2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Cooperative Navigation for Heterogeneous Autonomous Vehicles via Approximate Dynamic Programming

Silvia Ferrari, Michael Anderson, Rafael Fierro, and Wenjie Lu

Abstract— **Unmanned ground and aerial vehicles are becoming crucial to many applications because of their ability to assist humans in carrying out dangerous missions. These vehicles can be viewed as networks of heterogeneous unmanned robotic sensors with the goal of exploring complex environments, to search for and, possibly, pursue moving targets. The robotic vehicle performance can be greatly enhanced by implementing future sensor actions intelligently, based both on prior knowledge and on the information obtained by the sensors on line. In this paper, we present an approximate dynamic programming (ADP) approach to cooperative navigation for heterogeneous sensor networks. The mobile sensor network consists of a set of robotic sensors modeled as hybrid systems with processing capabilities. The goal of the ADP algorithm is to coordinate a team of heterogeneous autonomous vehicles (*i.e.*, ground robot and quadrotor UAV) to navigate within an obstacle populated environment while satisfying collision avoidance constraints and searching for stationary and mobile targets. It is assumed that the ground vehicle has a small sensor footprint with high resolution. The quadrotor, on the other hand, has a large sensor field-of-view but low resolution. The UAV provides a low resolution look-ahead map to the ground robot which in turn uses this information to plan its actions. The proposed navigation strategy combines artificial potential functions for target pursuing with ADP for learning *C-obstacles* on line. The efficacy of the proposed methodology is verified through numerical simulations.**

I. INTRODUCTION

Developments in autonomous vehicles and sensor technologies are producing systems with increased capabilities. These unmanned ground and aerial vehicles are becoming crucial to many applications because of their ability to assist or replace humans in carrying out dangerous yet vital missions. The paradigm that emerges from these applications is a set of hidden targets, mobile or stationary, that must be measured and possibly pursued by multiple heterogeneous sensors installed on mobile platforms. In all of these applications, the sensor network's performance can be greatly enhanced by coordinating and implementing future sensor actions intelligently, based both on prior knowledge and on the information obtained by the sensors *on line*.

Several authors have recognized the hybrid nature of coordinated robot control [1]–[5], and have utilized the three-layer *hybrid* architecture proposed in [6], [7] to model and analyze the interactions of a continuous-state vehicle with a discrete-event controller that makes high-level decisions on its optimal sequence of behaviors. Other approaches to the coordination of robotic networks include distributed control with an emphasis on communication protocols synchronous motion coordination [8]. A hybrid modeling approach to maintaining connectivity in a mobile multi-agent network was recently presented in [9]. A hybrid modeling framework for robust maneuver-based motion planning in nonlinear systems with symmetries was proposed in [10], and a cell decomposition approach to geometric sensor motion planning was described in [11]. Existing hybrid and distributed control approaches are very effective at maintaining a desired formation or connectivity in a sensor network, but they do not typically account for the geometries of the targets, obstacles, and sensors' fields-of-view (FOVs).

On the other hand, several robot motion planning approaches, described in [12], [13], have been developed to account for the geometries of the robot and the obstacles, and avoid collisions. However, these approaches are not directly applicable to robotic sensors because they are designed to prevent intersections between the vehicle's geometry and the obstacles to avoid collisions, while the sensor's FOV must intersect the geometry of the targets in order to obtain measurements. Most of the research relating sensor measurements to robot motion planning so far has focused on the effects that the uncertainty in the geometric models of the environment has on the motion strategies of the robot [14]–[17]. But, what remains to be addressed are the effects that these models have on the sensor's strategies, *i.e.*, how to plan and adapt the motion strategies that support the optimal *sensor* measurement sequence, subject to nonlinear dynamics [18].

Controlling a continuous-state nonlinear dynamical system, such as a robotic sensor, so as to optimize a desired performance metric is a difficult problem that requires the solution of a two-point boundary value problem. Solving this problem off line is not always an effective solution, because the actual initial conditions may be different from their assumed value and the system model may be imperfect. Approximate dynamic programming (ADP) approaches [19] seek to optimize the specified performance metric conditioned upon knowledge of the actual system. They reduce the computational complexity of the optimal control problem by modeling the control law and the estimated future cost as parametric structures. The control law is adapted subject to online measurements of the state and knowledge of the probability distribution of uncertainties, thereby optimizing the performance of the real system incrementally over time.

S. Ferrari and W. Lu are with the Laboratory for Intelligent Systems and Controls, Department of Mechanical Engineering & Materials Science, Duke University, Durham, NC 27708-0300, USA {sferrari, wenjie.lu}@duke.edu

M. Anderson and R. Fierro are with the MARHES Lab, Electrical & Computer Engineering Department, University of New Mexico, Albuquerque, NM 87131-0001, USA {msander, rfierro}@ece.unm.edu

978-1-61284-801-3/11/$26.00 ©2011 IEEE

In this paper, the advantages of hybrid control and geometric motion planning algorithms are combined by designing a continuous-discrete interface (CDI) for the hybrid system that is comprised of a novel sensor motion planning algorithm. Using a new concept known as *C-targets*, this CDI maps sensor's states from the vehicle's continuous configuration space to a sampled discrete-event graphical representation that is void of obstacles and enables target detections. According to a novel approach integrating roadmap methods with potential fields, the CDI is able to utilize a probability density function that is obtained by marginalizing the potential field defined from the information value of the targets. As a result, the potential function used by the CDI is also used to design the continuous-state controllers for the vehicles. A cooperative navigation ADP approach is implemented to adapt the robotic sensor policies subject to unstructured and uncertain environments, in order to avoid collision, while observing and pursuing hidden targets based on the sensor data that becomes available over time.

The remainder of the paper is organized as follows. Section II provides the problem formulation and assumptions. The mathematical models of robotic sensors and targets are described in Section III. The proposed cooperative navigation methodology is presented in Section IV. Section V contains simulation results and Section VI discusses some conclusions and future work to be performed.

II. PROBLEM FORMULATION AND ASSUMPTIONS

In this paper, the hybrid framework developed in [20] is extended to describe networked hybrid dynamical systems (NHDS), comprised of N continuous-state plants (CSPs), representing N robotic sensors with dynamic equations,

$$\text{CSP}_i\text{:} \quad \begin{cases} \dot{x}_i(t) = f_i[x_i(t), u_i(t)] \\ y_i(t) = h_i[x_i(t), \ell(t)] \end{cases} \quad i = 1, \ldots, N \qquad (1)$$

where $x_i \in X_i \subset \mathbb{R}^n$, $u_i \in U_i \subset \mathbb{R}^m$, and $y_i \in Y_i \subset \mathbb{R}^p$ are the state, control, and output of the i^{th} vehicle, respectively. The discrete-event parameter $\ell(t) \in \{1, 2, \ldots\}$ is a deterministic scalar index that is incremented by an event function when a continuous-state event occurs. The reference trajectory and discrete-event parameter are computed by a continuous-discrete interface (CDI) logic, described in Section IV based on the planned mission's tasks. Each vehicle is equipped with a continuous-state controller (CSC) given by

$$\text{CSC}_i\text{:} \quad u_i(t) = c_i[x_i(t), y_i(t), r_i(t), \ell(t)], \qquad (2)$$

where $i = 1, \ldots, N$ and $r_i \in \mathbb{R}^p$ is the reference trajectory.

In this paper, the mission planner is initially modeled by a centralized discrete-event plant (DEP) and controller (DEC) that reside on a remote computational facility or onboard one of the vehicles. The DEP and DEC model and control, respectively, the coordination of future sensors' tasks or *behaviors*. The three-layer hybrid system framework previously analyzed in the literature [6], [20] considers a deterministic DEP represented by difference equations, or Petri nets. However, because of the probabilistic nature of the sensing tasks, in this paper, the DEP is described by a partially-observable Markov decision process (POMDP). As was recently shown in [11], the coupled robotic sensor motion and observation process is a POMDP described by the tuple,

$$\begin{aligned} \text{DEP:} \quad \Sigma = \{T, \mathcal{X}, \mathcal{U}(\xi, z), \text{P}(\xi(k+1) \mid \xi(k), a(k)), \\ \text{P}(z(k) \mid \xi(k), u(k), R[\xi(k), z(k), a(k), u(k)])\} \end{aligned} \qquad (3)$$

where k is the discrete time index, and $T = \{1, \ldots, f\}$ is a finite set of decision epochs. The state space $\mathcal{X}$ is a finite set of mutually-exclusive events, the action space $\mathcal{U}(\xi, z)$ is the space of admissible action and test decisions $a(k)$ and $u(k)$, and P denotes a transition probability function. The subset of the DEP state denoted by $\xi(k)$ represents the physical state of the robotic sensors and, thus, it is observable, and determined by $\xi(k-1)$ and $a(k-1)$. The subset of the DEP state denoted by $z(k)$ represents the target characteristics and, thus, it is hidden up to k, and all of its possible outcomes must be propagated up to k according to the above transition probabilities. The DEC outputs are classified as either action decisions, a, when they change the physical state of the system ξ (such as the decision on where to move the robotic sensor), or as test decisions, u, when they affect the system's knowledge about the hidden state z (such as the decision to obtain a sensor measurement). Then, the utility of test decisions is given by the value of information associated with z, and the utility of action decisions is the reward associated with the physical state ξ.

Let $R : \Omega(\pi(v)) \to \mathbb{R}$ denote the total utility or reward function associated with the chosen modes of behavior. Then, the discrete-event controller (DEC) is described by a class of admissible policies or control laws, referred to as *strategy*,

$$\text{DEC:} \quad \sigma = \{\gamma_0, \gamma_1, \ldots, \gamma_{f-1}\} \qquad (4)$$

where γ_k maps the discrete state and output variables $\{\xi(k), z(k)\}$ into the admissible decisions,

$$\begin{aligned} \{a(k), u(k)\} = \gamma_k[\xi(0), a(0), u(0), \ldots, \\ \xi(k-1), a(k-1), u(k-1)], \end{aligned} \qquad (5)$$

such that $\gamma_k[\cdot] \in \mathcal{R}$ for all $\xi(k)$ and $z(k)$. Based on the mutual-information theoretic function derived in [11], [21], the robotic sensors' utility can be assumed to be an additive reward function and, therefore, a value function,

$$V(k) = \sum_{k=0}^{f-1} \mathbb{E}\left\{R[\xi(k), z(k), a(k), u(k)]\right\}, \qquad (6)$$

can be defined as the expected utility or cost-to-go of the DEC in (4), where $\mathbb{E}$ denotes the expectation with respect to the state variables. Then, the *optimal strategy* at k is the sequence of policies, σ^*, that maximizes $V(k)$ given the POMDP in (3).

Therefore, for the NHDS proposed here, the CSP is the tuple $S = \{\mathcal{I}, X_i, U_i, Y_i, f_i, h_i\}_{i \in \mathcal{I}}$, where $\mathcal{I} = 1, \ldots, N$ is the set of unique identifiers representing the robotic sensors

in the network. The continuous-discrete interface (CDI) that connects S and Σ becomes

$$\Sigma\backslash S: \quad \begin{cases} r_i(t) = \alpha_i[\xi(k), a(k), u(k)] \\ \ell(t) = \beta[\xi(k), a(k), u(k)] \end{cases} \tag{7}$$

$$S\backslash\Sigma: \quad z(k) = v_i[y_i(t), r_i(t)], \quad i = 1, \ldots, N \tag{8}$$

where, the functions $\alpha_i : \mathcal{X} \times \mathcal{R} \rightarrow \mathbb{R}^p$ and $\beta : \mathcal{X} \times \mathcal{R} \rightarrow \{1, 2, \ldots\}$ are mappings in $\Sigma\backslash S$. The event function $v_i : Y_i \times \mathbb{R}^p \rightarrow Z$ is a mapping in $S\backslash\Sigma$ that tells the discrete-event system and controller which event caused $\ell(t)$ to change, such that knowledge about the hidden state $z(k)$ can be updated based on real-time measurements and location of the vehicles. By utilizing this novel hybrid dynamical system framework, the POMDP hidden state $z(k)$ can be used to represent unknown target(s) and workspace characteristics that influence high-level decisions regarding mission planning. At the same time, the reference trajectory and event parameter are computed by the CDI mappings in (7).

III. MATHEMATICAL MODELS

The mathematical models utilized here are divided into two subsections. In the first subsection, we present the hybrid models of the robotic sensors comprised of ground and air vehicles that are deployed to cooperatively detect, track, and pursue multiple moving targets. In the second subsection, we present the probabilistic models of the targets, which are updated over time, based on the sensor data obtained and processed by the sensor network.

A. Modeling of Robotic Sensors

This papers focuses on the online adaptation required for the sensor network to operate in a partially-observable workspace, with no prior knowledge of the targets and partial knowledge of the obstacles. Let $\mathcal{W} \subset \mathbb{R}^3$ denote an Euclidean robotic sensor workspace populated with a set of fixed obstacles $\{\mathcal{B}_j\}_{j \in \mathcal{I}_\mathcal{B}}$ with geometries and positions that are partially unknown, and are estimated on line using real-time sensor measurements. In addition to obstacles to be avoided, the robotic sensor workspace also is populated with a set of moving targets $\{\mathcal{T}_l\}_{l \in \mathcal{I}_\mathcal{T}}$ (described in Section III-B) that must be detected, tracked, and then pursued by the sensors on line, as by the DEC. We use a geometric modeling approach inspired by classical robot motion planning [12], [13] that characterizes a robotic sensor by its vehicle dynamics (1), and by discrete geometries representing the vehicle's geometry $\mathcal{A}_i \subset \mathbb{R}^3$, and the sensor's field-of-view (FOV) $\mathcal{S}_i \subset \mathbb{R}^3$. Examples of these discrete geometries are illustrated in Fig. 1 using static sensors (*e.g.*, VICON motion capture), and mobile (ground and aerial) robotic sensors from the test bed available at the MARHES lab. The sensor's FOV, $\mathcal{S}_i$, is assumed to have fixed position and orientation with respect to a moving Cartesian frame $\mathcal{F}_{\mathcal{A}_i}$, embedded in $\mathcal{A}_i$. Thus, the robot configuration $q_i \subset x_i$ specifies the FOV's position and orientation with respect to a fixed Cartesian frame $\mathcal{F}_\mathcal{W}$. The ground vehicle dynamics in $\mathcal{W}$ are represented by a nonholonomic, nonlinear model [22], [23],

$$M_i(q_i)\ddot{q}_i + B_i(q_i, \dot{q}_i) + G_i(q_i) = u_i, \quad i \in \mathcal{I}, \tag{9}$$

where $M_i(q_i)$ is the i^{th} robotic sensor's inertia matrix, $B_i(q_i, \dot{q}_i)$ is the fictitious force, $G_i(q_i)$ is the gravitational force, and u_i is the torque input. The aerial vehicle dynamics are provided by a 6 DOF nonlinear quadrotor model described in [24], with collective, roll, pitch, and yaw input commands, and a six-dimensional state vector comprised of the vehicle's inertial position and orientation in $\mathcal{F}_\mathcal{W}$. Thus, all robotic sensors can be described by the CSP in (1), and in addition to requiring a stabilizing feedback control law to follow the reference trajectory, they must avoid the obstacles in $\mathcal{W}$ by preventing intersections between $\mathcal{A}_i$ and $\mathcal{B}_j$, *i.e.*, by guaranteeing that $\mathcal{A}_i \cap \mathcal{B}_j = \emptyset$, for all i, j, and t.

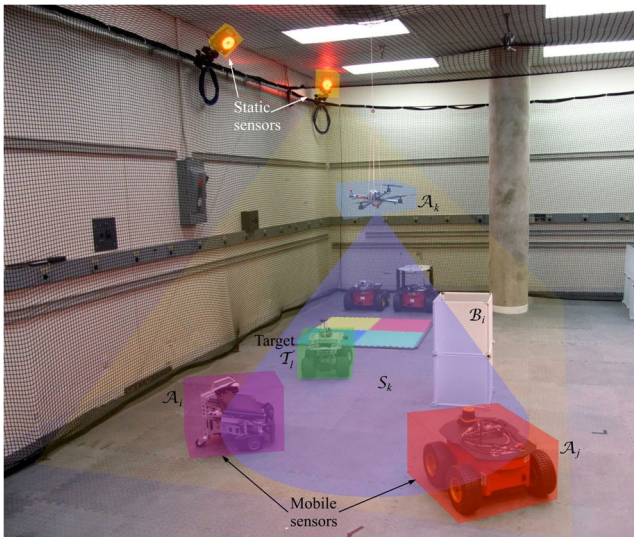

Fig. 1. Vehicle geometry $\mathcal{A}_i, \mathcal{A}_j, \mathcal{A}_k$, sensor's FOV $\mathcal{S}_k$, static sensors (VICON motion capture system), a target geometry $\mathcal{T}_l$, and a fixed obstacle $\mathcal{B}_i$ are shown.

Dual to the obstacle-avoidance problem is the target-measurement problem, as an intersection between the FOV geometry $\mathcal{S}_i$ and the geometry of a target $\mathcal{T}_l$ must take place, *i.e.*, $\mathcal{S}_i \cap \mathcal{T}_l \neq \emptyset$, in order for the i^{th} sensor to be able to obtain a vector of measurements m_i about z_l. In standard estimation theory, a sensor that obtains a vector of measurements is typically modeled as a deterministic, possibly nonlinear, vector function of the state and of a random vector representing the sensor noise or measurement errors. In many sensor applications, however, the target state to be estimated, and the sensor measurements also are random variables with arbitrary probability distributions [25], [26]. Therefore, a more general measurement model that has been proposed in the sensor networks literature is the

conditional probability mass function (PMF),

$$\Pr(m_i(k)|z_l(k),\lambda_i(k)) = \tag{10}$$

$$\begin{cases} \frac{\Pr(m_i(k),z_l(k),\lambda_i(k))}{[\Pr(z_l(k))\Pr(\lambda_i(k))]}, & \text{if } \mathcal{S}_i \cap \mathcal{T}_l \neq \emptyset \\ 0, & \text{if } \mathcal{S}_i \cap \mathcal{T}_l = \emptyset \end{cases}$$

for $k = 1, \ldots, f$, where λ_i is a vector of sensor characteristics, such as sensor mode and measurement errors. This sensor model assumes that m_i, z_l, and λ_i are discrete random variables with finite ranges $\mathcal{M}$, Z, and Λ, respectively. The joint PMF $\Pr(m_i(k), z_l(k), \lambda_i(k))$ is a probabilistic description of the measurement process that can be learned from data [11], [26], [27]. The priors $\Pr(z_l(k))$ and $\Pr(\lambda_i(k))$ are computed from environmental maps of $\mathcal{W}$ and target information, when available. Otherwise, they are assumed to be uniformly distributed. Various sensors, including infrared, ground penetrating radars, and synthetic aperture radars, have been modeled off line using (10) for demining, surveillance, and radar target tracking applications [25], [26], [28].

B. Modeling of Mobile Targets

There is considerable precedence in the sensor tracking and estimation literature for modeling target tracks by piecewise Markov motion models in order to estimate the target state from multiple, distributed sensor measurements [29]. Let $\mathcal{T}_l \subset \mathcal{W}$ denote the geometry of the l^{th} target, which is assigned a unique identifier $l \in \mathcal{I}_\mathcal{T}$ using an existing multisensor-multitarget assignment algorithm [29]. Each target track is estimated from the set of sensor measurements, $M(k) = \{m_1(1), m_2(1), \ldots, m_{N-1}(k), m_N(k)\}$, obtained by the network up to time k, when $\mathcal{T}_l \cap \mathcal{S}_i \neq \emptyset$.

The motion of each target is modeled as a continuous-time Markov process, where, we say that x_t is a *continuous-time Markov process* if for $0 \leq t_0 < \cdots < t_{k-1} < t_k < t$ we have $\Pr(x_t \in B | x_k = s_k, x_{k-1} = s_{k-1}, \cdots, x_0 = s_0) = \Pr(x_t \in B | x_k = s_k)$ where Pr denotes the probability function, and $s_1, \ldots, s_k \in \mathcal{X}$ are realizations of the state space $\mathcal{X}$. Now, let the random variables $\theta^{\mathcal{T}_l}$ and $v^{\mathcal{T}_l}$ represent the l^{th} target's heading and velocity, respectively. Assuming the target's heading and velocity are constant during every interval $\Delta t_\jmath = (t_{\jmath+1} - t_\jmath)$, $\jmath = 1, 2, \ldots$, where $\Delta t_\jmath$ is not necessarily constant, the target motion can be modeled as a continuous-time Markov process with a family of random variables $\{x^{\mathcal{T}_l}, \theta^{\mathcal{T}_l}, v^{\mathcal{T}_l}\}$, where $x^{\mathcal{T}_l} \in \mathcal{W}$ is the l^{th} target position at t. Then, a three-dimensional real-valued vector function maps the family of random variables $\{\theta^{\mathcal{T}_l}(t), v^{\mathcal{T}_l}(t)\}$ into the random vector $x^{\mathcal{T}_l}(t)$ at every time $t \in [t_0,\ t_f]$, such that the value of the target motion process is given by,

$$\dot{x}^{\mathcal{T}_l}(t) = v^{\mathcal{T}_l}(t) \begin{bmatrix} \cos\theta^{\mathcal{T}_l}(t) \\ \sin\theta^{\mathcal{T}_l}(t) \end{bmatrix}, \tag{11}$$

and, therefore, the motion of target l is a Markov process. The third component of the vector function is the identity function. It follows that $\theta^{\mathcal{T}_l}$ and $v^{\mathcal{T}_l}$ are piece-wise constant, while $x^{\mathcal{T}_l}$ has discontinuities at the time instants $t_\jmath$, $\jmath = 1, \ldots, \rho$, when target l changes its heading and velocity.

Let $\{x_\jmath^{\mathcal{T}_l}\}_{\jmath=1,\ldots,\rho}$ denote the set of target positions at which these discontinuities occur. Then, by integrating the linear differential equation (11) over every interval $\Delta t_\jmath$ with initial condition $x_\jmath^{\mathcal{T}_l}$, the position of the l^{th} target at any time t can be obtained as a function of the sequence of random variables $\{x_\jmath^{\mathcal{T}_l}, v_\jmath^{\mathcal{T}_l}, \theta_\jmath^{\mathcal{T}_l}\}_{\jmath=1,\ldots,\rho} \equiv \{\mathcal{P}_\jmath\}_{\jmath=1,\ldots,\rho}$ also known as Markov motion parameters,

$$x^{\mathcal{T}_l}(t) = x_\jmath^{\mathcal{T}_l} + v_\jmath^{\mathcal{T}_l}(t - t_\jmath) \begin{bmatrix} \cos\theta_\jmath^{\mathcal{T}_l} \\ \sin\theta_\jmath^{\mathcal{T}_l} \end{bmatrix}, \ t_\jmath \leq t < t_{\jmath+1} \tag{12}$$

where the Markov motion parameter values $\mathcal{P}_\jmath$ only depend on the values of the previous time step $\mathcal{P}_{\jmath-1}$, and remain constant during the time interval $\Delta t_\jmath$. It follows that the target motion is properly represented by the joint probability density function on $\mathcal{P}_\jmath$, denoted by $\Pr(\mathcal{P}_\jmath | \mathcal{P}_{\jmath-1})$ for $\jmath = 1, \ldots, \rho$.

The first two components of $\mathcal{T}_i$ are piece-wise linear, while the third component has discontinuities at time instants when a change in direction happens. The implemented time span between two consecutive instants is uniformly distributed in interval $[T_{i,\min}, T_{i,\max}]$ determined by the user's preferences. A change of direction is uniformly distributed in the interval $(\theta^{\mathcal{T}_i}(t_0) - \pi/2,\ \theta^{\mathcal{T}_i}(t_0) + \pi/2)$. Therefore, $\theta^{\mathcal{T}_i}(t)$ is a uniformly distributed random variable for countably many time instants t (*i.e.*, a sequence of independent identically distributed random variables), and otherwise is deterministic (holds a value of the last change of the target's direction). The maximum translational speed of all sensors and targets is known, and that while the sensors can move with any speed in $[0, v_{\max}]$, it is assumed that the speed of every target is uniformly distributed in $[v_{\min}^{\mathcal{T}}, v_{\max}^{\mathcal{T}}]$, with $v_{\min}^{\mathcal{T}} > 0$ and $v_{\max}^{\mathcal{T}} < v_{\max}$.

In summary, the set of targets detected by the sensor network in $\mathcal{W}$ is the tuple $\{\mathcal{I}_\mathcal{T}, \mathcal{T}_l, Z, \Pr(\mathcal{P}_\jmath | \mathcal{P}_{\jmath-1})\}_{\forall \jmath,\ l \in \mathcal{I}_\mathcal{T}}$. Under these assumptions, the spatio-temporal geometry of a *C-target* can be obtained by sliding $\mathcal{T}_l$ along the track (12), and the joint PDF on $\mathcal{P}_\jmath$ used to evaluate the value of information, as explained in the next section.

IV. METHODOLOGY

A fundamental paradigm utilized by obstacle avoidance algorithms [13] is that of *C-obstacle*, which consists of the subset of $\mathcal{C}$ that causes collisions with at least one obstacle in $\mathcal{W}$, *i.e.*, $\mathcal{CB}_i \equiv \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{B}_i \neq \emptyset\}$, where $\mathcal{A}(q)$ denotes the subset of $\mathcal{W}$ occupied by the vehicle geometry $\mathcal{A}$ when the robot is in the configuration q. The union $\bigcup_{j=1}^{n} \mathcal{CB}_j$ is the *C-obstacle region*, and the obstacle-free robot configuration space $\mathcal{C}_{free}$ is defined as the complement of the *C-obstacle* region in $\mathcal{C}$. Therefore, any robot configuration chosen from $\mathcal{C}_{free}$ avoids intersections between the vehicle geometry $\mathcal{A}$ and the obstacles' geometries. In the presence of targets, q must both avoid intersections between $\mathcal{A}$ and the obstacles, and enable intersections between the sensor's FOV and the targets' geometries in order to make sensor measurements. In

this paper, a target is treated as the dual of an obstacle, and the subset of $\mathcal{C}$ that enables intersections with $\mathcal{T}_l$ is identified and computed by introducing the following definitions.

The *field-of-view* of a sensor mounted on $\mathcal{A}$ is a closed and bounded subset $\mathcal{S}(q) \subset \mathcal{W}$ such that the measurement set of a target located at any point $p \in \mathcal{S}(q)$ can be obtained by the sensor when the robot occupies the configuration $q \in \mathcal{C}$. Then, the target $\mathcal{T}_l$ in $\mathcal{W}$ maps in the robot's configuration space, $\mathcal{C}$, to the *C-target region* $\mathcal{CT}_l = \{q \in \mathcal{C} \mid \mathcal{S}(q) \cap \mathcal{T}_l \neq \emptyset\}$. That leads to the following definition and proposition.

Definition 4.1: The target $\mathcal{T}_j$ in $\mathcal{S}$ maps in the i^{th} sensor configuration space $\mathcal{C}$ to the *C-target* region $\mathcal{CR}_j = \{q_i \in \mathcal{C} \mid \Pr\{\mathcal{S}_i \cap \mathcal{T}_i\} > \epsilon, \forall t \geq \tau, i \in \mathcal{I}_\mathcal{P}, j \in \mathcal{I}_\mathcal{T}\}$.

It was shown in [30] that the following proposition holds under the aforementioned assumptions:

Proposition 4.2: A C-target can be approximated by a cone-like area.

The sensors' goals are represented by an adaptive and additive potential function U^ι that is either attractive or repulsive, and is indexed by $\iota \in \mathcal{I}_K$. The goal of avoiding obstacles is indexed by $\iota = 1$, and the goal of target tracking by intersecting the target trajectory is indexed by $\iota = 2$. Where the information value of $\mathcal{T}_l$, denoted by $\mathcal{H}_l$, is defined as the expected entropy reduction of z_l, conditioned on $M(k)$, and can be computed from (10) as shown in [21]. Then, all attractive potentials generated by the targets are combined with all repulsive potentials generated by the obstacles, and the sensor's potential field is given by,

$$U(q,t) = \sum_{\iota \in \{a^*, u^*\}} U^\iota(q,t), \tag{13}$$

$$U(q,t) = \sum_{j \in \mathcal{I}_\mathcal{B}} U_j^{(\iota=1)}(q,t)_{rep} + \sum_{l \in \mathcal{I}_\mathcal{T}} U_l^{(\iota=2)}(q,t)_{att}.$$

Based on the concept of *C-targets*, a novel attractive potential function is defined for the targets as follows

$$U_l(q,t)_{att} = \eta\ \beta \mathcal{H}_l^\alpha(t) \left\{ 1 - \exp\left[-\frac{\rho_l^2(q,t)}{2\beta \mathcal{H}_l^\alpha(t)} \right] \right\}, \tag{14}$$

where

$$\rho_l(q,t) \equiv \min_{x \in \mathcal{CT}_l} ||W(x-q)||,$$

η, α, and β are user-defined scaling parameters, $||\cdot||$ is the Euclidean norm, and W is a diagonal and positive definite weighting matrix representing the importance of changes in position and orientation. It can be shown that this novel potential function satisfies the following properties: (i) It is an increasing function of the shortest distance from $\mathcal{T}_l$, denoted by ρ_l; (ii) when $\rho_i \to \infty$ all targets generate the same potential; and, (iii) given the same distance between two targets, the target with the higher information value has lower (more attractive) potential, as well as a steeper gradient for the same robot configuration q.

The potential functions express the local navigation objectives, also known as immediate reward in the ADP equations. At the mission-planning level each robotic sensor is represented by a physical graph, or *roadmap*, that captures the physical and geometric constraints on the dynamics, control, and sensing of each robotic sensor globally over the workspace. Using the method of cell decomposition proposed in [21], the sensor's roadmap can be obtained from the geometries of the *C-obstacles* and *C-targets* in $\mathcal{W}$. The roadmap is a directed graph, $G_i = (C_i, E_i)$, where $C_i = \{c_1, c_2, \ldots\}$ is a set of nodes representing robot configurations sampled from the free configuration space, and E_i is a set of edges, where an edge $(c_i, c_j) \in E_i$ represents an obstacle-free reachable path from c_i to c_j. Given the initial i^{th} robotic sensor's configuration $q_{i_0} = q_i(t = k)$ for $k = 1$, the corresponding node in G_i can be tagged, and used as the root of two connectivity trees $T_i = (\mathcal{N}_i, \mathcal{E}_i)$ that are grown by connecting q_{i_0} to its adjacent nodes in the respective graphs. Each adjacent node thereafter is connected to its adjacent nodes, and each tree is pruned incrementally using the principle of optimality [11], [21].

The roadmap is then folded into a *dynamic Bayesian network* (DBN) representation of the physical state evolution, with transition probabilities that are either equal to zero or one depending on the values in the adjacency matrix of G_i. Nodes sampled from a *C-target* are also tagged and augmented with a child node comprised of the corresponding target variable, indicating it can be measured when the sensor is in its parent state. Finally, by augmenting the DBN with the action and decision nodes, and the potentials defined in (13) and (14), the final model of the DEP in (3) is obtained and utilized to compute the optimal strategy for future action and test decisions using, for example, the following Q-learning rule,

$$Q[\xi(k), a(k), u(k)] \leftarrow (1-\phi) Q[\xi(k), a(k), u(k)] + \tag{15}$$
$$\phi\{U[\xi(k), k] + \gamma^k \max_{u(k), a(k) \in \mathcal{U}} Q[\xi(k+1), a(k), u(k)]\},$$

where $\phi \in (0,1)$, $R[\cdot] = U[\cdot]$ is the immediate reward evaluated by the potential field in (13), and γ^k is the discount factor. Then, assuming two functions $Q(\cdot)$ and $V(\cdot)$ that satisfy Bellman equation exist, they specify an optimal *greedy* navigation policy σ^*, where:

$$V[\xi(k+1)] = \tag{16}$$
$$\max_{u(k+1), a(k+1) \in \mathcal{U}} Q[\xi(k+1), a(k+1), u(k+1)],$$

and

$$\gamma_k^* = \arg\max\{Q[\xi(k), a(k), u(k)]\}.$$

The mobile sensor's roadmap and potential field in (13) are used to estimate the optimal future reward Q at $k+1$ and the immediate reward at k, respectively.

V. SIMULATION RESULTS

To illustrate the above methodology and cooperation between heterogeneous agents, we present a scenario whereby the effect of the sensors' *field-of-view* (FOV) on the performance of the system utilizing the methods presented in this paper can be explored. We employ two agent types. The first is a quadrotor type unmanned aerial vehicle (UAV), modeled in simulation using holonomic kinematics, which

is unimpeded by obstacles in the environment and has a downward looking sensor with large range and wide FOV. The second type of agent is an unmanned ground vehicle (UGV), modeled with nonholonomic kinematics. The UGV has a forward looking directional sensor with limited FOV. Both sensors are modeled on a normal distribution, with the UGV sensor more accurate than the UAV sensor, but with reduced range.

Both sensors are working in a square environment 10×10 m^2 to search an area as directed by a higher level motion planner for potential targets at certain high probability locations in the environment. These simulations focus on cooperative navigation of the UGV to the points of interest while avoiding obstacles using combined measurement data from both sensors.

In the scenario, the parameters for the agents are picked to simulate cheaper sensors that are less accurate, such as might be used on a production unit. The UAV sensor has a coverage radius of 1.5 m, a FOV of 360° (in the 2D projection), and zero-mean measurement noise with $\sigma = 0.15$. The UGV sensor parameters are a coverage radius of 0.75 m, a 45° FOV, and zero-mean measurement noise with $\sigma = 0.05$.

The UGV utilizes a simplified version of the potential field navigation strategy given by (13). The UAV simply follows a proportional controller to *pre-fly* the waypoint path to be traversed by the ground vehicle, at the same time the ground vehicle navigates the environment. Because the UAV has a greater velocity than the ground vehicle it visits the waypoints first, receiving noisier initial readings of the obstacle locations. We currently implement a *last measurement only* format for the UGV navigation, where, in order of primary selection, a measurement is a UGV sensor measurement, a UAV sensor measurement if no UGV measurement is available, and if no previous measurement is available, no measurement data is stored. We intend to show that the dynamic programming approach to this hybrid system framework will improve the performance of the UGV navigation.

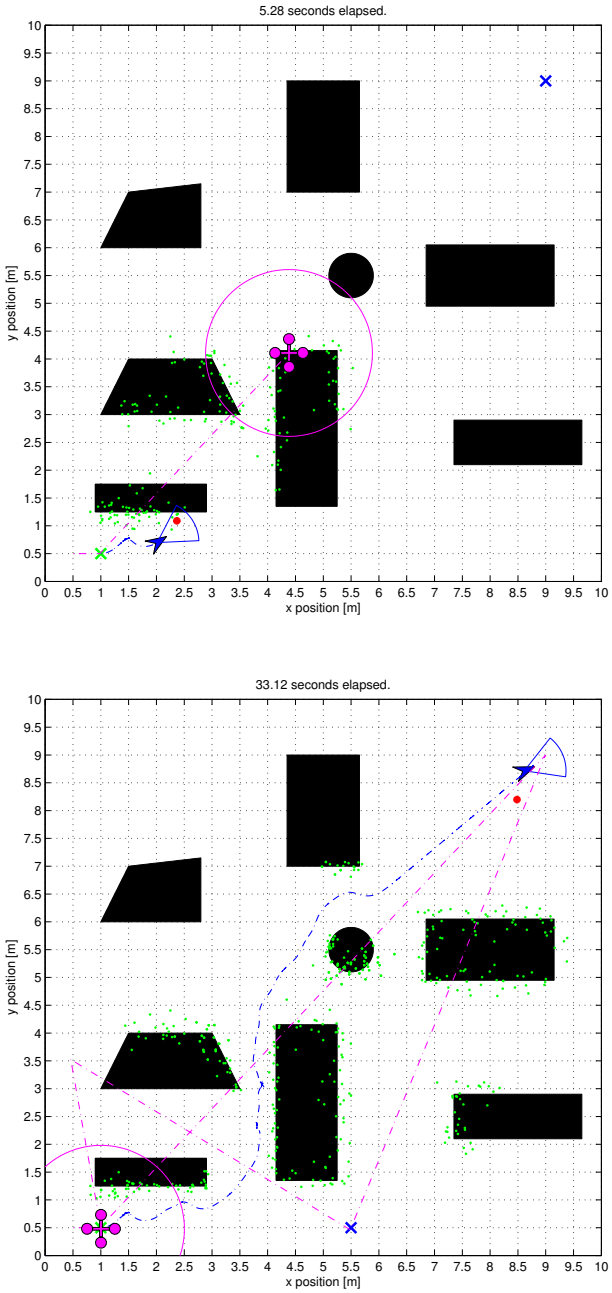

Fig. 2. **Scenario:** Black objects are obstacles, green dots are measurements of the obstacles, the blue arrow represents the UGV with a directional FOV outlined in blue, magenta is the quadrotor UAV with circular coverage area in the 2D projection outlined by the magenta circle. Both UAV and UGV started in the bottom left corner, with the dotted lines showing their path from initial position. After 33 s, note the noisier measurements around obstacles that the UAV has seen but the UGV has not, for example the rectangular object at the bottom right corner. Also note the higher accuracy measurements of obstacles that the UGV has visited (rectangle in the bottom left).

VI. CONCLUSIONS

In this paper we present a hybrid system framework and outline an approximate dynamic programming approach to cooperative navigation for heterogeneous sensor networks. It is assumed that the ground vehicle has a small sensor footprint compared with the sensor onboard the aerial vehicle. The proposed navigation strategy combines artificial potential functions for target pursuing with ADP for learning *C-obstacles* on line. The controller for the UGV is developed from the potential functions computed using the available information provided by heterogeneous sensors, and then iteratively refined as the potential functions are refined when more precise information becomes available.

In this work, the mission planner is initially modeled by a centralized discrete-event coordinator that resides on a remote computational facility or onboard one of the vehicles. Investigating multiple, decentralized mission planners is part of our future research agenda. Finally, we are also interested in considering more advanced ADP and online learning algorithms that can be integrated in the networked hybrid dynamical system proposed herein.

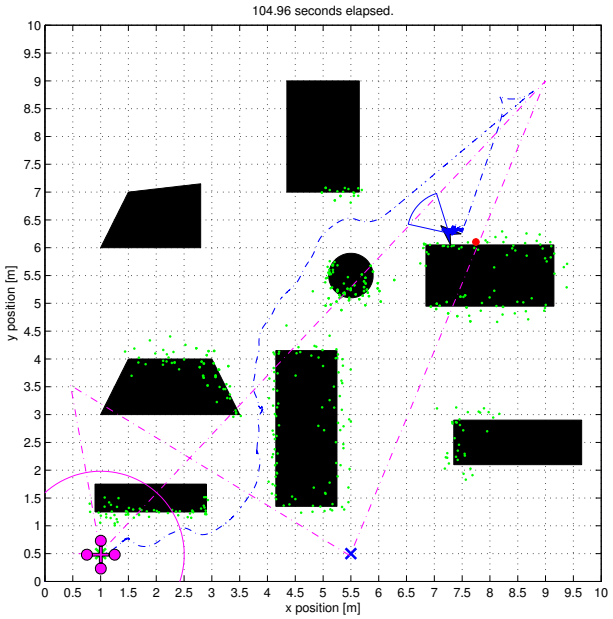

Fig. 3. **Scenario:** With limited FOV, the ground vehicle is slow navigating the obstacle course. After 105 s seconds have elapsed, note that UGV is still attempting to navigate around the obstacle between the first and second goal point. The ground vehicle is stuck in this location due to local minima along the obstacle's edge. The UGV was not able to visit all the target/goal points.

ACKNOWLEDGMENTS

This work was supported by NSF ECCS grant #1027775, and by the Department of Energy URPR Grant #DE-FG52-04NA25590.

REFERENCES

[1] R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J. P. Ostrowski, G. Pappas, C. J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic, and J. Southall, "A framework and architecture for multi-robot coordination," *Int. J. Robot. Research*, vol. 21, no. 10-11, pp. 977–995, Oct-Nov 2002.

[2] M. Boccadoro, P. Valigi, and Y. Wardi, "A method for the design of optimal switching surfaces for autonomous hybrid systems," in *Lecture Notes in Computer Science*, 2007, pp. 650–655.

[3] M. Shaikh and P. Caines, "On the optimal control of hybrid systems: Optimization of trajectories, switching times, and location schedules," *Lecture Notes in Computer Science*, vol. 2623, pp. 466–481, 2003.

[4] C. Cassandras, C. Pepyne, and Y. Wardi, "Generalized gradient algorithms for hybrid system models of manufacturing systems," in *IEEE Conference on Decision and Control*, 1998, pp. 2627–2632.

[5] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "Optimal control synthesis of a class of nonlinear systems using single network adaptive critics," *Neural Networks*, vol. 19, no. 1, pp. 1648–1660, 2006.

[6] P. Antsaklis, M. Lemmon, and J. Stiver, "Modeling and design of hybrid control systems," *Proc. IEEE Mediterranean Symp. New Directions Control Automation,*, p. 440447, 1994.

[7] R. Brockett, "Hybrid models for motion control systems," in *Essays on Control: Perspectives in the Theory and its Applications*, Boston, MA, 1993, pp. 29–53.

[8] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "On synchronous robotic networks. part i: Models, tasks and complexity notions," in *Proc. IEEE Conf. on Decision and Control, and the European Control Conf.*, Seville, Spain, December 12-15 2005, pp. 2847–2852.

[9] M. Zavlanos and G. Pappas, "Distributed hybrid control for multiple pursuer multiple evader games," in *10th International Conference on Hybrid Systems: Computation and Control*, Pisa, Italy, April 2007, pp. 787–789.

[10] R. Sanfelice and E. Frazzoli, "A hybrid control framework for robust maneuver-based motion planning," in *Proc. American Control Conference*, Seattle, WA, 2008, pp. 2254–2259.

[11] C. Cai and S. Ferrari, "Information-driven sensor path planning by approximate cell decomposition," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. 39, no. 2, 2009.

[12] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[13] J. C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[14] S. Koenig, C. Tovey, and Y. Smirnov, "Performance bounds for planning in unknown terrain," *Artificial Intelligence*, vol. 147, pp. 253–279, 2003.

[15] N. Rao, "Robot navigation in unknown generalized polygonal terrains using vision sensors," *IEEE Transactions on System, Man, and Cybernetics*, vol. 25, no. 6, pp. 947–962, 1995.

[16] S. Thurn, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, pp. 21–71, 1998.

[17] J. C. Latombe, A. Lazanas, and S. Shekhar, "Robot motion planning with uncertainty in control and sensing," *Artificial Intelligence*, vol. 52, pp. 1–47, 1991.

[18] E. U. Acar, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *International Journal of Robotic Research*, vol. 22, pp. 7–8, 2003.

[19] W. B. Powel, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, ser. Probability and Statistics. Hoboken, NJ: WILEY, 2007.

[20] R. Fierro and F. L. Lewis, "A framework for hybrid control design," *IEEE Trans. on Syst., Man, and Cyber.*, vol. 27-A, no. 6, pp. 765–773, Nov. 1997.

[21] S. Ferrari and C. Cai, "Information-driven search strategies in the board game of CLUE®," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. 39, no. 2, 2009.

[22] S. Ge and Y. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.

[23] E. Rimon and D. Kodischek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[24] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems Magazine*, vol. April, pp. 51–64, 2008.

[25] C. Kreucher, K. Kastella, and A. Hero, "Sensor management using an active sensing approach," *Signal Processing*, vol. 85, pp. 607–624, 2005.

[26] S. Ferrari and A. Vaghi, "Demining sensor modeling and feature-level fusion by bayesian networks," *IEEE Sensors*, vol. 6, pp. 471–483, 2006.

[27] G. Zhang, S. Ferrari, and M. Qian, "An information roadmap method for robotic sensor path planning," *Journal of Intelligent and Robotic Systems*, vol. 56, pp. 69–98, 2009.

[28] C. Cai, S. Ferrari, and Q. Ming, "Bayesian network modeling of acoustic sensor measurements," in *Proc. IEEE Sensors*, Atlanta, GA, 2007, pp. 345–348.

[29] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: Algorithms and software for information extraction*. John Wiley and Sons, 2001.

[30] D. Tolic, R. Fierro, and S. Ferrari, "Cooperative multi-target tracking via hybrid modeling and geometric optimization," in *16th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, June 2009, pp. 440–445, invited.