Adaptive Tracking of Underwater Targets with Autonomous Sensor Networks

Donald P. Eickstedt (corresponding author)
Massachusetts Institute of Technology
Room 5-204
77 Massachusetts Avenue
Cambridge, MA 02139
email: eicksted@mit.edu FAX: 617-253-2350
(W): 617-253-1032

Michael R. Benjamin
Massachusetts Institute of Technology
Room 5-204
77 Massachusetts Avenue
Cambridge, MA 02139
email: mikerb@csail.mit.edu FAX: 617-253-2350

Henrik Schmidt
Massachusetts Institute of Technology
Room 5-204
77 Massachusetts Avenue
Cambridge, MA 02139
email: henrik@mit.edu FAX: 617-253-2350

John J. Leonard
Massachusetts Institute of Technology
Room 5-204
77 Massachusetts Avenue
Cambridge, MA 02139
email: jleonard@@mit.edu FAX: 617-253-2350

Adaptive Tracking of Underwater Targets with Autonomous Sensor Networks

Abstract

This work describes an ongoing investigation at MIT into the use of a distributed network of autonomous sensor platforms to adaptively track, classify, and prosecute underwater targets in littoral areas. The paper has two major objectives, to quantify the significant advantages that a network of mobile sensors has over a single-sensor paradigm for tracking and classification of underwater targets and to describe a framework for adaptive and cooperative sensor operation in such a network. This framework has three major components, a logical sensor which is used to provide high-level state information to a behavior-based autonomous vehicle control system, a new approach to behavior-based control of autonomous sensor platforms using multiple objective functions which allows reactive control in complex environments with multiple constraints, and the adaptive behaviors and tracking algorithms which use the estimated target state information to control the sensor platform. Experimental results are presented for a 2-D target tracking application using a network of autonomous surface craft (a proxy for our autonomous underwater vehicles) with simulated bearing sensors. Results from both one and two-bearing tracking experiments are given. From these results, the advantages of a multi-sensor, networked approach to target tracking are clearly shown.

I. Introduction

Autonomous oceanographic sampling networks are useful in many cases where large volumes of ocean must be monitored for transitory phenomena including not only the changing physical properties of the ocean itself but perhaps also man-made phenomena like the acoustic fields emitted by underwater objects of interest [1]. This work is motivated by an interest in a fundamental problem in sensor system design and operation for sensing in the ocean which is also applicable to sensing systems on land and in space. That is, how can one sense processes or the characteristics of processes which are intentionally or unintentionally difficult to sense using a single sensor which can only sample the process from a single spatial location at a given instant in time. In this work we define a process as the field generated by a physical event or phenomenon which can be sensed by a physical sensor. Many processes of interest are time-varying and not spatially isotropic and, therefore, either the process itself or some of its characteristics may not be observable from a single sensor platform. For other processes, spatially distributed sensors can add significant processing gain, reducing the sensing time and improving our estimates of the process parameters.

For example, in a passive sonar context, single sensor platforms can localize contacts using passive bearing information but require temporal diversity to do so while multiple, distributed bearing sensors can immediately form a solution. In an active sonar context, the scattering of acoustic energy off of objects of different shapes is highly directional and is dependent on the spatial relationship between the source, receiver, and target. It is impossible for a single sensor platform carrying both source and receiver to capture the full scattered field which is useful in classifying the target shape. A group of distributed sensors would be able to capture the spatial distribution of this scattering for use in classification. For sampling transient oceanographic phenomena such as frontal dynamics, the spatial sampling resolution is related to the frequency content of the frontal process. Synoptic sampling coverage by multiple sensors can help avoid the temporal smearing that would occur in the data sampled by a single sensor platform. These examples, and numerous others, encompass a class of problems in marine sensing that can benefit from a multiple sensor approach.

In addition to being able to address a number of problems that cannot be solved using a single sensor, the use of mobile sensor platforms working in coordination offers several additional advantages. They may each have different payloads, sensors, and endurance capabilities. A network of small, inexpensive platforms with low-performance sensors may be able to use its spatial diversity to outperform systems using single, very expensive, high-performance sensors. The use of multiple platforms also may allow one platform to stay at the surface, with a higher bandwidth link to other robotic or human operated vehicles, while one or more other platforms operate under the surface at varying depths to optimize their sensor-oriented tasks. Network survivability is also enhanced as the loss of one or even possibly several inexpensive sensors can be absorbed with the redundancy inherent in such a network.

While coordinated sensor platforms have their advantages, they present challenges in their joint control to reach their combined potential. Inter-vehicle communication, if possible, is limited in bandwidth, and carefully allocated. Any kind of central continuous control is likely infeasible. In multi-vehicle joint exercises involved with sensing dynamic phenomena, it may not be practical or effective to think in terms of a single vehicle state space to which proper actions can be assigned a priori. The question then remains as to how we can accomplish this sensing of dynamic phenomena in the ocean with multiple sensors. What would such a system look like and how would it behave? As difficult as it is to sense phenomena in the ocean with a single sensor, coordinating multiple sensors seems a daunting challenge. In Section II we outline a number of the requirements necessary for implementing marine sensor networks. In the balance of this work we address a number of these challenges by presenting a novel architecture for cooperative control of autonomous marine sensors in the context of two bearings-only target tracking

scenarios.

Experimental results are presented for a 2-D target tracking application in which fully autonomous surface craft using simulated bearing sensors acquire and track a moving target in open water. In the first example, a single sensor vehicle adaptively tracks a target while simultaneously relaying the estimated track to a second vehicle acting as a classification platform. In the second example, two spatially distributed sensor vehicles adaptively track a moving target by fusing their sensor information to form a single target track estimate. In both cases the goal is to adapt the platform motion to minimize the uncertainty of the target track parameter estimates. The link between the sensor platform motion and the target track estimate uncertainty is fully derived and this information is used to develop the behaviors for the sensor platform control system. The experimental results clearly illustrate the significant processing gain that spatially distributed sensors can achieve over a single sensor when observing a dynamic phenomenon as well as the viability of behavior-based control for dealing with uncertainty in complex situations in marine sensor networks.

II. Requirements for a Marine Sampling Network

In this section we attempt to define some of the major requirements for a marine sampling network with the goal of being able to sense and characterize dynamic ocean phenomena both natural and man-made using multiple, cooperating sensor platforms.

A. Mobility

One way to provide coverage of an area with multiple sensors would be to lay out a grid of fixed sensors all communicating back to a central data processing location. The grid spacing of the fixed sensors would be related to the process under observation. In fact, this method is used in many ocean sensing systems. For example, there is currently a network of ocean buoys that monitors parameters such as sea surface temperature, currents, conductivity, and ocean wave statistics for use in weather and hurricane forecasting systems. However, for many problems of interest, laying out a fine grid of fixed sensors is clearly impractical. This would be the case, for example, in applications where sensing must take place over a wide area with fine resolution or in deep water where the installation and maintenance costs of a sensor grid of the necessary size would be prohibitive. Fixed sensor systems are also not appropriate for applications where a temporary monitoring system is needed or in applications in which some action must be taken when certain conditions are sensed. The mobility of the sensor platforms is a key aspect of the adaptive sampling scenario, allowing dynamic optimization of the sensor locations with respect to

the reduction in uncertainty of the process parameters we are attempting to estimate. A mobile sensor paradigm also allows resource optimization in scenarios where specialized sensors on mobile platforms can be brought to bear on a problem when more generalized sensors have made initial determinations. For example, in a mine countermeasures scenario, a network of low-frequency sonar platforms could localize a potential target and then call in additional sensor platforms with chemical sensors or sidescan sonars to gather additional information. In military target tracking applications, kill vehicles could be vectored to a target by a network of sensors which are simultaneously tracking and classifying the target.

B. Adaptivity

In the absence of a fine grid of sensors which can spatially sample a phenomenon simultaneously from multiple points, the sensors must not only be mobile but they must also be able to autonomously adapt their motion in real time according to the sampled data. This requires tight coordination between the sensors and the vehicle control. Given that a sensor platform may carry multiple heterogeneous sensors, this requires a sensor integration model that abstracts sensory data for use by the sensor platform control system. In Section III-B we describe a sensor integration model that makes use of the concept of a logical sensor that abstracts away the details of the physical sensor. In Sections IV and V we use this model in two experiments using simulated bearing sensors where the output of the logical sensor is a target track.

Once a sensor platform receives sensory data, the platform control system must use this environmental state data to maneuver. Typically, our goal will be to maneuver the platform in such a way as to gain additional information about the process we are observing. This requires some sort of mapping between the environmental state data and the vehicle control parameters (rudder, elevator, speed, etc.) In general, there are two major methods used for robotic control, the world model-based and the behavior-based methodologies. In the world model-based methodology, one large model is used to map the environmental state data to the control parameters. However, the very large state space inherent to a marine vehicle operating in any reasonably complex application is prohibitive for such an approach in my view. A sensor platform may be dealing not only with sensor data from an application specific sensor like an acoustic array but also with tasks like obstacle avoidance, path planning, and navigation. A direct mapping of sensor states to vehicle control variables is infeasible with such a large state space. A behavior-based control system, in contrast, uses a number of modular computing units termed ”behaviors”, all operating in parallel, to decide the vehicle’s course of action during each control cycle. During a control cycle, each behavior will use the current sensor state data to compute its opinion on the next course of action. For example, each behavior may output its preferred course, speed, and depth for the vehicle. The issue

then arises about how to select the preferred action when multiple behaviors disagree. One method would be to simply pick the output of the behavior with the highest priority. This scheme was used by Brooks in his original layered control method [2]. This method, however, does not allow for the possibility of compromise between the preferred actions of different behaviors. In this work, we use a method for behavior-based control in which each behavior outputs its preferred action as an objective function over the vehicle control variables. During each control cycle, the preferred action is decided by performing a multi-function optimization over all of the objective functions. The optimization is performed using the Interval Programming Method (IvP) developed by Benjamin [3] to perform the optimization in a computationally efficient manner.

The current state of the art in sampling with underwater vehicles is primarily limited to the use of non-adaptive, preplanned sampling missions where the collected data is stored for offline retrieval and analysis. The sensor platforms have no capability to react to data received from their environmental sensors other than the navigation sensors which keep them on their preplanned courses.

C. Communications

It seems an obvious conclusion that in order for multiple sensors to coordinate their actions and share state information, they must be able to communicate. This is easier said than done in the ocean environment however. On land, RF or fiber optic communications systems are capable of transmitting information on the order of megabits per second or greater. In the ocean, where RF energy is unusable over any distance and acoustic data transmission must be used, data transmission rates are orders of magnitude lower due to the propagation constraints imposed by the ocean environment (in particular, the relatively slow phase speed of acoustic waves). This directly impacts the amount of information that can be shared in a marine network and the types of network connectivity that can be used. Since all sensor platforms must share the same acoustic channel, this may also limit the number of platforms that can be active.

The amount of bandwidth needed in a cooperative sensor network is related to the sampling requirements of the process under observation. Processes with high frequency content require correspondingly high bandwidth. For processes with low frequency content, bandwidth requirements may be traded off for an increased sampling period. This issue is complicated by the fact that the acoustic channel may also be used simultaneously by sensors and navigation systems. Along with sensor platform navigation, a robust communications capability is one of the two critical supporting technologies needed to implement an effective sensor network.

D. Cooperation

While coordinated marine vehicles have their advantages, they present challenges in their joint control to reach their combined potential. Inter-vehicle communication is limited in bandwidth and carefully allocated. Any kind of central continuous control is likely infeasible. In multi-vehicle joint exercises involved with sensing dynamic phenomena, it may not be practical or effective to think in terms of a single vehicle state space to which proper actions can be assigned a priori. In Section III-A, we describe a behavior-based control approach where a number of behaviors operating in parallel use the sensed environmental state data to maneuver the sensor platform. In this work, we use an approach to cooperation in which some of the behaviors on the sensor platforms are specifically designed to use state data from other sensor platforms in order to form a decision on preferred platform maneuvers. This state data is shared via the communications network. This is a form of highly decentralized cooperative control in which there is no central planner dictating actions to the sensor platforms. This in keeping with the spirit of behavior-based control in which there is a tight coupling between control and the perceived environment. This scheme has an obvious advantage with respect to network survivability in that any network with central planning is vulnerable to to the loss of the planner, whether that function resides on another sensor platform or on the surface. A network with decentralized control is more able to gracefully degrade with the loss of particular sensor nodes.

E. Sensor Fusion

Data fusion is the synergistic combination of information from different sources such as sensors in order to provide a better understanding of the state of the world [4]. In our marine sensor network application, sensor data from multiple, distributed sensor platforms must be combined. These sensors may be heterogeneous and may have different resolutions. For example, data from both range and bearing sensors may need to be combined in surveillance and target tracking applications. In the distributed target tracking example discussed in Section V, two independent bearing observations from distributed sensor platforms must be combined to estimate a target track. A significant issue in fusing data from multiple sources is in determining that distributed measurements correspond to the same environmental feature [4]. This is known as the data association problem. This issue arises for example in tracking applications where multiple targets may be present.

In order to properly fuse data from multiple sources and possibly heterogeneous sensors, accurate models of the process we want to observe and our sensor characteristics are imperative. Accurate process models allow us to derive the proper sensor platform behaviors given the state of the environment. These

models must view the process from a probabilistic standpoint. It is not good enough to provide an estimate of a process parameter without also providing a notion of the uncertainty associated with the estimate. This also allows accurate simulation of adaptive sensor platform operation. The uncertainty of our estimates is related to a number of factors but primarily on the uncertainty of our sensor measurements, the uncertainty of the spatial location where the measurement was taken, and the time the measurement was taken. The uncertainty of the sensor measurements can be dealt with by having an accurate sensor model. The uncertainty in the measurement time can be easily dealt with by precision time synchronization between the sensor platforms. A method for doing this is discussed in [5]. It should be clear then that sensor platform navigation is one of the critical supporting technologies required in a sensor network. At the very least, the navigation uncertainty must remain bounded over the time period the sensor platform is in operation. If the navigation uncertainty grows over time this will introduce a growing uncertainty in our sensor measurements and hence in our estimates. This is clearly undesirable.

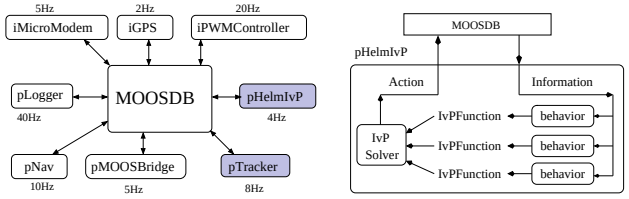
III. Technical Approach

In this section we present our general autonomy architecture for autonomous sensor platform control and how the particular components that reflect the contribution of this work fit into that architecture. The outline for experimental validation is also discussed.

A. The MOOS-IvP Autonomy Architecture

This work uses the MOOS-IvP architecture for autonomous marine vehicle control. MOOS-IvP is composed of the Mission Oriented Operating Suite (MOOS), a open source software project for coordinating software processes running on an autonomous platform, typically under GNU/Linux. MOOS-IvP also contains the IvP Helm, a behavior-based helm that runs as a single MOOS process and uses multi-objective optimization with the Interval Programming (IvP) model for behavior coordination, [6], [7]. See [8] and [9] for other examples of MOOS-IvP on autonomous marine vehicles.

A MOOS community contains processes that communicate through a database process called the MOOSDB, as shown in Fig. 1(a). MOOS ensures a process executes its “Iterate” method at a specified frequency and handles new mail on each iteration in a publish and subscribe manner. The IvP Helm runs as the MOOS process pHelmIvP (Fig. 1(b)). Each iteration of the helm contains the following steps: (1) mail is read from the MOOSDB, (2) information is updated for consumption by behaviors, (3) behaviors produce an objective function if applicable, (4) the objective functions are resolved to produce a single action, and (5) the action is posted to the MOOSDB for consumption by low-level control MOOS



(a) A MOOS Community (b) The pHelmIvP process

Fig. 1. The IvP Helm runs as a process called pHelmIvP in a MOOS community. MOOS may be composed of processes for data logging (pLogger), data fusion (pNav), actuation (iPWMController), sensing (iGPS), communication (pMOOSBridge, iMicroModem), and much more. They can all be run at different frequencies as shown.

processes. The behaviors responsible for sensor platform control in the tracking application are discussed in Section VI.

B. The Logical Sonar Sensor

As discussed in Section II, adaptive sensor platform motion requires tight integration between the sensors and the control system. However, the concept of a "logical sensor" [10] [11] [12] allows an abstract view of a sensor that allows the actual details of the physical sensor to be hidden or abstracted away in much the same way as an abstract data type does in software engineering [4]. This is especially useful if multiple physical sensors contribute to forming a piece of sensory information. The logical sonar sensor (see Fig. 2) consists of the physical acoustic sampling hardware as well as algorithms that abstract the real-time data into higher forms of information suitable for the behavior-based control system. Because of the distributed MOOS architecture, the actual sensor and processing algorithms (MOOS processes) of the logical sensor may well reside in a separate vehicle payload from the main vehicle control computer. The tracking vehicles in this work use a set of tracking algorithms that run in a single MOOS process called pTracker (see Fig. 1(a)). This process subscribes to target bearing data from the MOOS database as input to the tracking algorithms. The bearing data is either produced by another MOOS process interfaced with a physical bearings-only sensor, or the bearing data is produced by an alternative MOOS process that simulates bearings-only sensor data. The pTracker process then produces and posts track solution information to the MOOSDB to be consumed by any other MOOS process including inter-vehicle communications processes like pMOOSBridge or iAcousticModem or the behaviors in the vehicle control system. Feedback from the platform behaviors is available for dynamically changing the sensor parameters in response to the platform state. More information on the algorithms for the pTracking process is given in Section IV for single bearing tracking and in V for two bearing tracking.

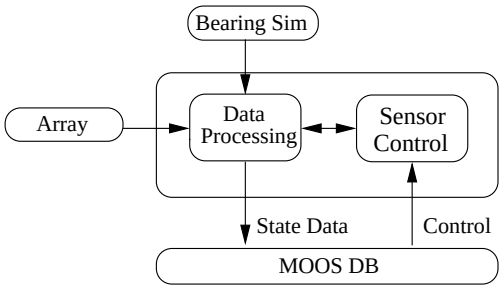


Fig. 2. The logical sonar sensor. Rather than passing raw acoustic data directly to the platform control system, the sensor processes the acoustic data into a higher level of abstraction suitable for a behavior-based control system. Feedback from the platform behaviors is available for dynamically changing the sensor parameters in response to the platform state.

C. Validation with Experimental Data

Experimental validation of this work is presented using three autonomous kayaks rather than actual underwater vehicles. This is largely due to the convenience of using lightweight surface craft as proxies to the larger AUVs which are more expensive and time-consuming to operate.

IV. Single Bearing Track and Classify

We are motivated by the following scenario: two heterogeneous vehicles are in operation, the first is fitted with a passive, bearings-only towed sensor array and takes on the role of tracking other moving underwater objects of unknown trajectory and type.

The second vehicle is fitted with a different sensor more appropriate for detecting acoustic signatures of underwater objects and takes on the role of classifying other underwater objects. The two vehicles work together to track and classify underwater objects by communicating track solution information from the tracking vehicle to the classify vehicle via acoustic modem. The latter vehicle uses the track information to close its position on the object of interest to the benefit of its classification sensors. Each vehicle optimizes its trajectory to balance their sensing responsibilities alongside mutual relative position responsibilities.In this chapter, we will first derive the mathematical basis for target tracking with a single mobile bearing sensor which will allow us to design the proper behaviors for the vehicle motion. Next, we will derive the target localization and tracking algorithms which reside on the intelligent sensor. Finally, we will present experimental validation of these concepts using three autonomous surface craft in Section VIII-B.

In order to track a moving object from a set of discrete sensor observations, one must first decide on the kinematic model used to describe the object's motion. In this work, a constant-velocity model was chosen because it is one of the simplest to describe mathematically and because estimating the motion of a constant velocity target using a bearings-only sensor is a classical problem in target motion analysis.

Also termed "passive localization" or "passive ranging" this problem arises, for example, when trying to estimate the motion of a submarine moving at constant velocity from another submarine observing the target using a linear towed array sensor.

A. State Estimator Derivation

In formulating this problem, we follow a classical analysis as given in [13]. Consider a Cartesian coordinate frame having an object with position $[x_t[n]\ y_t[n]]^T$ and constant velocity $[\dot{x_t}\ \dot{y_t}]^T$ being tracked by a bearing sensor on a sensor platform with position $[x_p[n]\ y_p[n]]^T$ moving in the same plane with measurement observations taken at the discrete time intervals $n = 0, 1, \ldots, N$. The state equations for the target motion can be written in discrete time as

$$x_t[n] = x_t[0] + \dot{x_t}t[n] \tag{1}$$

$$y_t[n] = y_t[0] + \dot{y_t}t[n] \tag{2}$$

Given (1) and (2) we define the state parameter vector

$$\hat{x_t} \triangleq [x_t[0]\ y_t[0]\ \dot{x_t}\ \dot{y_t}]^T \triangleq [x_0\ x_1\ x_2\ x_3]^T \tag{3}$$

All of the parameters in the state parameter vector are assumed to be statistically independent. The measurements are target bearings relative to the sensor platform given by

$$z[n] = h[n, x] + w[n] \tag{4}$$

where

$$h[n, x] \triangleq tan^{-1}\frac{y_t[n] - y_p[n]}{x_t[n] - x_p[n]} \tag{5}$$

and $w[n]$ is the measurement noise assumed to be a Gaussian white noise sequence with variance q. Our sensor makes a sequence of bearing measurements which we combine into a single measurement vector Z.

Given our assumption of a constant velocity target, estimating the parameters in (3) from a sequence of observations completely defines the target motion. A number of techniques are available to perform the parameter estimation. The maximum likelihood estimator was chosen in order to form the optimal estimate (in a least-squares sense).

B. The Likelihood Function

Given the Gaussian noise assumption for our measurement, we define the negative log-likelihood function as

$$\lambda(x) \triangleq \frac{1}{2q} \sum_{n=1}^{N} [z[n] - h[n, x]]^2 \tag{6}$$

The maximum likelihood estimate is then formed by

$$\hat{x} = arg \min_{x} \lambda(x) \tag{7}$$

The state parameter vector which satisfies (7) is the *maximum likelihood estimate*. The minimization required to satisfy (7) was accomplished using the Broyden-Fletcher-Goldfarb-Shanno algorithm, a quasi-Newton method requiring the first derivatives of (6) with respect to the state parameters. The derivatives (with irrelevant constants removed) necessary for the minimization are

$$\frac{\partial \lambda(x)}{\partial x_0} = \sum_{n=1}^{N} \frac{2(z[n] - h[x, n])(y_t[n, x] - y_p[n])}{(x_t[n, x] - x_p[n])^2 + (y_t[n, x] - y_p[n])^2} \tag{8}$$

$$\frac{\partial \lambda(x)}{\partial x_1} = \sum_{n=1}^{N} \frac{-2(z[n] - h[x, n])(x_t[n, x] - x_p[n])}{(x_t[n, x] - x_p[n])^2 + (y_t[n, x] - y_p[n])^2} \tag{9}$$

$$\frac{\partial \lambda(x)}{\partial x_2} = \sum_{n=1}^{N} -t[n] \frac{\partial \lambda(x)}{\partial x_0} \tag{10}$$

$$\frac{\partial \lambda(x)}{\partial x_3} = \sum_{n=1}^{N} -t[n] \frac{\partial \lambda(x)}{\partial x_1} \tag{11}$$

C. The Cramer-Rao Lower Bound

The Cramer-Rao lower bound (CRLB) stipulates that the variance of our parameter estimates cannot be lower (on average) than a certain value determined by the shape of the likelihood function. The derivation and proof of the CRLB can be found in a number of textbooks on estimation theory including [13]. Formally, we say that

$$E\left[(\hat{x}(Z) - x)^2\right] \geq I_y(x)^{-1} \tag{12}$$

where $I_y(x)$ is known as the Fisher information matrix (FIM). The elements of the FIM are measures of the amount of "information" available about each parameter. Given our measurement vector Z and the Gaussian noise assumption, the diagonal elements of the FIM for this problem are

$$h_{x0}[n, x] = \sum_{n=1}^{N} \left[\frac{-(y_t[n] - y_p[n])}{(x_t[n] - x_p[n])^2 + (y_t[n] - y_p[n])^2} \right]^2 \tag{13}$$

$$h_{x1}[n,x] = \sum_{n=1}^{N} \left[\frac{(x_t[n] - x_p[n])}{(x_t[n] - x_p[n])^2 + (y_t[n] - y_p[n])^2} \right]^2 \tag{14}$$

$$h_{x2}[n,x] = \sum_{n=1}^{N} [t[n] h_{x0}]^2 \tag{15}$$

$$h_{x3}[n,x] = \sum_{n=1}^{N} [t[n] h_{x1}]^2 \tag{16}$$

In this analysis, we won't consider the cross-correlations in the FIM as significant in terms of how we approach the analysis of the preferred sensor platform motion. The Cramer-Rao lower bound on the variance of each of our parameters is then found by inverting the FIM. By examining the elements of the FIM, several important issues can be noted. First, it is readily apparent that the number of observations N in our observation vector Z is a critical parameter determining the variance of our parameter estimates. Second, it is also apparent that the relative positions of the sensor and target over time also play a critical role as is explored in section IV-D.

D. Parameter Observability

A well known constraint in tracking a constant-velocity target from a moving sensor platform is that, if the sensor platform also moves with constant velocity, the target motion parameters are unobservable. Therefore, the sensor platform must undergo an acceleration with respect to the target. A simple change of course can satisfy this condition. The degree to which the sensor motion improves the observability and, hence, the variance of the parameter estimates can be quantified by the condition number J of the FIM [13]. If J is too large, the FIM is ill-conditioned and the parameters are unobservable. Even if the FIM is invertible, the parameters may be marginally observable depending on the actual value of J. The vehicle behaviors described in section VI are designed to produce a well-conditioned FIM.

V. Two Bearing Cooperative Tracking

We are motivated by the following scenario: two networked sensor vehicles are in operation, both fitted with passive, towed, acoustic sensor arrays. Both vehicles will detect and cooperatively track moving targets of unknown trajectory and type. Both vehicles begin in patrol mode in separate portions of the operating area in order to optimize their sensor coverage. The two vehicles work together to track underwater objects by communicating target bearing and track estimate information between themselves via acoustic modem. The vehicles will then position themselves with respect to the target in a track and trail formation designed to minimize the uncertainty in the target track estimate.

In this chapter, we will follow a similar technical approach to that followed in Section IV for target tracking with a single sensor platform. In typical passive ranging applications, however, the state parameters for the target track are estimated using a set of observations from a single moving sensor platform. With only one sensor, both temporal and spatial diversity in the sensor measurements are needed to estimate the target track. In this work, we will estimate the target track parameters using simultaneous measurements from two spatially distributed sensors from which an immediate solution of the target position can be formed. Successive position estimates will then be used to estimate the target's velocity components. By comparing the tracking results obtained with two distributed sensor platforms with those obtained in Section IV using a single sensor platform, it will be clear that spatially distributed sensors have the potential to offer significant advantages.

A. 2D Target Position Triangulation

Triangulating the position of an object using passive angle measurements is common in a number of fields including optics. Most analysis, however, assumes fixed sensors triangulating fixed or moving targets or moving sensors estimating the position of a fixed target [14]. In this work we now consider the position estimation for a moving target from a moving sensor platform. In this section, we will follow the analysis as developed in [14] for the 2D target position estimation and the subsequent error analysis. Given

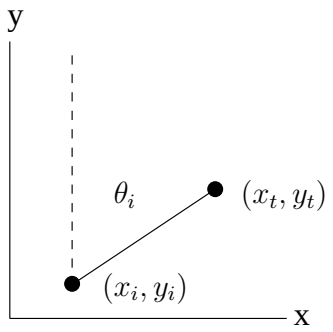


Fig. 3. Coordinate frame for 2D multi-sensor tracking.

the coordinate frame shown in Fig. 3 with target location $(x_t[n], y_t[n])$ and sensor positions $(x_i[n], y_i[n])$ for the discrete time interval $n = 0, 1, \ldots, N$, the relationship between the position of the i^{th} sensor and its measured target bearing θ_i at time n is given by

$$\tan\theta_i[n] = \frac{x_t[n] - x_i[n]}{y_t[n] - y_i[n]} \tag{17}$$

The solution to (17) for the general case of I sensors can be written in matrix form as

$$\begin{bmatrix} \cdot \\ x_i[n] - y_i[n]\tan\theta_i \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ 1 & -\tan\theta_i[n] \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \hat{x}_t[n] \\ \hat{y}_t[n] \end{bmatrix} \tag{18}$$

This system of nonlinear equations can be solved using general least-squares methods such as Gauss-Newton and Levenberg-Marquardt. For the problem under consideration in this work, we limit ourself to the case of two sensors for which the exact solution at any time step n can be be written as

$$\hat{x}_t = \frac{x_2\tan\theta_1 - x_1\tan\theta_2 + (y_1 - y_2)\tan\theta_1\tan\theta_2}{\tan\theta_1 - \tan\theta_2} \tag{19}$$

$$\hat{y}_t = \frac{y_1\tan\theta_1 - y_2\tan\theta_2 + x_2 - x_1}{\tan\theta_1 - \tan\theta_2} \tag{20}$$

B. Variance of the Target Position Estimate

One of the most important pieces of information needed to develop the proper behaviors for a sensor-adaptive system is the relationship between the target motion and the variance of the parameter estimates for the process under observation. From (19) and (20) it is apparent that the uncertainty in the target position estimates will be influenced by three factors:

1) The uncertainty of the sensor positions $(x_i[n], y_i[n])$
2) The uncertainty of the bearing measurements $\theta_i[n]$
3) The positions of the sensors with respect to the target

The sensor position uncertainties we model as Gaussian distributions with variance σ^2_{pos} equal and uncorrelated in both the x and y directions. The bearing measurement uncertainties we also model as Gaussian distributions with variance σ^2_θ equal and independent of sensor platform. The usual method for finding the variances of (19) and (20) would be to take the expectation

$$var(\hat{x}) = E[(\hat{x} - x)^2)] \tag{21}$$

Given the complexity of the functional forms for (19) and (20) however, no closed form solution for (21) can be calculated. In this case, one can derive the error propagation equations by performing Taylor series expansions of (19) and (20) as given in detail for this application in [14]. Using the above assumptions with regards to the uncertainties for sensor position and bearing measurements, a first-order

approximation to the target position uncertainties can be given as

$$\sigma_{x_t}^2 \approx C_1\sigma_{pos}^2 + C_2\sigma_\theta^2 \tag{22}$$

$$\sigma_{y_t}^2 \approx C_3\sigma_{pos}^2 + C_4\sigma_\theta^2 \tag{23}$$

where C_1, C_2, C_3, and C_4 are coefficients given as

$$C_1 = \left(\frac{\partial x_t}{\partial x_1}\right)^2 + \left(\frac{\partial x_t}{\partial x_2}\right)^2 + \left(\frac{\partial x_t}{\partial y_1}\right)^2 + \left(\frac{\partial x_t}{\partial y_2}\right)^2 \tag{24}$$

$$C_2 = \left(\frac{\partial x_t}{\partial \theta_1}\right)^2 + \left(\frac{\partial x_t}{\partial \theta_2}\right)^2 \tag{25}$$

$$C_3 = \left(\frac{\partial y_t}{\partial x_1}\right)^2 + \left(\frac{\partial y_t}{\partial x_2}\right)^2 + \left(\frac{\partial y_t}{\partial y_1}\right)^2 + \left(\frac{\partial y_t}{\partial y_2}\right)^2 \tag{26}$$

$$C_4 = \left(\frac{\partial y_t}{\partial \theta_1}\right)^2 + \left(\frac{\partial y_t}{\partial \theta_2}\right)^2 \tag{27}$$

The derivatives needed to calculate (24) through (27) are derived in [14]. Coefficients C_1 and C_3 measure the contribution of the sensor position error to the target location error while coefficients C_2 and C_4 measure the contribution of the bearing measurement error to the target location error. Coefficients C_1 and C_2 are plotted in Figures (4) and (5). Plots for coefficients C_3 and C_4 (not shown) are similar with a 90 degree rotation. Fig. (6) is a plot of coefficient C_2 with a sensor to target range of 20 meters versus the range of 10 meters used in Fig. (5). From an analysis of these plots, the following observations can be made with regard to the effect of sensor platform motion on the variance of the target position estimates:

1) The largest influence on $\sigma_{x_t}^2$ and $\sigma_{y_t}^2$ is the sensor separation angle $(\theta_1 - \theta_2)$ with minimum variance at a separation angle of 90 degrees rising to infinity at separation angles of 0 degrees and 180 degrees.
2) The influence of the bearing measurement error rises linearly with the sensor to target range. The bearing measurement error will also rise with the sensor to target range due to the reduction in the received signal to noise ratio when using a real acoustic array.
3) The 90 degree rotation between the plots of the coefficients for the variances of $\hat{x}_t$ and $\hat{y}_t$ indicate that uncertainty in one spatial direction can be minimized with a corresponding increase in uncertainty in the other spatial direction.

These observations will be used in Section VI to develop the autonomous vehicle behaviors designed to cooperatively track a moving target with two sensor platforms with a goal of minimizing the target localization errors subject to other constraints on the platform motion.

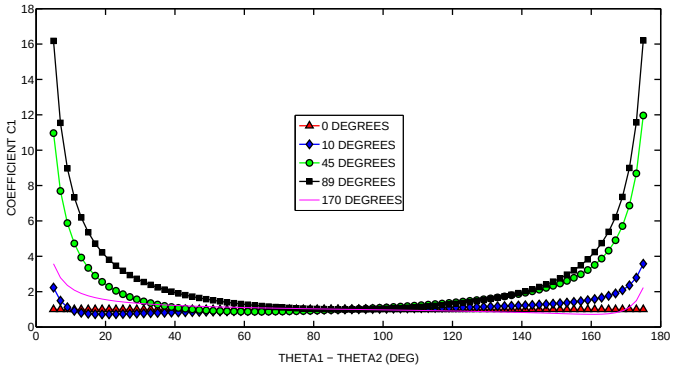


Fig. 4. Coefficient C_1. This plot shows shows coefficient C_1 in (24) as a function of θ_1 and $(\theta_1 - \theta_2)$. It is clearly seen that C_1 is minimized for $(\theta_1 - \theta_2) = 90$ degrees.

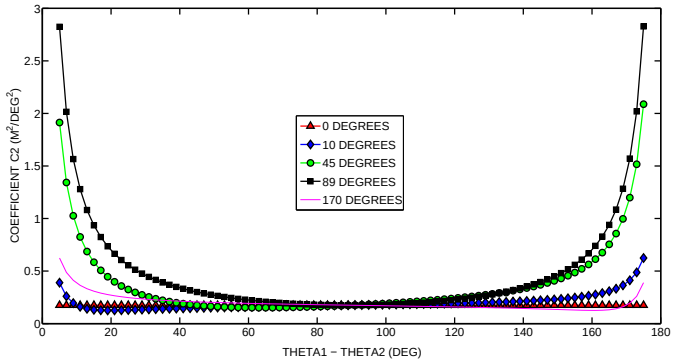


Fig. 5. Coefficient C_2 $(10m)$. This plot shows shows coefficient C_2 in (25) as a function of θ_1 and $(\theta_1 - \theta_2)$ for a sensor to target range of 10 meters. It is clearly seen that C_2 is minimized for $(\theta_1 - \theta_2) = 90$ degrees.

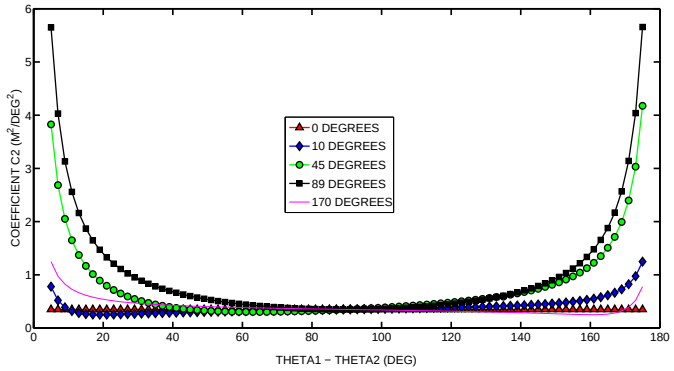


Fig. 6. Coefficient C_2 $(20m)$. This plot shows shows coefficient C_2 in (25) as a function of θ_1 and $(\theta_1 - \theta_2)$ for a sensor to target range of 20 meters. By comparison with Fig. 5, it is clearly seen that C_2 is linearly dependent on the sensor to target range.

C. Target Velocity Component Estimation

Having derived the necessary analysis to be able to estimate the instantaneous position of a target from two simultaneous bearing measurements, we would like to filter these noisy measurements as well as estimate the target's velocity components from successive position estimates. A number of techniques are

available to do this but the extended Kalman filter was chosen for its speed, with available CPU cycles being limited on small, autonomous platforms. Even though this is a non-optimal estimation technique, good performance was obtained as shown in Section VIII-B. We start by modeling the target motion with the discrete time state equation

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \tag{28}$$

where $\mathbf{x}_k$ is the state vector for the target motion given by

$$\mathbf{x}_k \triangleq [\dot{x}_t \;\; x_t \;\; \dot{y}_t \;\; y_t]_k^T \tag{29}$$

with

$$x_{tk} = x_{tk-1} + \dot{x}_t \, dt \qquad y_{tk} = y_{tk-1} + \dot{y}_t \, dt \tag{30}$$

and $\mathbf{w}_k$ the process noise vector given as $[qx \;\; 0 \;\; qy \;\; 0]^T$ where qx and qy are independent and equally distributed, zero mean, Gaussian random variables. Given the assumption of a constant velocity target, the state transition matrix $\mathbf{F}_k$ is computed as the Jacobian of (29)

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ dt & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & dt & 1 \end{bmatrix} \tag{31}$$

$\mathbf{Q}_k$ is the covariance matrix of the process noise $\mathbf{w}_k$ given by

$$\mathbf{Q}_k = E\{\mathbf{w}_k^2\} = E\left[\int_0^{dt} \mathbf{F}_k \mathbf{w}_k \, d(\epsilon) \int_0^{dt} (\mathbf{F}_k \mathbf{w}_k)^T \, d(\eta)\right] \tag{32}$$

resulting in

$$\mathbf{Q}_k = qq \begin{bmatrix} dt^2 & \frac{dt^3}{2} & 0 & 0 \\ \frac{dt^3}{2} & \frac{dt^4}{4} & 0 & 0 \\ 0 & 0 & dt^2 & \frac{dt^3}{2} \\ 0 & 0 & \frac{dt^3}{2} & \frac{dt^4}{4} \end{bmatrix} \tag{33}$$

where qq is the variance of the process noise. At each time step k, an observation $\mathbf{z}_k$ of $\mathbf{x}_k$ is made according to

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{34}$$

where $\mathbf{H}_k$ is the Jacobian of the observation model

$$\mathbf{h} = \left[atan\left(\frac{y_t - y_1}{x_t - x_1}\right) atan\left(\frac{y_t - y_2}{x_t - x_2}\right) \right]^T \tag{35}$$

and $\mathbf{v}_k$ is the measurement noise vector given as $r[1\ 1]^T$ with r a zero mean normally distributed random variable. In this application, $\mathbf{z}_k$ corresponds to a pair of simultaneous bearing measurements $\mathbf{z}_k = [z_1\ z_2]_k^T$ and the resulting matrix $\mathbf{H}_k$ is

$$\mathbf{H}_k = \begin{bmatrix} 0 & \dfrac{-(y_t - y_1)}{d_1} & 0 & \dfrac{(x_t - x_1)}{d_1} \\ 0 & \dfrac{-(y_t - y_2)}{d_2} & 0 & \dfrac{(x_t - x_2)}{d_2} \end{bmatrix} \tag{36}$$

where d_i is the squared distance from sensor i to the target position estimate given by $(y_t - y_i)^2 + (x_t - x_i)^2$. We consider the measurement noise of the bearing measurements from each sensor platform to be equal and independent of platform, therefore the covariance matrix of the measurement noise $\mathbf{v}_k$ is given as

$$\mathbf{R}_k = rr \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{37}$$

where rr is the variance of our bearing measurements. Given these definitions of our estimation model, our estimation proceeds in classical fashion in two steps. In the first step, we calculate the predicted state $\mathbf{x}_{k|k-1}$ and the predicted state covariance $\mathbf{P}_{k|k-1}$ for the current time step k given the information from the previous time step $k-1$ as follows:

$$\mathbf{x}_{k|k-1} = \mathbf{F}_k \mathbf{x}_k \tag{38}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \tag{39}$$

In the second step we refine this prediction using our observations. We proceed by first calculating the measurement residual $\tilde{\mathbf{y}}_k$ and the covariance residual $\mathbf{S}_k$ as

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{h}_k \tag{40}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{41}$$

The Kalman gain is then computed as

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{42}$$

The Kalman gain is then used with the measurement residual to update the current state estimate $\mathbf{x}_{k|k}$ and the state covariance matrix $\mathbf{P}_{k|k}$ as follows

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \tag{43}$$

$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{44}$$

VI. The IvP Helm and Vehicle Behaviors

Here we describe the use of multi-objective optimization with interval programming and the primary behaviors used in this experiment for the one and two bearing tracking experiments. For further examples of this approach, although with different missions and behaviors, see [8], [9].

A. Behavior-Based Control with Interval Programming

By using multi-objective optimization in action selection, behaviors produce an *objective function* rather than a single preferred action ([6], [15], [16]). The IvP model specifies both a scheme for representing functions of unlimited form as well as a set of algorithms for finding the globally optimal solution. All functions are piecewise linearly defined, thus they are typically an *approximation* of a behavior's true underlying utility function. Search is over the weighted sum of individual functions and uses branch and bound to search through the combination space of pieces rather than the decision space of actions. The only error introduced is in the discrepancy between a behavior's true underlying utility function and the piecewise approximation produced to the solver. This error is preferable compared with restricting the function form of behavior output to say linear or quadratic functions. Furthermore, the search is much faster than brute force evaluation of the decision space, as done in [16]. The decision regarding function approximation accuracy is a local decision to the behavior designer, who typically has insight into what is sufficient. The solver guarantees a globally optimal solution and this work validates that such search is feasible in a vehicle control loop of 4Hz on a 600MHz computer.

To enhance search speed, the initial decision provided to the branch and bound algorithm is the output of the previous cycle, since typically the optimal prior action remains an excellent candidate in the present, until something changes in the world. Indeed when something *does* change dramatically in the world, such as hitting a way-point, the solve time has been observed to be up to 50% longer, but still comfortably under practical constraints.

Although the use of objective functions is designed to coordinate multiple simultaneously active behaviors, helm behaviors can be easily conditioned on variable-value pairs in the MOOS database

to run at the exclusion of other behaviors. Likewise, behaviors can produce variable-value pairs upon reaching a conclusion or milestone of significance to the behavior. In this way, a set of behaviors could be run in a plan-like sequence, or run in a layered relationship as originally described in [17].

B. The OpRegion Behavior

The OpRegion behavior is a safety behavior responsible for insuring the sensor platform remains in a predetermined safe operating area. The behavior is congured with a single polygon and will result in an all-stop signal (THRUST=0) to the low level controllers if the vehicle leaves the operation area. The OpRegion behavior does not produce an objective function. It just informs the helm that there is a critical condition that should trump all behaviors and produce the action of all-stop. In this sense, the OpRegion behavior is not unlike a behavior in Brooks' layered approach [2], where a critically important module can trump all others without seeking compromise.

C. The Waypoint Behavior

The waypoint behavior is responsible for moving the sensor platform from one point to another along the shortest path. The behavior is configured with a list of waypoints and produces objective functions that favorably rank actions with smaller detour distances along the shortest path to the next waypoint. This behavior is used by the target vehicle in the experiments to form a constant velocity motion, for example, and multiple waypoints can be sequenced together to form platform motion along arbitrary polygons. Every vehicle is typically configured with an instance of this behavior having a single waypoint just off the starting area, conditioned on both a "mission=complete" or "return=true" condition for returning all vehicles upon mission completion or recalling them mid-mission should the need occur. The objective function for this behavior is three-dimensional over course, speed, and time.

D. The Orbit Behavior

The Orbit behavior is responsible for providing a patrol capability in which the vehicle will orbit a fixed point. Given an orbit center, the behavior dynamically determines a list of waypoints to form the orbit. Parameters to this behavior allow the choice of clockwise/counter-clockwise orbits as well as the number of waypoints in the orbit path and the vehicle speed. The objective functions for this behavior are identical to the standard waypoint objective functions described in Section VI-C. The Orbit behavior can be conditioned to be active when no target is being tracked and to deactivate itself upon target detection.The Orbit behavior always has a weighting of 1.0.

E. The ArrayTurn Behavior

The ArrayTurn behavior is responsible for providing a vehicle turning motion such that sensor platforms with acoustic line arrays can determine which side of the array the target is on. This behavior requires tight integration with the acoustic sensor which signals when the left/right ambiguity has been cleared. The objective function for this behavior is one-dimensional over course and bimodal, with the modes centered around the two possible course choices which are ninety degrees from the vehicle's course when the behavior is activated (he course fix). The mode that is centered at the course closest to the vehicle's current course is weighted in order to prevent frequent oscillation between the two modes. Fig. 7 shows a plot of the objective function for this behavior for a course fix of zero degrees and a current course of five degrees. Note how the mode closest to the current course is weighted slightly higher. Fig. 8 shows a plot of the objective function for the ArrayTurn behavior for a course fix of zero degrees and a current course of fifty degrees. Note how the mode closest to the current course has increased its weight relative to the other mode for the situation shown in Fig 7. The ArrayTurn behavior has a constant weighting of 1.0.

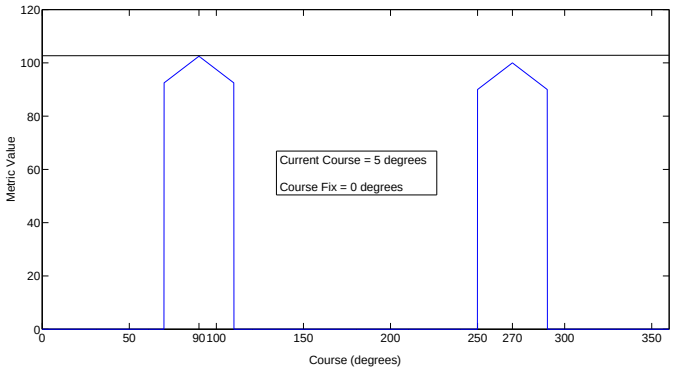


Fig. 7. Objective function for the ArrayTurn behavior. This figure shows a plot of the objective function for the ArrayTurn behavior for a course fix of zero degrees and a current course of five degrees. Note how the mode closest to the current course is weighted slightly higher.

F. ArrayAngle Behavior

The ArrayAngle behavior is responsible for holding a vehicle course such that sensor platforms with acoustic line arrays will have the array as close as possible to broadside with the target given the other constraints on vehicle motion. The objective function for this behavior is one-dimensional over course and bimodal, with the modes centered around the two possible course choices that keep the array oriented at broadside with respect to the target. The mode that is centered at the course closest to the vehicle's current course is weighted in order to prevent frequent oscillation between the two modes. Fig 9 shows

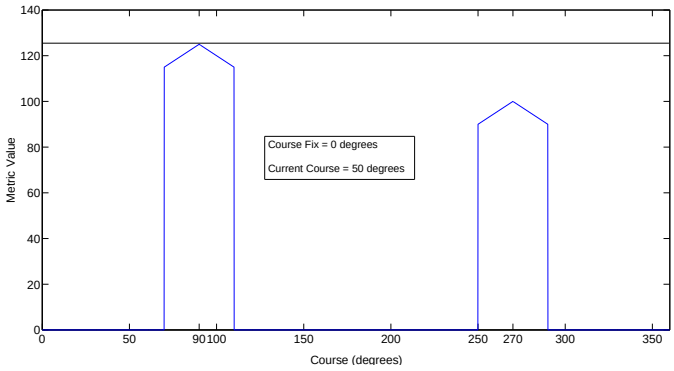


Fig. 8. Objective function for the ArrayTurn behavior. This figure shows a plot of the objective function for the ArrayTurn behavior for a course fix of zero degrees and a current course of fifty degrees. Note how the mode closest to the current course has increased its weight relative to the other mode for the situation shown in Fig 7.

a plot of the objective function for the ArrayAngle behavior for a target bearing of zero degrees and a current course of fifty degrees. Note how the mode closest to the current course is weighted slightly higher. Fig 10 shows a plot of the objective function for the ArrayAngle behavior for a target bearing of zero degrees and a current course of minus fifty degrees. Note how the mode closest to the current course has increased its weight relative to the other mode for the situation shown in Fig 9. Beyond a specified maximum range, the weighting of the ArrayAngle behavior is 0.1 otherwise it is weighted at 1.0.

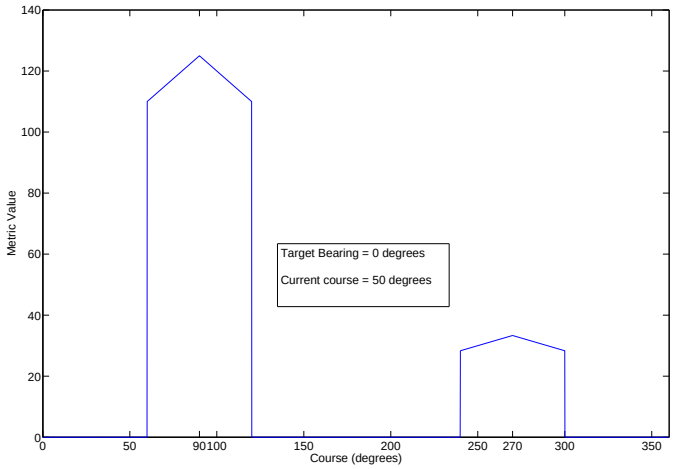


Fig. 9. Objective function for the ArrayAngle behavior. This figure shows a plot of the objective function for the ArrayAngle behavior for a target bearing of zero degrees and a current course of fifty degrees. Note how the mode closest to the current course is weighted slightly higher.

G. CloseRange Behavior

The CloseRange behavior is designed to close the distance to a target being tracked by the on board sensor subject to a minimum approach distance. The behavior produces objective functions that are three-

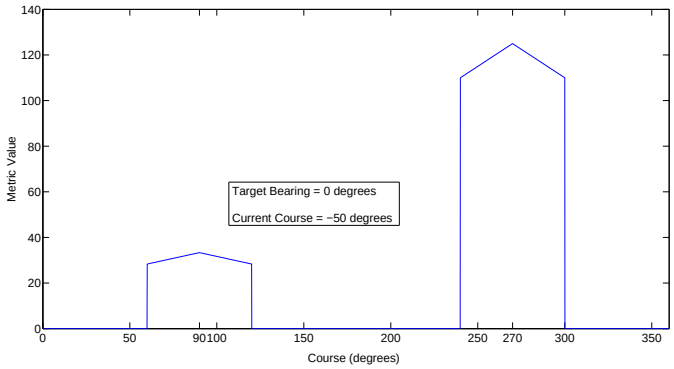


Fig. 10. Objective function for the ArrayAngle behavior. This figure shows a plot of the objective function for the ArrayAngle behavior for a target bearing of zero degrees and a current course of minus fifty degrees. Note how the mode closest to the current course has increased its weight relative to the other mode for the situation shown in Fig 9.

dimensional over course, speed, and time and rates actions favorably that have a smaller closest point of approach (CPA).

H. Classify Behavior

The Classify behavior used in this demonstration is active on the classify vehicle and is identical to the CloseRange behavior described in VI-G with the exception that the target track information is provided from an external source (in this case the tracking vehicle), instead of an on board sensor.

I. Formation Behavior

The formation behavior (see Fig. 11 is responsible for maintaining two sensor platforms in formation in a track and trail scenario behind the target using the current target position estimate as a virtual leader. The optimal formation consists of the sensor platforms maintaining a ninety degree angle with respect to the target position estimate while trailing at a fixed trail distance r. The objective functions for this behavior are three dimensional over course, speed and time. shows a plot of the metric applied to a proposed combination of course, speed, and time, that results in a value for the separation angle between this sensor platform and its partner sensor platform. It should be noted that the separation is computed using the current position of the other sensor platform which is also calculating the separation angle. This can lead to dynamic instability problems if there is not enough damping in the vehicle motion. The formation behavior always has a weighting of 1.0.

VII. Experimental Scenario and Configuration

Experimental validation of the architecture and algorithms for cooperative sensor platform control in the sensor-adaptive tracking application was conducted using two autonomous kayaks as mobile

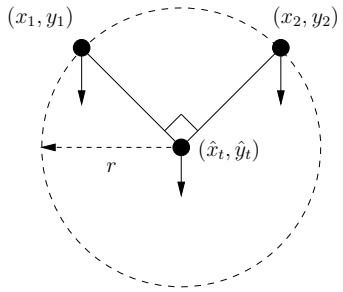


Fig. 11. Formation behavior for 2-vehicle cooperative target tracking. The formation behavior is responsible for maintaining two sensor platforms in formation in a track and trail scenario behind the target using the current target position estimate as a virtual leader. The optimal formation consists of the sensor platforms maintaining a ninety degree angle with respect to the target position estimate while trailing at a fixed trail distance r.

sensor platforms and a third kayak acting as a moving object to be tracked. The kayaks are proxies for autonomous underwater vehicles (AUVs) used in upcoming follow-on experiments. Except for the physical acoustic sensor, the control architecture, algorithms, and software are identical between the autonomous surface craft and the AUVs. The use of the autonomous surface craft allows a much more productive experimental approach versus using the very resource-intensive AUVs and almost all of the tracking and behavior-based control algorithms can be tested except those dealing directly with the physical sensor. The experiments were conducted using a test range available on the Charles River near MIT.

A. Simplifying Assumptions

Two significant simplifying assumptions were made. First, as a proxy for the towed acoustic array sensor, the GPS position of the sensed vehicle was communicated over an 802.11b wireless connection to the sensor vehicles. The sensor vehicles converted (diminished) this information into bearings-only sensor data using a simulator which provided bearing data to the MOOS database just as the intelligent sensor currently in use on the AUVs would do. Although a bearing simulator of this nature does not have the same characteristics as a real acoustic array, the performance is acceptable within the ranges used in this experiment. The second simplification was the use of the 802.11b wireless connection as a proxy for communications via acoustic modem between the sensor vehicles. Given that acoustic communications is much slower than the wireless system used in this experiment, the simplification allowed the compression of the experiment in time in order to fit within the allowed physical boundaries of the test range.

B. The Marine Vehicle Platforms

The autonomous surface crafts used in this experiment are based on a kayak platform (Fig. 12). Each is equipped with a Garmin 18 GPS unit providing position and trajectory updates at 1 Hz. The vehicles are

also equipped with a compass but the GPS provides more accurate heading information, and is preferred, at speeds greater than 0.2 m/s. Each vehicle is powered by 5 lead-acid batteries and a Minn Kota motor

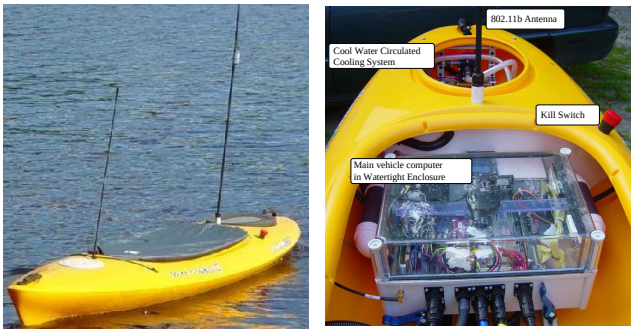


Fig. 12. The kayak-based autonomous surface craft.

providing both propulsion and steerage. The vehicles have a top speed of roughly 2.5 meters per second. See [18] for more details on this platform. Each kayak is equipped with the distributed MOOS architecture and IvP Helm as described in Section III-A.

C. Scenario

The two experimental scenarios begin with the deployment of two sensor vehicles into separate patrol orbits where they will remain until a target detection occurs. At some point, the target kayak will begin its motion into the target area. When it enters into the target area (Fig. 13 and Fig. 16), it will begin broadcasting its GPS location to the sensor vehicles whose sensor simulators will convert the position information into target bearings. In the single bearing scenario, the sensor vehicle will use the bearing information to compute the target track. This track will then be used by the sensor vehicle to maneuver as well as broadcast to the classification vehicle which will maneuver toward the target. In the two bearing scenario, sensor vehicle two's bearing data will then be transmitted to vehicle one where it will be combined with vehicle one's bearing information to form the target track. The target track information will then be broadcast back to vehicle two and both vehicles will use the track information to position themselves with respect to the target using the formation described in Section VI-I. After a predetermined amount of tracking time, tracking will be declared over and the sensor vehicles will return to their patrol orbits to await another target. The target vehicle will return to its starting location.

D. Behavior Configurations

The sensor vehicles were configured with the following behaviors and preconditions. A condition is a "variable=value" pair in the MOOS Database. A mission is started by broadcasting "deploy=true" to

all vehicles and ended when the "return=true" message is broadcast. A broadcast is over 802.11b and changes a particular MOOS variable in the database resident on the vehicle. The broadcast could also be made via acoustic modem. All vehicle helms were configured with the OpRegion behavior as a safety measure. This behavior is active upon mission startup indicated by "deploy=true".

In the one bearing scenario, the helms on the sensor vehicles were configured with Orbit behaviors which are active immediately upon mission startup indicated by "deploy=true". The Orbit behavior is conditioned on not receiving bearing sensor data, i.e., "sensor_data=inactive". It was also configured with the ArrayTurn, ArrayAngle, and CloseRange behaviors described in Section VI. These three behaviors are conditioned on the vehicle receiving bearings-only sensor data, indicated by "sensor_data=active" in the MOOS Database. The classification platform was configured with a Classify behavior which is identical in nature to the CloseRange behavior. In the two bearing scenario, the helms were configured with the Orbit, ArrayTurn, ArrayAngle, and Formation behaviors.

The target vehicle was configured to follow a simple set of waypoints. Deployment of the target vehicle was done via human command over wireless link when the other two vehicles had been on-station for an arbitrarily sufficient time.

E. Kalman Filter Initialization

The Kalman filter must be initialized before use. Before the first measurement is processed in the two bearing scenario, the state covariance matrix P_0 , the measurement noise variance rr and the process noise variance qq must be initialized. In all missions described in this work, P_0, rr, and qq were initialized to the following values:

$$\mathbf{P}_0 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 20,000 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 20,000 \end{bmatrix} \tag{45}$$

$$rr = 0.01\ rad^2 \qquad qq = 0.002\ m^2 \tag{46}$$

The performance of the Kalman filter will of course depend on the relative difference between the given values for the process and measurement noise and the actual values.

VIII. Experimental Results

A. One Bearing Track and Classify Results

Fig. 13 shows the vehicle motion for an experimental "track and classify" mission with autonomous kayaks (see Fig. 12) with one tracking kayak, one classify kayak, and one target kayak. The objective of this mission is for the tracking vehicle to acquire and track the target vehicle while relaying target track solutions to the classify vehicle which then executes a simulated classification run.

In (a) the track vehicle and classify vehicle are deployed and executing their Orbit behavior to loiter in two separate regions. In (b) the target vehicle is deployed and has just entered the "sensor region" where it begins to transmit its position data to the track vehicle. The track vehicle has just activated its ArrayTurn behavior for determining which side of the sensor array the target is on. In (c) the track vehicle has just sufficiently resolved the left-right ambiguity and has begun transmitting track solutions to the classify vehicle. The classify vehicle has begun its CloseRange behavior to facilitate classification of the target. The track vehicle has activated its CloseRange and ArrayAngle behaviors. In (d) both the track and classify vehicle are dominated by CloseRange behaviors to the target. In (e), the classify vehicle has performed the classification of the target and both vehicles are returning a back to their loiter regions. In (f) both vehicles are back on-station and awaiting any further unknown objects or vehicles to come through its sensor field. The target vehicle has returned to the dock.

Fig. 14 depicts the target position estimates produced by the MOOS process pTracker overlaid onto the actual target track. It is readily seen in the figure that the initial estimates were poor due to a small value for N as discussed in section IV-D. As the number of observations increases, a convergence of the estimate near to the actual track can be seen. Of special note is the large increase in convergence labeled "Vehicle Turn" in the figure. This is the point at which the sensor vehicle's CloseRange behavior became active and made a sharp course change between the positions shown in Fig. 4(c) and 4(d). Some increasing error can be seen in the estimates near the end of the experiment for two primary reasons. First, this highlights the difficulty in trying to use a single bearings-only sensor to track a target of nearly the same or faster speed. In this configuration, the target is ahead of and moving away from the sensor and it is difficult to position the sensor to produce a better FIM as discussed in section IV-D. Second, this error is due to a need to further optimize the vehicle behavior parameters to produce a better FIM.

B. Two Bearing Cooperative Tracking Results

Fig. 16 shows the vehicle motion for an experimental tracking mission with autonomous kayaks (see Fig. 12) with two tracking vehicles and one target vehicle. The objective of this mission is to execute the

scenario described in Section V where two sensor vehicles cooperatively track the target vehicle. This mission took place in the Charles River test range on December 1st, 2005. In (a) two tracking vehicles are deployed and executing their Orbit behaviors to patrol in two separate regions. Note that tracking vehicle two is exhibiting signs of a rudder control problem. In (b) the target vehicle is deployed and has just entered the "sensor region" where it begins to transmit its position data to the tracking vehicles for use in the bearing simulators. The tracking vehicles have just activated their ArrayTurn behaviors for determining which side of the sensor array the target is on. In (c) the tracking vehicles have just sufficiently resolved the left-right ambiguity and have begun executing their Formation behaviors using the target position estimate as a virtual leader. In (d) both the tracking vehicles have moved into formation behind the target. In (e), the target unexpectedly turned for home before the sensor vehicles have finished tracking, violating the constant velocity assumption and confusing the tracking system. In (f) both vehicles are back on-station and awaiting any further unknown objects or vehicles to come through its sensor field. The target vehicle has returned to the dock.

Fig. 17 depicts the target position estimates produced by the MOOS process pTracker overlaid onto the actual target track for the period in which the target vehicle was operating in a constant velocity scenario. As can be seen, excellent position estimates were obtained, especially compared with the tracking results obtained using a single sensor platform to track a constant velocity target. The gaps in the estimates as seen in the figure were due to communications breaks when no bearing estimates from vehicle two were received by vehicle one. Fig. 18 shows the error in the target position estimate as a function of mission run time. As can be seen, even with the communications breaks, position estimation results were generally very good.

IX. Conclusions

In this work we have demonstrated a method for sensor-adaptive control of autonomous marine vehicles in an autonomous oceanographic sampling network and shown its suitability for controlling multiple, cooperating heterogeneous sensor platforms. We have also demonstrated the advantage that multiple, cooperating sensor platforms have over single sensor platform scenarios and we have shown experimental results from a passive tracking application which support our assertions. The results show that our proposed method combining a behavior-based, multiple objective function control model with a sensor providing high-level state information about the process being sampled is a viable method for adaptive sampling of transitory ocean phenomena in which fast reaction time is necessary. For example, a group of autonomous surface craft could provide area monitoring with some vehicles carrying radar sensors that

then vector vehicles with optical sensors toward any potential targets. In complex environments where such vehicles may have to contend with unknown and situations like obstacle avoidance while still maintaining sensing performance, the state space for the vehicle control is much too large for a world-model approach and a behavior-based approach such as described in the paper is indicated. This approach does not come without penalty, however. The parameter tuning and weighting needed for multiple, interacting behaviors to provide reasonable performance under complex conditions is not trivial at this stage. Our work in this area continues with an application requiring autonomous underwater vehicles with real array sensors to detect and track moving underwater targets as well as tracking applications using N sensor platforms possibly tracking multiple simultaneous contacts.

REFERENCES

[1] T. Curtin, J. Bellingham, J. Catipovic, and D. Webb, "Autonomous Oceanographic Sampling Networks," *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.

[2] R. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. 2, 1986.

[3] M. Benjamin, "Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control," Ph.D. dissertation, Brown University, 2002.

[4] M. Abidi and R. Gonzales, *Data Fusion in Robotics and Machine Intelligence*, 1992.

[5] D. Eickstedt and H. Schmidt, "A Low-Frequency Sonar for Sensor-Adaptive, Multi-Static, Detection and Classification of Underwater Targets with AUVs," in *Oceans 2003*, San Diego, CA, 2003.

[6] M. R. Benjamin, "Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control," Ph.D. dissertation, Brown University, Providence, RI, May 2002.

[7] M. R. Benjamin and J. Curcio, "COLREGS-Based Navigation in Unmanned Marine Vehicles," in *AUV-2004*, Sebasco Harbor, Maine, June 2004.

[8] M. Benjamin, J. Curcio, J. Leonard, and P. Newman, "Navigation of Unmanned Marine Vehicles in Accordance with the Rules of the Road," in *International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006.

[9] M. Benjamin, M. Grund, and P. Newman, "Multi-objective Optimization of Sensor Quality with Efficient Marine Vehicle Task Execution," in *International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006.

[10] T. Henderson and E. Shilcrat, "Logical Sensor Systems," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 169–193, 1984.

[11] T.C. Henderson, C. Hansen, and B. Bhanu, "A Framework for Distributed Sensing and Control," in *Proc. 9th Int. Joint Conf. Artificial Intell.*, 1985, pp. 1106–1109.

[12] ——, "The Specification of Distributed Sensing and Control," *J. Robotic Syst.*, vol. 2, no. 4, pp. 387–396, 1985.

[13] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques, and Software*, 1998.

[14] J. N. Sanders-Reed, "Error Propagation in Two-sensor 3D Position Estimation," *Optical Engineering*, vol. 40, no. 4, 2001.

[15] P. Pirjanian, "Multiple Objective Action Selection and Behavior Fusion," Ph.D. dissertation, Aalborg University, 1998.

[16] J. K. Rosenblatt, "DAMN: A Distributed Architecture for Mobile Navigation," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 1997.

[17] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14–23, April 1986.

[18] J. Curcio, J. Leonard, and A. Patrikalakis, "SCOUT - A Low Cost Autonomous Surface Platform for Research in Cooperative Autonomy," in *OCEANS 2005*, Washington DC, September 2005.

[19] D. Eickstedt and M. Benjamin, "Adaptive Control of Heterogeneous Marine Sensor Platforms in an Autonomous Sensor Network," MIT Computer Science and AI Laboratory, Tech. Rep. 0000, 2006.

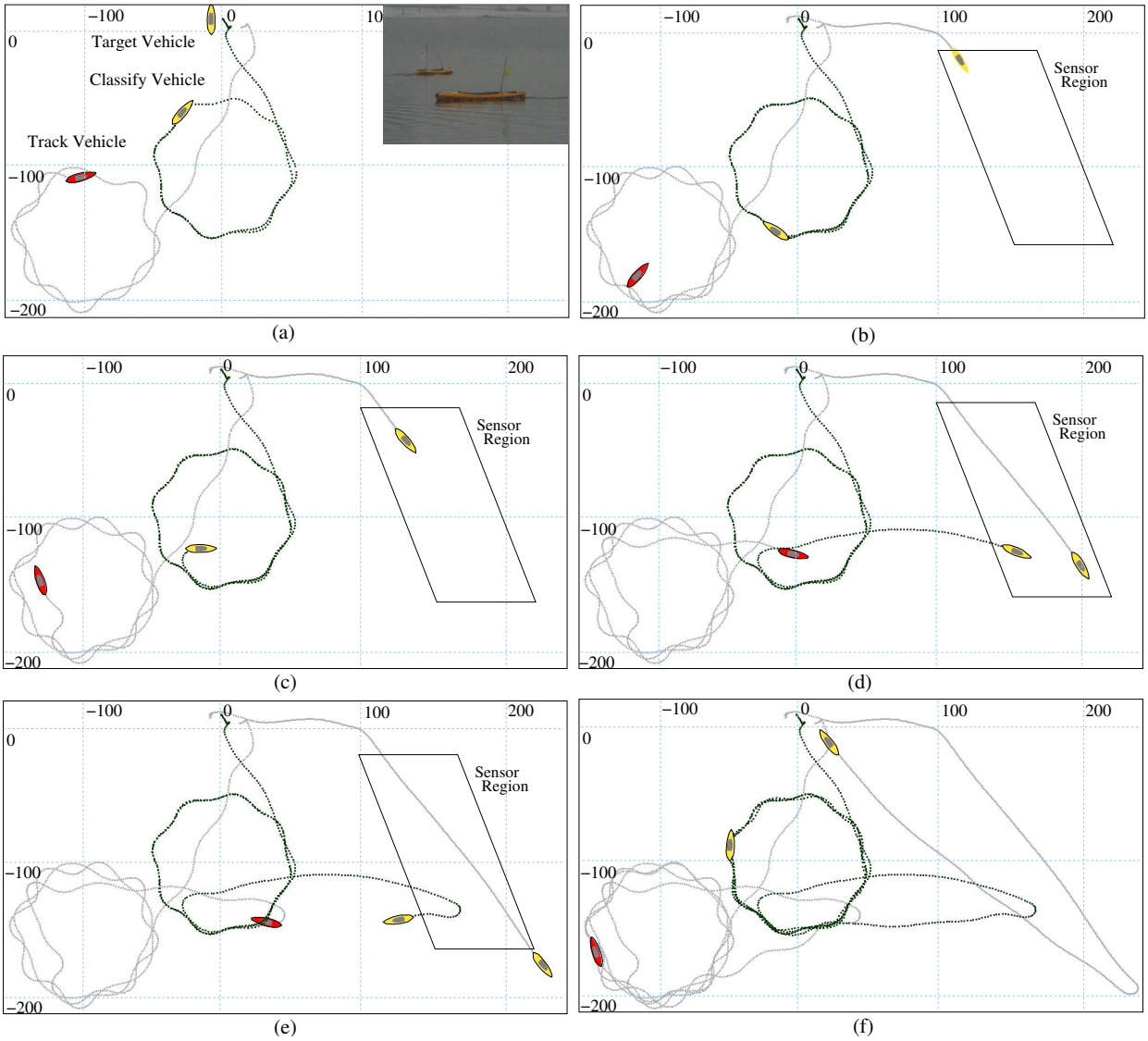


Fig. 13. A rendering of the experimental results. In (a) the track vehicle and classify vehicle (both autonomous kayaks, see Fig. 12) are deployed and executing their Orbit behavior to loiter in two separate regions. In (b) the target vehicle is deployed and has just entered the "sensor region" where it begins to transmit its position data to the track vehicle. The track vehicle has just activated its ArrayTurn behavior for determining which side of the sensor array the target is on. In (c) the track vehicle has just sufficiently resolved the left-right ambiguity and has begun transmitting track solutions to the classify vehicle. The classify vehicle has begun its CloseRange behavior to facilitate classification of the target. The track vehicle has activated its CloseRange and ArrayAngle behaviors. In (d) both the track and classify vehicle are dominated by CloseRange behaviors to the target. In (e), the classify vehicle has performed the classification of the target and both vehicles are returning back to their loiter regions. In (f) both vehicles are back on-station and awaiting any further unknown objects or vehicles to come through its sensor field. The target vehicle has returned to the dock.

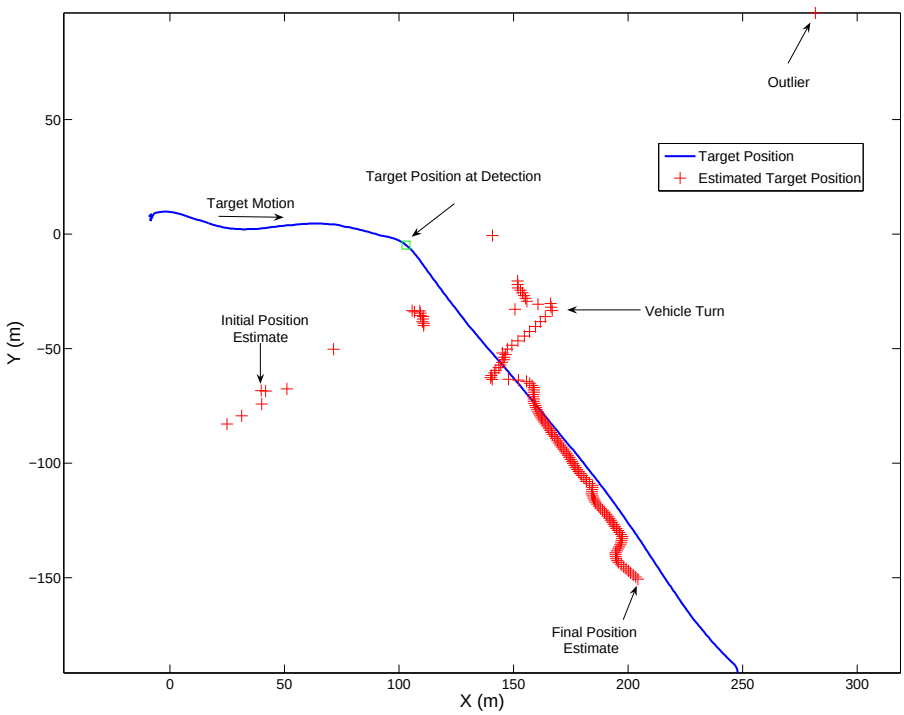


Fig. 14. This figure depicts the target position estimates produced by the MOOS process pTracker overlaid onto the actual target track. It is readily seen in the figure that the initial estimates were poor due to a small value for N as discussed in section IV-D. As the number of observations increases, a convergence of the estimate near to the actual track can be seen. Of special note is the large increase in convergence labeled "Vehicle Turn" in the figure. This is the point at which the sensor vehicle's CloseRange behavior became active and made a sharp course change between the positions shown in Fig. 4(c) and 4(d). Some bias can be seen in the estimates near the end of the experiment due to a need to further optimize the vehicle behavior parameters to produce a better FIM as discussed in section IV-D.

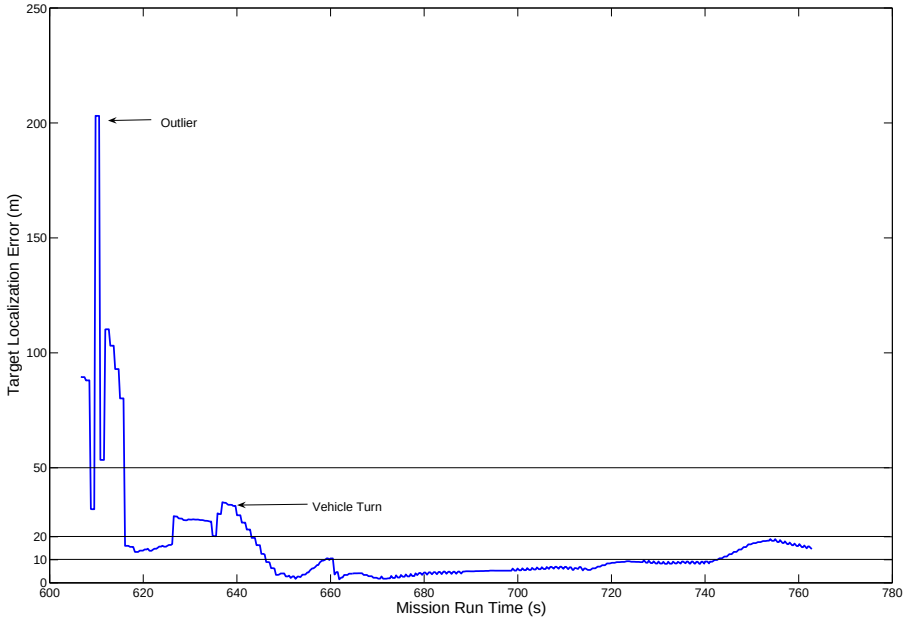


Fig. 15. Target localization error. This figure shows the error between the target position estimates and the actual target location as a function of mission run time. The point on the figure labeled "Vehicle Turn" corresponds to the point in the mission labeled "Vehicle Turn" in Fig. 14.

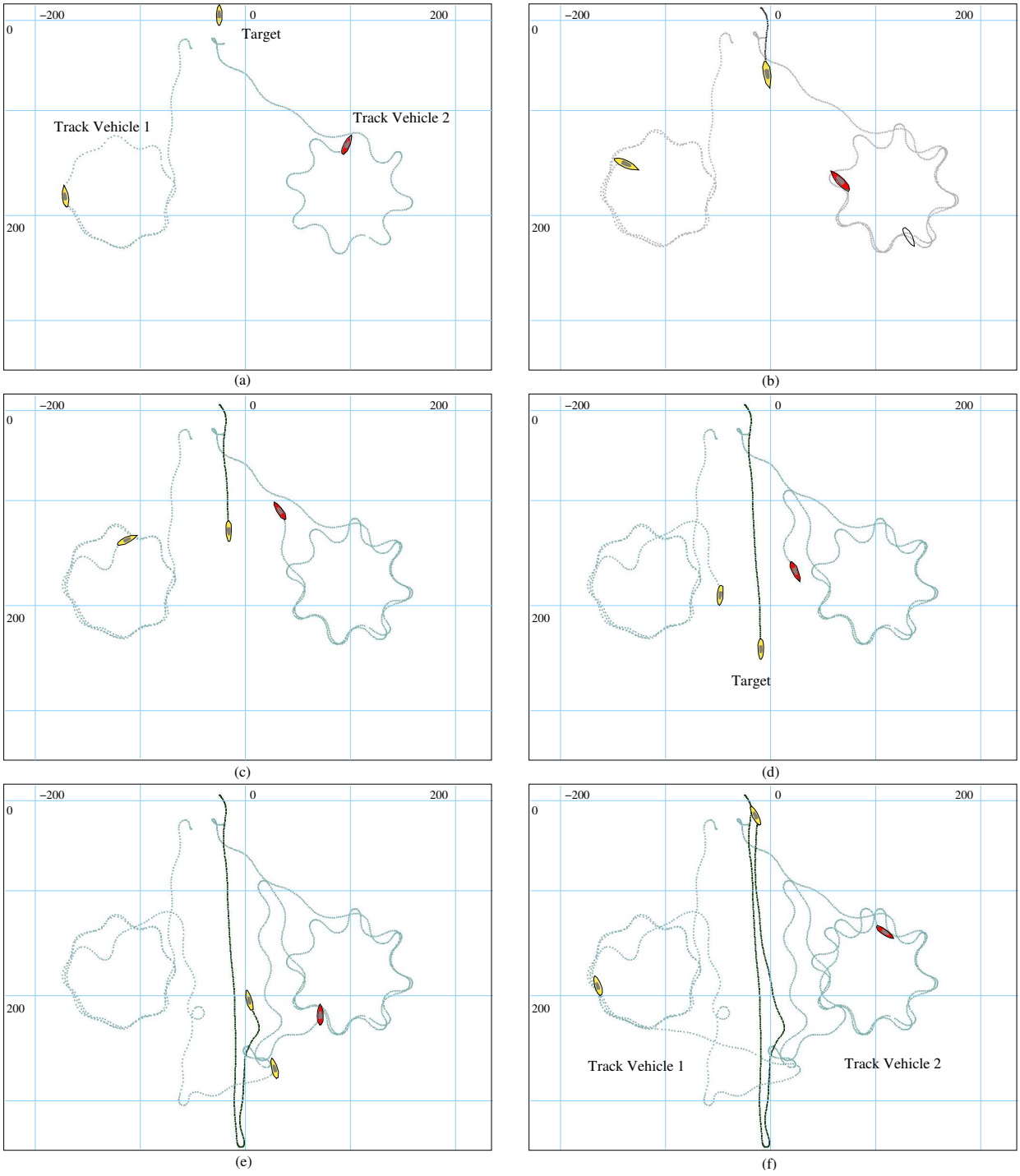


Fig. 16. A rendering of the experimental results. In (a) two tracking vehicles (both autonomous kayaks, see Fig. 12) are deployed and executing their Orbit behaviors to patrol in two separate regions. Note that tracking vehicle two is exhibiting signs of a rudder control problem. In (b) the target vehicle is deployed and has just entered the "sensor region" where it begins to transmit its position data to the tracking vehicles for use in the bearing simulators. The tracking vehicles have just activated their ArrayTurn behaviors for determining which side of the sensor array the target is on. In (c) the tracking vehicles have just sufficiently resolved the left-right ambiguity and have begun executing their Formation behaviors using the target position estimate as a virtual leader. In (d) both the tracking vehicles have moved into formation behind the target. In (e), the target unexpectedly turned for home before the sensor vehicles have finished tracking, violating the constant velocity assumption and confusing the tracking system. In (f) both vehicles are back on-station and awaiting any further unknown objects or vehicles to come through its sensor field. The target vehicle has returned to the dock.

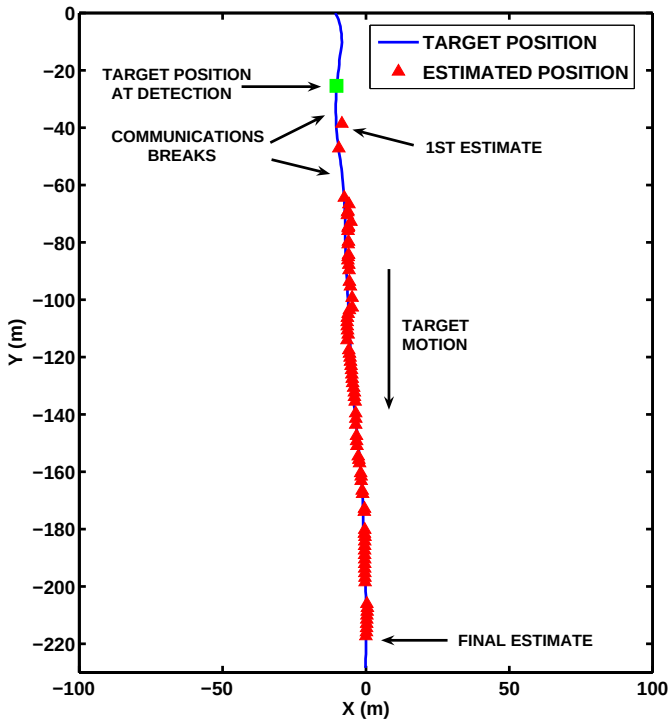


Fig. 17. Target track solution results. This figure depicts the target position estimates produced by the MOOS process pTracker overlaid onto the actual target track for the period in which the target vehicle was operating in a constant velocity scenario. As can be seen, excellent position estimates were obtained, especially compared with the tracking results obtained using a single sensor platform to track a constant velocity target as detailed in [19]. The gaps in the estimates as seen in the figure were due to communications breaks when no bearing estimates from vehicle two were received by vehicle one.

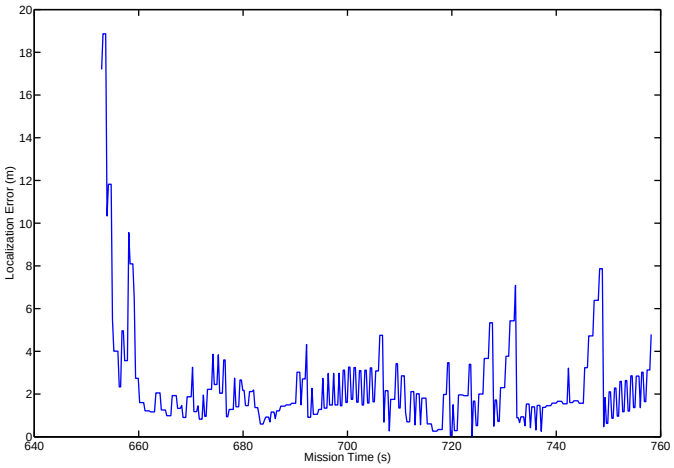


Fig. 18. Target Position Estimate Error. This figure shows the error in the target position estimate as a function of mission run time. As can be seen, even with the communications breaks, position estimation results were generally very good.