

Adaptive Control of Heterogeneous Marine Sensor Platforms in an Autonomous Sensor Network

Under submission for peer-review publication

Donald P. Eickstedt

Laboratory for Autonomous Marine Sensing
Center for Ocean Engineering
Massachusetts Institute of Technology
Cambridge MA 02139
Email: eicksted@mit.edu

Michael R. Benjamin

NAVSEA Division Newport
Newport RI 02841
Center for Ocean Engineering, MIT
Cambridge MA 02139
Email: mikerb@csail.mit.edu

Henrik Schmidt

John J. Leonard

Center for Ocean Engineering
Massachusetts Institute of Technology
Cambridge MA 02139
Email: henrik@mit.edu, jleonard@mit.edu

Abstract—This paper describes an investigation into the control of autonomous mobile sensor platforms in a marine sensor network used to provide monitoring of transitory phenomenon over a wide area. A distributed network of small, inexpensive vehicles with heterogeneous sensors allows us to build a robust monitoring network capable of real-time response to rapidly changing sensor data. The major objective of this paper is to describe a framework for adaptive and cooperative control of the autonomous sensor platforms in such a network. This framework has two major components, a sensor that provides high-level state information to a behavior-based autonomous vehicle control system and a new approach to behavior-based control of autonomous vehicles using multiple objective functions that allows reactive control in complex environments with multiple constraints. Experimental results are presented for a 2-D target tracking application using a network of autonomous surface craft in which one platform with a simulated bearing sensor tracks a moving target and relays the target state information to a second vehicle that is moving in a classification mode. From these results, it is readily seen that there is the potential for potent synergy from the cooperation of multiple sensor platforms which can each view an event of interest from a different vantage point.

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].

Vehicles working in coordination offer distinct advantages. They may each have different payloads, sensors, and endurance capabilities. They may be able to share sensor data to perceive phenomena a single vehicle itself may not be able to perceive alone. The use of multiple vehicles also may allow one vehicle to stay at the surface, with a higher bandwidth link to other robotic or human operated vehicles, while one or more other vehicles operate under the surface operating at varying depths to optimize their sensor-oriented tasks.

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

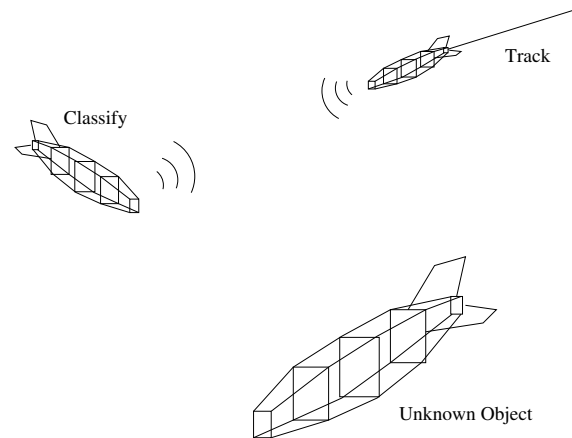


Fig. 1. Two heterogeneous unmanned marine vehicles are in operation together. The first tracks the position and trajectory of unknown underwater objects using a towed linear array, and communicates track solution information via acoustic modem to a second vehicle with different sensors more suitable for classifying underwater objects.

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 position to 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.

While coordinated marine vehicles 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.

In this work we address these challenges by presenting a novel architecture and set of vehicle behaviors and present experimental validation of this work using three fully autonomous kayaks.

II. TECHNICAL APPROACH

In this section we present our general autonomy architecture 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 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, [2], [3]. See [4] and [5] 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. 2(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. 2(b)). Each

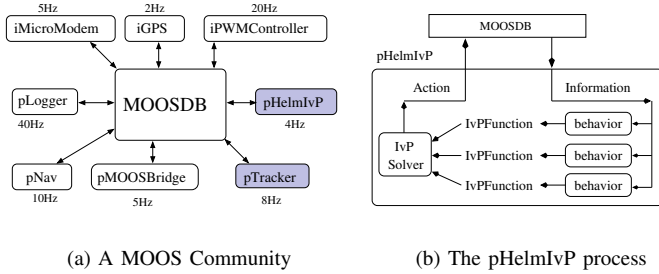


Fig. 2. The IvP Helm runs as a process called pHelmIvP in a MOOS community. MOOS may be composed of processes for data logging (pLogger), data fusion (pNav), actuation (iPWMController), sensing (iGPS), communication (pMOOSBridge, iMicroModem), and much more. They can all be run at different frequencies as shown.

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 processes. The behaviors responsible for control in the tracking and classification vehicles are discussed in Section IV.

B. Autonomous Bearings-Only Object Tracking

The tracking vehicle in this work uses a set of tracking algorithms that run in a single MOOS process called pTracker

(see Fig. 2(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 MOOS process then produces and posts track solution information to the MOOSDB to be consumed by any other MOOS process including an inter-vehicle communications process like pMOOSBridge or iAcousticModem for consumption by another platform working in coordination. More on the algorithms for the pTracker process is given in Section III.

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 AUV’s which are more expensive and time-consuming to operate.

III. BEARINGS-ONLY OBJECT TRACKING

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 [6]. 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 $[x_p[n] \ y_p[n]]$ moving in the same plane with measurement observations taken at the discrete time intervals $n = 0, 1, \dots, N$. The state equations for the target motion can be written in discrete time as

$$x[n] = x[0] + \dot{x}_t t_n \quad (1)$$

$$y[n] = y[0] + \dot{y}_t t_n \quad (2)$$

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

$$\hat{x} \triangleq [x[0] \ y[0] \ \dot{x}_t \ \dot{y}_t]^T \triangleq [x_0 \ x_1 \ x_2 \ x_3]^T \quad (3)$$

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

$$z[n] = h[n, x] + w[n] \quad (4)$$

where

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

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

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

The maximum likelihood estimate is then formed by

$$\hat{x} = \arg \min_x \lambda(x) \quad (7)$$

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

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

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

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

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

C. The Cramer-Rao Lower Bound

The Cramer-Rao lower bound (CRLB) stipulates that the variance of our parameter estimates cannot be lower 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 [6]. Formally, we say that

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

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 elements of the FIM for this problem are

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

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

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

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

Given that the parameters are statistically independent, this gives the following FIM

$$I_y(x) = \frac{1}{q} \begin{bmatrix} h_{x0} & 0 & 0 & 0 \\ 0 & h_{x1} & 0 & 0 \\ 0 & 0 & h_{x2} & 0 \\ 0 & 0 & 0 & h_{x3} \end{bmatrix} \quad (17)$$

The Cramer-Rao lower bound on the variance of each of our parameters is then found by inverting (17). 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 III-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 [6]. 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 IV are designed to produce a well-conditioned FIM.

IV. 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 further examples of this approach, although with different missions and behaviors, see [4], [5].

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 ([2], [7], [8]). 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 [8]. 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 [9].

B. The Orbit Behavior

The Orbit behavior is designed to provide a patrol capability in which the vehicle will orbit a fixed point while waiting for an event to occur. 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 IV-G.

C. The ArrayTurn Behavior

The ArrayTurn behavior is designed to provide 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. 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.

D. ArrayAngle Behavior

The ArrayAngle behavior is designed to hold 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.

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

F. Classify Behavior

The Classify behavior used in this demonstration is active on the classify vehicle and is identical to the CloseRange behavior described in IV-E 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.

G. The Waypoint Behavior

The waypoint behavior is configured with a set 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. Furthermore, a vehicle can be configured with multiple instances of the same behavior type, and every vehicle is typically configured with an instance of this behavior having a single waypoint just off the dock, 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.

H. The OpRegion Behavior

The OpRegion behavior is configured 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.

V. EXPERIMENT SETUP AND RESULTS

Experimental validation of the architecture and algorithms for autonomous bearings-only tracking, was conducted using two autonomous kayaks as the tracking and classify vehicles, and a third kayak as a moving object to be tracked and classified. The kayaks are proxies for autonomous underwater vehicles (AUV's) used in upcoming follow-on experiments.

A. Simplifying Assumptions

Three simplifying assumptions were made. First, as a proxy for the towed array bearings-only sensor, the GPS position of the sensed vehicle was communicated over an 802.11b wireless connection to the sensing vehicle. The sensing (tracking) vehicle converted (diminished) this information into bearings-only sensor data. The second simplification was the use of the 802.11b wireless connection as a proxy for communications via acoustic modem. The third simplifying assumption concerns the classify vehicle, which is not fitted with sensors suitable for classifying another marine vehicle. As a proxy for this aspect of the scenario, the classify vehicle will solely aspire to close the range between itself and the object of interest once it starts receiving track-solution information from its robotic counterpart.

B. The Marine Vehicle Platforms

The autonomous surface crafts used in this experiment are based on a kayak platform (Fig. 3). 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

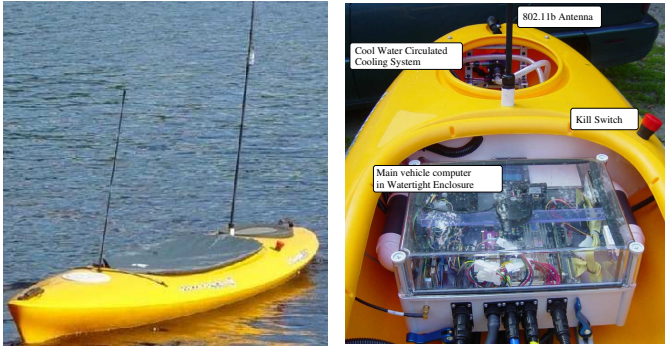


Fig. 3. The kayak-based autonomous surface craft.

is powered by 5 lead-acid batteries and a Minn Kota motor providing both propulsion and steering. The vehicles have a top speed of roughly 2.5 meters per second. See [10] for more details on this platform.

C. Behavior Configurations

The three vehicles were configured with the following behaviors and pre-conditions. 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”.

The tracking vehicle helm was configured with an Orbit behavior which is 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 ArrayTurn, ArrayAngle, and CloseRange behaviors. These three behaviors are conditioned on the vehicle receiving bearings-only sensor data, indicated by “sensor_data=active” in the MOOS Database.

The classify vehicle helm was also configured with an Orbit behavior that activated at mission startup. Additionally, the helm on this vehicle was configured with a Classify behavior which went active when target track solution data was received from the tracking vehicle. The Classify behavior was configured to deactivate itself when the CPA to the target vehicle reached 30 meters.

The target vehicle was configured to follow a simple set of waypoints, and was further configured to communicate its GPS position to the sensor vehicle. This communication only occurred when the target vehicle was within a certain specified region referred to as the “Sensor” region (Fig. 4). Deployment of the target vehicle was done via human command over wireless link when the other two vehicles had been on-station for an arbitrary sufficient time.

D. Experimental Results

Fig. 4 shows the vehicle motion for an experimental “track and classify” mission with autonomous kayaks (see Fig. 3) 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. 5 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 III-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 III-D. Second, this error is due to a need to further optimize the vehicle behavior parameters to produce a better FIM.

VI. 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. 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.

ACKNOWLEDGMENTS

This work was funded in part by ONR under Pennsylvania State University subcontract S05-06, “PLUSNET: Persistent Littoral Undersea Surveillance Network”. Special thanks go to the personnel of the MIT Sailing Pavilion whose continuing support is instrumental to our marine vehicle experiments. The authors also wish to thank Joe Curcio, the designer and

builder of the kayak platforms used in our experiments, and Paul Newman of Oxford University for the existing MOOS software libraries as well as for help in developing new and improved MOOS modules needed for this effort.

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] M. R. Benjamin, “Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control,” Ph.D. dissertation, Brown University, Providence, RI, May 2002.
- [3] M. R. Benjamin and J. Curcio, “COLREGS-Based Navigation in Unmanned Marine Vehicles,” in *AUV-2004*, Sebasco Harbor, Maine, June 2004.
- [4] 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.
- [5] 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.
- [6] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques, and Software*, 1998.
- [7] P. Pirjanian, “Multiple Objective Action Selection and Behavior Fusion,” Ph.D. dissertation, Aalborg University, 1998.
- [8] J. K. Rosenblatt, “DAMN: A Distributed Architecture for Mobile Navigation,” Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 1997.
- [9] 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.
- [10] 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.

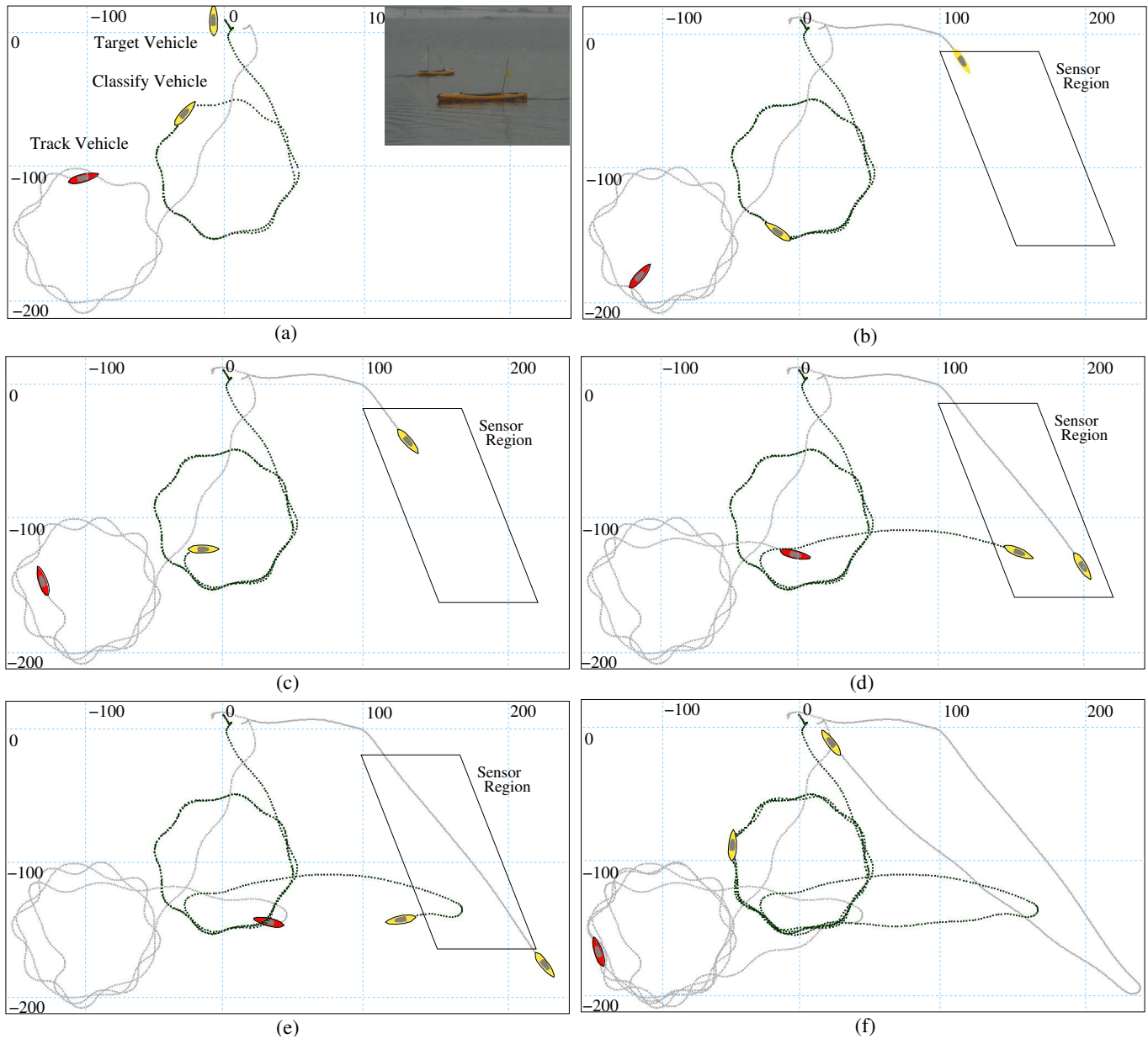


Fig. 4. A rendering of the experimental results. In (a) the track vehicle and classify vehicle (both autonomous kayaks, see Fig. 3) are deployed and executing their Orbit behavior to loiter in two separate regions. In (b) the target vehicle is deployed and has just entered the “sensor region” where it 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.

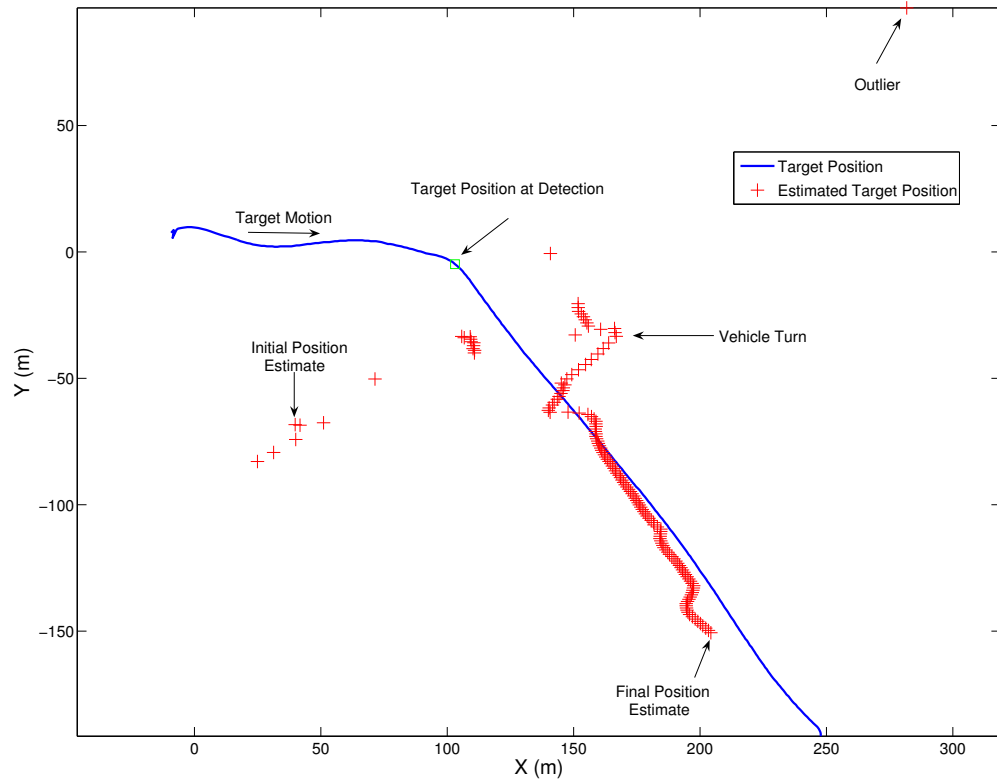


Fig. 5. Target track solution results. This figure depicts the target position estimates produced by the MOOS process pTracker overlaid onto the actual target track. It is readily seen in the figure that the initial estimates were poor due to a small value for N as discussed in section III-D. As the 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 III-D

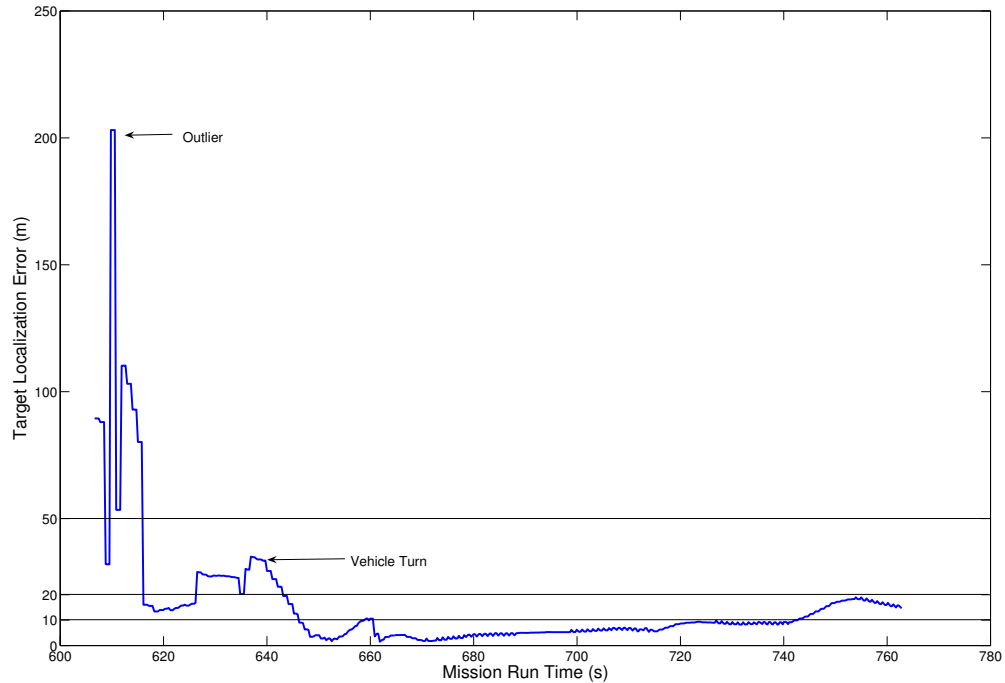


Fig. 6. Target localization error. This figure shows the error between the target position estimates and the actual target location as a function of mission run time. The point on the figure labeled “Vehicle Turn” corresponds to the point in the mission labeled “Vehicle Turn” in Fig. 5