state estimate:

$$\hat{\mathbf{x}}_k = \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{x}}_k^{(i)} \quad (3.16)$$

Table 3.3: The RBPF algorithm for linear hybrid systems with Gaussian noise and stochastic mode transitions.

function of the observation noise and the mode-matched models' uncertainty (process noise).

Hybrid estimation techniques for stochastic systems distinguish between modes by using both filter residuals *and* their covariances. These are compared using observation likelihood functions conditioned on the current discrete mode of the system (or specific mode-branching histories in particle filtering algorithms). Thus, the degree of distinguishability between modes depends on several factors, including the separation of predicted means under differing mode assumptions and the relative overlap in the observation likelihood functions due to the shape of those likelihoods as functions of the uncertainty levels in the observation models and/or process models.

Identical to the RBPF algorithm of Table 3.3, except for step 3(a)ii, which is replaced by:

3(a)ii. (Mode-matched filtering)
 Perform Unscented Kalman update using system and Gaussian parameters matching mode $\hat{m}_k^{(i)}$:

$$\left(\hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{\mathbf{S}}_{k|k-1}^{(i)}, \hat{\mathbf{x}}_k^{(i)}, \hat{\mathbf{\Sigma}}_k^{(i)} \right) \leftarrow UKF \left(\hat{\mathbf{x}}_{k-1}^{(i)}, \hat{\mathbf{\Sigma}}_{k-1}^{(i)}, \Theta(\hat{m}_k^{(i)}), \mathbf{y}_k \right)$$

Table 3.4: The GPF algorithm for nonlinear hybrid systems with Gaussian noise and stochastic mode transitions.

Conditions for observability of both the discrete mode and continuous states of a hybrid system over a wide range of system types and assumptions have been derived for deterministic linear hybrid systems [4, 73] and stochastic hybrid systems [28–30]. Distinguishing between modes can be formalized as the ability of a hybrid estimator to detect all mode transitions within a finite time. For stochastic hybrid systems, conditions for convergence of the mode probabilities to the correct mode by a hybrid estimator were derived in [29]. Here, those conditions are revisited and a new, simplified condition for proving the lack of mode observability for some or all modes is derived. After the conditions under which transition to some modes are not observable are given, they are applied to an example system that is a very simplified version of the system model that will be applied later for tracking active marine animals.

3.3.1 Necessary Conditions for Mode Transition Detection

The conditions for observability of mode transitions for stochastic hybrid systems have been derived by analysis of the residual means and covariances as they affect the mode probability recursions [29]. These were used to prove that a mode transition will be detected by a hybrid estimator within a finite time after the transition (assuming the sojourn time after the transition is greater). A handful of key equations are repeated here for clarity, but an alternate, simpler form most useful for showing that a mode transition is *not* observable will be derived shortly.

First, from (3.4) and (3.5), the mode probability recursion for hybrid estimation with one filter per mode (such as the MMAE and IMM) is given by:

$$\mu_i(k+1) = \frac{\Lambda_i(k)}{c(k)} \sum_{l=1}^{N_{modes}} h_{il} \mu_l(k-1) \quad (3.17)$$

where $c(k)$ is a normalizing constant that ensures that the sum of all mode probabilities at time k add to 1. The likelihood function $\Lambda_i(k)$ is given by the following expression when

Gaussian residuals are assumed:

$$\Lambda_i(k) = (2\pi)^{-n/2} |\mathbf{S}_i|^{-1/2} \exp\left(-\frac{1}{2} \mathbf{r}_i(k)^T \mathbf{S}_i^{-1} \mathbf{r}_i(k)\right) \quad (3.18)$$

with n equal to the dimension of the observation, and $\mathbf{r}_i(k)$ and \mathbf{S}_i the residual and predicted residual covariance, respectively, of the i^{th} filter at time k .

In [29], conditions for the mode probability at time k to match the correct mode T were derived using an equation that combined (3.17) and (3.18) with the requirement that $\mu_T(k) > \mu_i(k), \forall i \neq T$, resulting in the following:

$$\begin{aligned} 0 &\leq \mathbf{r}_T(k)^T \mathbf{S}_T^{-1} \mathbf{r}_T(k) \\ &< \mathbf{r}_i(k)^T \mathbf{S}_i^{-1} \mathbf{r}_i(k) + \ln\left(\frac{|\mathbf{S}_i|}{|\mathbf{S}_T|}\right) + 2 \ln\left(\frac{\bar{\mu}_T(k)}{\bar{\mu}_i(k)}\right) \end{aligned} \quad (3.19)$$

with $\bar{\mu}_i(k) = \sum_{l=1}^N h_{il} \mu_l(k)$, thus including both the mode probabilities' previous values in the recursion and the rows of the mode transition model matrix, H . Hence, the last term in (3.19), requiring $\bar{\mu}_i(k)$ to be computed, is difficult to calculate other than by computing the full estimator recursion.

Instead, a similar (and necessary but not sufficient) condition for convergence is re-derived here based only on the requirement that the maximum likelihood value of the true mode's likelihood function be greater than that of all other modes at steady-state after a mode transition:

$$\bar{\Lambda}_T(k) > \bar{\Lambda}_i(k), \quad \forall i \neq T \quad (3.20)$$

Without this condition holding, the hybrid estimator cannot converge to all possible mode transitions correctly, as the right-most term in (3.17) (renamed as $\bar{\mu}_i$ earlier) is driven by the constant Markov matrix, H . That matrix can only bias the mode estimate in certain directions based on the previous mode probabilities – the typical diagonally dominant H biases the mode probabilities such that the currently dominant mode tends to remain that way. Hence, the convergence of the recursion on mode probabilities toward any new mode after a transition requires that (3.20) holds after a finite time from the transition.

By substituting (3.18) into (3.20), eliminating the common term of $(2\pi)^{-n/2}$ and taking the natural logarithm of both sides, the following results:

$$-\frac{1}{2} \ln |\mathbf{S}_T| - \frac{1}{2} \bar{\mathbf{r}}_T(k)^T \mathbf{S}_T^{-1} \bar{\mathbf{r}}_T(k) > -\frac{1}{2} \ln |\mathbf{S}_i| - \frac{1}{2} \bar{\mathbf{r}}_i(k)^T \mathbf{S}_i^{-1} \bar{\mathbf{r}}_i(k) \quad (3.21)$$

$$\bar{\mathbf{r}}_T(k)^T \mathbf{S}_T^{-1} \bar{\mathbf{r}}_T(k) < \bar{\mathbf{r}}_i(k)^T \mathbf{S}_i^{-1} \bar{\mathbf{r}}_i(k) + \ln \left(\frac{|\mathbf{S}_i|}{|\mathbf{S}_T|} \right) \quad (3.22)$$

Because the residual covariance is by definition symmetric positive semidefinite, (3.22) can be augmented to show both terms greater than or equal to zero for observability of that mode transition:

$$\begin{aligned} 0 &\leq \bar{\mathbf{r}}_T(k)^T \mathbf{S}_T^{-1} \bar{\mathbf{r}}_T(k) \\ &< \bar{\mathbf{r}}_i(k)^T \mathbf{S}_i^{-1} \bar{\mathbf{r}}_i(k) + \ln \left(\frac{|\mathbf{S}_i|}{|\mathbf{S}_T|} \right) \end{aligned} \quad (3.23)$$

At steady state (a sufficiently long time after a transition), the residuals may be characterized by their steady state mean, $\bar{\mathbf{r}}_{i,ss}$, and (3.23) becomes

$$\begin{aligned} 0 &\leq \bar{\mathbf{r}}_{T,ss}^T \mathbf{S}_T^{-1} \bar{\mathbf{r}}_{T,ss} \\ &< \bar{\mathbf{r}}_{i,ss}^T \mathbf{S}_i^{-1} \bar{\mathbf{r}}_{i,ss} + \ln \left(\frac{|\mathbf{S}_i|}{|\mathbf{S}_T|} \right) \end{aligned} \quad (3.24)$$

If the conditions of (3.24) do not hold for all $i \neq T$, the mode probabilities for mode transitions to mode T will not converge. Thus (3.24) not holding implies that the mode transition is not observable using a hybrid estimator with the assumed dynamic and observation models. This tends to occur when the mean residual error of the mismatched models is fairly small relative to the predicted residual covariance, which is a function of both measurement noise assumptions and the process noises modeled.

3.3.2 A Target Tracking Example with One-Dimensional Motion

To illustrate this effect, an example of the use of hybrid estimation for target tracking in one dimension is presented here. This example can be thought of as a very simplified version of the dynamics of the targets of the underwater robotic observation system. By example here, given a set of uncertainty parameters about process and measurement noise for each mode of the system, transitions from quiescent to active modes will be demonstrated to be observable only if the mean target velocity is sufficiently high. This is demonstrated both empirically by simulation using a hybrid IMM estimator and by evaluating the conditions derived in the previous section.

The model used here is summarized as follows. The target is constrained to move in only one direction and all process and observation models assumed here are linear and

Gaussian. The state vector, \mathbf{x} , consists solely of the target's velocity, \dot{p} . The modal aspect of the system is modeled using two kinematic models, one for resting (driven toward zero velocity) and one nominally constant velocity model which can adapt towards any constant velocity exhibited by the target. The measurement vector, \mathbf{y} , is a simple measurement of the object's velocity, corrupted by zero-mean Gaussian noise.

The hybrid estimator's system models are thus defined as follows:

$$\begin{aligned} \mathbf{x} &= \dot{p} \\ m &\in \{Rest, ConstVel\} \end{aligned} \quad (3.25)$$

defines the hybrid state of the system. A stochastic mode transition model of a constant Markov matrix, H , defined by (3.3) is assumed to be given by:

$$H = \begin{bmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{bmatrix} \quad (3.26)$$

The process model and measurement functions are given by the following:

$$\begin{aligned} \dot{x} = f(m, \mathbf{x}) = \ddot{p} &= \begin{cases} -\tau + n_r; & m = Rest \\ n_{cv}; & m = ConstVel \end{cases} \\ n_r &\sim \mathcal{N}(0, \sigma_r^2) \\ n_{cv} &\sim \mathcal{N}(0, \sigma_{cv}^2) \end{aligned} \quad (3.27)$$

$$\begin{aligned} y = h(x) &= \dot{p} + n_{\dot{p}} \\ n_{\dot{p}} &\sim \mathcal{N}(0, \sigma_{\dot{p}}^2) \end{aligned} \quad (3.28)$$

The state truth data was generated using the following simplified function:

$$\dot{p} = \begin{cases} 0; & m = Rest \\ \dot{p}_0; & m = ConstVel \end{cases} \quad (3.29)$$

Two representative simulations are carried out, with identical noise parameters but different steady-state velocities of the target, \dot{p}_0 . The simulated mode history scenarios are identical – the target begins at rest, switches to a velocity of \dot{p}_0 , then later returns to rest. In the first case, the target's active velocity is relatively large, making these two modes easy to discern using hybrid estimation. In the second case, the velocity is much lower, at a level typically exhibited by marine animals tracked by the robotic observation system. At this lower velocity, the residuals of mismatched models are smaller and the

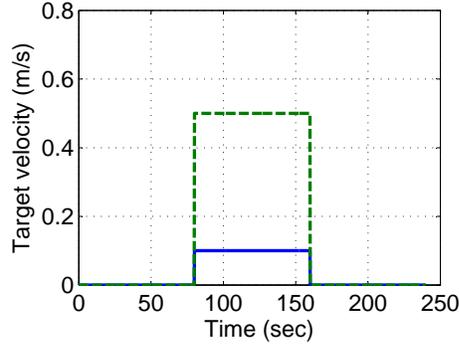


Figure 3.1: Velocity trajectories for two simulated cases. Dotted – high velocity of 0.5m/s, Solid – low velocity of 0.1m/s.

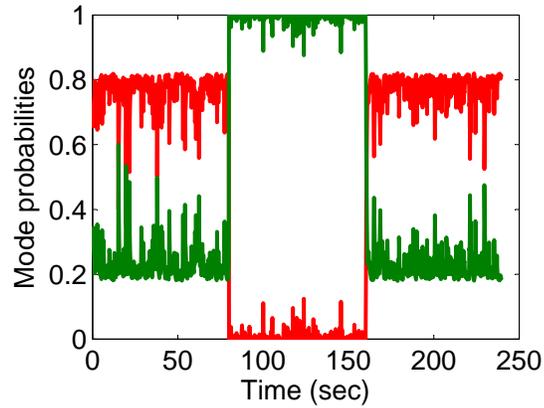
hybrid estimator cannot correctly identify the mode change from rest to active motion. The two velocity trajectories are plotted in Figure 3.1. The parameters used for these examples to completely specify the models in (3.25)–(3.28) are specified in Table 3.5. The velocity sensor noise is chosen to match the noise properties of the Doppler Velocity Log instruments used by the robotic observation system.

| Parameter | Value | Units |
|--------------------|-------|-------------------|
| σ_r | 3.1 | cm/s ² |
| σ_{cv} | 7.1 | cm/s ² |
| $\sigma_{\dot{p}}$ | 3 | cm/s |
| τ | 2 | s |

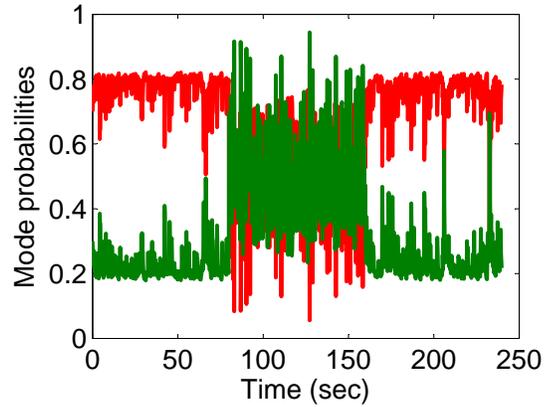
Table 3.5: Summary of model parameters assumed for one-dimensional example.

Figure 3.2 shows the mode probabilities calculated by the IMM. For both cases, the *Rest* mode can be correctly observed by the hybrid estimator. However, the ability to detect transitions to the *ConstVel* mode depend on the mean velocity of the target. At 0.5m/s, that transition is observable, but at the lower value of 0.1m/s it is not.

Table 3.6 tabulates the terms of (3.24) for each mode of this system. This corroborates the empirical demonstration of mode observability properties shown in Figure 3.2. The low target velocity case cannot be detected despite the fact that the *Rest* mode mean residuals are non-zero. This is because the separation of the residual means is small and therefore the shapes of the residual covariances dominate. Thus, the narrower likelihood function for *Rest* still dominates that of the wider *ConstVel* mode’s likelihood. These effects are illustrated graphically in Figure 3.3.



(a)



(b)

Figure 3.2: Mode probability estimates for 1D example. Legend: red - *Rest*, green - *ConstVel*. In (a) the target is moving with velocity of 0.5m/s. In (b) the target is moving with velocity of 0.1m/s.

3.3.3 Application to the Robotic Observation System

The example in the previous section provides a great deal of insight into the performance of hybrid estimation algorithms using the continuous measurements available to the robotic observation system. In that system, the degree of noise in the key measurement of vehicle velocity from the Doppler Velocity Log, combined with the large unknown disturbances on the tracking vehicle make the modes of the target very difficult to distinguish using the available continuous sensors.

Mode tracking results when applying the Gaussian Particle Filter (GPF) to that system are very similar to that of the one dimensional example when tracking the lower velocity target shown in Figure 3.2(b). These results are covered in detail in Chapter 6. However,

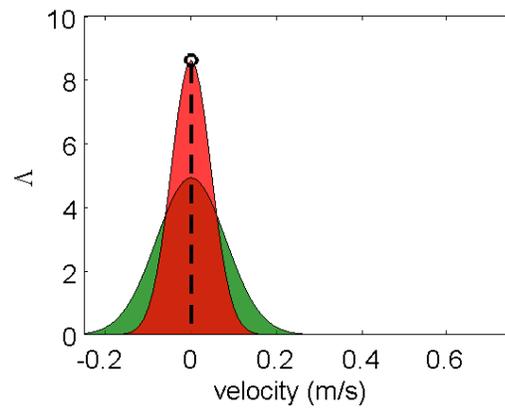
| Quantity | Mode (T) transitioned to | | |
|---|------------------------------|--------------------------|--------------------------|
| | <i>Rest</i> | <i>ConstVel</i> , 0.5m/s | <i>ConstVel</i> , 0.1m/s |
| $\bar{\mathbf{r}}_{i,ss}$ | 0 | 0.097 | 0.02 |
| $\bar{\mathbf{r}}_{T,ss}$ | 0 | 0 | 0 |
| $ \mathbf{S}_i $ | 0.0066 | 0.002155 | 0.002155 |
| $ \mathbf{S}_T $ | 0.002155 | 0.0066 | 0.0066 |
| $\ln\left(\frac{ \mathbf{S}_i }{ \mathbf{S}_T }\right)$ | 1.12 | -1.12 | -1.12 |
| $\bar{\mathbf{r}}_{i,ss}^T \mathbf{S}_i^{-1} \bar{\mathbf{r}}_{i,ss}$ | 0 | 4.366 | 0.186 |
| $\bar{\mathbf{r}}_{i,ss}^T \mathbf{S}_i^{-1} \bar{\mathbf{r}}_{i,ss} + \ln\left(\frac{ \mathbf{S}_i }{ \mathbf{S}_T }\right)$ | 1.12 | 3.25 | -0.94 |
| $\bar{\mathbf{r}}_{T,ss}^T \mathbf{S}_T^{-1} \bar{\mathbf{r}}_{T,ss}$ | 0 | 0 | 0 |
| Condition (3.24) met? | Yes | Yes | No |

Table 3.6: Mode observability test quantities for one dimensional example.

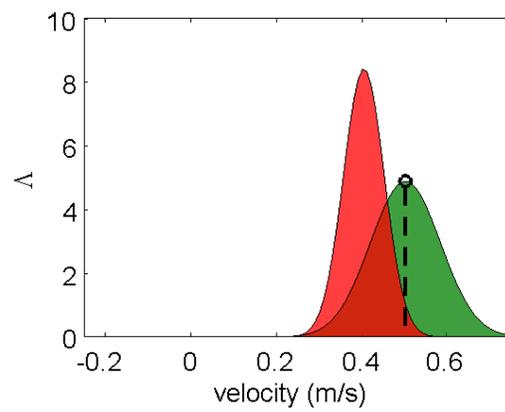
the conditions of (3.24) are tabulated for the full system here in Table 3.7, demonstrating that given the typical velocity levels of the target relative to the noise parameters of the sensors and process, some mode transitions of that system cannot be detected with a hybrid estimator. Additional information beyond the traditional sensors available to the observation system is required to discern the modes – this will be demonstrated in detail in Chapter 6 by augmenting a hybrid estimator with mode-related perceptual observations.

| Quantity | Mode (T) transitioned to | |
|---|---|---|
| | <i>Rest</i> ($i1 = VelTrans, i2 = ConstVel$) | <i>ConstVel</i> ($i1 = VelTrans, i2 = Rest$) |
| $ \mathbf{S}_{i1} $ | 1.0e-6 | 1.0e-6 |
| $ \mathbf{S}_{i2} $ | 4.4e-7 | 3.5e-7 |
| $ \mathbf{S}_T $ | 3.5e-7 | 4.4e-7 |
| $\ln\left(\frac{ \mathbf{S}_{i1} }{ \mathbf{S}_T }\right)$ | 1.05 | 0.82 |
| $\ln\left(\frac{ \mathbf{S}_{i2} }{ \mathbf{S}_T }\right)$ | 0.23 | -0.23 |
| $\bar{\mathbf{r}}_{i1,ss}^T \mathbf{S}_{i1}^{-1} \bar{\mathbf{r}}_{i1,ss}$ | 0 | 0 |
| $\bar{\mathbf{r}}_{i1,ss}^T \mathbf{S}_{i1}^{-1} \bar{\mathbf{r}}_{i1,ss} + \ln\left(\frac{ \mathbf{S}_{i1} }{ \mathbf{S}_T }\right)$ | 1.05 | 0.82 |
| $\bar{\mathbf{r}}_{i2,ss}^T \mathbf{S}_{i2}^{-1} \bar{\mathbf{r}}_{i2,ss}$ | 0 | 1.4e-6 |
| $\bar{\mathbf{r}}_{i2,ss}^T \mathbf{S}_{i2}^{-1} \bar{\mathbf{r}}_{i2,ss} + \ln\left(\frac{ \mathbf{S}_{i2} }{ \mathbf{S}_T }\right)$ | 0.23 | -0.23 |
| $\bar{\mathbf{r}}_{T,ss}^T \mathbf{S}_T^{-1} \bar{\mathbf{r}}_{T,ss}$ | 0 | 0 |
| Condition (3.24) met? | Yes | No |

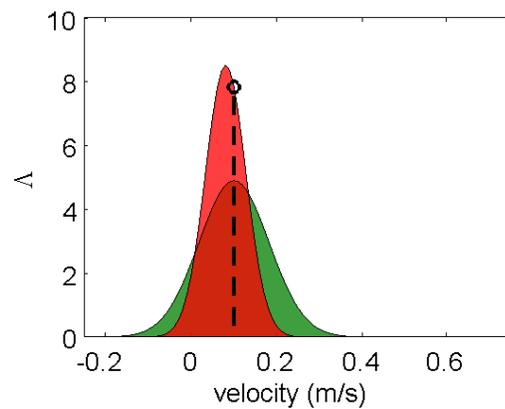
Table 3.7: Mode observability test quantities for underwater tracking system.



(a)



(b)



(c)

Figure 3.3: Measurement likelihood functions for 1D example. Legend: red - *Rest*, green - *ConstVel*, dotted stem - steady-state mean measurement. In (a) the target is at rest. In (b) the target is moving with velocity of 0.5m/s. In (c) the target is moving with velocity of 0.1m/s.

Chapter 4

Estimation of Water-Relative Motion

The essential product of this dissertation is a framework for estimating the mode and continuous states of a hybrid system by fusing two very different kinds of available information: (1) “conventional” sensors which observe the system’s continuous states and (2) perceptual information about the system’s mode. The development of this framework focuses on the motivating application of automatic tracking of underwater creatures such as jellyfish, whose motion behaviors are strongly modal. Achieving successful tracking in many situations hinges on knowledge of the creature’s mode.

This chapter focuses on the first main element of the estimation framework, the estimation of the target’s water-relative motion, which is driven primarily by its swimming mode. For the agile specimens of the robotic observation system, the key new information required (compared to the first fielded version of the observation system) is the water-relative motion of the target, from which its swimming mode can potentially be inferred. This motion term had been unnecessary and therefore neglected in the first version by assumption that targets were relatively passive. However, a motivation for this dissertation is to extend that system’s capabilities toward tracking of much more agile targets whose water-relative accelerations can be significant. To make this water-relative motion term observable, the system is augmented with an additional sensor, a Doppler Velocity Log (DVL). This chapter presents the design of a new estimator for the observation system that fuses this DVL with the stereo camera sensor readings being used previously to estimate water-relative motion in addition to the target’s vehicle-relative position.

The estimator introduced in this chapter does not estimate the mode of the target, but it does provide the water-relative motion estimates from which mode can be inferred.

Thus, this estimator provides a key building block for the hybrid estimation approaches that are presented later in this dissertation, which involve more complex and expressive system models that include discrete target modes.

The chapter is organized as follows: first, an overview is provided for the problem of estimating water-relative motion of the target tracked by the robotic observation system. This is followed by a review of the estimation approach that was employed previously to estimate only the position and velocity of the vehicle with respect to the target. Finally, the new estimator that incorporates the additional information from the DVL to estimate water-relative motion states is presented.

4.1 Overview

When tracking targets that exhibit behaviors that result in significant accelerations or decelerations, the assumption made in the earlier versions of the observation system [51, 54–57, 59] of zero target acceleration is no longer appropriate. Since the stereo camera system provides only a relative measurement, additional information must be used to observe the absolute velocities of the target in the water frame, the frame where swimming motions occur. One such way to observe the target’s motion is to supplement the relative positions observed by a vision system with an estimate of the motion of the camera in the water frame. A source for this information on the vehicle is the Doppler Velocity Log (DVL), an acoustic sensor which, in the midwater, measures the velocities of the sensor with respect to the water column. Hence, with relative motion between the target and vehicle captured by the stereo camera pair, and water-relative motion of the vehicle measured by the DVL, the velocity of the target relative to the water column becomes observable.

This chapter describes the estimation of relative position between the vehicle and target and the absolute velocities (with respect to the water column) of both the vehicle and target. The multi-sensor approach complements the stereo vision with measurements from other sensors typically found on underwater robotic vehicles, as previously documented in [50]. These measurements are fused by a dynamic estimator that naturally handles multiple sample rates and brief unavailability of subsets of the sensors. The design of a robust estimator for this problem requires consideration of several issues including the assumptions about the specimen’s motion behavior, uncertainty in the dynamic models of the vehicle, and toleration of substantial unmodeled disturbances. A nonlinear multi-rate estimator implemented with a Sigma Point Kalman Filter (SPKF) (Unscented Kalman Filter (UKF) implementation) is used here.

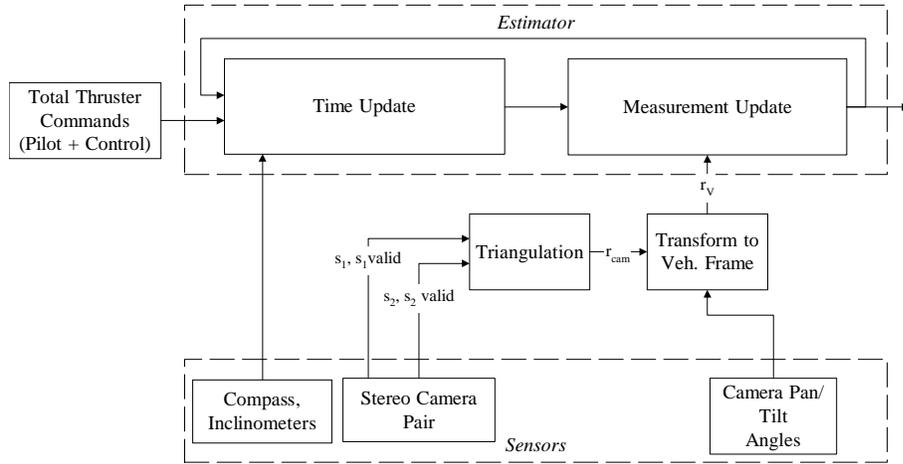


Figure 4.1: Estimator architecture for relative states only.

Although this approach estimates the target’s velocity in the water column, only low bandwidth estimates are achievable due to the high noise content and low sample rate of the DVL water column velocity measurement and the lack of high-fidelity predictive models of the vehicle, which is subject to significant unmeasured disturbances. Therefore this estimation approach alone does not provide the high bandwidth detection of accelerations needed for fast recognition of mode changes as a basis for vigorous control actions. However, the form and underlying equations of this estimator provide the building blocks for the hybrid estimation approaches that are presented in Chapter 6.

4.2 Estimation of Relative States Only

Before introducing the estimator which includes the water-relative motions of the tracked target, this section provides a brief review of the estimator that has been used previously for estimation of target-relative states (see [54] for a more complete discussion). Several noteworthy, simplifying assumptions were made in designing that estimator. These assumptions include that the target being tracked is non-accelerating, that vehicle motions are small enough that linear dynamic models are sufficiently accurate, and that stereo vision measurements of bearing coupled with a triangulation function can be abstracted into a linear sensor model that measures relative position of the target from the vehicle. These assumptions made it possible to use a linear state observer for state estimation. The architecture of this estimator is pictured in Figure 4.1.

Because the system was initially designed to track fairly large and passive ocean animals, the tracked target was assumed to be non-accelerating for the purposes of estimation

and control in the implementations of [54, 59] and also that of Chapter 2 of this dissertation. With this assumption, any actual accelerations by the target are treated as a component of the disturbances to the relative positioning control system.

Two important linearity assumptions were also made. First, the motions of the vehicle’s position relative to the target and in orientation with respect to the Earth were assumed to be small and perturbational in nature. Under these assumptions, linearized models of the dynamics of the vehicle in the estimator were adequate and helpful. Second, the sensor model was also assumed to be linear, and consisted of a “measurement” of relative position – the output of a triangulation function which processes the pixel locations passed from the visual tracking algorithms. This relative position is transformed into vehicle frame and assumed to be corrupted by Gaussian noise.

The linear estimator was included in the system for two main reasons, both driven by the desire to improve relative positioning control performance. The first is to provide better velocity estimates from the noisy stereo triangulation results than could be achieved by using a simpler low pass filter coupled with a numerical derivative function. Thus, all lead information on the relative position between the vehicle and target was derived from noisy stereo vision measurements complemented by a low fidelity linearized model of the vehicle’s dynamics. The second purpose of this estimator was to estimate the force bias on the vehicle to enable experiments in smooth transfer between pilot and automatic control (although this feature was dropped from the implementation of Chapter 2). This relied upon information about the thruster levels commanded by the vehicle control system and ROV pilot, and the linearized dynamic model of the vehicle, including terms approximating first order dynamics for the thrusters’ response to their commands.

The estimation approach presented in this chapter makes a number of important changes in assumptions, approximations and goals. Most fundamentally, the target’s velocity in the water frame is now a key component of the estimator’s state vector. Therefore, the target is no longer assumed to be non-accelerating. These velocities are made observable by adding a water-relative velocity measurement to the system. Further, the sensor models are no longer linearized – the pixel measurements are used directly as measurements, and the nonlinear functions relating these to the relative position between the target and camera are incorporated into the estimator. This choice yields some significant advantages. The biasing effects of the Gaussian covariance assumption about the triangulated relative position measurement are avoided [64], and the effects of pixel measurement noise on relative position error covariance are more accurately modeled [24, 64]. Also, by moving the triangulation function from an external function of noisy bearings to

an implicit portion of the estimator sensor model, sensor dropouts are handled naturally and more consistently than had been possible using the ad-hoc dropout handling methods of the previous architecture. Another new consideration addressed by the nonlinear estimator is the nonlinear coupling between the vehicle's angular velocity and the newly added water-relative velocity measurements, which is significant due to the long moment arm to the rear-mounted velocity sensor. Finally, the low fidelity linear dynamic model was dropped in favor of a kinematic model with white noise acceleration process noise (covering disturbances and control inputs). These design choices and the overall design of the nonlinear estimator that is implemented is covered in detail by the remainder of this chapter.

4.3 Estimation of Water-Relative Motions of the Target and Vehicle

In this section, a new estimator is proposed for tracking of the target and vehicle's relative position and each of their velocities with respect to the water column. To achieve this, additional sensor information is brought in from the vehicle. Furthermore, the estimator architecture is changed to one that handles nonlinear models for the system equations and sensor models.

4.3.1 State Vector and Process Model Equations

The discussion of the state vector and process model equations in this subsection begins with the enumeration of the sensors available to the system, the reference frame definitions required to specify the vehicle, target and sensor models and the vectors that define their configuration. This is followed by definitions of the estimator state vector, the equations of motion that govern those states, and the assumptions made here and the rationale behind them.

To achieve estimation of relative positions and the water-relative velocities of the target and vehicle, other vehicle sensors are available to provide the necessary additional information. ROVs such as *Ventana* and *Tiburón* typically are instrumented with an array of sensors, such as velocity sensors, sonar, current meters, compass, inclinometers and rate gyros.

To supplement the vision-based system described in Chapter 2 with the information available from these other vehicle sensors, all measurements and states must be brought into a consistent reference frame. The vision-based system operates in a target-centered

frame, i.e., all measurements and observable states are relative to the target frame, j . A heading measurement is used to align the target frame's orientation with the earth-fixed north-east-down (NED) frame. However, when tracking the target through the water, both the target and the tracking vehicle are moving independently through the water column, which is itself not fixed with respect to the earth¹. The water frame is assumed to align with the earth-fixed NED frame, but with a constant current velocity with respect to the earth.

Figure 2.3(b) includes information necessary to specify the configuration of the system. \mathbf{p} is the position of the target in the water frame, \mathbf{q} is the position of the vehicle in the water frame, $\boldsymbol{\lambda}$ is the orientation of the vehicle in the water frame, and \mathbf{r} is the relative position of the target with respect to the vehicle. As in the linear estimation implementation, the target is given a three-DOF representation (no orientation states are modeled), but the vehicle is represented by a six-DOF model. Thus, the complete configuration of the system is specified by $\langle \mathbf{p}, \mathbf{q}, \boldsymbol{\lambda} \rangle$. However, because the positions \mathbf{p} and \mathbf{q} are set with respect to an arbitrary origin and are not directly relevant to the tracking task, all the relevant information therein is captured by their difference, \mathbf{r} . Hence, the system is modeled with the following state vector and equations of motion:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_v^T & \dot{\mathbf{q}}_v^T & \dot{\mathbf{p}}_w^T \end{bmatrix}^T \quad (4.1)$$

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}^v, \mathbf{u}^w, \boldsymbol{\lambda}, \boldsymbol{\omega}, \mathbf{n}^f, \mathbf{n}^p) = \begin{cases} \frac{d}{dt} \mathbf{r}_v & = {}^v \mathbf{R}^w(\boldsymbol{\lambda}) \cdot \dot{\mathbf{p}}_w - \dot{\mathbf{q}}_v - \boldsymbol{\omega} \times \mathbf{r}_v \\ \frac{d}{dt} \dot{\mathbf{q}}_v & = g_v(\mathbf{u}_v + \mathbf{n}^f, \dot{\mathbf{q}}_v) - \boldsymbol{\omega} \times \dot{\mathbf{q}}_v \\ \frac{d}{dt} \dot{\mathbf{p}}_w & = \mathbf{n}^p \end{cases} \quad (4.2)$$

$$\mathbf{n}^p \sim \mathcal{N}(\mathbf{0}, \text{diag}(\sigma_p^2))$$

In (4.2), coordinate systems are indicated with subscripts, w and v , indicating the water frame and vehicle frame body coordinates respectively. The function $g_v(\cdot)$ represents the translational dynamics of the vehicle in the water, which can contain terms accounting for the relationships between the thrusters, the vehicle's dry and added masses and damping forces.

Note also that the orientation is assumed to have only small pitch and roll angles, therefore angular velocity may be approximated as equal to Euler rates. ${}^v \mathbf{R}^w(\boldsymbol{\lambda})$ is the rotation matrix, based on the vehicle's orientation, $\boldsymbol{\lambda}$, which rotates vectors from the

¹An assumption is made, and is carried through this dissertation, that the water current is steady and therefore the water frame may be treated as inertial.

water frame to the vehicle frame. $\boldsymbol{\omega}$ is the angular rate of the vehicle, expressed in body coordinates.

Note that both $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ are not treated as states of the estimator, but rather used directly as inputs to the process model². The choice to remove them from the state vector was made to reduce the order of the estimator, and is possible due to the reliability and quality of the sensors used on the vehicle to measure these quantities, as well as their availability at a high sample rate.

The design choices regarding the function for the vehicle dynamics, $g_v(\cdot)$, and for the target dynamics are discussed later in Section 4.3.3.

4.3.2 Sensor Models

Cameras

The camera set chosen includes two identically designed cameras, model Pegasus by Insite Pacific, Inc. (shown in Figure 4.2(b)). These have a horizontal field-of-view of 48 degrees, and a vertical field-of-view of 37 degrees. They are mounted with a short baseline of 10 cm, converging slightly at an angle of 10 degrees. The vision processing component of the system executes at a 10 Hz rate. For each camera (superscript i), the measurement model is given by:

$$\begin{aligned} \mathbf{s}^{cam,(i)} &= h^{cam,(i)}(\mathbf{r}_v) = d^{(i)} \left\{ n^* p \left(s^{(i)} \mathbf{T}^v \cdot \mathbf{r}_v \right) \right\} + \mathbf{n}^{cam,(i)}; \\ \mathbf{n}^{cam,(i)} &\sim \mathcal{N} \left(\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_{cam,(i)}^2) \right) \end{aligned} \quad (4.3)$$

Here, $p(\cdot)$ is the projection function for a pinhole camera model, and $d^{(i)}\{\cdot\}$ is a lens distortion model as calibrated for the i^{th} camera [7, 23]. The cameras were calibrated in air and the underwater refraction effect is handled by scaling the bearings by n^* , the sea water/air refraction index of 1.339 [39]. The matrix $s^{(i)} \mathbf{T}^v$ is the homogeneous transformation matrix [11] which transforms vectors expressed in vehicle coordinates to the coordinates of camera i . This matrix is based on sensed pan and tilt angles from the stereo camera mount and the measured location of that mount with respect to the vehicle's center.

²These quantities are sensed by an Octans gyrocompass and Attitude and Heading Reference System (AHRS) (a product of Photonetics [46]), which includes fiber optic gyros that measure angular rates and computes vehicle angles with respect to the earth. The gyrocompass is monitored at 10 Hz (the fastest sample rate used within this system).

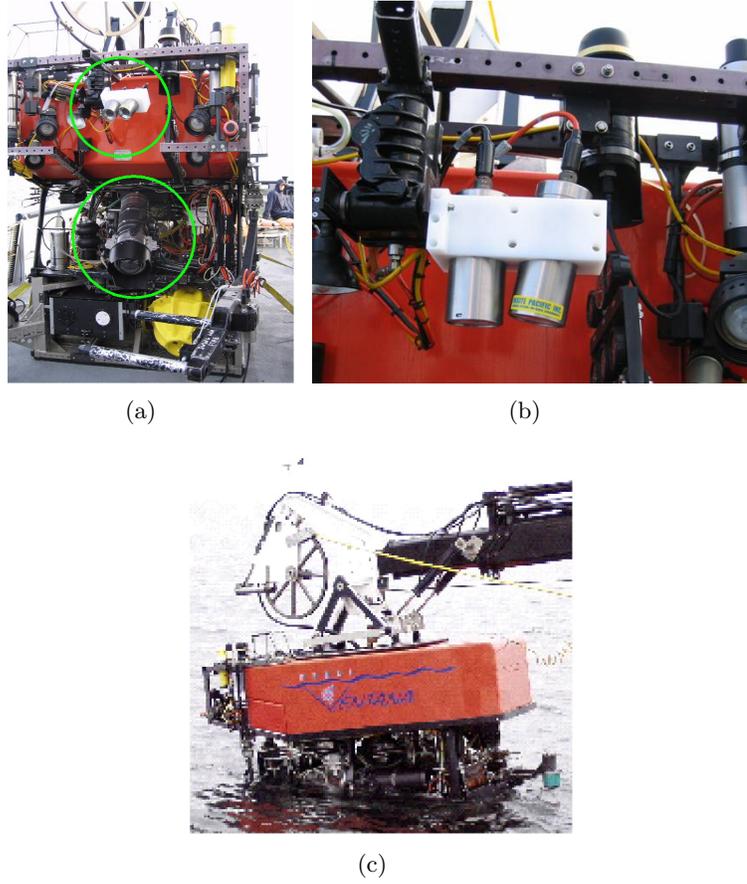


Figure 4.2: (a) The ROV *Ventana*, configured with cameras for midwater activities (circled, top center) – the high definition science camera is indicated in the lower circle, (b) close-up view of the tracking cameras, (c) *Ventana* being lowered into the ocean – the DVL instrument is at the rear of *Ventana* (lower right).

Doppler Velocity Log (DVL)

This instrument, an RD Instruments Workhorse DVL [53], is mounted at the rear of the ROV *Ventana* (as shown in Figure 4.2(c)). This instrument provides measurements of its velocity with respect to the water at a maximum rate of 5 Hz. The DVL is modeled here by (4.4), which includes the coupling of the vehicle's angular velocity into this instrument's readings. Note that the DVL is assumed to be aligned with the vehicle's axes, but located at position \mathbf{r}_v^{dvl} from the vehicle's center.

$$\begin{aligned} \mathbf{v}^{dvl} &= h^{dvl}(\mathbf{x}, \mathbf{n}^{dvl}) = \dot{\mathbf{q}}_v + \boldsymbol{\omega} \times \mathbf{r}_v^{dvl} + \mathbf{n}^{dvl}; \\ \mathbf{n}^{dvl} &\sim \mathcal{N}(\mathbf{0}, \text{diag}(\sigma_{dvl}^2)) \end{aligned} \quad (4.4)$$

4.3.3 Estimator Design

With the process and sensor models defined in the previous subsections, several design choices, both quantitative and qualitative, must still be made in designing this estimator. These include the type of estimation structure, the rate of operation of each component, and handling of the unavailability of some or all sensors for short periods of time.

Due to the highly nonlinear sensor models and the nonlinearities in the process model, a Sigma-Point Kalman Filter (SPKF), using the Unscented Kalman Filter (UKF) implementation [32, 69] was chosen here. The SPKF is an estimator that provides the ability to naturally handle nonlinearity in both the process and sensor models without linearizing those models and without incurring significant computational costs compared to an Extended Kalman Filter (EKF), which requires model linearization about the mean state estimate. Note that the same assumptions of Gaussian uncertainty as are made with the EKF are made with the SPKF. However, while the EKF linearizes the models and propagates the Gaussian uncertainties through the equations linearly, the SPKF approximates the state estimates and their Gaussian uncertainties numerically using a deterministically selected set of “sigma” points. These points are propagated through the nonlinear process and sensor models, and the resulting mean estimates and covariances are calculated using those outputs.

The architecture relating the process and sensor models for the robotic observation system and their interaction with the system’s sensors is illustrated in Figure 4.3. This diagram illustrates a number of features of this estimator. The handling of sensor validities and the multiple rates at which sensors are sampled is represented by the block labeled “Multi-Rate/Validity Processing”. This process dynamically assembles available sensor measurements into the vector, \mathbf{y} , and constructs a measurement covariance matrix, \mathbf{R}_n based on the subset of measurements available. Thus the estimator runs at the fastest rate at which sensor measurements are collected (10 Hz). Via this block, the measurement update dynamically adjusts its structure to prevent sensor readings that have been flagged as “not valid” from being applied to the estimation process. This happens often, particularly with the vision system, whose recognition component is designed to reject false tracking matches [54], and also occurs during out-of-frame events when the tracking specimen leaves the view of the cameras, or when the specimen is visible in only one camera. This same mechanism also handles the slower update rate of the DVL, which is nominally 5 Hz, by only adding \mathbf{v}^{dvl} and its associated covariance to \mathbf{y} and \mathbf{R}_n when new, valid data are available. The flexibility of the Kalman filter architecture that is carried

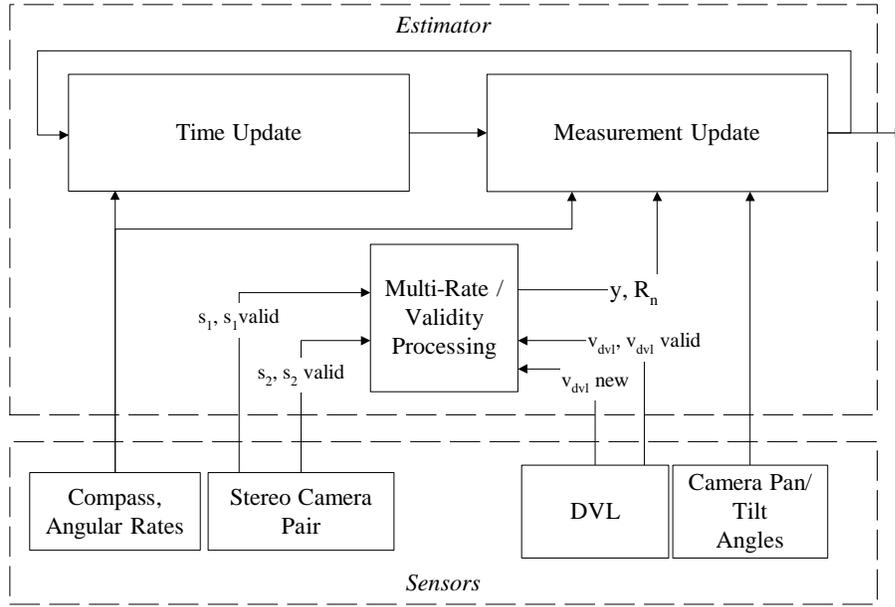


Figure 4.3: Architecture of UKF estimator for water-relative states.

through to the SPKF allows this adjustment to happen naturally without modification to the core estimation algorithms.

Figure 4.3 also illustrates the interaction between auxiliary sensors (those used directly by the process and sensor models of (4.2)-(4.4)) and the estimator's components. Specifically, the vehicle's angles and angular rates from the Octans gyrocompass are used by both the process and sensor models, while the sensed camera pan and tilt angles are used only by the sensor models to predict camera measurements, per (4.3).

The function $g_v(\cdot)$ in (4.2), which represents the translational dynamics of the vehicle in the water is defined here in (4.5) to be a purely kinematic model driven by white noise acceleration.

$$\begin{aligned} g_v(\mathbf{u}_v + \mathbf{n}^f, \dot{\mathbf{q}}_v) &= \mathbf{n}^f; \\ \mathbf{n}^f &\sim \mathcal{N}(\mathbf{0}, \text{diag}(\sigma_f^2)) \end{aligned} \quad (4.5)$$

This choice is made for several reasons, including the presence of large disturbances from tether forces and variable, non-zero thruster null settings, and often varying dynamics of the tracking vehicle due to thruster hydraulics variability and the specific science equipment installed on the vehicle on a given day. This large degree of uncertainty in the dynamic modeling motivates the choice of the simplistic but flexible white noise acceleration model of (4.5).

Less information is available to model in detail the dynamics of the target specimen, some of which can accelerate actively in the water column. Thus, the target model portion of (4.2), a white noise Gaussian acceleration model, was chosen with the magnitude of σ_p set to be large enough to allow the estimator to track accelerations by the target. The implications of this choice manifest as lags by the estimator when large accelerations or decelerations occur. This effect must be traded off with the desire to filter out the substantial noise in the sensed water-relative velocities from the DVL that make the target's velocity observable. This model will be augmented later in this dissertation to consider behavior modes by the target specimen and the different dynamic regimes associated with each mode.

4.4 Results from the Sigma Point Kalman Filter

Results from applying this estimator to both simulated and field data have been presented previously [50] and will also be covered later in this dissertation in Chapter 6. There, parameters such as noise standard deviations and *Ventana*-specific ROV parameters are specified. Results from this Sigma Point Kalman Filter estimator will be given and compared as a baseline to the more complex estimators of Chapter 6 that build upon this one.

Chapter 5

Visual Classification of Body Motion Modes

The estimation framework contributed by this dissertation enables the estimation of the mode of a hybrid system by fusing two different kinds of information: (1) “conventional” sensors which observe the system’s continuous states and (2) perceptual information about the system’s mode. Chapter 4 covered the first path for the underwater robotic observation system by fusing water-relative velocities from a Doppler Velocity Log with stereo vision. This chapter provides the second information path, the perceptual information related to mode. The fusion of these two sets of information will be covered later in Chapter 6.

This chapter documents specific methods for eliciting perceptual information about the current behavior mode of tracked marine animals. For this application, visually apparent information about the specimen’s behavior mode is derived by quantifying and classifying patterns of deformation of the body of the target. This classification is inspired by the perception of the human observer when observing the specimen. The output of the algorithms presented is a fast *visual* indication of the mode of the target, providing a higher bandwidth source of mode information than the velocity information derived in Chapter 4. Although this visual mode observation is not accurate enough to be used to trigger drastic control actions, it still proves to be extremely helpful and crucial when filtered together with other observations via the augmented hybrid estimation framework that will be presented in Chapter 6.

This chapter is organized as follows: first, the motion modes of animals tracked by the robotic observation system are established and characterized in terms of the body

deformation patterns and the resulting propulsion dynamics. Then, a mode model characterizing the body motions exhibited by these animals is introduced, including decision criteria that enable the use of this finite automaton to classify the current mode of the animal based on the body motion patterns detected. The remainder of the chapter focuses on recognition of mode changes by applying techniques in real-time computer vision and supervised machine learning in the form of a support vector machine (SVM) to evaluate the guard conditions (transition criteria) of the proposed finite automaton. Methods are presented to distinguish between active and resting modes, and to detect and measure rhythmic patterns in the body motions of these animals [47, 48].

5.1 Motion Behavior of Tracked Marine Animals

5.1.1 Characteristics of the Motion Behaviors Encountered

In this section, the motion behaviors exhibited by gelatinous animals are enumerated, and a body motion mode model derived from those behaviors is established¹.

Gelatinous animals generally effect propulsive forces by deforming some part of their bodies (or the entire body) in a pulsing motion. Pumping water in and out of the bell portion of their bodies generates a thrust force on the surrounding water. Many species exhibit periodic bell pulsing behaviors in order to propel themselves through the water. Others exhibit these pulsing actions in non-periodic patterns. Table 5.1 gives some example statistics for several species in terms of body pulsing frequencies (gait) and the resulting cruise and maximum speeds.

Figure 5.1 shows some image sequences with examples of these motions. For medusa jellyfish, an expression for dynamic thrust given by (5.1) shows thrust, T , related to water density, ρ , and generated by the time rate of change in the volume of the bell, V , and the velar area, A_v , as proposed by Daniel [13].

$$T = -\frac{\rho}{A_v} \frac{\delta V}{\delta t} abs \left(\frac{\delta V}{\delta t} \right) \quad (5.1)$$

Observable body motions often, but not always, lead to self-propulsion. Figure 5.1 shows three image sequences with examples of these motions. In (a), the *Praya* siphonophore

¹While the automatic tracking system is most commonly used to track gelatinous animals, other types of animal such as small squid and some (low swimming speed) fish species have been tracked. The visible cues related to their motions are also discernible by the vision algorithms of this chapter. However, (5.1) does not apply to those classes of animal.

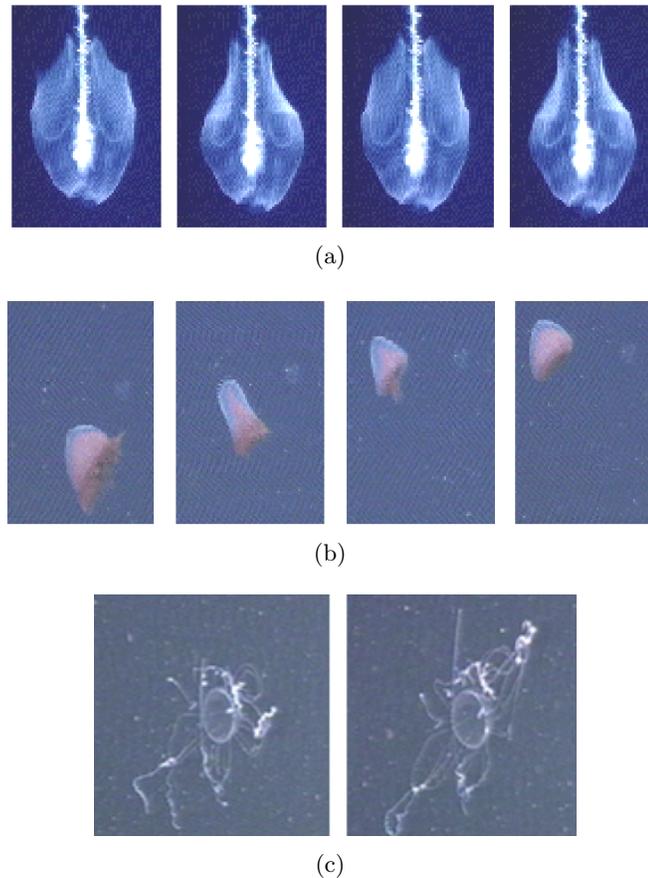


Figure 5.1: Examples of body motion behaviors: (a) the head of a *Praya* siphonophore exhibiting repetitive pulsing motions with period of about 0.9 sec. time-lapse at 0.43 sec intervals and stabilized, (b) a *Benthocodon* jellyfish making a single pulse swimming motion, time-lapse at 0.27 sec intervals, (c) a *Colobonema* jellyfish changing its configuration, images at 1.3 seconds apart. (Images courtesy of MBARI.)

is swimming by the repetitive pulsing motions of its “head”. In (b), a *Benthocodon* jellyfish makes a sudden swimming motion by pulsing its entire body once, then returns to resting. In (c), a *Colobonema* jellyfish changes the pose of its body, displaying a wide variation in its tentacle configuration over a short period of time. Unlike the first two examples, this body motion does not result in any self-propulsion.

It is useful to classify these observed motions into four types: resting, single pulsing, repetitive pulsing, and changing configuration. “Resting” is defined as the mode in which the body is not deforming significantly. This usually means the animal is passively floating with whatever currents are present or perhaps moving vertically due to a non-neutral buoyancy force. “Changing Config/Pose” means the body is deforming but not effecting

| Species | Cruise (cm/s) | Max (cm/s) | Gait (Hz) | Source |
|---|------------------|---------------|-----------|----------|
| <i>P. camtschatica</i> | 1-2 | 2 | 0.11-0.33 | [65] |
| <i>Stomolophus meleagris</i> | 4-12 | 16 | * | [37] |
| <i>Solmissus incisa</i> and <i>Solmissus marshalli</i> | 1.0-2.7 | * | 0.31-0.79 | [38, 43] |
| <i>Aegina digitale</i> | 2-5 | 30-40 | * | [10, 16] |
| <i>Aegina grimaldi</i> | 1.5-2.8 | * | 0.5-1.0 | [38] |

Table 5.1: Example statistics on motion behavior of several species of gelatinous animal. * indicates that data was not available in given references. (Table adapted from [54].)

any propulsion, as in Figure 5.1(c). “Single Pulsing” represents an isolated pulsing motion, differentiated from “Changing Config/Pose” by the fact that the body returns to its original configuration at the end of the motion. “Repetitive Pulsing” signifies a sustained version of the “Single Pulsing” mode, and for most species represents a sustained swimming motion.

5.1.2 A Mode Model for Real-Time Monitoring of Animal Motions

Figure 5.2(a) shows a finite automaton that expresses an observer’s perception of mode in real-time. This model differs slightly from the list enumerated in the preceding subsection because it is drawn from an observer’s forward-looking (causal) perspective. For instance, it is difficult to tell the difference between a single pulsing motion and a body configuration change until the motion is complete. Also, a periodic motion is not clearly periodic until multiple cycles are complete. Because of these ambiguities, Figure 5.2(a) shows one “Moving” mode in place of single pulsing and changing configuration. This body motion mode effectively means to the observer that the body is deforming, but it is unclear if there is a pattern to the deformations.

The model of Figure 5.2(a) can be used as the reference model when monitoring a gelatinous animal. To identify which mode the subject is in, an observer must answer three questions. The switching events in Figure 5.2(a) are numbered to match the question that must be answered for an observer to recognize the event.

1. ‘Is the body deforming significantly?’ If it is not, the animal may be considered to be resting. (Because this question is related to four switches in the mode logic diagram, those switches are labeled 1(a)-1(d).)

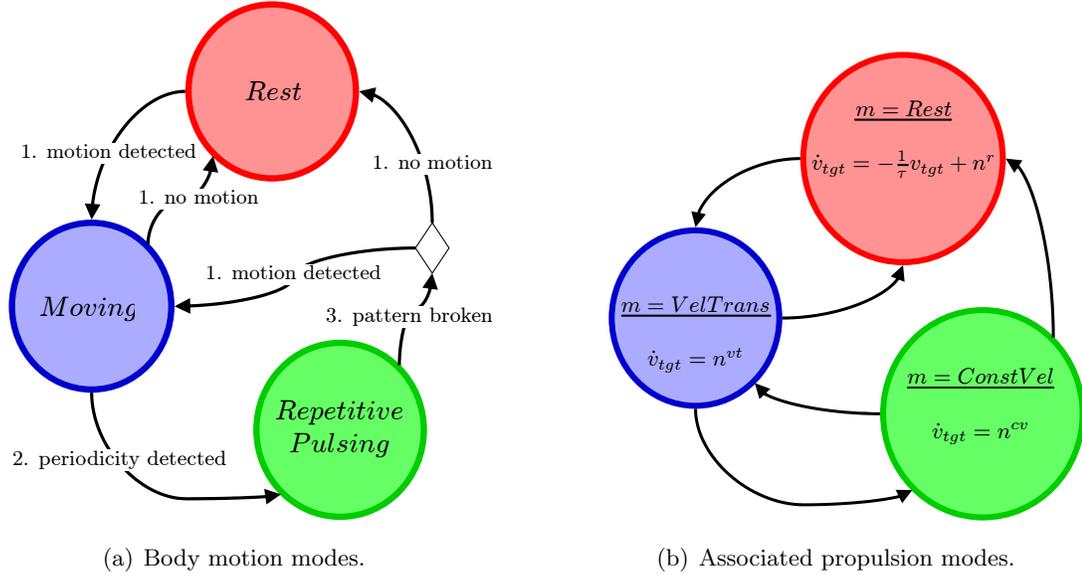


Figure 5.2: Finite automata representing motion behaviors of tracked specimens.

2. ‘If so, is the motion rhythmic?’ If it is, the animal is pulsing repetitively (which will usually be associated with swimming).
3. ‘If it is pulsing repetitively, and its pattern changes, what is it doing?’ This leads to the question, ‘Is it moving in a different way or is it switching to a resting mode?’

Automated decision algorithms that answer each of these questions are introduced in Section 5.3. Fig. 5.2(a) shows a finite automaton that expresses an observer’s perception of body motion mode, including these criteria used to detect switching events. Algorithms for applying this finite automaton to a stream of video images of an object in real time were presented in [47] and [48]. The output of this vision algorithm is a classification of the motion type exhibited by the body of the observed animal at time t :

$$\sigma(t) \in \{Resting, Moving, Repetitive Pulsing\} \quad (5.2)$$

5.2 Rationale for Algorithms

5.2.1 Algorithms Required for Mode Determination

For an automated sensing system to determine the body motion mode of a gelatinous animal, it must evaluate the logic specified in Figure 5.2(a). Three algorithms are required to perform that evaluation, one for each item in the enumerated list from the previous

section. An algorithm that detects the presence or absence of significant body deformations is required to evaluate the switching criteria labeled 1(a)-1(d). To identify repetitive pulsing (switching criterion 2), a means to detect periodicity in the motion is required. To recognize deviations from periodic motion (switching criterion 3), an algorithm is required that determines if the body is moving as expected based on the recent pattern.

The algorithms in this chapter use computer vision and supervised machine learning to achieve these objectives. They are based on analysis of self-similarity measures of the tracked animal's image over time. To measure the evolution of self-similarity, stabilized image sequences that isolate the animal's body deformations from its overall rigid-body motions are required. The self-similarity data are then analyzed for indications of motion in general and for indications of periodicity.

5.2.2 Properties of Body Deformations in Sequences of Images

A human observer using primarily visual information can accurately determine which of the modes of Figure 5.2(a) the animal is in with little difficulty or delay. For many of the animals of interest, the motions look distinct enough that they can be identified with numerical vision-processing techniques.

Figure 5.1(a) shows some sequential images of a *Praya*, a type of siphonophore. This shows the repetitive nature of typical pulsing motions, with the resulting images looking nearly identical every 0.9 seconds. In this sequence, the siphonophore's head has been tracked and stabilized, meaning that rigid-body translation, scaling and rotation of the subject in the image plane has been removed.

This example illustrates some basic properties of a stabilized sequence of images of a gelatinous animal. If, as in this example, relative three-dimensional rotations are small or slow, some general statements can be made. If the body is not significantly deforming over time, a short sequence will consist of images that look very similar. If it is deforming, each image in the sequence will look significantly dissimilar to recently preceding images. If there is a repetitive pulsing motion behind the deformations of the body with a period of p samples, then every pair of stabilized images in the sequence that are p frames apart should look very similar.

These properties of image sequences of a gelatinous animal provide a foundation for the computer vision algorithms used for mode identification. Quantitative methods that test those properties can be used for recognition of body motion and the detection of periodicity in that motion.

5.2.3 Self-Similarity Metrics for Quantification of Body Motion

The properties of image sequences of a gelatinous animal are all based on the similarity between the images in the sequence. Analysis of the motion of an animal based on those properties requires a quantitative way to measure its self-similarity over time.

Area-based correlation metrics are chosen to provide this measurement of self-similarity. By comparing the entire area of an object's projection, many kinds of body deformations can be captured. Those include deformations that involve changes in the projected contour of the body, as well as deformations that are internal to its boundaries. Such changes might be thought of as variations in the object's visual "texture", and are not captured by tracking changes in object edges or contours. Also, good signal-to-noise ratio properties are associated with using the entire area of the object to measure changes in images [40]. Signal to noise ratio is a significant issue for this problem, because of the noise levels in typical underwater video and the small pixel area that the tracked animals often occupy in the image frame.

Table 5.2 provides a qualitative comparison of the area-based correlation approach with other general types of imaging approaches with respect to these considerations. The table summarizes the capabilities and limitations that could be expected for some commonly used vision approaches when tracking changing subjects. The entries in the table are filled out with a check mark (\checkmark) when the class of algorithm for the column can be expected to perform well for the requirement enumerated for the row. A blank signifies that the approach cannot meet that requirement. Finally, a question mark indicates that the performance of the algorithm class with respect to the requirement is unknown or will vary based on more specific application details.

Cutler and Davis [12] used analysis of self-similarity of a tracked object over time to detect periodic motion in image sequences of people walking, dogs running, etc. Matching of patterns in the similarity data allowed computation of the period of the body changes and identification of the type of motion (e.g., walking vs. running, biped vs. quadruped). Periodicity was detected by placing similarity data into a "similarity matrix" and examining the "texture" of that matrix for regularity.

The computer vision processing methods used in this paper are similar to those of [12], but with extensions to extract additional information from the self-similarity metrics. Recognition of periodic motion is important for this application, but identification of the modes of gelatinous animals also requires detection of non-periodic body motion and rapid recognition of deviations from repetitive motions. These are all accomplished through analysis of the animal's self-similarity over time.

| | Active Contours | Pixel Area / Moments of Inertia | Feature Tracking w/ 3D Model | Area-based correlation |
|--|--------------------|---------------------------------------|------------------------------------|---------------------------|
| Detect body shape changes | ✓ | ✓ | ✓ | ✓ |
| Detect body deformations internal to boundary (visual texture) | | | ✓ | ✓ |
| Robust to object of small pixel area | | ? | | ✓ |
| High S/N ratio | | ? | ? | ✓ |
| Low computational complexity | ? | ✓ | ? | ✓ |
| Robust to changes in 3D viewpoint | ? | ? | ✓ | ? |

Table 5.2: Comparison of potential vision approaches.

5.3 Algorithms and Example Results

This section describes in detail the algorithms used to identify the motion mode of an observed gelatinous animal with respect to the finite automaton in Figure 5.2(a). The methods are demonstrated by application to several example video clips.

Section 5.3.1 describes the object tracking and stabilization methods used to generate sequences of images of the tracked animal. Section 5.3.2 details the computation of self-similarity measures for pairs of stabilized images. Section 5.3.3 provides results from applying a support vector machine to the binary classifier problem of deciding whether the animal being tracked is moving its body or not. Section 5.3.4 discusses detection of periodicity in the tracked animal’s motion and calculation of period. Section 5.3.5 then discusses detection of deviations from the periodic pattern when periodicity is present. Figure 5.3 is a block diagram that shows how these algorithms combine to determine the motion mode of the tracked animal.

5.3.1 Object Tracking and Stabilization

Measurement of self-similarity requires that the subject’s relative scaling and translational and rotational motions be corrected. The result is a sequence where the subject’s apparent size and orientation does not change and its centroid does not move. Refer again to Figure 5.1(a) for an example of a stabilized sequence.

By segmenting the object from each image and tracking its centroid’s position over time, translations in the image plane are corrected. Rescaling based on range measurements normalizes the object’s size. Range calculated by stereo triangulation is preferred to

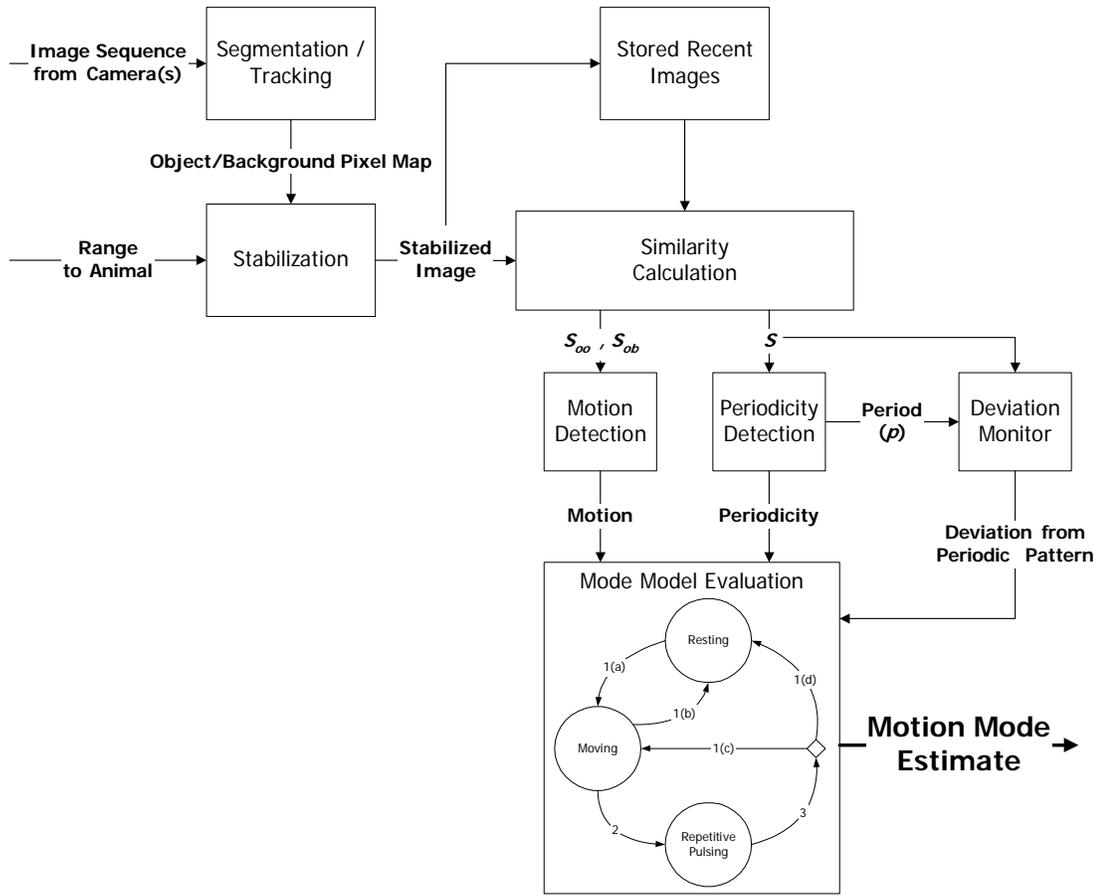


Figure 5.3: Block diagram tracing steps taking an image stream of a tracked animal to determination of the body motion mode of the animal.

pixel area-based methods for this application because body deformations often cause false fluctuations in range measurements computed from object pixel area. After translation and scaling are corrected, rotations are corrected by testing rotational correlations over a small set of angles. For details of the segmentation and tracking methods used in the Stanford University-MBARI jellyfish-tracking project, refer to [54, 56, 58].

Many characteristics of the motion of underwater animals, the motion of the camera, and the underwater environment conspire to make object stabilization from this video an imperfect process. This stabilization process takes a two-dimensional view of the subject. Therefore, the most significant errors are due to the changes in viewpoint caused by the motions of the camera and the potentially unconstrained rotations of the subject in three dimensions which violate that assumption. The vision algorithms here mitigate this issue primarily by the limitation to a small window backward in time within which image

comparisons are deemed valid. This window can be as small as 1 second for animals whose motions are not periodic, or about 2 periods for those whose motions of interest are repetitive.

5.3.2 Computation of Self-Similarity

To measure the similarity between two images of the tracked object, a normalized sum-of-squared-differences (SSD) is computed (over a common region denoted R_{ab}):

$$S_{ab}(t_1, t_2) = \frac{\sum_{x,y \in R_{ab}} (O_{t_1}(x, y) - O_{t_2}(x, y))^2}{mn(255)^2} \quad (5.3)$$

Here $O_t(x, y)$ represents the grayscale value at pixel coordinate (x, y) of the stabilized image O at time t and m and n are the dimensions in pixels of the images. For simplicity of notation, the region specifier ab is often dropped, thus this matrix will typically be referred to as S . Note that by using a measure of image differences, $S(t_1, t_2)$ is really a measure of dissimilarity. Comparing two identical images gives a score of zero and comparing a purely black image with a purely white image gives a score of one. $S(t_1, t_2)$ is normalized to be a per-pixel average difference, in units of the maximum possible grayscale difference. It is useful to store similarity data in a matrix with coordinates formed by pairs of time steps.

The intent of measuring self-similarity is to provide a quantitative measure of how much an animal's image projection has changed between two snapshots. By computing $S(t_1, t_2)$ as in (5.3), differences due to the background also contribute to this measurement, effectively as noise. The difference contribution for those pixels is typically quite small, therefore the number of pixels that make up the background can have a strong effect on the magnitude of $S(t_1, t_2)$ through the per pixel normalization term. This distorts the magnitude of S entries. For algorithms that process S using the same parameters for arbitrary animals that occupy different percentages of the stabilized image, this variation in magnitude becomes problematic.

To measure changes in the animal's appearance in a more uniform way, segmentation results are also utilized in this process. Comparison measures are calculated for each of three pixel classification regions. The pixel area is divided into three regions; pixels that are segmented in both images (the object), pixels that are not segmented in both images (the background), and the remaining pixels that are classified differently by the segmenter in each image. This is illustrated in Figure 5.4 for a pair of images. Self-similarity is

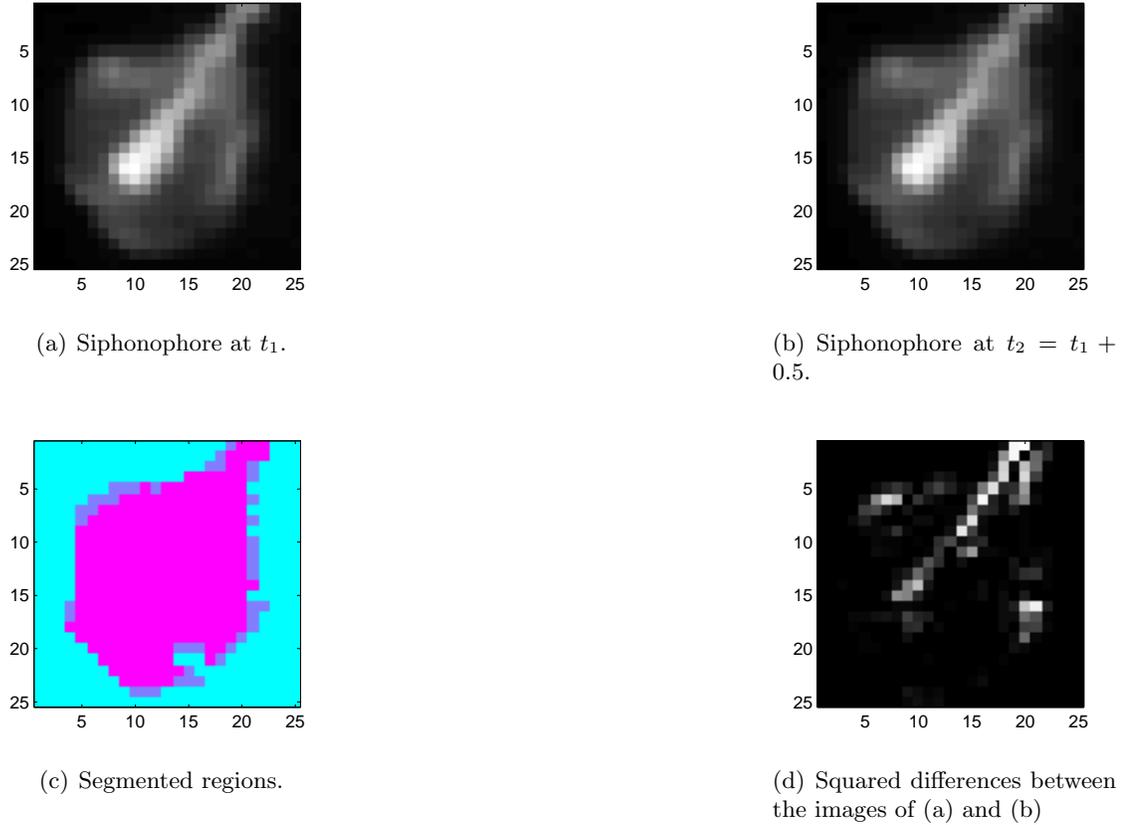


Figure 5.4: Example of image regions when calculating similarity over regions defined by segmentation process. (a) and (b) are stabilized images 0.5 seconds apart. (c) shows the regions defined by comparing the segmented results from the two images. Color key: Object-object: magenta, background-background: cyan, object-background: blue. (d) squared differences between the images in (a) and (b) with large differences lightly colored and small differences very dark.

computed using (5.3) separately for the pixels in these regions and will be referred to as S_{oo} , S_{bb} , and S_{ob} , where subscript o refers to the object region and b refers to the background region. The numbers of pixels in each region are defined as n_{oo} , n_{bb} , and n_{ob} , respectively.

Because confidence in longer-term stabilization results is low, a limit is placed on the range of time backwards for which similarity is computed. If w is the maximum similarity horizon, then the similarity matrices S , S_{oo} , S_{bb} , and S_{ob} are sparse, with entries only within w of the main diagonal. For underwater clips, w is typically chosen to limit

comparisons to the last five to ten seconds worth of images. The pattern recognition algorithms that process these matrices include options to further limit the window backwards in time for self-similarity data which is deemed valid or relevant.

The computation of self-similarity here is close to the method used in [12], with several extensions made. The normalization of similarity scores to a per pixel scale, the additional statistics based on segmented image regions, and the short time horizon were added to address the additional analysis objectives and more challenging image stabilization issues.

Typical gelatinous animal motions register very clearly in a topographical view of the resulting S matrix. This view of the motion is efficient to compute while still providing a clear insight into the animal's body activity level. Figure 5.5(a) shows an example similarity matrix from underwater footage of a *Ptychogena* jellyfish, while Figure 5.5(b) is a matrix computed on footage of a *Benthocodon* jellyfish. The forward progression of time can be read from these plots by moving diagonally from top-left (0 s) to lower-right (40 seconds for Figure 5.5(a), 45 seconds for Figure 5.5(b)). Large differences between the current image at t_1 and a past image at t_2 are indicated by dark colors, while nearly identical images have a low difference measure and indicated by very lightly colored points in the matrix. During the *Ptychogena* clip, the animal is resting initially, and then begins swimming by repetitive pulsing of its bell, beginning at about 6 seconds into the clip and ceasing at about 13 seconds, then resting until the repetitive pulsing motion resumes at 31 seconds into the clip. The resting periods (0-6 seconds and 13-31 seconds) are indicated in Figure 5.5(a) by the largely flat and light colored nature of those areas. The swimming periods (6-13 seconds and 31-40 seconds) show very regular, lattice-like textures of light and dark.

The matrix for the *Benthocodon* can be read in similar fashion. In this example, the specimen never exhibits periodic motion, only single pulses, which show up in Figure 5.5(b) as narrow dark lines at approximately 8, 17, 24, 29, 35 and 42 seconds into the footage. For all other portions, the jellyfish is at rest, shown by the light, flat and textureless areas that make up the remainder of the matrix.

The remainder of this section documents pattern recognition algorithms which process this self-similarity data to permit the evaluation of the mode model of Figure 5.2(a), thus providing automated methods for recognizing patterns in the matrices that were just described. The first classification to be made is whether or not the specimen is actively moving its body, i.e., telling the difference between flat, light areas of the matrix (no motion) and darker local areas (motion). The second is to detect and measure periodicity in the motion when present, i.e., recognizing the textured areas of the matrices. Finally,

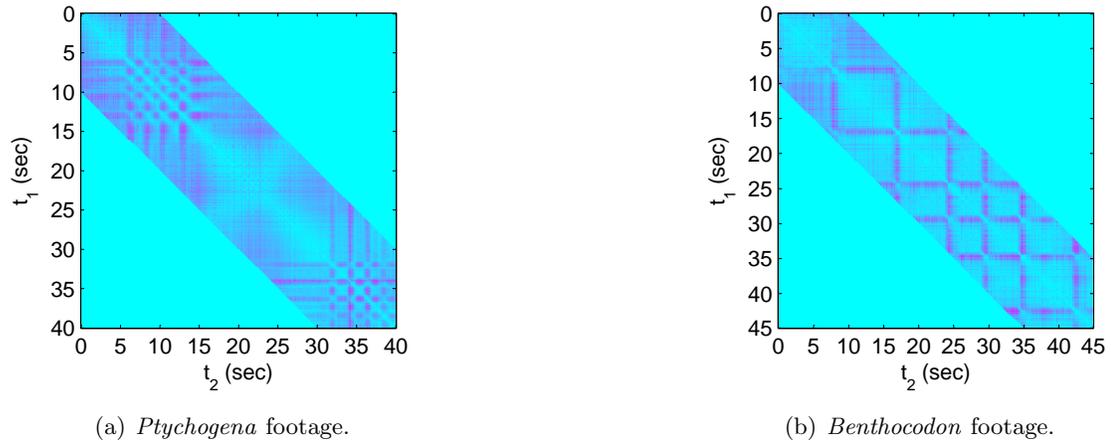


Figure 5.5: Similarity matrices for (a) *Ptychogena* footage and (b) *Benthocodon* footage. In (a), this animal began swimming by repetitively pulsing its bell, paused for several seconds then began swimming again. The regions highlighted show a clear texture in the similarity matrix during repetitive pulsing motions. In (b) the *Benthocodon* jellyfish made several single pulse motions.

when periodic motion is present, changes in the motion pattern are detected by exploiting the periodicity information as a prediction of the body's image.

5.3.3 Presence or Absence of Body Motion using a SVM

Use of the mode model of Figure 5.2(a) requires a reliable method for detecting the presence or absence of body motion. The degree and style of motion varies significantly for each animal. To be useful to an underwater vehicle that can encounter any type of animal, an algorithm that is effective with a constant structure and parameters is needed.

The general goal in detecting motion is to approximate a rate of change of the body of the subject and determine if it is sufficiently high to classify as in motion. In terms of the similarity matrix, such as those in Figure 5.5, looking at the point on the diagonal for the current time, the question to answer is whether the nearby matrix entries going upwards or left from there are sufficiently dark. A straightforward method was tested [47] and yielded accurate results with a carefully chosen threshold, but did not facilitate the use of a fixed threshold when testing with different video clips, making it impractical for use on an autonomous vehicle. It was based on thresholding a moving average of w similarity scores between the current image and w most recent images. The algorithm was based on the premise that if the animal were not deforming significantly, a sequence of stabilized

images of it would all appear to be very similar. This is quantified by (5.4),

$$\bar{s}_w = \frac{1}{w} \sum_{k=0}^{w-1} S(t - kT, t) > T_m \rightarrow MOTION \quad (5.4)$$

where T is the image sample time period, w is the length in samples of the window backwards to consider self-similarity scores, and T_m is a decision threshold.

To achieve better results which do not require parameter adjustments for each tracked specimen, a supervised machine learning approach is chosen for this classification task. This entails training a classifier using several labeled example clips of a variety of animals in motion or resting. A support vector machine was utilized for this classifier.

Binary Classification and Support Vector Machines

Supervised learning for a binary classifier problem refers to algorithms that, given a set of k feature vectors and their corresponding labeled values, find a decision function that is optimal by some definition such as how many test samples are classified correctly or the maximum margin between samples and the decision boundary. In this case, the labels represent “moving” or “not moving”, and are supplied by manually labeling several test image sequences.

Support vector machine (SVM) classifiers (also known as maximum margin classifiers) have been established as a mature and effective learning method in recent years [6, 70]. Their structure is very general and numerous implementations are readily available, such as the *LIBSVM* library utilized here [9]. An advantage of the SVM lies in its ability to efficiently learn in a very high (potentially infinite) dimensional space with feature vectors that can be expressed in a much smaller dimension. A separating hyperplane is found in the high dimensional space that results in a maximum margin classifier. Explicit provision is made to allow some samples to be misclassified with a penalty, thus mitigating the tendency to over fit training data. The resulting decision function is efficiently computed in the same lower dimensional feature space of the training vectors.

The decision function of an SVM takes the form of:

$$h(\mathbf{x}) = \sum_{i=1}^m \alpha_i y^{(i)} K(\mathbf{x}^{(i)}, \mathbf{x}) + \mathbf{b} \quad (5.5)$$

where m is the number of support vectors, K is a kernel function, α_i is a weight computed for the i^{th} support vector, $y^{(i)}$ is the label of the i^{th} support vector and \mathbf{b} is a computed constant offset for the hyperplane. Vectors \mathbf{x} and $\mathbf{x}^{(i)}$ are feature vectors; \mathbf{x} is the sample

being classified by $h(\mathbf{x})$, while $\mathbf{x}^{(i)}$ is the i^{th} support vector. The set of support vectors is the set of training samples closest to the separating hyperplane, and is typically much smaller than the training set. A decision is made by $h(\mathbf{x})$ classifying \mathbf{x} as having a predicted label of +1 if $h(\mathbf{x}) > 0$, or a predicted label of -1 otherwise.

The results are computed from the following optimization:

$$\begin{aligned} \max_{\alpha} W(\alpha) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ \text{s.t. } 0 &\leq \alpha_i \leq C, \quad i = 1, \dots, m \\ &\sum_{i=1}^m \alpha_i y^{(i)} = 0 \end{aligned} \quad (5.6)$$

Here C is a constant that allows regularization; i.e. by using this constraint, samples are allowed to be misclassified with a penalty. Without regularization, a hyperplane that perfectly separates the labeled data is found. While this optimally separates the training data, if the data is not expected to be completely separable, the result is an over fit to the training samples that does not work well when classifying other test data.

Application to Motion Detection

A support vector machine was trained and tested using a set of labeled sequences from clips of different animals in motion, with promising results. Using an SVM requires choices to be made about what feature vector to process, which kernel function to use to map the feature vector to a high dimensional space, how to perform cross-validation and how to identify best choices for parameters.

The feature vector chosen for classification are per-pixel similarity values separated for the two pixel regions of interest. Instead of using S values, which difference entire stabilized images, segmentation results are incorporated, S_{oo} , S_{ob} , and S_{bb} . The feature vector is constructed using a weighted per-pixel average of differences over two pixel regions is computed for every pair of images compared, S' from (5.7):

$$S'(t_1, t_2) = \frac{\begin{bmatrix} n_{oo}(t_1, t_2) S_{oo}(t_1, t_2) \\ n_{ob}(t_1, t_2) S_{ob}(t_1, t_2) \end{bmatrix}}{n_{oo}(t_1, t_2) + n_{ob}(t_1, t_2)} \quad (5.7)$$

Note that the third region, S_{bb} , is disregarded; this is the region where the segmentation algorithm classified pixels as background in both images. This region contributes only

noise to the process. The j^{th} feature vector, $\mathbf{x}^{(j)}$, consists of this vector S' computed between each image and w most recent images backwards in time.

$$x^{(j)} = \begin{bmatrix} S'(j, j - T) \\ S'(j, j - 2T) \\ \vdots \\ S'(j, j - wT) \end{bmatrix} \quad (5.8)$$

A Gaussian kernel function is a common choice for design of an SVM [9], and was used here. This kernel function is of the form:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2); \quad \gamma > 0 \quad (5.9)$$

This kernel maps the feature vectors to an infinite dimensional space where the optimal separating hyperplane is calculated.

Cross-validation was performed by randomly choosing half of the training data and leaving that set out of the training process. Then optimal values of C and γ are found by a designing an SVM using a grid of (C, γ) pairs and choosing the pair of values with the best cross-validation accuracy on the samples that were set aside for testing.

False mode switch indications are limited by requiring consecutive indicators of a mode switch before validating the indication. Because the animals in question do not change modes at a high frequency, it makes sense to only trust that a mode change has occurred if it is identified for m (typically 2) consecutive time steps. For mode transition detection, this introduces a small amount of lag, but is worthwhile as a tradeoff in terms of rejecting false (extremely brief) mode change indicators. If applied for collecting statistics about the animal's motion behavior, no cost is incurred by applying this extra criterion, since the first m mode switch indications can be relabeled after the fact.

Table 5.3 summarizes some results from applying this to a set of four underwater clips of motion of four different animals. Figure 5.6 plots results for the first two clips in the table.

5.3.4 Detection of Cyclic Motion and Calculation of Period

To decide that the motion of a gelatinous animal is a repetitive pulsing motion (switching criterion 2 of Figure 5.2(a)), periodicity must be detected in the body motion. If periodicity is detected, the period of motion can be calculated. In terms of interpreting the similarity matrix, such as those of Figure 5.5, this is the process of detecting regular textures in a

| Animal | Clip Length (sec) | Behavior during clip | Success Rate (%) | Success Rate (%) with 2 consec indicators reqt |
|---------------------|-------------------|---------------------------------|------------------|--|
| <i>Benthocodon</i> | 45 | Intermittent single pulses | 91.1 | 91.1 |
| <i>Ptychogena</i> | 50 | Intermittent repetitive pulsing | 86.3 | 91.4 |
| <i>Siphonophore</i> | 30 | Constant repetitive pulsing | 93.1 | 96.1 |
| <i>Solmissus</i> | 20 | Intermittent repetitive pulsing | 79.0 | 76.5 |

Table 5.3: Summary of results from applying SVM to several clips. Used Gaussian kernel with $C = 8$, $\gamma = 1$ and window w of 10 steps at 10 Hz sample rate. Test/training data separation done by splitting data set in two in random sequence.

portion of the matrix around the current time down the main diagonal up, and measuring the the spacing of the texture pattern.

This algorithm is subject to a number of objectives. One is to detect periodicity quickly, therefore from a limited amount of data. That goal is also driven by the constraint of the short window in time for which image stabilization results are trusted to be accurate. Another objective is accuracy and smoothness of the calculated period. While that is not necessary for detecting entry into a repetitive pulsing mode, it becomes important when detecting the exit from this mode.

Similarity data may be analyzed from either one-dimensional or two-dimensional views. A one-dimensional approach examines the self-similarity of the current object image with its past images. A two-dimensional analysis of the data in the matrix S looks for patterns in the comparisons between many images and past images.

The one-dimensional view uses the most recent column of similarity matrix S , which contains the similarity measurements between the current image and recent past images (a column in S starting from the diagonal and moving up for a number of steps back in the time history). The motions of interest are typically at a single frequency. Therefore, the simplest approach might be to simply find the peaks and measure their separation. Fourier analysis can also be used by computing a one-dimensional power spectrum of that signal, and examining the results for a dominant frequency.

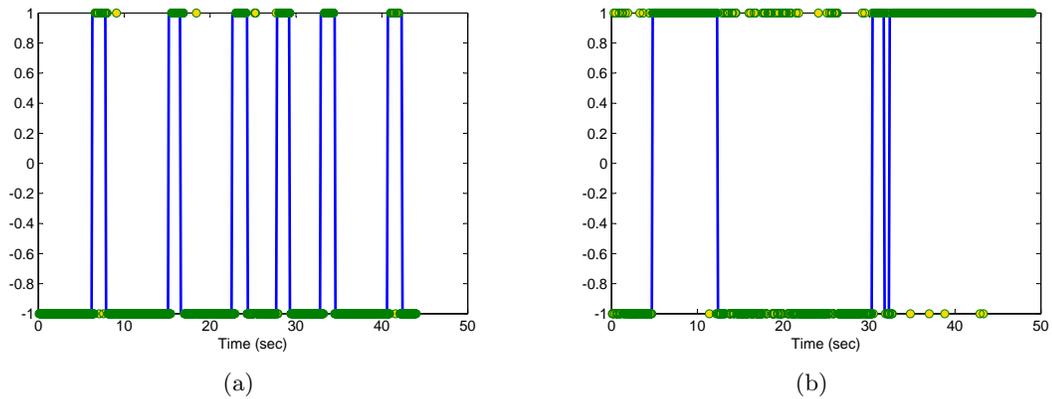


Figure 5.6: Body motion detection results for (a) *Benthocodon* jellyfish footage and (b) *Ptychogena* jellyfish footage. A value of 1 indicates motion is present, a value of -1 indicates that no motion is present. Solid line is the labeled value by human, circles are the result of the SVM decision function. A consecutive mode change indicator of 2 consecutive steps is enforced.

A more robust method of detecting periodicity is achieved using a two-dimensional view of the similarity matrix. The two-dimensional approach is based on the idea of using the texture apparent in the 2D view of the similarity matrix. Rather than deciding if the current image has periodicity in similarity with past images, this method asks if all (or most) of the past n frames show strong similarity to their corresponding frames p frames ago. If so, the object has a body motion period of p time steps.

To answer this question, the 2D autocorrelation of a local portion of the similarity matrix, A , is computed and searched for local peaks. If the motion is periodic with period p , the peaks should form a lattice structure with points every p steps. Candidate lattice structures of spacing d are compared with the layout of the local peaks of A . Local peaks of A and the lattice points are matched one-to-one by closest proximity and a check is made that an acceptably high percentage of available peaks and lattice points are matched. If any lattice structure matches, periodicity is considered present, and the calculated period p is equal to the value of d that resulted in the minimum total matching distance error.

These periodicity detection methods are based on those used in [12]. There, lattice structures that span the entire area of the autocorrelation matrix A were used. However, tests on underwater video found that the peaks along the center axes were consistently spaced as expected, but often the off-axis peaks were less regularly spaced. Because of this, the matching requirements were relaxed somewhat to use a modified lattice structure that forms a “cross” shape along the axes of A . Figure 5.7 shows an example result from

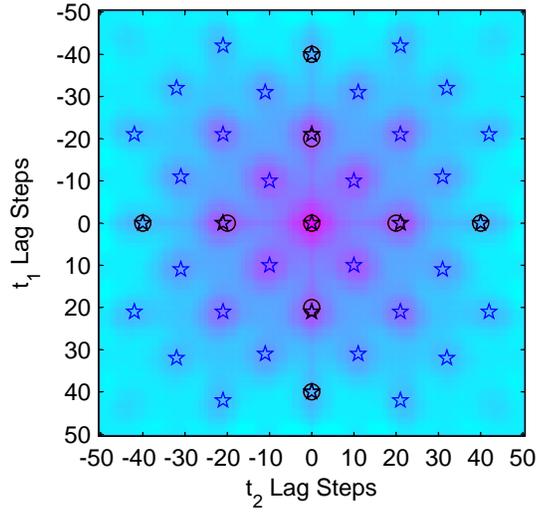


Figure 5.7: Example of lattice-matching to local peaks of autocorrelation of S for *Ptychogena* footage. “Cross” pattern matched to peaks at period of 2.0 sec. Stars represent local peaks in autocorrelation of the local portion of S , circles represent matched lattice points.

the siphonophore footage. This figure shows a match to a “cross”-shaped lattice with spacing of 9 steps, indicating periodicity is present with a 0.9 second period.

Previous studies [12, 47] showed better results using the 2D approach to finding significant periodicity in similarity data. The calculated periods were shown to be smoother, more consistent and more accurate.

5.3.5 Deviation from Expected Body Pose

To decide that the repetitive pulsing motion of a gelatinous animal has ended (switching criterion 3 of Figure 5.2(a)), a deviation from the motion pattern must be detected. If the pattern is broken, either transition 1(c) or 1(d) is then taken based on the presence of motion detection algorithm described earlier.

The algorithm uses a recent image as a prediction of what the animal should look like if the repetitive pulsing motion holds. When the animal is pulsing periodically, the sequence can be thought of as containing reference images that strongly resemble the current one. The computed period, p , is used to choose a reference image from the periodic motion image sequence. The reference image at time $t - p$ represents what the object is expected

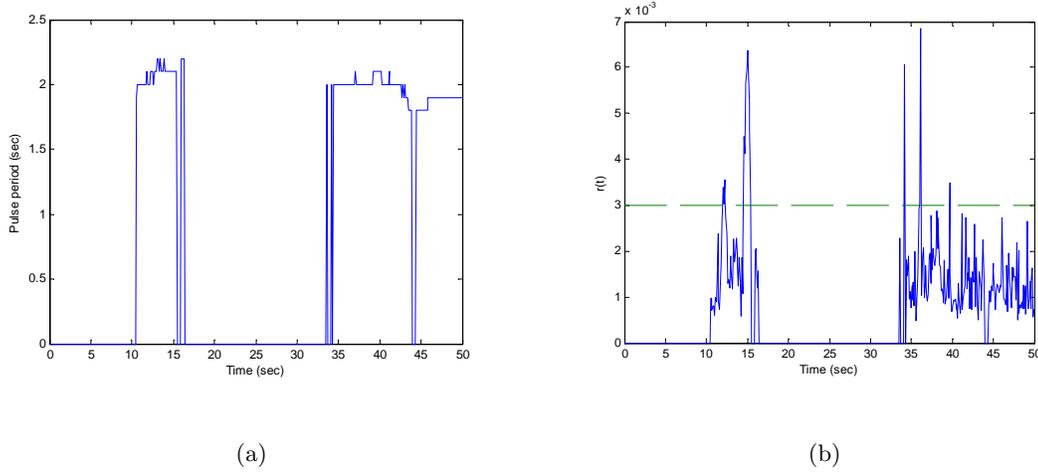


Figure 5.8: (a) Period and (b) deviation from periodic pattern results for *Ptychogena* jellyfish footage. In (a), a period of zero indicates that no periodicity is detected at that time. In (b) the dotted line shows the threshold value T_r ; when $r(t)$ is above T_r , a pattern deviation is detected.

to look like if the pulsing pattern is still in effect. An implementation is given by:

$$r(t) = S(t - p, t) > T_r \rightarrow \text{DEVIATION} \quad (5.10)$$

where T_r is a threshold value. Note that for good performance, short-term variations in period must be tracked well by the periodicity analysis algorithm. That requirement was addressed by the design objectives defined in the previous section for that algorithm.

Although the mode switch event from “Repetitive Pulsing” can also be determined by the loss of indication that periodicity is present, this predictor of animal pose reduces the lag in identifying the change. This method can typically detect the change in pattern within only 2 or 3 time steps, substantially faster than it takes for the periodicity indicator to no longer register periodicity (about half to one period). In the case of a switch from “Repetitive Pulsing” to “Resting”, the body motion detection algorithm can also identify the event by detecting a lack of motion. Again however, use of this predictor can reduce the lag involved.

Results of the period calculation along with the deviation from period pattern results for the *Ptychogena* footage are plotted in Figure 5.8. The results of combining the motion, periodicity and pattern deviation indications are plotted in Figure 5.9. Here the overall mode indication from evaluating the mode model using the indications of each of the algorithms is illustrated.

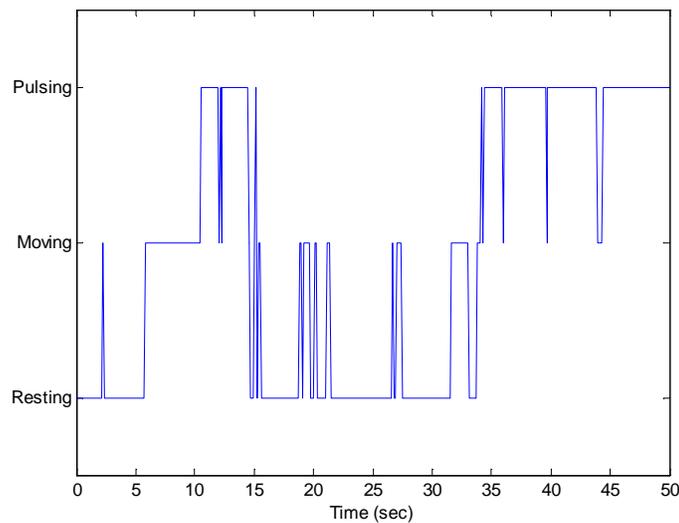


Figure 5.9: The overall mode indication based on all indicators for *Ptychogena* jellyfish footage.

5.3.6 Real-Time Considerations

The augmenting vision algorithms of this chapter have not yet been implemented in real-time code for this application, although that task is the first item of future work enumerated in Chapter 7. Instead, the results in this chapter were computed using Matlab to post-process video and data collected in the ocean. However, real-time compatibility is not expected to be an issue, based on the comparability of the algorithms here to the real-time demonstrated algorithms by Cutler and Davis [12] from which these have been extended. The highest computational loads are incurred in computing the similarity matrices and the autocorrelation of the similarity matrix used for periodicity detection. Those components of the vision algorithms are nearly identical to the Cutler and Davis versions – it is the extra pattern recognition steps (the SVM and pattern deviation monitor of sections 5.3.3 and 5.3.5 respectively) and the incorporation of segmentation data that are the main extensions added to enable classification of the motion of tracked marine animals. Those additional steps are all cheap to compute (the SVM is designed off-line). Since the core algorithms were implemented in real-time in the year 2000 on a Pentium III 550 MHz processor [12], no issues with real-time implementation on today’s more powerful processors are expected.

5.4 Summary

The algorithms presented in this chapter provide a source of perceptual information about the mode of tracked marine animals, with modes typically recognized visually within about 2-3 frames (or 0.2-0.3 seconds at 10 Hz). This information provides fast indications of mode changes, and is modeled loosely after the perceptual approach of human observers when watching these creatures. However, no information about the actual velocity of the target in response to mode changes is derived here. Further, this perceptual “measurement” related to mode is uncertain enough that it should not be used to trigger drastic control actions. Hence, the remainder of this dissertation presents methods to use this perceptual information and the information from more traditional sensors together in a complementary way to derive estimates of both mode and velocity with higher quality than could be done with either source alone. The final estimation algorithms result in improved mode and velocity estimates, with mechanisms provided to the design engineer to tune the results to tradeoff higher bandwidth versus low incidence of false switches in the overall mode estimate.

Chapter 6

Estimation of Target Mode and Velocity Using Perceptual Information

The preceding chapters presented the derivation of visual (body motion) mode observations for tracked deep ocean animals, and relatively low bandwidth, Kalman filter-based estimation of water-relative velocities of the animals using a combination of stereo vision measurements and acoustic measurements of the tracking vehicle's velocity. Hybrid estimation algorithms that have been used to track maneuvering targets' motion and maneuvering modes were summarized in Chapter 3 and shown to be unable to detect all mode transitions in this application.

In this chapter, a new approach is presented that estimates the propulsion mode of the target accurately and quickly, enabling potentially drastic mode-based control actions by a robotic observation system, and improves velocity estimation for the target by anticipating velocity changes before they become apparent in very noisy sensor readings. The approach consists of a hybrid estimation framework with underlying models taken from the Kalman filter based estimation framework of Chapter 4, but now complemented with the uncertain visual mode observation signal, $\sigma(t)$, from the classification system presented in Chapter 5.

Because existing hybrid estimation techniques are unable to detect all behavior mode transitions by the specimens of this system, the perceptual information from the visual body motion mode classifier is brought in to make the mode discernable in a fast and accurate manner. Thus, these components complement each other: the visual mode

observation is not accurate enough to base drastic control actions upon it, and a hybrid estimator processing the continuous-valued measurements cannot successfully derive mode. By using the visual mode observation within the framework for hybrid estimation, the resulting estimate of target mode is effectively filtered by balancing the evidence from the body motion classifier with the evidence in the continuous-valued sensors. Furthermore, the estimate of target velocity is improved, using visual cues from the body motion classifier to anticipate velocity changes before they are clear in the noisy sensor data.

The visual mode observation, $\sigma(t)$, proves to be the key element that permits successful estimation of the target’s mode and velocities. However, using this information in an estimator poses some unique challenges. This signal is discrete-valued, having only a small, finite number of possible values, yet is still uncertain and error-prone. Thus, the uncertainty model of a Gaussian distribution, with a mean and covariance, which is most naturally compatible with most estimation approaches, cannot be used to describe the uncertainty of σ . State estimation involving distributions that include non-Gaussian components is typically accomplished using particle filters. However, these can become prohibitively expensive computationally for systems with large or even modestly sized state dimensions, particularly when required to numerically model large degrees of uncertainty. In fact, previous work [49] on the robotic observation system combined the methods from [42] and [34] to achieve quality estimates of mode and velocities using the visual mode observations as part of a classical particle filter, but required far too many samples to accurately represent the system’s state distributions in real-time.

The solutions presented in this chapter involve two important steps. The first is the use of a more general method to specify the relationships between and evolutionary models for the system’s variables, a dynamic Bayesian network (DBN) [15], including an observation model relating $\sigma(t)$ to the target’s true mode. In [49], the classical particle filter used for state estimation on this DBN was very expensive computationally. Thus, in this dissertation, the second important step taken is to carry out state estimation on this DBN model by making an extension to a form of Rao-Blackwellized particle filter, the Gaussian Particle Filter (GPF) [25, 26], dubbed here to be the new *mode-observed* Gaussian Particle Filter. “Rao-Blackwellization” [8, 18] can be used to significantly reduce the computational requirements of a particle filter when certain assumptions about the underlying system models can be made. Rao-Blackwellized particle filters represent their system being tracked as a dynamic Bayesian network partitioned into two sides: one non-Gaussian and possibly nonlinear, the other linear and Gaussian. The states in the non-Gaussian partition are sampled using sequential Monte Carlo techniques (as in classical particle

filtering) while the statistics of the states on the linear Gaussian side are computed analytically with the updates of the Kalman filter, conditioned on the sampled values of the non-Gaussian states. Thus, many fewer samples are required to accurately represent the distributions than in a classical particle filter, and a significant improvement in computational efficiency is achieved without sacrificing accuracy. This technique, for instance, has been applied to estimation on linear stochastic hybrid systems [14]. The Gaussian Particle Filter is a variant of this technique which extends the approach to stochastic hybrid systems with moderately nonlinear component dynamical and/or observation models by modifying the underlying machinery on the Gaussian side of the DBN to use the Unscented Kalman Filter (UKF). For the application in this dissertation, the underlying UKF is taken from Chapter 4 but with mode-specific target motion models substituted for the generic model. The new *mode-observed* Gaussian Particle Filter extends the Gaussian Particle Filter to incorporate an uncertain observation related to a hybrid system's mode (for this application, the visual observation, $\sigma(t)$).

This chapter is organized as follows: first, the full system models for the robotic observation system are presented, including a hybrid model for the target's dynamics in its various propulsion modes. A family of dynamic Bayesian network models for stochastic hybrid systems with mode-related discrete-valued observations is introduced and the robotic observation system's variables are cast into this framework. Then, an observation model for the visual mode observation, $\sigma(t)$, and its associated uncertainty is proposed. A state estimation framework using the mode-observed Rao-Blackwellized Particle Filter for stochastic linear hybrid systems and the mode-observed Gaussian Particle Filter for stochastic nonlinear hybrid systems is then presented. Simulated tracking data is generated and results are compared for (1) the UKF estimator from Chapter 4, (2) a Gaussian Particle Filter (which uses only the continuous-valued observations available to the system), and (3) the mode-observed Gaussian Particle Filter. Finally, these estimation comparisons are made again on example field data and video collected during experiments in Monterey Bay, California.

6.1 System Model Equations

Before presenting the new estimation approach for hybrid systems with discrete mode-related observations, the system model presented for the underwater robotic observation system in Chapter 4 is revisited here and extended to include the full hybrid target model for the joint vehicle/target system. Then, the system model is cast into the framework of

a dynamic Bayesian network (DBN) and the classifier output of the subsystem described in Chapter 5 is incorporated into the system model in the context of that framework. Thus, the equations that describe the robotic observation system and its targets are used as an example to both motivate and demonstrate the estimation methods presented in this chapter.

6.1.1 Process and Sensor Models

The continuous state used for estimation in the robotic observation system, as well as (single model) process and sensor models were given in Chapter 4 in (4.1)-(4.4), with coordinates and vectors illustrated in Figure 2.3. Here, they are reproduced and extended such that the state is now a hybrid state consisting of the tuple $\langle \mathbf{x}, m \rangle$, specified for the robotic observation system in (6.1). Correspondingly, the process model now includes a hybrid model for the target dynamics, as now specified in (6.2) and (6.4). The stochastic mode transition model is defined in (6.5).

The hybrid state is defined as:

$$\begin{aligned} \mathbf{x} &= \left[\mathbf{r}_v^T \quad \dot{\mathbf{q}}_v^T \quad \dot{\mathbf{p}}_w^T \right]^T \\ m &\in \{Rest, VelocityTransition, ConstVel\} \end{aligned} \quad (6.1)$$

The equations used in estimation for the hybrid process model, mode transition model and sensor model follow. Note that the equations of this section contain several noise terms of the form $\mathbf{n}^{(\cdot)}$ which represent sensor noise or process noise terms. All are assumed to be zero-mean Gaussian random variables, distributed with covariances that will be specified later in this chapter when example data sets from simulation and collected field data are processed.

The process model is given by:

$$\dot{\mathbf{x}} = f(\mathbf{x}, m, \mathbf{u}^v, \mathbf{u}^w, \boldsymbol{\lambda}, \boldsymbol{\omega}, \mathbf{n}^f, \mathbf{n}^p) = \begin{cases} \frac{d}{dt} \mathbf{r}_v &= v \mathbf{R}^w(\boldsymbol{\lambda}) \cdot \dot{\mathbf{p}}_w - \dot{\mathbf{q}}_v - \boldsymbol{\omega} \times \mathbf{r}_v \\ \frac{d}{dt} \dot{\mathbf{q}}_v &= g_v(\mathbf{u}_v + \mathbf{n}^f, \dot{\mathbf{q}}_v) - \boldsymbol{\omega} \times \dot{\mathbf{q}}_v \\ \frac{d}{dt} \dot{\mathbf{p}}_w &= g_t(m, \mathbf{x}) \end{cases} \quad (6.2)$$

with,

$$\begin{aligned} g_v(\mathbf{u}_v + \mathbf{n}^f, \dot{\mathbf{q}}_v) &= \mathbf{n}^f; \\ \mathbf{n}^f &\sim \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_f^2)) \end{aligned} \quad (6.3)$$

In (6.2), $\boldsymbol{\lambda}$ is the orientation of the vehicle in the water frame, and $\boldsymbol{\omega}$ is the angular rate of the vehicle, expressed in body coordinates. Both vectors are taken directly from

the Octans gyrocompass [46] onboard the ROV. For full descriptions and discussions of the model (6.3) and the first two lines of (6.2), see Chapter 4, sections 4.3.3 and 4.3.1, respectively.

The following equation encodes the tracked target's component subsystem dynamical models depicted in the finite automaton of Figure 5.2(b):

$$g_t(m, \mathbf{x}) = \begin{cases} -\frac{1}{\tau}\dot{\mathbf{p}}_{\mathbf{w}} + \mathbf{n}^r, \tau > 0; & m = Rest \\ \mathbf{n}^{vt}; & m = VelocityTransition \\ \mathbf{n}^{cv}; & m = ConstVel \end{cases} \quad (6.4)$$

$$\mathbf{n}^r \sim \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_r^2))$$

$$\mathbf{n}^{vt} \sim \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_{vt}^2))$$

$$\mathbf{n}^{cv} \sim \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_{cv}^2))$$

This model treats the target as a particle which can move in any direction. The models are kinematic only, with the exception of a damping term applied within the “*Rest*” mode equation. This term is present to drive the target's velocity toward zero when it is resting, although the velocity can still change due to the white noise acceleration term, \mathbf{n}^r . If chosen with a physically realistic magnitude, estimation results in the “*Rest*” mode will track well during deceleration transients. The other modes are modeled with only white noise acceleration terms that allow an estimator to adapt to changes in the velocity to different degrees based on the magnitude of the covariances that define \mathbf{n}^{vt} and \mathbf{n}^{cv} . By definition \mathbf{n}^{vt} is larger in magnitude than \mathbf{n}^{cv} to allow an estimator to adapt more quickly to changing velocities in the “*VelocityTransition*” mode than in the “*ConstVel*” mode.

The mode transition model defining $p(m_k|m_{k-1})$ is represented by a Markov chain whose properties are specified by a square stochastic transition matrix, H , of dimension equal to the number of modes of the hybrid system. Thus, each element of H represents a transition probability given by:

$$h_{ij} = p(m_k = i | m_{k-1} = j) \quad (6.5)$$

$$i = 1, \dots, N_{modes}; \quad j = 1, \dots, N_{modes}$$

The sensor models for the cameras and Doppler Velocity Log (DVL) are unchanged from those of the single model estimator presented in Chapter 4 in (4.3)-(4.4). For clarity, these are reproduced again in (6.6)-(6.7) (see section 4.3.2 for the detailed definition of all terms).

$$\begin{aligned} \mathbf{s}^{cam,(i)} &= h^{cam,(i)}(\mathbf{r}_v) = d^{(i)} \left\{ n^* p \left(s^{(i)} \mathbf{T}^v \cdot \mathbf{r}_v \right) \right\} + \mathbf{n}^{cam,(i)}; \\ \mathbf{n}^{cam,(i)} &\sim \mathcal{N} \left(\mathbf{0}, \text{diag}(\sigma_{cam,(i)}^2) \right) \end{aligned} \quad (6.6)$$

$$\begin{aligned} \mathbf{v}^{dvl} &= h^{dvl}(\mathbf{x}, \mathbf{n}^{dvl}) = \dot{\mathbf{q}}_v + \boldsymbol{\omega} \times \mathbf{r}_v^{dvl} + \mathbf{n}^{dvl}; \\ \mathbf{n}^{dvl} &\sim \mathcal{N} \left(\mathbf{0}, \text{diag}(\sigma_{dvl}^2) \right) \end{aligned} \quad (6.7)$$

6.1.2 A Dynamic Bayesian Network System Representation

In this section, the concepts of Bayesian networks and dynamic Bayesian networks are briefly introduced as an intuitive way to specify and visualize the interrelationships between random variables such as subsets of the states of a stochastic dynamic system as they evolve with time. The aspects of dynamic Bayesian networks most relevant to this dissertation are the means they provide for specifying types of systems to which certain estimation algorithms apply, and the sparse topographical map of conditional distribution functions that they provide for efficient evaluation by those estimation algorithms. As an example, the representation of a linear Gaussian stochastic dynamic system as a dynamic Bayesian network (DBN) is presented and related to the familiar Kalman filter. Then, the model of the underwater robotic observation system is recast as a DBN, including an additional observation model to represent the motion classifier that bridges the visually derived classification into the framework of the dynamical and observation models of (6.1)–(6.7).

A Bayesian network (BN) is a graphical representation of a model of the probabilistic relationships and conditional independence of a set of variables [45]. The networks in Figure 6.1 are an example. An arrow connecting two nodes in the network (an edge) indicates that a conditional dependence exists for the child variable (the node pointed to) upon the parent variable. Each edge is therefore associated with a conditional probability function relating the parent and child node variable. Based on the (typically sparse) set of conditional dependencies defined by the network and the distributions associated with the variables and their parents, full joint distributions may be efficiently computed for any particular assignments to the variables in the network. That is, if each of N variables in the network are X_i, \dots, X_N , then the joint probability that they are assigned to x_i, \dots, x_N , respectively, is given by

$$p(x_i, \dots, x_N) = \prod_{i=1}^N p(x_i | \text{parents}(X_i)) \quad (6.8)$$

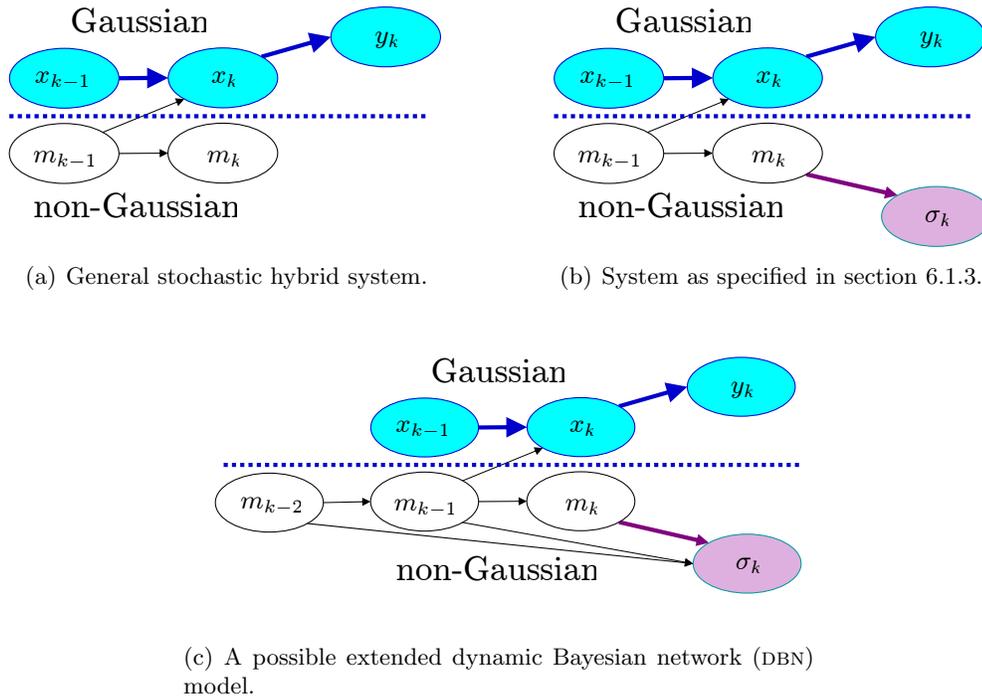


Figure 6.1: A set of dynamic Bayesian network (DBN) models relating the robotic observation system’s variables. **Nomenclature:** x : continuous state vector, y : continuous measurement vector, m : system (target) mode, σ : perceptual mode observation.

A dynamic Bayesian network (DBN) [15] is a Bayesian network describing the relationships between dynamic state variables that evolve over time. In this case, the network illustrates the dependencies between these variables at given instances in time. A dynamic Bayesian network is a graphical, intuitive and extremely flexible way to depict and specify the relationships between a system’s variables as they evolve temporally. A DBN can include both discrete- and continuous-valued variables simultaneously. It captures, both qualitatively and quantitatively, how the uncertainties associated with each variable propagate to the subset of variables it directly influences. They provide a great deal of flexibility in specifying conditional distributions, as each edge can be associated with any kind of probabilistic influence - e.g., linear Gaussian, non-Gaussian, a conditional probability table (CPT), etc.

Within the context of estimation of the states of complex systems, the DBN view of a system provides a useful and intuitive way to view and to partition the variable set such that specialized estimation approaches are known to be compatible with the system’s model. This network view of the system provides a convenient way to specify families

of systems that can be represented by general characteristics of the topographies of the network. They provide a map for evaluation of the conditional distributions specified by each edge when the system’s states are tracked by a particle filter [34].

When the network view of a system model is structured such that the network can be partitioned into linear Gaussian “leaf” nodes and non-Gaussian “root” nodes, the family of Rao-Blackwellized particle filters can be utilized to greatly reduce the computational load involved in tracking complex systems [18]. The ability to specify families of system models based on how they may be partitioned in a DBN view is leveraged in this dissertation to specify the sets of system models for which the new estimation approach presented applies.

It is instructive to relate the DBN view of a system to the underlying system models of some well-established estimators. The Kalman filter (KF) [33] for linear Gaussian state-space systems is one example. The continuous-time portion of the DBNs of Figure 6.1 (above the Gaussian/non-Gaussian separators) could represent the relationships between the states \mathbf{x}_{k-1} and \mathbf{x}_k and the observations \mathbf{y}_k as modeled by a discrete-time KF (with no control inputs). Then the conditional probabilities represented by the edges in that DBN are the conditional Gaussian distributions of \mathbf{x}_k given \mathbf{x}_{k-1} defined by the state transition matrix and the process noise covariance, and of \mathbf{y}_k given \mathbf{x}_k based on the observation matrix and the observation noise covariance.

The assumed model structure for stochastic hybrid systems utilized in estimators such as the Interacting Multiple Model (IMM) estimator [3], the multiple model bootstrap filter [42] and the Rao-Blackwellized Particle Filter for linear stochastic hybrid systems [14] that were summarized in Chapter 3 can all be represented by the network structure shown in Figure 6.1(a). Each of those estimators assumes a mode transition model of the form specified by (6.5) for the edge between m_{k-1} and m_k (although the DBN and particle filtering estimation algorithms do not limit models to only the form of (6.5)). The continuous state evolution model associated with the edge from \mathbf{x}_{k-1} to \mathbf{x}_k and the observation model for continuous sensors \mathbf{y}_k as influenced by \mathbf{x}_k are linear and Gaussian for those estimators, except for the implementation of [42] which permits nonlinear models (although still with Gaussian statistics assumed). The model structure depicted in Figure 6.1(a) also applies when the linear assumption is dropped, such as in the variant of the IMM that uses sets of EKFs instead of KFs, or when the UKF replaces the KF for the Gaussian updates of [14] in the Gaussian Particle Filter (GPF) [26].

The model equations (6.1)–(6.7) that describe the robotic observation system are mapped into the DBNs of Figure 6.1 as follows. The edge from \mathbf{x}_{k-1} to \mathbf{x}_k is mapped to the continuous state process models of (6.2)–(6.4). The target models of (6.4) are also

conditioned on the mode of the target, as depicted by the directed edge from m_{k-1} to \mathbf{x}_k . The mode transition model, (6.5) specifies the influence from m_{k-1} on m_k .

The influence on the visual classifier measurement σ_k by the true mode of the system m is depicted by the edge from m_k to σ_k in Figure 6.1(b), or alternatively the three edges from m_k , m_{k-1} and m_{k-2} as shown in Figure 6.1(c). A detailed model will be proposed for this edge of the network in the next subsection. The networks shown are two possible network topologies that will be shown later to be compatible with the mode-observed Gaussian Particle Filter – it is the partitioning of the mode measurement σ away from the continuous states of the system defines this family of network topologies.

6.1.3 A Conditional Probabilistic Model for σ

Figure 6.1(b) illustrates the state evolution and observation models of the target, cast in the form of a DBN. Note the inclusion of the discrete motion classifier, whose output, σ_k , is assumed to be dependent only upon m_k . The use of this DBN as the underlying model of a particle filter requires some conditional probability distribution (CPD) that can be numerically sampled to relate the observation σ_k to the value of m_k . Based on the error rates expected when running the visual classifier [48], a conditional probability table (CPT) can be constructed and utilized to approximate this relationship. Hence, the classification algorithms that make up the “Perception” block of the new system architecture in Figure 1.3(b) is characterized as if it were a unique type of “sensor”, and this CPT may be thought of as capturing its characteristics. The CPT is shown in Table 6.1.

$p(\sigma|m)$: Conditional Probability Table

| Observation | Propulsion mode, m | | |
|---------------------------------|----------------------|---------------------------|-----------------|
| | <i>Rest</i> | <i>VelocityTransition</i> | <i>ConstVel</i> |
| $p(\sigma = \textit{Resting})$ | 0.90 | 0.15 | 0.03 |
| $p(\sigma = \textit{Moving})$ | 0.08 | 0.80 | 0.09 |
| $p(\sigma = \textit{RepPulse})$ | 0.02 | 0.05 | 0.88 |

Table 6.1: CPT relating σ to m

Although a relatively simple relationship between the visual mode observation, σ_k , and the true mode of the system, m_k , is assumed here, the framework provided by the dynamic Bayesian network is extremely flexible. For instance, a more complex model could be proposed that encodes some of the lag characteristics of the visual classifier, as depicted in Figure 6.1(c), where σ_k is shown to be dependent on m_k , m_{k-1} and m_{k-2} . However, as will be shown later in this chapter, even the simple model of the classifier as a

random process modeled by the CPT of Table 6.1 enables good estimation of the target's modes and velocities when incorporated into the mode-observed Gaussian Particle Filter.

6.2 Estimation Assisted by Mode-Related Observations

Estimation on the models represented by the networks in Figure 6.1(b) and 6.1(c) can be achieved with a particle filter [34]. These models include both Gaussian and non-Gaussian uncertainty models for state evolution and observations, including the observation model for the independent mode classifier. This was demonstrated in [49] using a classical particle filter but required far too many samples to be considered computationally realistic for real-time implementation. To achieve computational efficiency, particle filters whose models support certain assumptions may be Rao-Blackwellized. In this section, Rao-Blackwellized Particle Filters for estimation on stochastic hybrid systems are extended in two steps to become applicable to the robotic observation system. First, the Rao-Blackwellized Particle Filter for stochastic linear hybrid systems [14] is extended to incorporate the mode-related observation. Then, this same extension is applied to the Gaussian Particle Filter [26] (for stochastic nonlinear hybrid systems) with the same change to the algorithm. With these extensions, computationally efficient and high quality estimation of the target's mode and velocities using both traditional sensor information and information from a separate classifier of mode is enabled.

Some previous work exists in solving this problem. For instance, Evans and Evans presented the Image-Enhanced Interacting Multiple Model estimator (IE-IMM) which augmented the IMM with direct visual classifications of the mode of a maneuvering target (aircraft) [20]. This estimator assumes a stochastic linear hybrid system, with direct (but uncertain) mode observations. Sworder, et. al. presented the Gaussian Wavelet Estimator (GME), using a different estimation algorithm but under the same system and observation assumptions [66–68]. These limiting assumptions are relaxed with the approach presented here, such that nonlinear component dynamics are allowable, and more complex models for mode-related observations are supported.

6.2.1 The Mode-Observed Rao-Blackwellized Particle Filter

The mode and states of linear stochastic hybrid systems can be estimated from continuous sensor measurements based on the system's continuous states using a variety of techniques, including the efficiently implemented Rao-Blackwellized Particle Filter [14] when measurement and process noises may be assumed to be Gaussian. In this section, that method

is extended to incorporate an augmenting observation related to the mode of the hybrid system.

For particle filters applied to hybrid systems with only continuous variables being observed, the measurement update consists of a computation of the probability of \mathbf{y}_k , the continuous-time observation, given the continuous state of each sample, $\mathbf{x}_k^{(i)}$. Now, with a second observation available, σ_k , the measurement update requires the computation of a joint probability of (\mathbf{y}_k, σ_k) given (\mathbf{x}_k, m_k) for the sample. In the Rao-Blackwellized particle filter for stochastic hybrid systems, each sample has instantiated values for m_k and any past values of m that are carried along (per the DBN structure of the model). Because the values of the mode variables, m , are instantiated to specific values, the probabilities of \mathbf{y}_k and σ_k given \mathbf{x}_k and m_k become independent. Therefore the joint probability is given by the product of the two separately conditioned probabilities:

$$p\left(\mathbf{y}_k, \sigma_k | \hat{\mathbf{x}}_k^{(i)}, M_{0:k}^{(i)}\right) = p\left(\mathbf{y}_k | \hat{\mathbf{x}}_k^{(i)}\right) p\left(\sigma_k | M_{0:k}^{(i)}\right) \quad (6.9)$$

Note that all previously described hybrid estimation algorithms in Chapter 3 assumed that only the previous time step's mode and the current mode has any influence on any relevant variables. Here, the requirement for this Markov assumption is relaxed, and the use of DBN models such as in Figure 6.1(c) are possible. Thus, the notation for a mode history, e.g., from time 0 to time k , is introduced here as $M_{0:k}$ and is included in the general forms of the estimation algorithms of this chapter.

If, as in the DBN of Figure 6.1(b), the influence on σ_k is assumed to be only exerted by the current mode of the system, m_k , then the second term of (6.9) simplifies to:

$$p\left(\sigma_k | M_{0:k}^{(i)}\right) = p\left(\sigma_k | m_k^{(i)}\right) \quad (6.10)$$

and therefore the expression in (6.9) becomes:

$$p\left(\mathbf{y}_k, \sigma_k | \hat{\mathbf{x}}_k^{(i)}, M_{0:k}^{(i)}\right) = p\left(\mathbf{y}_k | \hat{\mathbf{x}}_k^{(i)}\right) p\left(\sigma_k | m_k^{(i)}\right) \quad (6.11)$$

Because the expression $p(\sigma_k | m_k^{(i)})$ exists on the sampled, non-Gaussian side of the DBN model, any relationship that can be numerically sampled may be utilized. For the implementations used to compute the results that will be presented later in this chapter, this relationship is assumed to be a CPT of the form specified by Table 6.1.

Identical to the RBPF algorithm of Table 3.3, except for step 3(a)iii, which is replaced by:

Compute the weight of particle $\hat{p}^{(i)}$:

$$w_k^{(i)} \sim p\left(\mathbf{y}_k, \sigma_k | M_{0:k}^{(i)}, \hat{\mathbf{x}}_k^{(i)}\right) =$$

$$p\left(\mathbf{y}_k, \sigma_k | M_{0:k}^{(i)}, \hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{\mathbf{S}}_{k|k-1}^{(i)}\right) = \mathcal{N}\left(\mathbf{y}_k; \hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{\mathbf{S}}_{k|k-1}^{(i)}\right) p\left(\sigma_k | M_{o:k}^{(i)}\right)$$

Table 6.2: The mode-observed RBPF algorithm.

Identical to the GPF algorithm of Table 3.4, except for step 3(a)iii, which is replaced by:

Compute the weight of particle $\hat{p}^{(i)}$:

$$w_k^{(i)} \sim p\left(\mathbf{y}_k, \sigma_k | M_{0:k}^{(i)}, \hat{\mathbf{x}}_k^{(i)}\right) =$$

$$p\left(\mathbf{y}_k, \sigma_k | M_{0:k}^{(i)}, \hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{\mathbf{S}}_{k|k-1}^{(i)}\right) = \mathcal{N}\left(\mathbf{y}_k; \hat{\mathbf{y}}_{k|k-1}^{(i)}, \hat{\mathbf{S}}_{k|k-1}^{(i)}\right) p\left(\sigma_k | M_{o:k}^{(i)}\right)$$

Table 6.3: The mode-observed GPF algorithm.

The steps of the algorithm for the Rao-Blackwellized Particle Filter for piecewise linear hybrid systems were given in Table 3.3. For the general family of linear hybrid systems with mode-related observations of the forms depicted in Figure 6.1, the algorithm is modified only in the likelihood weighting step, which now accounts for the mode-related observation likelihood per (6.9). The modified algorithm, resulting in the mode-observed Rao-Blackwellized Particle Filter for linear hybrid systems is given in Table 6.2.

6.2.2 The Mode-Observed Gaussian Particle Filter

The mode-observed Gaussian Particle Filter is derived by making the same change to the Gaussian Particle Filter as was made in the preceding subsection to the Rao-Blackwellized Particle Filter for linear stochastic hybrid systems. This is formalized in Table 6.3.

The steps of the RBPF (GPF) and mode-observed RBPF (GPF) are illustrated graphically in Figures 6.2 and 6.3. In Figure 6.2, an example iteration of the RBPF (GPF) is shown, with steps executing from left to right. Here individual samples of the RBPF (GPF) are shown by the circular particles, and are color-coded to indicate the sample's current mode, with the color scheme for the illustration based on the three mode model for underwater animals of Figure 5.2(b) (red-*Rest*, blue-*VelocityTransition*, green-*ConstVel*). This diagram begins from left to right on an iteration at time step k , with the leftmost column illustrating the numerical representation of the prior state distribution at time $k - 1$ and progressing to the rightmost column with the posterior state distribution at time k .

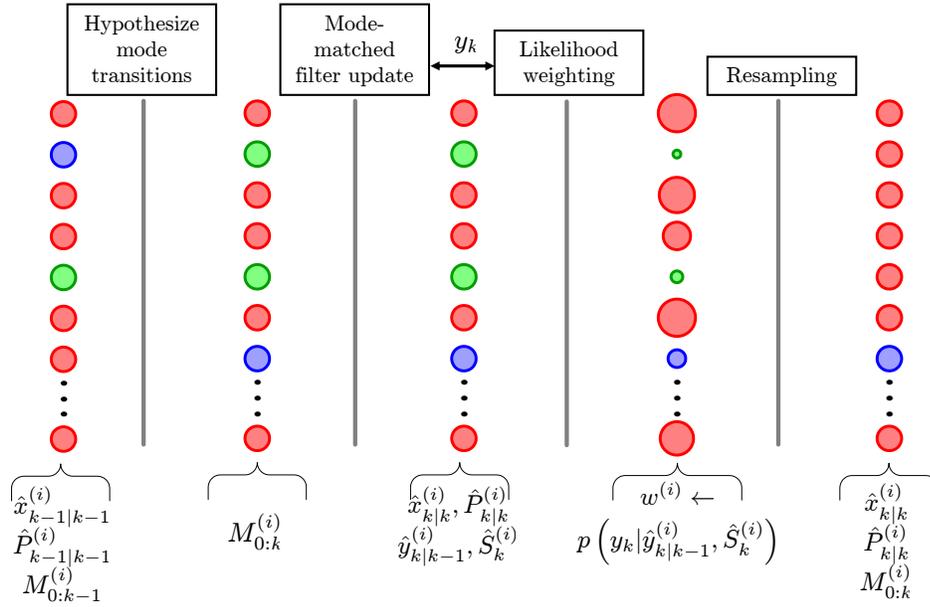


Figure 6.2: Graphical depiction of Rao-Blackwellized (or Gaussian) Particle Filter for hybrid linear (nonlinear) systems.

Following a single sample of the set, the one in the top row, the algorithm illustration may be described as follows. This sample has current mode of *Rest* at time $k-1$. The next step is to hypothesize mode transitions; for this sample, no transition is made. Then a KF (UKF) update is done using the observation \mathbf{y}_k and the model associated with the *Rest* mode. Using the predicted observation and residual covariance from the filter update, the sample is then weighted, per Table 3.3 or 3.4. The weight of this particle is illustrated by the size of the sample in the fourth column after the likelihood weighting step – in this case the sample was assigned a high likelihood weight and therefore is shown large with respect to the other samples. Finally, the last step toward creating the posterior distribution estimate for time k is the resampling step, where samples with high weight are duplicated with probability proportional to their weight and samples with very low probability are dropped entirely. The top row’s sample is duplicated into the last column due to its high weight.

The second version of the diagram, in Figure 6.3, illustrates the same process, but this time for the *mode-observed* RBPF (GPF). The process is identical except for the likelihood weighting step, which now augments the sensor vector, \mathbf{y}_k , with the mode-related discrete-valued observation, σ_k (shown on top with an arrow pointing to the left as input to the likelihood weighting step). Thus, the first three columns of samples are identical to those of Figure 6.2. In the illustration, all priors and the sensor measurement vector are shown

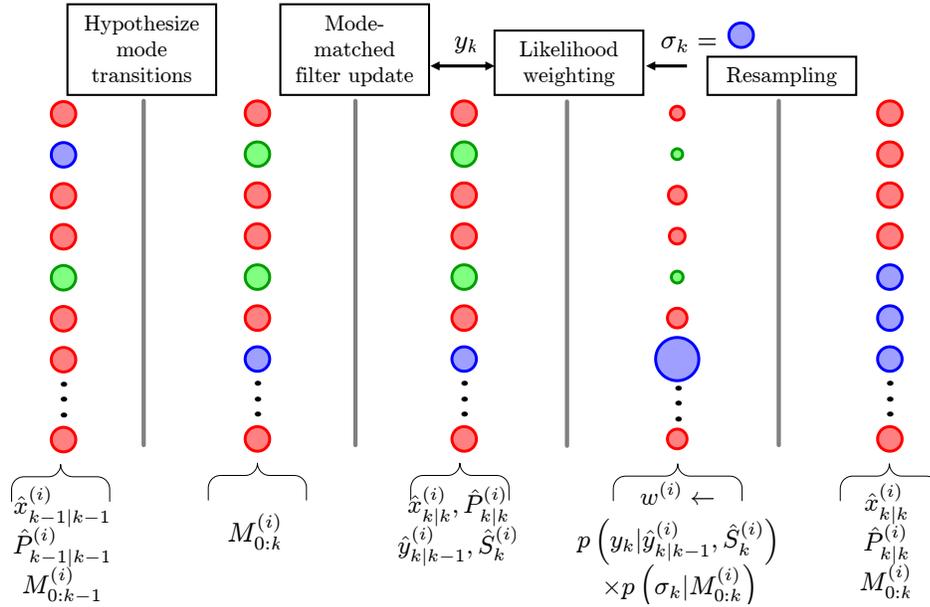


Figure 6.3: Graphical depiction of Mode-observed Rao-Blackwellized (or Gaussian) Particle Filter for hybrid (nonlinear) systems.

to be the same as in Figure 6.2 – however, because the mode-related observation, σ_k , indicates that the current mode is probably that of the blue color (*VelocityTransition*), the samples in the proposed and updated sample set in the third column whose mode hypothesis supports that observation are weighted more heavily. Thus, for this example, this augmented estimator is able to favor the correct mode in its posterior statistics much more quickly.

6.2.3 Applying Mode Observations to the Multiple Model Bootstrap Filter

The same extensions made to the Rao-Blackwellized estimators for stochastic hybrid systems can be made to the multiple model bootstrap filter [42]. This was implemented for the robotic observation system [49] but has been superseded for that application by the more computationally efficient mode-observed Gaussian Particle Filter to exploit the normal distribution assumptions about the process and continuous sensor noises. However, for the sake of completeness, the extension is applied to the multiple model bootstrap filter here.

Identical to the multiple model bootstrap algorithm of Table 3.2, except for step 3(a)iii, which is replaced by:
 Compute the weight of particle $\hat{p}^{(i)}$:
 $w_k^{(i)} \sim p(\mathbf{y}_k, \sigma_k | M_{0:k}^{(i)}, \mathbf{x}_k^{(i)}) = p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\sigma_k | M_{o:k}^{(i)})$

Table 6.4: The mode-observed multiple model bootstrap filter algorithm.

6.3 Example Results

To compare the performance of the new estimator presented in the previous section with that of previously existing methods, two representative example data sets are presented in this section, one generated in simulation and the other taken from field data collected while tracking a small and agile jellyfish with the robotic observation system employed on the ROV *Ventana* in Monterey Bay, California. These data sets are used in this section to compare the performance of three estimators: (i) the UKF estimator of Chapter 4, (ii) a Gaussian Particle Filter (GPF), which (by definition) operates only on continuous state observations, and (iii) a mode-observed Gaussian Particle Filter, which is supplemented by data from the online body motion mode classifier, $\sigma(t)$, as defined in Chapter 5.

6.3.1 Parameterization for the Estimator Equations

The parameters used in generating the results of this section are enumerated and commented on here. These include the normally distributed process and sensor noise covariances of (6.2)– (6.4), (6.6), and (6.7), the mode transition Markov matrix, H , and the CPT defining $p(\sigma_k | m_k)$. The parameter settings are summarized in Table 6.6.

The continuous-valued sensors (camera pixel locations and Doppler Velocity Log readings) and the assumed process noise on the tracking vehicle are all assumed to be independent of the target’s mode. The sensor noise standard deviation values are chosen based on field experience with the vision sensing system and the DVL and are set to 2 pixels for each camera measurement (in a 160x120 image) and 3 cm/s on the DVL water velocities. Vehicle disturbance process noise standard deviation is set to 0.5 volts on all axes (on a scale with limits at +/- 5 volts for thruster command levels).

Several parameters specify the target models of Figure 5.2(b) for the estimators which utilize hybrid dynamical models. Target process noise terms were specified with standard deviations of 2 cm/s² (*Rest*, *ConstVel*) and 10 cm/s² (*VelocityTransition*). The exponential decay (damping) term of the *Rest* mode dynamics was specified with a time constant, τ_r , of 2 sec.

For the UKF, which uses a single model design, the target model is a constant velocity model, with a single value for white noise acceleration standard deviation of 4 cm/s². This choice lies between the settings within the multiple model estimators for quiescent modes and the maneuvering mode, and is the result of the compromise required such that a single mode estimator will track adequately through more than one type of motion behavior.

The mode transition probability model, H , and the CPT used to model $p(\sigma_k|M_{0:k}^{(i)})$ must be populated for the hybrid estimators. These models are approximations of the true mode transition mechanisms and classifier error behavior, respectively, which are crude but effective in practice. If the true system components matched the forms of these models, and hard data for those models was readily available, that data would be used. In reality, the models are very approximate and their parameterizations can be chosen to match the average characteristics over a wide variety of examples. Since the effects of the parameters on estimator performance are well-known, a sound approach to populating them are to start from these averages and adjust from there to achieve the desired response characteristics. For instance, using an extremely diagonally dominant H reduces the bandwidth of the mode estimate of a stochastic hybrid estimator [3]. Similarly, a CPT for $p(\sigma_k|M_{0:k})$ with values near 1.0 for the correct m - σ association results in σ being given a very strong influence on the mode estimate of perception-assisted hybrid estimators, resulting in a large sensitivity to errors in σ . Thus, the values for these two models are populated here based on average characteristics which are then tuned with these considerations in mind.

The Markov mode switching probabilities are given by H below, where h_{ij} represents the probability of switching to mode i from mode j . The modes are indexed from 1 to 3 in the order presented in (6.1).

$$H = \begin{bmatrix} 0.85 & 0.33 & 0.10 \\ 0.15 & 0.34 & 0.05 \\ 0.0 & 0.33 & 0.85 \end{bmatrix}$$

The diagonal elements for the first and third modes are chosen such that the target is assumed to stay in those modes for reasonably long periods of time, while still allowing the possibility of mode switches from each to have reasonable credence (totaling 15% probability for all exit possibilities). In practice, the use of values for h_{11} and h_{33} between about 0.8 and 0.95 yield similar results. Another noteworthy aspect of H is that h_{31} is zero, indicating that the direct transition from mode 1 (*Rest*) to mode 3 (*ConstVel*) is not possible, as is also specified in the state machine diagram of Figure 5.2(b). Also note

that the terms in the second column are tuned to give about equal probability of switches from mode 2 (*VelocityTransition*) to any other mode as well as remaining in mode 2, indicating that this mode is intended to be a brief transitory mode bridging between the other two modes where the bulk of all time spent is assumed to lie.

The direct use of the CPT probabilities from Table 6.1 in (6.11) were found to make the estimator too sensitive to errors in the classifier output, σ , forcing the estimator to be too trusting in σ over evidence in the continuous measurements and the priors. To blunt this effect and achieve a better balance between the discrete classifier outputs and the prior belief states, a more uncertain version of the CPT was used in the mode-observed GPF estimator, given in Table 6.5.

$p(\sigma|m)$: Conditional Probability Table

| Observation | Propulsion mode, m | | |
|---------------------------------|----------------------|---------------------------|-----------------|
| | <i>Rest</i> | <i>VelocityTransition</i> | <i>ConstVel</i> |
| $p(\sigma = \textit{Resting})$ | 0.75 | 0.20 | 0.10 |
| $p(\sigma = \textit{Moving})$ | 0.18 | 0.67 | 0.18 |
| $p(\sigma = \textit{RepPulse})$ | 0.07 | 0.13 | 0.72 |

Table 6.5: CPT relating σ to m , tuned to reduce the modeled certainty of the mode-related measurement, σ .

Finally, to end this subsection, the parameters that have been specified are collected in Table 6.6.

| Parameter | Value | Units |
|----------------------------------|---|----------------------------|
| σ_f | $[0.5 \ 0.5 \ 0.5]^T$ | Volts (ROV thruster scale) |
| σ_r | $[2 \ 2 \ 2]^T$ | cm/s ² |
| σ_a | $[10 \ 10 \ 10]^T$ | cm/s ² |
| σ_{cv} | $[2 \ 2 \ 2]^T$ | cm/s ² |
| σ_p (UKF only) | $[4 \ 4 \ 4]^T$ | cm/s ² |
| σ_{cam} (for all cameras) | $[2 \ 2 \ 2]^T$ | pixels |
| σ_{dvl} | $[3 \ 3 \ 3]^T$ | cm/s |
| τ_r | 2 | s |
| H | $\begin{bmatrix} 0.85 & 0.33 & 0.10 \\ 0.15 & 0.34 & 0.05 \\ 0.0 & 0.33 & 0.85 \end{bmatrix}$ | probabilities |
| $p(\sigma_k m_k)$ | see Table 6.5 | probabilities |
| \mathbf{r}_v^{dvl} | $[-1.77 \ 0.0 \ 0.39]^T$ | m |

Table 6.6: Summary of model parameters assumed for estimation.

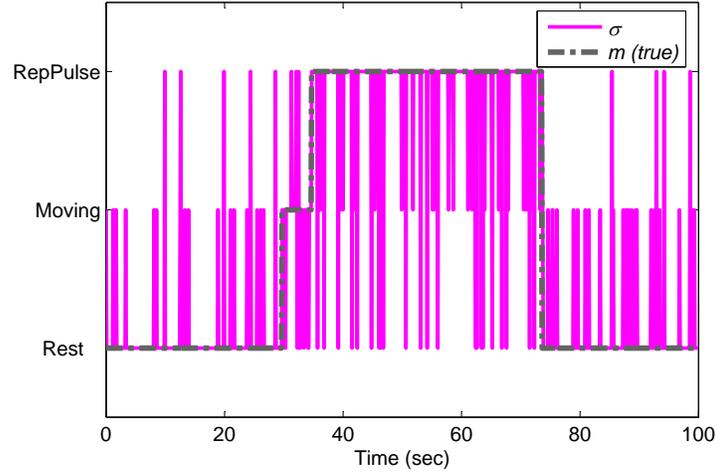


Figure 6.4: σ and m (true mode) for simulated case.

6.3.2 Simulated Case

To test the performance of the new estimator and pre-existing ones, test data was generated using the baseline tracking control software (as documented in Chapter 2) to track a simulated moving target. This target was tracked through a sequence of propulsive modes, m , in the following order: resting, acceleration, steady swimming, resting. Noisy and distorted pixel measurements for each camera of the stereo pair as well as compass and angular rate readings were generated at the 10 Hz update rate of the observation system. DVL velocities with additive zero-mean Gaussian noise as specified previously were generated at 5 Hz (the maximum update rate of the DVL deployed at MBARI), with angular rates of the vehicle coupled into the measurements based on the location of the DVL on the ROV *Ventana*.

A simulated mode observation vector, σ , the noisy and imperfect mode classification, was generated based upon the expected error rates of the body motion classifier vision algorithm as applied to the “true” mode, m , of the target. This was specified earlier as a CPT with values shown in Table 6.1. The time history of m and the generated vector σ for this trajectory are plotted in Figure 6.4.

Results Summary

A marked improvement in overall tracking quality of mode and target velocity is demonstrated by the mode-observed GPF estimator over the other two estimators. The test trajectory may be partitioned into phases by the motion of the target, and the estimator

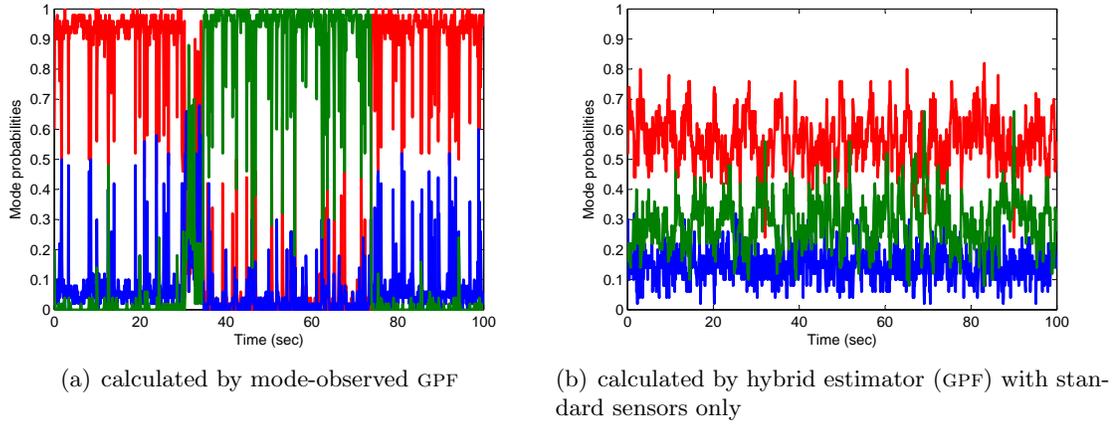


Figure 6.5: Mode probability results on simulated data. Mode probability key: *Rest*–red, *VelocityTransition*–blue, *ConstVel*–green.

errors for each algorithm are tabulated in Table 6.7. The modal trajectory of the target consists of a sequence of $\{Rest, VelocityTransition, ConstVel, Rest\}$, and for the purposes of judging the performance of the estimators, the transition from *ConstVel* to *Rest* is broken down into two phases (deceleration and resting, where velocity is nearly zero).

The mode probabilities calculated for each multiple model estimator are shown in Figure 6.5, with (a) showing results from the estimator of (ii) which uses continuous measurements only, and (b) showing the results from estimator (iii). These results demonstrate that the uncertainties of the measurements and of the vehicle and target dynamics are too high to discern modal information successfully without the extra information from the online classifier. The noisiness in the mode probabilities in (b) are primarily in response to errors in σ . This response is momentary, countered by the evidence in the priors and continuous measurements, keeping overall state tracking errors due to classifier errors small.

In addition to tracking the target’s mode well, the estimator that utilizes the output of the perceptual algorithms of Chapter 5 (iii) outperforms both of the other estimators in velocity tracking for all motion phases. The performance improvement is particularly notable in the deceleration portion of the trajectory, where this estimator is able to anticipate the deceleration based on visual cue of the ceasing of body motions by the target (as recognized by the body motion classifier). Velocity tracking results for the target’s velocity in the vertical direction and the 2-norm of the overall velocity error are presented in Figures 6.6 and 6.7, respectively.

| Estimator | <i>Rest1</i> | <i>VelTrans</i> | <i>ConstVel</i> | <i>Decel</i> | <i>Rest2</i> | Overall |
|-------------------------|--------------|-----------------|-----------------|--------------|--------------|---------|
| (i) UKF | 1.1 | 2.4 | 1.4 | 3.1 | 1.7 | 1.8 |
| (ii) GPF | 0.7 | 4.1 | 4.5 | 1.2 | 1.3 | 2.5 |
| (iii) mode-observed GPF | 0.4 | 2.3 | 1.2 | 1.1 | 0.9 | 1.0 |

Table 6.7: Average of 2-norm of error in target velocity estimates (cm/s), by target motion phase.

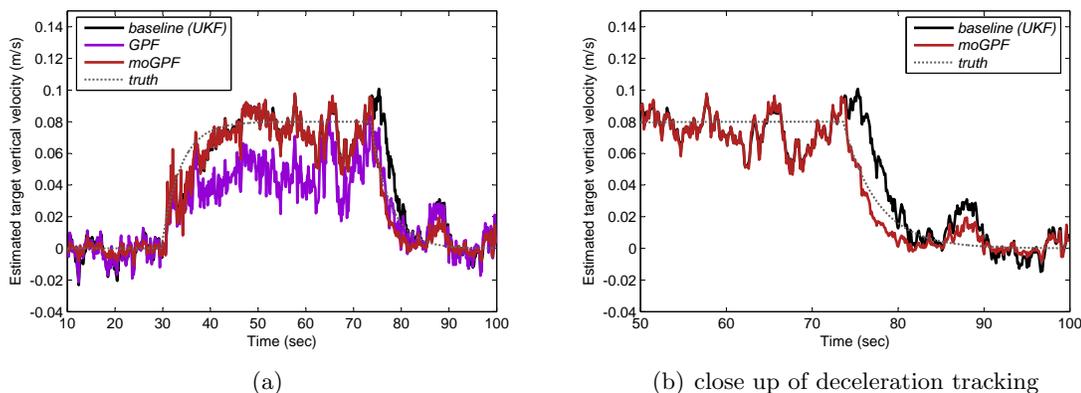


Figure 6.6: Vertical velocity of target, estimated for simulated case.

6.3.3 Application to Field Data

Figure 1.4(a) showed a series of snapshots of a small agile medusa jellyfish evading the robotic observation system. This animal was tracked on September 15, 2005, and provided a good source of test data by changing its behavior modes to and from active and resting modes often, and sometimes generating enough acceleration to evade the observation system. In this subsection, an example of this specimen where it was tracked continuously through some mode changes (i.e., it did not evade the robotic observation system) is used to demonstrate the effectiveness of the mode-observed Gaussian Particle Filter in tracking its mode and velocities.

A representative data set is given in this section that includes the specimen swimming actively downward through the water column, then stopping swimming. The transition to a resting mode first involves a couple of last active pulsing motions before finally settling into a rest mode. The video from this data set was processed off-line using the body motion mode classification algorithms presented in Chapter 5. The results are shown in Figure 6.8. As seen in this plot, there are some errors in the classifier's output, but none are sustained over any long period of time and in general the target mode is identified correctly by the visual classification algorithms. The data recorded by a subset of the

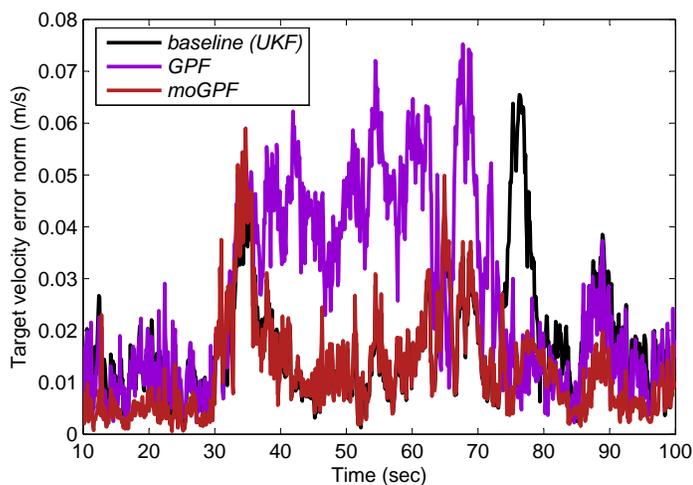


Figure 6.7: 2-norm of velocity estimation error for simulated case.

continuous-valued sensors (one camera’s pixel location measurements and the vertical velocity measured by the DVL) are shown in Figure 6.9.

Figures 6.10–6.12 show the mode probabilities calculated by the hybrid estimators and the velocity estimates of all estimators as applied to this data set, respectively. Figure 6.12 is a zoomed-in version of Figures 6.10(a) and 6.11, focusing on the time frame near the switch from actively swimming to resting and the resulting deceleration of the specimen in the water frame. These plots show good tracking of the mode of the specimen when assisted by the body motion classifier signal, σ .

The ability to track the specimen’s mode improves velocity tracking as well. Although the true velocity of the specimen is not available to allow the computation of velocity error, some qualitative comparisons between velocity estimation can be made. In particular, when looking at the tracking of the deceleration motion of the target shown in Figure 6.12, the output of the mode-observed GPF is more rapidly heading to zero velocity than the other estimates. Like in the simulated case, the mode-observed GPF is able to exploit the higher bandwidth information stream from the visual body motion mode classifier, which visually recognizes that the specimen has stopped moving. Hence, the estimator can predict the deceleration of the target toward zero velocity with a good deal of confidence.

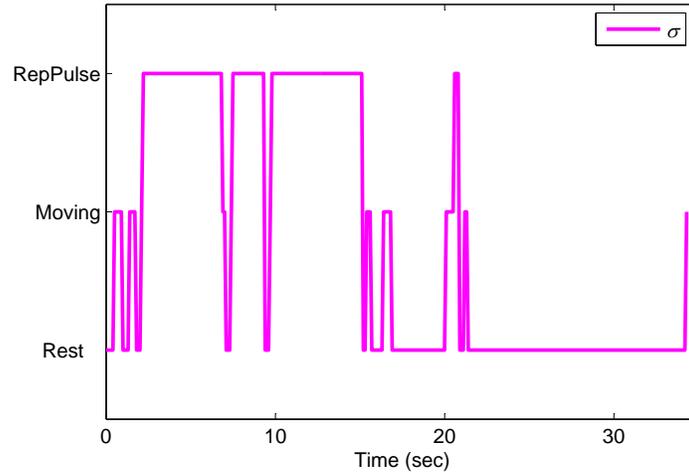


Figure 6.8: Visual mode observation, σ , as applied to field data.

6.3.4 Mode Estimation Bandwidth Comparison

The mode estimation abilities of the three estimators compared in this section, as well as the visual mode observation, σ , can be summarized in terms of the time it takes to recognize mode changes, as well as the general ability (or lack thereof) to do so.

First, in principle, the UKF which does not have any intrinsic concept of mode, could be coupled to a mode monitor that tests velocity estimates against a threshold to classify the mode of the target. As in the one-dimensional example used in Chapter 3, the ability of this scheme to detect any mode change signal in the noise of the velocity measurements would require a high steady-state velocity by the target. Assuming this to be the case, the time required to recognize mode changes can be approximated as the rise time of the estimator to changes in velocity. From data such as that of Figure 6.6, that rise time is approximately 1 second. Thus, the average mode estimation delay using the UKF coupled to a velocity monitor would be about 1 second, assuming the ideal conditions required for that scheme to work could be met. (If those conditions were met, then a more standard hybrid estimator could be used here, with somewhat faster mode recognition times. As has been documented, this is not possible in practice.)

The visual classification signal, σ , is a mode estimate of sorts that provides indications of mode changes much more quickly. As documented in Chapter 5, mode changes by the targets are typically registered in σ within 0.2-0.3 seconds.

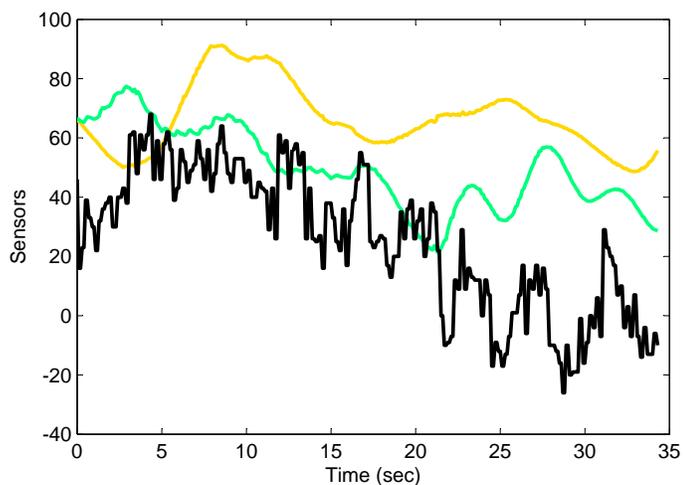


Figure 6.9: Sensor data during field test. Green and yellow are pixel measurements, in black is vertical velocity measured from the DVL in mm/s.

The mode-observed GPF presented in this chapter, lands in between these two extremes by blending the information from each source. The example method of using a CPT to represent $p(\sigma|m)$ provides the engineer a mechanism to tradeoff speed in recognizing mode changes via the visual classification path with avoidance of false mode change indications. For the examples given here, the mode indication of the mode-observed GPF lags only slightly behind that of σ , giving an approximate end-to-end mode detection time of 0.3-0.4 seconds.

6.3.5 Real-Time Considerations

The full hybrid estimators for this application have not been implemented in real-time code and deployed in the ocean, although that task is the first item of future work enumerated in Chapter 7. The results in this chapter were computed using Matlab for both simulation and post-processing of field data. However, the UKF of Chapter 4, which has been used here as a baseline comparison and which forms the building block of the hybrid estimators tested here, has been implemented in C++ and deployed on the ROV *Ventana*. Therefore, its real-time load on a benchmark processor can be measured and extrapolated for the estimators of this chapter.

The extrapolation of the UKF to these estimators is as follows. The UKF requires 22 “sigma points”, or deterministically chosen samples, which are each propagated through the process and observation equations. This provides a scaling point to compare to the

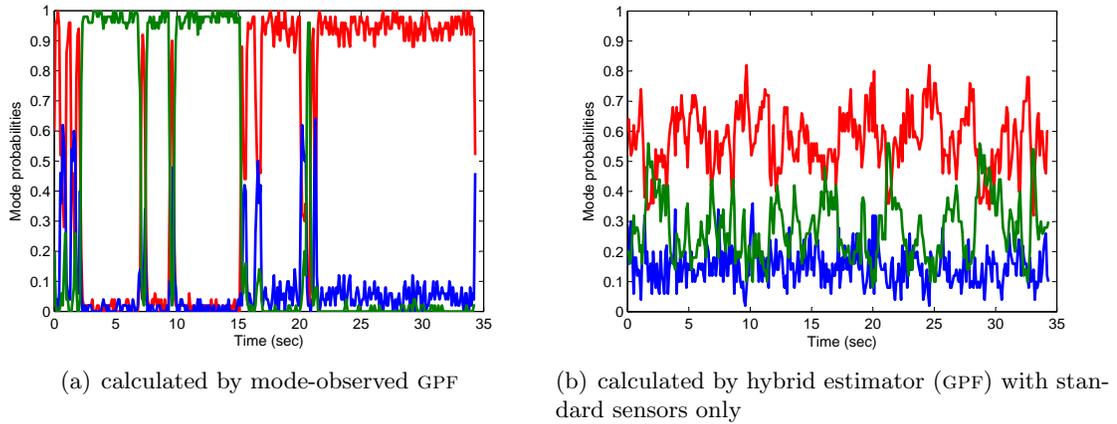


Figure 6.10: Mode probability results on field data. Mode probability key: *Rest*–red, *VelocityTransition*–blue, *ConstVel*–green.

mode-observed Gaussian Particle Filter and the analogous classical particle filter (the mode-observed multiple model bootstrap filter from Table 6.4). The mode-observed GPF is implemented here with 50 samples, each containing a version of that UKF, therefore incurring approximately 50 times the load. The mode-observed multiple model bootstrap filter, whose results [49] are effectively identical to the mode-observed GPF, required about 50,000 samples to achieve that accuracy. Hence, an approximate scaling factor to extrapolate from the UKF’s real-time load is $50,000/22 \approx 2273$.

The UKF was benchmarked on a Dell Inspiron 700m, with a 1.60 GHz Intel Pentium M processor, as requiring 0.04ms to execute. Extrapolating this to the GPF gives an approximately 2ms execution time, well within budget for 10Hz operation. The multiple model bootstrap filter extrapolates to 91ms, not fast enough to run at 10Hz in the presence of the real-time vision processing. Thus, a factor of over 45 has been achieved in computational efficiency by using the Rao-Blackwellized mode-observed GPF instead of the classical particle filter equivalent, with no noticeable degradation of accuracy.

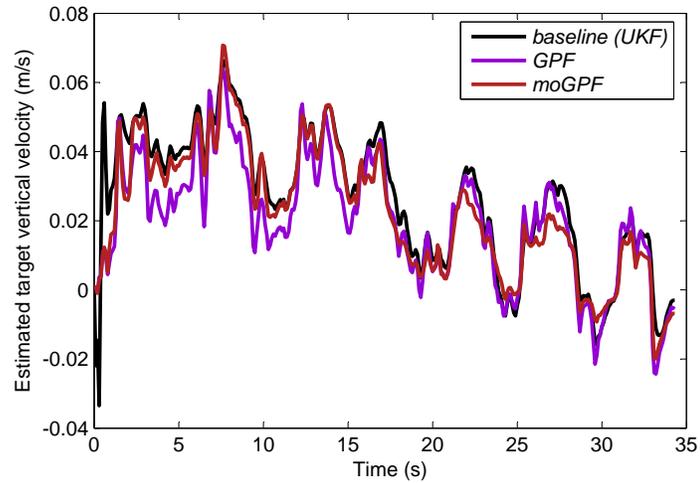
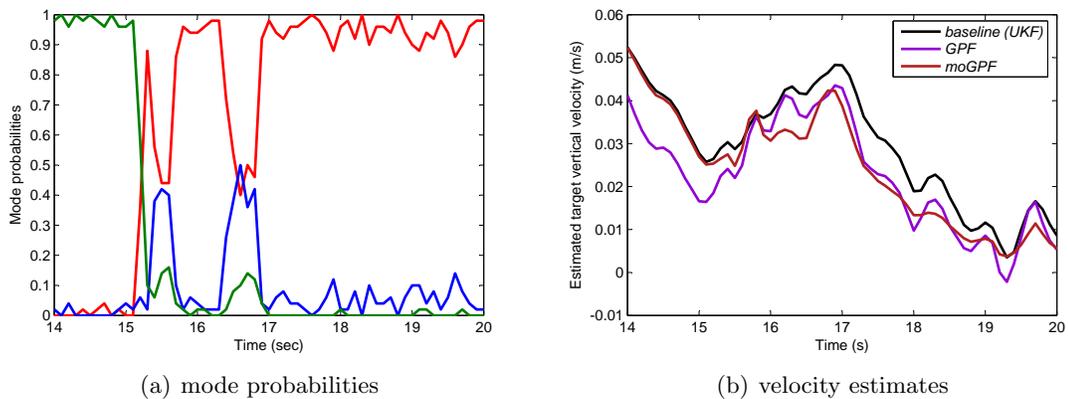


Figure 6.11: Vertical velocity of target, estimated, for field data.



(a) mode probabilities

(b) velocity estimates

Figure 6.12: Velocity and mode probability estimates on field data (deceleration phase). Mode probability key: *Rest*–red, *VelocityTransition*–blue, *ConstVel*–green.

Chapter 7

Conclusions and Future Research

This dissertation has presented a framework for the estimation of the mode and continuous states of stochastic hybrid systems using both traditional sensors and perceptual information about the system's mode. This is particularly advantageous when estimating the states of hybrid systems which includes modes that are not observable (or poorly so) using the available information from traditional sensors, and which have a potential for mode observation using other perceptual but uncertain methods. These estimation and perception algorithms have been demonstrated on a motivating application – the tracking of behavior modes and velocities of ocean animals using standard underwater vehicle sensors combined with computer vision – using both simulated data sets and data and video collected on actual animals in the ocean.

A motivation for this work is the desire to extend a robotic system for automatically tracking and filming marine animals in their natural environment to handle targets whose behavior mode changes result in fast accelerations and decelerations with respect to the water. When the tracking and filming task is done under the control of a human pilot, the ability to recognize visually and respond to changes in the behavior modes of the tracked animals quickly is an important ability which enables the human pilot to maintain sight of agile animals very naturally. The contributions of this dissertation could provide an approximation of this ability to a robotic observation system, and thereby open up the possibilities of mode-based control strategies to react quickly and intelligently to mode changes by the system's specimens.

The perception algorithms in this dissertation are designed for visual recognition of the motion modes of marine animals, and the estimators use parameters specific to the underwater robotic observation system and its targets. However, the overall framework

developed in this dissertation for both perception and estimation using perceptual observations is very flexible. The framework can be easily extended to other hybrid systems by substituting other forms of robotic perception, sensors and mode-dependent dynamical models. Furthermore, any of the component algorithms presented for the robotic observation system application could potentially be improved or modified incrementally without changing any fundamental aspect of the overall architecture.

7.1 Extensions of the Underwater Robotic Observation System

Integration with Real-Time Software and Mode-based Control Strategies

This dissertation has provided the tools for the robotic observation system to perceive what its specimen is doing, in the form of recognition of behavior modes and improved velocity estimation. Armed with this additional information, potential control strategies are now enabled that may utilize this knowledge to its advantage. The system is most likely to lose tracking continuity when the tracked target accelerates or decelerates quickly – thus, when recognizing such mode changes, the primary objective should be to reduce the chance of losing sight of it. The most direct way to do so is to increase range to the target or zoom out the control cameras to enlarge the volume of water viewable and thereby improve the observation system’s chance of maintaining track continuity through relative positioning error transients. This, in fact, is the most typically observed first reaction by human ROV pilots when performing the tracking task upon the more agile specimens. This approach is the first control strategy that can be tried and can be tied directly to the mode estimate of the perception-augmented hybrid estimator.

To date, the perception and estimation algorithms in this dissertation have been tested only off-line in post-processing settings. An important prerequisite step will be to convert them into a fully real-time and recursive form and in software that can be placed in line with the existing real-time control and sensing software. All of the new algorithms have been designed with computational constraints in mind, and the transition to on-line, recursive real-time software should be fairly straightforward. Cutler and Davis [12], whose algorithms for visual recognition of periodic motion provided a starting point for the perceptual algorithms of this dissertation, offer some insights for efficient real-time implementation of the vision algorithms that are still applicable here.

Advanced Vision Algorithms and Target Modeling

The proposed robotic observation system as augmented by the algorithms of this dissertation seeks to extract more information from the video stream than did the baseline version – namely monitoring gross activity levels in its body visually and using that data to classify the target into a small set of discrete modes. There is, however, much more information present in the visual scene. For instance, another particularly useful capability of the human ROV pilot is the ability to visually observe the orientation of the specimen. The typical target can thrust primarily in one direction relative to its body’s orientation – hence, with knowledge of orientation, the human observer can easily anticipate the direction of velocity changes when they occur due to body motion mode changes. If more advanced vision algorithms are developed to visually recognize the orientation of the specimen’s body, the target model can be extended from the current 3-DOF model to a 6-DOF model which then leverages visual orientation information to predict velocity changes more accurately.

Science Data

The algorithms in this dissertation provide a means to derive the behavior mode and to quantify the water-relative motion of the specimens of the observation system. This has been driven by the desire to react in real-time to behavior mode changes and thereby avoid loss of visual lock on the target. However, the quantities that are output by these algorithms are valuable in themselves from an experimental biology standpoint. The collection of experimentally derived statistics on behavior modes and dynamic motion data is difficult, expensive and laborious, but this data will be computed by the observation system once augmented by the algorithms of this dissertation. Furthermore, the algorithms here, which have been designed with a heavy emphasis on computational efficiency, could be extended for a post-processing application. In a post-processing setting, the relaxation of the real-time execution and causal constraints opens the door to the use of more powerful and accurate vision, estimation and identification algorithms than have been considered in this dissertation.

Species-Specific Tracking Approaches and Target Classification

The mode models and associated dynamical models used in this dissertation have been chosen to capture the essential characteristics of a large set of potentially tracked species by their generality. However, models with parameters that are specific to the type of target

being tracked could be utilized as well. The data in Table 5.1 provides several examples. The use of such specific knowledge such as expected velocity magnitude of a species when swimming actively, or a richer set of modes tailored to the target could provide enhanced tracking capabilities. The additional data computed by the visual classifier such as body pulse period could be linked to expected water-relative velocities for some species. It also could be advantageous to utilize different mode-based control responses based on the species being tracked.

The activation of species-specific estimation or control approaches would require some method of classifying the species being tracked, either by autonomous methods or via operator selection. In the context of automatic classification online by the system, the behavior mode, associated velocities, and body pulse periods estimated by the algorithms of this dissertation could act as important distinguishing characteristics by which species might be classified. This dynamic and mode-related data may significantly augment other pre-existing classification algorithms using more static characteristics of the target being observed [19].

Positioning with Respect to Moored Platforms

The technologies developed for automatic tracking of deep ocean animals are beginning to be transformed to provide the basis for new capabilities in relative-positioning with respect to moored instrument platforms [52]. As several deep ocean cabled observatory networks near deployment to enable data sampling of the ocean at high temporal and vertical resolutions, deployment and servicing challenges are being examined. One component of these observatories is moored platforms which are to be deployed and serviced using Remotely Operated Vehicles (ROVs). As an alternative to the risks involved when making a fixed attachment to these often massive platforms to enable precise manipulation of their subcomponents, the relative-positioning technology of the deep ocean animal observation system is being recast to provide a mooring-relative positioning system. The approach provides a shared control system in which the ROV hovers automatically with respect to the mooring leaving the pilot free to focus only on the manipulation tasks.

The system requirements for the tasks of positioning with respect to moorings and marine animals are similar in many ways, but some important extensions must be made. The animal observation system was designed to sense the relative position of the specimen of interest and servo the vehicle to maintain a constant range to the animal while pointing directly at it (thus absolute heading control is not a core requirement). Several extensions

are necessary to enable station-keeping with respect to the mooring. The primary differences are the needs (a) to estimate the orientation (and rotational velocity) of the tracked object, (b) to servo the vehicle such that it is always pointed at a specific face of the object as it rotates and (c) to handle intrusions of vehicle appendages (such as manipulators) in the camera's view. Preliminary field tests of the mooring-relative shared control system were carried out in Monterey Bay, California in 2006 using the ROV *Ventana* and research activities to more fully develop this system are continuing.

7.2 Perception-Augmented Estimation for Hybrid Systems

The estimation examples used in Chapter 6 do not take full advantage of the flexibility of the dynamic Bayesian network model for which the mode-observed Gaussian Particle Filter can admit. Some of the modeling assumptions made there could apply within simpler hybrid estimators such as the IMM or the IE-IMM, which admits error-prone direct observations of mode, with an observation model similar to the conditional probability tables used in that chapter. However, without increasing computational loads significantly, the mode-observed GPF provides a great deal of flexibility to the estimator designer for the modeling of the mode-related perceptual observation algorithms and the mode transitions of the target. Some possible options to exploit this extra flexibility are enumerated in this section.

More Advanced Modeling of the Perceptual Observation

Good results were achieved using the simple CPT model proposed for the error properties of the perceptual observation, σ . However, that model neglects some significant attributes of the perception algorithms. The lag characteristics of the algorithms, in particular, are not included. Doing so would fit very naturally into the estimation framework supported by the mode-observed GPF and could improve its accuracy. Also, it may be worthwhile to study the error mechanisms of the perceptual algorithms and model those within the estimator framework. For instance, sustained errors by the classifier have not yet been acknowledged in the estimator model. This is a possibility not supported by the error model of the CPT used in Chapter 6, which assumes errors simply occur randomly and independent of any previous errors. Acknowledgment of sustained classifier errors could permit them to be effectively overridden by conflicting evidence in the more traditional sensor observations.

Advanced Mode Transition Dynamics

Again due to the flexibility of the dynamic Bayesian network model assumed by the estimation framework, more complex mode transition models can be included here than are possible with the Markov matrix models used in most hybrid estimators. For instance, in the context of tracking marine animals, the idea of an animal getting “tired” could be incorporated by tracking current statistics on the discrete mode side of the model and assuming transitions to the “Rest” mode would be more likely if the target had been observed to be swimming for a long period of time.

Also, the mixing of stochastic and autonomous transitions, for instance, with mode transition probabilities changing as a function of the continuous state estimates (e.g., depth in the case of the underwater tracking application) could be incorporated. Related work in Rao-Blackwellized particle filtering for autonomous hybrid systems [21] could be combined with the algorithms presented in this dissertation to model these kinds of dynamics.

7.3 Additional Applications of Perception-Augmented Estimation

The framework for estimating the mode and continuous states of a hybrid system by augmenting the continuous state observations with discrete, classifier-style observations related to the system’s mode could be readily applied to other systems with such observations available. Some natural applications of these algorithms might be in the surveillance of aircraft or ballistic missiles where some available observations are of continuous states (e.g., radar) while other observations might be the product of discrete on-line classification, based on the processing of imagery from visible spectrum cameras, infrared sensors or laser radar. The estimation algorithms of this dissertation could fuse this sort of data, merging range, azimuth, elevation data from radar with mode-related observations related to events such as thruster burns, bank angles or aerosurface deployment regimes that might be derived from online image classification.

Bibliography

- [1] H. Balakrishnan. *Target-Tracking and Identity Management Algorithms for Air Traffic Surveillance*. PhD thesis, Stanford University, Stanford, CA 94305, 2006.
- [2] H. Balakrishnan, I. Hwang, J. S. Jang, and C. J. Tomlin. Inference methods for autonomous stochastic linear hybrid systems. In *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science (LNCS 2993)*. Springer-Verlag, 2004.
- [3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley, 2001.
- [4] A. Bemporad, G. Ferrari, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10):1864 – 1876, October 2000.
- [5] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto. Recognition of human gaits. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume II, pages 52–58, 2001.
- [6] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, July 1992.
- [7] J.-Y. Bouget. Camera calibration toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.
- [8] G. Casella and C. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.
- [9] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

- [10] S. Colin and J. Costello. Morphology, swimming performance and propulsive mode of six co-occurring hydromedusae. *The Journal of Experimental Biology*, 205:427–437, 2002.
- [11] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [12] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, Aug 2000.
- [13] T. Daniel. Mechanics and energetics of medusan jet propulsion. *Canadian Journal of Zoology*, 61:1406–1420, 1983.
- [14] N. de Freitas. Rao-Blackwellised particle filtering for fault diagnosis. In *Aerospace Conference Proceedings*, volume 4, pages 1767–1772. IEEE, 2002.
- [15] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.
- [16] S. Donaldson, G. Mackie, and A. Roberts. Preliminary observations on escape swimming and giant neurons in *Aglantha digitale* (Hydromedusae: Trachylina). *Canadian Journal of Zoology*, 58:549–552, 1980.
- [17] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- [18] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic Bayesian networks. In *Uncertainty in Artificial Intelligence (UAI2000)*, pages 176–183, 2000.
- [19] D. R. Edgington, I. Kerkez, D. E. Cline, J. Mariette, M. Ranzato, and P. Perona. Detecting, tracking and classifying animals in underwater video. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), demonstration*, New York, New York, 2006.
- [20] J. Evans and R. Evans. Image-enhanced multiple model tracking. *Automatica*, 35(11):1769–1786, November 1999.
- [21] S. Funiak and B. Williams. Multi-modal particle filtering for hybrid systems with autonomous mode transitions. In *SafeProcess*, 2003.

- [22] S. Haddock, C. Dunn, and P. Pugh. A reexamination of siphonophore terminology and morphology, applied to the description of two new prayine species with remarkable bio-optical properties. *Journal of the Marine Biological Association of the United Kingdom*, 85:695–707, 2005.
- [23] J. Heikkilä and O. Silvén. A four-step camera calibration procedure with implicit image correction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, San Juan, Puerto Rico, 1997.
- [24] A. Huster. *Relative Position Sensing by Fusing Monocular Vision and Inertial Rate Sensors*. PhD thesis, Stanford University, Stanford, California, July 2003.
- [25] F. Hutter and R. Dearden. Efficient on-line fault diagnosis for nonlinear systems. In *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2003.
- [26] F. Hutter and R. Dearden. The Gaussian Particle Filter for diagnosis of non-linear systems. In *Proceedings of the 14th International Conference on Principles of Diagnosis (DX'03)*, pages 65–70, Washington, DC, USA, June 2003.
- [27] I. Hwang. *Air Traffic Surveillance and Control Using Hybrid Estimation and Protocol-Based Conflict Resolution*. PhD thesis, Stanford University, Stanford, CA 94305, 2003.
- [28] I. Hwang, H. Balakrishnan, and C. Tomlin. Observability criteria and estimator design for stochastic linear hybrid systems. In *Proceedings of the IEE European Control Conference*, Cambridge, UK, September 2003.
- [29] I. Hwang, H. Balakrishnan, and C. Tomlin. Performance analysis of hybrid estimation algorithms. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003.
- [30] I. Hwang, J. Hwang, and C. Tomlin. Flight-mode-based aircraft conflict detection using a Residual-Mean Interacting Multiple Model algorithm. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Austin, August 2003.
- [31] Insite Tritech, Inc., <http://www.insitetritech.com>.
- [32] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, pages 1628–1632, 1995.

- [33] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.
- [34] D. Koller and U. Lerner. *Sequential Monte Carlo Methods in Practice*, chapter 21. Springer-Verlag, 2001.
- [35] D. Koller, U. Lerner, and D. Angelov. A general algorithm for approximate inference and its application to hybrid Bayes nets. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in AI (UAI)*, pages 324–333, 1999.
- [36] X. Koutsoukos, J. Kurien, and F. Zhao. Estimation of distributed hybrid systems using particle filtering methods. In *Hybrid Systems: Computation and Control*, 2003.
- [37] R. Larson. Costs of transport for the scyphomedusa *Stomolophus meleagris* L. Agassiz. *Canadian Journal of Zoology*, 65:2690–2695, 1987.
- [38] R. Larson, C. Mills, and G. Harbison. In situ foraging and feeding-behavior of narcomedusae (cnidaria, hydrozoa). *Journal of the Marine Biological Association of the United Kingdom*, 69:785–794, 1989.
- [39] J.-M. Lavest, G. Rives, and J.-T. Lapreste. Underwater camera calibration. In *European Conference on Computer Vision*, volume 2, pages 654–668, 2000.
- [40] R. Marks. *Experiments in Visual Sensing For Automatic Control of an Underwater Robot*. PhD thesis, Stanford University, Stanford, California, 1995.
- [41] P. Maybeck. *Stochastic Models, Estimation, and Control*, volume 2. Academic Press, New York, 1982.
- [42] S. McGinnity and G. W. Irwin. *Sequential Monte Carlo Methods in Practice*, chapter 23. Springer-Verlag, 2001.
- [43] C. Mills and J. Goy. In situ observations of the behavior of mesopelagic solmissus arcomedusae (Cnidaria, Hydrozoa). *Bulletin of Marine Science*, 43(3):739–751, 1988.
- [44] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002.
- [45] J. Pearl and T. Verma. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

- [46] Photonetics, Inc., <http://www.photonetics.com>.
- [47] A. M. Plotnik and S. M. Rock. Quantification of cyclic motion of marine animals from computer vision. In *Proceedings of the IEEE OCEANS Conference*, pages 1575–1581, 2002.
- [48] A. M. Plotnik and S. M. Rock. Improving performance of a jelly-tracking underwater vehicle using recognition of animal motion modes. In *Proceedings of the Unmanned Untethered Submersible Technology Conference (UUST)*, Durham, NH, August 2003. AUSI.
- [49] A. M. Plotnik and S. M. Rock. Improved estimation of target velocity using multiple model estimation and a dynamic Bayesian network for a robotic tracker of ocean animals. In *Proceedings of the 12th International Symposium of Robotics Research*, San Francisco, CA, Oct 2005.
- [50] A. M. Plotnik and S. M. Rock. A multi-sensor approach to automatic tracking of midwater targets by an ROV. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, San Francisco, CA, Aug 2005.
- [51] A. M. Plotnik and S. M. Rock. Relative position sensing and automatic control for observation in the midwater by an underwater vehicle. In *Proceedings of the Unmanned Untethered Submersible Technology Conference (UUST)*, Durham, NH, August 2005. AUSI.
- [52] A. M. Plotnik and S. M. Rock. Visual servoing of an ROV for servicing of tethered ocean moorings. In *Proceedings of the MTS/IEEE OCEANS Conference*, Boston, MA, September 2006.
- [53] Teledyne RD Instruments, Teledyne Technologies, Inc., <http://www.rdinstruments.com>.
- [54] J. Rife. *Automated Robotic Tracking of Gelatinous Animals in the Deep Ocean*. PhD thesis, Stanford University, Stanford, California, December 2003.
- [55] J. Rife and S. M. Rock. A pilot-aid for ROV based tracking of gelatinous animals in the midwater. In *Proceedings of the OCEANS 2001 Conference*, Honolulu, HI, November 2001.

- [56] J. Rife and S. M. Rock. Visual tracking of jellyfish in situ. In *Proc. of the 2001 International Conference on Image Processing*, Thessaloniki, Greece, October 2001. IEEE.
- [57] J. Rife and S. M. Rock. Field experiments in the control of a jellyfish tracking ROV. In *Proceedings of the IEEE OCEANS Conference*, pages 2031–2038, 2002.
- [58] J. Rife and S. M. Rock. Segmentation methods for visual tracking of deep-ocean jellyfish using a conventional camera. *IEEE Journal of Oceanic Engineering*, 28(4):595–608, October 2003.
- [59] J. Rife and S. M. Rock. Design and validation of a robotic control law for observation of deep-ocean jellyfish. *IEEE Transactions on Robotics*, 22(2):282–291, April 2006.
- [60] B. H. Robison. Midwater research methods with MBARI’s ROV. *Marine Technology Society Journal*, 26(4):32–39, 1992.
- [61] B. H. Robison. The coevolution of undersea vehicles and deep-sea research. *Marine Technology Society Journal*, 33(4):65–73, 1999.
- [62] B. H. Robison. Deep pelagic biology. *Journal of Experimental Marine Biology and Ecology*, 300:253–272, 2004.
- [63] K. Roy, B. Levy, and C. J. Tomlin. Target tracking and estimated time of arrival (ETA) prediction for arrival aircraft. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Keystone, Aug 2006. AIAA Paper 2006-6324.
- [64] G. Sibley, L. Matthies, and G. S. Sukhatme. Bias reduction and filter convergence for long range stereo. In *Proceedings of the 12th International Symposium of Robotics Research*, San Francisco, CA, Oct 2005.
- [65] S. W. Strand and W. M. Hamner. Predatory behavior of *Phacellophora-camtschatica* and size-selection predation upon *Aurelia-aurita* (scyphozoa, cnidaria) in Saanich Inlet, British Columbia. *Marine Biology*, 99(3):409–414, October 1988.
- [66] D. D. Sworder and J. E. Boyd. *Estimation Problems in Hybrid Estimation*. Cambridge University Press, 1999.
- [67] D. D. Sworder, J. E. Boyd, and R. J. Elliott. Modal estimation in hybrid systems. *Journal of Mathematical Analysis and Applications*, 245:225–247, 2000.

- [68] D. D. Swarder, J. E. Boyd, R. J. Elliott, and R. Hutchins. Data fusion using multiple models. In *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1749–1753, Pacific Grove, CA, 2000.
- [69] R. van der Merwe, E. Wan, and S. Julier. Sigma-point Kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Providence, RI, Aug 2004.
- [70] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [71] V. Verma, G. Gordon, R. Simmons, and S. Thrun. Real-time fault diagnosis [tractable particle filters for robot fault diagnosis]. *Robotics & Automation Magazine*, 11(2):56 – 66, June 2004.
- [72] R. Vidal and B. Anderson. Recursive identification of switched ARX hybrid models: exponential convergence and persistence of excitation. In *43rd IEEE Conference on Decision and Control*, volume 1, pages 32–37, December 2004.
- [73] R. Vidal, A. Chiuso, S. Soatto, and S. Sastry. Observability of linear hybrid systems. In *Proceedings of the Workshop on Hybrid Systems Computation and Control*, 2003.