

## Research Article

# Guidance of Autonomous Amphibious Vehicles for Flood Rescue Support

Shankarachary Ragi,<sup>1</sup> ChingSeong Tan,<sup>2</sup> and Edwin K. P. Chong<sup>1</sup>

<sup>1</sup> Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA

<sup>2</sup> Faculty of Engineering, Multimedia University, Malaysia

Correspondence should be addressed to Shankarachary Ragi; [shankar.ragi@colostate.edu](mailto:shankar.ragi@colostate.edu)

Received 14 February 2013; Revised 6 May 2013; Accepted 7 May 2013

Academic Editor: Yongping Li

Copyright © 2013 Shankarachary Ragi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We develop a path-planning algorithm to guide autonomous amphibious vehicles (AAVs) for flood rescue support missions. Specifically, we develop an algorithm to control multiple AAVs to reach/rescue multiple victims (also called targets) in a flood scenario in 2D, where the flood water flows across the scene and the targets move (drifted by the flood water) along the flood stream. A target is said to be rescued if an AAV lies within a circular region of a certain radius around the target. The goal is to control the AAVs such that each target gets rescued while optimizing a certain performance objective. The algorithm design is based on the theory of *partially observable Markov decision process* (POMDP). In practice, POMDP problems are hard to solve exactly, so we use an approximation method called *nominal belief-state optimization* (NBO). We compare the performance of the NBO approach with a greedy approach.

## 1. Introduction

Various guidance algorithms for autonomous amphibious vehicles (AAVs) are being designed and tested to fight today's global warming disasters such as flooding, typhoon, and hurricane [1–3]. With this motivation, we present a guidance framework to control multiple AAVs to rescue multiple victims (henceforth called targets) in a flood situation, where the flood water (interchangeably called river) flows along a valley as shown in Figure 1. A target is said to be rescued when an AAV is within the circular region of radius  $d_{\text{dist-thresh}}$  on the 2D plane around the target. In general, AAVs are equipped with various advanced sensors such as polarized stereo vision, laser scanning, and SONAR [4–6]. The sensors onboard an AAV generate the (noisy) measurements corresponding to the targets and the river. Our goal is to design a path-planning algorithm that guides the AAVs so that every target gets rescued, while maximizing a performance measure (discussed later). The algorithm runs on a notional central fusion node, which collects the measurements from the sensors on-board each AAV, fuses them and updates the tracks on the targets and the river state (discussed later),

computes the control commands for the AAVs, and sends the control commands back to the AAVs.

Guidance control methods [1, 7–9] for AAVs are normally based on a standard three-layered system architecture that requires human-machine interactions. We design the guidance algorithm based on the theory of *partially observable Markov decision process* (POMDP) [10, 11]. There are several other autonomous control methods in the literature for AAVs and underwater vehicles, for example, [12–14]. Our approach differs from these existing approaches in that we place the guidance problem in the context of POMDP, wherein this approach has a look-ahead property, which trades off short-term for long-term performance.

## 2. Problem Specification

The AAV guidance problem is specified as follows.

**2.1. Targets.** In this study, we assume that there are multiple mobile targets (flood victims) located in a river, being drifted down by the flood water, as shown in Figure 1.

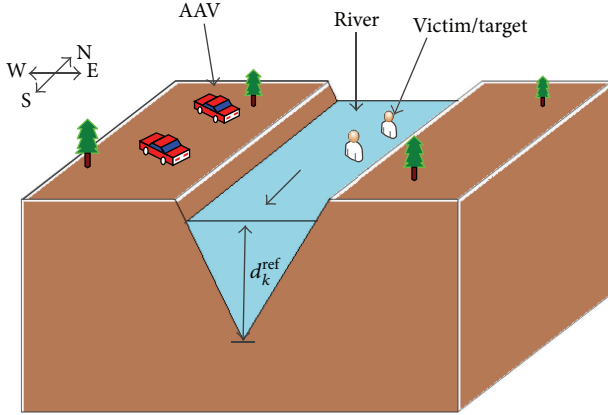


FIGURE 1: Flood scenario.

**2.2. Autonomous Amphibious Vehicles (AAVs).** There are multiple autonomous amphibious vehicles (AAVs) located on the shore, as shown in Figure 1. An AAV is controlled by the following kinematic controls: forward acceleration and steering angle. Each AAV is equipped with on-board sensors that generate measurements of targets and the river depth. In this problem, AAVs float when moving in the river. For the purpose of this study, we assume that the number of AAVs and the number of targets are the same.

**2.3. Environmental Conditions.** The elevation map of the region is known a priori. The landscape for this problem is shown in Figure 1, which shows a river flowing along a valley from the north toward the south. The state of the river includes the depth  $d_k^{ref}$  at a reference point on the map (lowest point in the landscape, e.g., some location at the bottom of the valley as shown in Figure 1).

**2.4. River Model.** Typically a river flows slowly near the coastlines (where the river is shallow) and flows quickly far from the coastlines (i.e., toward the center of the river where the river is deep). In this paper, we assume that the river flows from the north toward the south in a v-shaped channel as shown in Figure 1. We adopt the *logarithmic velocity profile* to model the velocity of the flow (see [15] for a detailed description). According to this model, the speed of the river, at the surface, at the location  $(p, q)$  at time  $k$  is given by

$$w_k(p, q) = C_1 [\log(d_k(p, q)) + C_2], \quad (1)$$

where  $d_k(p, q)$  is the depth of the river at the location  $(p, q)$  at time  $k$ , and  $C_1$  (a function of the viscosity and the density of flood water) and  $C_2$  are constants (see [15] for more details).

**2.5. Observations.** The sensors onboard an AAV generate noisy observations of target locations and the depth of the river directly beneath the vehicle, that is, the sensors generate the observations of the depth of the river only when the AAV is in the river.

**2.6. Objective.** A target is said to be rescued if there is an AAV within a circular region of radius  $d_{\text{dist-thresh}}$  around the target. The objective is to minimize the average rescue time, where the average is over the number of targets, and the rescue time of a target is defined as the time it takes to rescue the target.

### 3. Problem Formulation

We cast the AAV guidance problem into the framework of a *partially observable Markov decision process* (POMDP). A POMDP is a mathematical framework useful for solving resource control problems and enables us to exploit approximation methods for POMDPs to design our AAV guidance algorithm. A POMDP evolves in discrete time steps. We use  $k$  as the discrete-time index. To cast the AAV guidance problem into the POMDP framework, we need to define the following key components in terms of our guidance problem as follows.

**3.1. States.** Let  $x_k$  represent the state of the system at time  $k$ . The state of the system includes the state of the vehicles (AAVs)  $s_k$ , river state (depth of the river at a reference location)  $d_k^{ref}$ , target state  $\chi_k$ , and track states  $(\xi_k^{riv}, P_k^{riv}, \xi_k^{targ}, P_k^{targ})$ , that is,  $x_k = (s_k, d_k^{ref}, \chi_k, \xi_k^{riv}, P_k^{riv}, \xi_k^{targ}, P_k^{targ})$ . The vehicle state  $s_k$  includes the locations and the velocities of the AAVs at time  $k$ . The river state  $d_k^{ref}$  is the depth of the river at the reference point at time  $k$ . The reference point is the lowest point in the elevation map, that is, some location at the bottom of the valley in the landscape, as shown in Figure 1. Here, we assume that the flow direction of the river is the same everywhere and is known a priori. The target state  $\chi_k$  includes the locations and the velocities of the targets at time  $k$ . The track states represent the state of the tracking algorithm, where  $\xi_k^{riv}$  and  $P_k^{riv}$  are the mean and the variance, standard in Kalman filter equations, corresponding to the river state, and, similarly,  $\xi_k^{targ}$  is the mean vector and  $P_k^{targ}$  is the covariance matrix corresponding to the target state.

**3.2. Observations and Observation Law.** The vehicle and the track states are assumed to be fully observable. The river and the target states are only partially observable. The observation of the river state at an AAV is given by

$$z_k^{riv} = \begin{cases} d_k^{ref} + n_k^{riv} & \text{if AAV is in river,} \\ \text{no measurement} & \text{otherwise,} \end{cases} \quad (2)$$

where  $n_k^{riv} \sim \mathcal{N}(0, R_k)$ , and  $R_k$  is the measurement variance. The sensors at an AAV generate the measurement of the river state only when the AAV is in the river. In practice, the sensors on an AAV measure the depth of the river exactly below the AAV. We wrote the observation model (2) as if the sensors are generating the observations of the depth of the river at the reference point. The rationale behind this assumption is that we can always calculate the depth of the river at the reference point given the elevation map and the observed depth of

the river at a different location. The observation of the  $j$ th target at an AAV is given by

$$z_k^{\chi^j} = \begin{cases} H\chi_k^j + n_k^{\text{targ}} & \text{if there is line-of-sight,} \\ \text{no measurement} & \text{otherwise,} \end{cases} \quad (3)$$

where  $H$  is the target-state observation model,  $\chi_k^j$  is the state of  $j$ th target, and  $n_k^{\text{targ}} \sim \mathcal{N}(0, S_k)$ , where  $S_k$  is the measurement covariance matrix. The line-of-sight between the target and the AAV is blocked sometimes, for example, whenever the target sinks in the water.

**3.3. Actions.** The actions include the controllable aspects of the system. In this problem, the actions include the decisions on the assignment of AAVs to targets, and kinematic control commands for AAVs. Let  $u_k$  be the action tuple at time  $k$ , which is given by  $u_k = (g_k, a_k)$ , where  $a_k$  represents kinematic control vectors (includes forward acceleration and steering angle for each AAV), and  $g_k$  is a vector, which represents the assignment of AAVs to targets, that is,  $g_k(i) = j$  means that the  $i$ th AAV is assigned to the  $j$ th target. For the purpose of this study, the number of AAVs and the targets is the same. Each AAV is assigned to only one target, and each target gets assigned only one AAV, that is,  $g_k$  represents a one-to-one correspondence between the AAVs and the targets.

**3.4. State-Transition Law.** The state-transition law specifies the next-state distribution given the current state and the action. The transition function for the vehicle state is given by  $s_{k+1} = \psi(s_k, a_k, \xi_k^{\text{riv}})$ , where  $\psi$  (defined later) represents the AAV kinematic model,  $s_k$  is the vehicle state,  $a_k$  is the kinematic control vector (includes forward acceleration and steering angle), and  $\xi_k^{\text{riv}}$  is the estimated river state at time  $k$ . The river state evolves according to the following equation:

$$d_{k+1}^{\text{ref}} = d_k^{\text{ref}} + o_k, \quad o_k \sim \mathcal{N}(0, U_k^{\text{riv}}), \quad (4)$$

where  $U_k^{\text{riv}}$  is the process variance corresponding to the river state evolution. The target state evolves according to

$$\chi_{k+1} = F\chi_k + e_k, \quad e_k \sim \mathcal{N}(0, U_k^{\text{targ}}), \quad (5)$$

where  $F$  represents the target motion model, and  $U_k^{\text{targ}}$  is the process covariance matrix corresponding to the target state evolution. The track states evolve according to the Kalman filter equations given the observations from the sensors onboard the AAVs. When the observations are not available, the track states evolve according to the Kalman filter equations, where only the *prediction step* is performed and the *update step* is not performed.

**3.5. Cost.** The cost function represents the cost of performing an action at the current state. The cost function is given by

$$C(x_k, u_k) = \sum_{i=1}^N \mathbb{1} \left\{ \mathbb{E} \left[ \left\| s_{k+1}^{i, \text{pos}} - \xi_{k+1}^{g_k(i), \text{targ, pos}} \right\| \mid x_k, u_k \right] > d_{\text{dist-thresh}} \right\}, \quad (6)$$

where  $s_{k+1}^{i, \text{pos}}$  represents the 2D position coordinates of  $i$ th AAV,  $\xi_{k+1}^{j, \text{targ, pos}}$  represents the estimated 2D position coordinates of the  $j$ th target at time  $k+1$ ,  $\|\cdot\|$  is the Euclidean norm (everywhere in this paper), and  $\mathbb{1}\{\cdot\}$  is the indicator function which equals 1 when the expected distance between the AAV and the target at time  $k+1$  is greater than some threshold distance  $d_{\text{dist-thresh}}$  and 0 otherwise.

**3.6. Belief State.** The belief state  $b_k$  is the posterior distribution of the state at time  $k$ . The vehicle and the track states are assumed to be fully observable, that is, the belief state corresponding to the vehicle state is given by  $b_k^s(s) = \delta(s - s_k)$ , where  $\delta(\cdot)$  is the Kronecker delta function. Similarly, the belief states corresponding to the track states can be written in terms of the actual track states. The belief states corresponding to the river and the target are the posterior distributions of  $d_k^{\text{ref}}$  and  $\chi_k$ , respectively, given the history of observations.

## 4. Objective and Optimal Policy

The goal is to find the action sequence  $(u_0, u_1, \dots, u_{H-1})$  such that the expected cumulative cost over a time horizon  $H$  is minimized. The expected cumulative cost is given by

$$J_H = \mathbb{E} \left[ \sum_{k=0}^{H-1} C(x_k, u_k) \right]. \quad (7)$$

We can write the expected cumulative cost in terms of the belief states given the initial belief state  $b_0$  (similar to the treatment in [10, 11]) as follows:

$$J_H(b_0) = \mathbb{E} \left[ \sum_{k=0}^{H-1} c(b_k, u_k) \mid b_0 \right], \quad (8)$$

where  $c(b_k, u_k) = \int C(x, u_k) b_k(x) dx$ , and  $b_0$  is the belief state at time  $k=0$ . From Bellman's principle of optimality [16], the optimal objective function value is given by

$$J_H^*(b_0) = \min_u \{c(b_0, u) + \mathbb{E}[J_{H-1}^*(b_1) \mid b_0, u]\}, \quad (9)$$

where  $b_1$  is the random next belief state,  $J_{H-1}^*$  is the optimal cumulative cost over the horizon  $H-1$ ,  $k=1, 2, \dots, H-1$ , and  $\mathbb{E}[\cdot \mid b_0, u]$  is the conditional expectation given the current belief state  $b_0$  and the current action  $u$  at time  $k=0$ . Let us define the Q value of taking action  $u$  given the current belief state  $b_0$ :

$$Q_H(b_0, u) = c(b_0, u) + \mathbb{E}[J_{H-1}^*(b_1) \mid b_0, u]. \quad (10)$$

The optimal policy (from Bellman's principle) at time  $k=0$  can be written as

$$\pi_0^*(b_0) = \arg \min_u Q_H(b_0, u). \quad (11)$$

In general, it is hard to obtain the Q value exactly. There are several approximation methods in the literature: heuristic expected-cost-to-go (ECTG) [17], parametric approximation

[18], policy rollout [19], hindsight optimization [20], and foresight optimization [21]. In this paper, we use one such approximation method called *nominal belief-state optimization* (NBO), which was introduced in [11] along with other approximations and techniques specific to guidance problems. The rationale behind choosing NBO method over other methods to solve POMDP is that it is relatively inexpensive in terms of computation time, that is, the computational requirements are not prohibitive unlike other approximation methods. The following subsection provides a brief description of the NBO method.

**4.1. NBO Approximation Method.** The computational requirements of obtaining the optimal assignments of AAVs to targets ( $g_k$ ) over a long horizon are prohibitive. Also, we expect that the optimal assignment of AAVs to targets ( $g_k$ ) over a long horizon does not change with time. For these reasons, in the NBO method, we keep the assignment of AAVs to targets fixed. In other words, in approximating the expected cost-to-go in (10),  $g_k$  remains fixed over the planning horizon  $H$ . Therefore, we drop the subscript  $k$  from  $g_k$  in the objective function used in the planning based on (10), that is,  $g_k = g$  for all  $k$ . In the NBO approximation method, we use the following objective function, written in terms of belief states:

$$J_H(b_0) = \mathbb{E} \left[ \sum_{k=0}^{H-1} c(b_k, a_k, g) \mid b_0 \right], \quad (12)$$

where  $a_k$  represents the kinematic controls for the AAVs, and  $g$  is the assignment of AAVs to the targets.

The belief states corresponding to the river state and the target state are given by

$$\begin{aligned} b_k^{\text{riv}}(d) &= \mathcal{N}(d - \xi_k^{\text{riv}}, P_k^{\text{riv}}), \\ b_k^{\text{targ}}(\chi) &= \mathcal{N}(\chi - \xi_k^{\text{targ}}, P_k^{\text{targ}}), \end{aligned} \quad (13)$$

where  $(\xi_k^{\text{riv}}, P_k^{\text{riv}}, \xi_k^{\text{targ}}, P_k^{\text{targ}})$  are the track states corresponding to the river and the target states, respectively, which evolve according to the Kalman filter equations. In the NBO method, we approximate the objective function as follows:

$$J_H(b_0) \approx \sum_{k=0}^{H-1} c(\hat{b}_k, a_k, g), \quad (14)$$

where  $\hat{b}_1, \dots, \hat{b}_{H-1}$  is a *nominal* belief-state sequence, and the optimization is over an action sequence  $g, a_0, \dots, a_{H-1}$ . We obtain the *nominal* belief states by evolving the current belief state with exactly zero-noise sequence over the horizon  $H$  (similar to the treatment in [10, 11]). Therefore, the objective function from the NBO method is given by

$$J_{\text{NBO}}(b_0) = \sum_{k=0}^{H-1} \sum_{i=1}^N \mathbb{1} \left\{ \left\| \hat{s}_{k+1}^{i,\text{pos}} - \hat{\xi}_{k+1}^{g(i),\text{targ},\text{pos}} \right\| > d_{\text{dist-thresh}} \right\}, \quad (15)$$

where  $\hat{s}_{k+1}^{i,\text{pos}}$  is the *nominal* position of the  $i$ th AAV (defined below),  $\mathcal{N}(\hat{\xi}_{k+1}^{j,\text{targ}}, \hat{P}_{k+1}^{j,\text{targ}})$  is the *nominal* belief state of the  $j$ th

target at time  $k+1$ , where  $\hat{\xi}_{k+1}^{j,\text{targ},\text{pos}}$  (component of  $\hat{\xi}_{k+1}^{j,\text{targ}}$ ) represents the position estimate of the target. This *nominal* target belief state is obtained by evolving the track state component  $\hat{\xi}_k^{j,\text{targ}}$  with exactly zero-noise sequence as follows:

$$\hat{\xi}_{k+1}^{j,\text{targ}} = F \hat{\xi}_k^{j,\text{targ}}. \quad (16)$$

The evolution of vehicle state depends on the river state estimate  $\xi_k^{\text{riv}}$ . In the NBO method,  $\xi_k^{\text{riv}}$  is replaced with  $\hat{\xi}_k^{\text{riv}}$  in the AAV kinematic model  $\psi(\cdot)$ , where  $(\hat{\xi}_1^{\text{riv}}, \dots, \hat{\xi}_H^{\text{riv}})$  are the *nominal* track state components corresponding to the river state, and the obtained positions of the  $i$ th AAV  $\hat{s}_{k+1}^{i,\text{pos}}$  are called *nominal positions*.

Here, we adopt an approach called “receding horizon control,” according to which we optimize the action sequence for  $H$  time steps at the current time step, implement only the action corresponding to the current time step, and again optimize the action sequence for  $H$  time steps in the next time step. The length of the planning horizon  $H$  should be large enough for an AAV to receive a benefit by moving toward a target. Due to computational constraints, we cannot have an arbitrarily long horizon. Therefore, we truncate the length of the horizon to a few time steps (we set  $H = 6$  in our simulations) and append the cost function with an appropriate expected cost-to-go (ECTG). The following is a distance-based ECTG:

$$J_H^{\text{dist-ECTG}} = \sum_{i=1}^N \left\| \hat{s}_H^{i,\text{pos}} - \hat{\xi}_H^{g(i),\text{targ},\text{pos}} \right\|, \quad (17)$$

where  $\hat{s}_H^{i,\text{pos}}$  is the *nominal* position of the  $i$ th AAV, and  $\hat{\xi}_H^{j,\text{targ},\text{pos}}$  is the estimated location of the  $j$ th target (from NBO approach) at time  $k = H$ . Therefore, the objective function from the NBO method is given by

$$\begin{aligned} J_{\text{NBO}}(b_0) &= \sum_{k=0}^{H-1} \sum_{i=1}^N \mathbb{1} \left\{ \left\| \hat{s}_{k+1}^{i,\text{pos}} - \hat{\xi}_{k+1}^{g(i),\text{targ},\text{pos}} \right\| > d_{\text{dist-thresh}} \right\} \\ &\quad + J_H^{\text{dist-ECTG}}, \end{aligned} \quad (18)$$

where  $J_H^{\text{dist-ECTG}}$  is the distance-based ECTG.

**4.2. AAV Kinematics.** The kinematic equations of an AAV vary depending on whether the AAV is in the river or on the land. When the AAV is in the river, we take into account the speed of the river to write the kinematic equations. The steering and thrust generation of the vehicle are modeled based on the work done by the authors of [2, 22], which is designed using single drive system. The vehicle is front-wheel driven on land. When the AAV is in the river, it is propelled using the centrifugal pump from the front wheels. The following subsections describe the kinematics of AAV on the land and in the river.

**4.2.1. Kinematics of AAVs on the Land.** This subsection provides the definition of  $\psi$ , which was introduced in Section 3,



when the vehicle is on land. Let  $s_k = (p_k, q_k, v_k, \theta_k)$  be the state of the vehicle at time  $k$ , where  $(p_k, q_k)$  represents the location of the vehicle on the 2D plane,  $v_k$  represents the speed of the vehicle along the heading direction, and  $\theta_k$  represents the heading angle of the vehicle at time  $k$ . Let  $a_k = (f_k, \phi_k)$  represent the action vector of the vehicle, where  $f_k$  represents the acceleration along the direction of the front wheels, and  $\phi_k$  represents the steering angle of the front wheels. The (simplified) schematic of a basic four-wheeled vehicle is shown in Figure 2. The control variable  $f_k$  lies within the interval  $[-f_{\text{land}}, f_{\text{land}}]$ , where  $f_{\text{land}}$  (or  $-f_{\text{land}}$ ) is the maximum acceleration (or deceleration), and the control variable  $\phi_k$  lies within the interval  $[-\delta_{\text{land}}, \delta_{\text{land}}]$ , where  $\delta_{\text{land}}$  is the maximum steering angle. The function  $\psi$  can be specified by a set of nonlinear kinematic equations, as shown below:

$$\begin{aligned} p_{k+1} &= p_k + v_k T \cos(\theta_k), \\ q_{k+1} &= q_k + v_k T \sin(\theta_k), \\ v_{k+1} &= v_k + \mathbf{f}_k T \cos(\phi_k), \\ \theta_{k+1} &= \theta_k - \frac{2\mathbf{f}_k T^2 L}{W^2 + L^2} \sin(\phi_k), \end{aligned} \quad (19)$$

where  $T$  is the length of the time step,  $W$  is the width of the vehicle, and  $L$  is the distance between the front axle and the rear axle. The derivation of the heading angle update (19) is as follows. When the front wheels of the vehicle are oriented at a particular angle  $\phi_k$  with respect to the main axis of the vehicle (as shown in Figure 2), the heading direction of the vehicle at time  $k + 1$  is derived as follows:

$$\begin{aligned} \alpha &= \arctan\left(\frac{W}{L}\right), \\ \theta_{k+1} &= \theta_k + \frac{T^2}{\sqrt{L^2 + W^2}} (f_{k,1}^\theta - f_{k,2}^\theta) \\ &= \theta_k + \frac{f_k T^2}{\sqrt{L^2 + W^2}} [\sin(\alpha - \phi_k) - \sin(\alpha + \phi_k)] \\ &= \theta_k - \frac{2f_k T^2}{\sqrt{L^2 + W^2}} [\cos(\alpha) \sin(\phi_k)] \\ &= \theta_k - \frac{2f_k T^2 L}{W^2 + L^2} \sin(\phi_k). \end{aligned} \quad (20)$$

**4.2.2. Kinematics of AAVs on the River.** This subsection provides the definition of  $\psi$ , when the vehicle is in the river. The kinematic equations of the AAV motion are as follows:

$$\begin{aligned} p_{k+1} &= p_k + v_k T \cos(\theta_k) + \widehat{w}_k^x(p_k, q_k) T, \\ q_{k+1} &= q_k + v_k T \sin(\theta_k) + \widehat{w}_k^y(p_k, q_k) T, \end{aligned} \quad (21)$$

where  $\widehat{w}_k^x(p_k, q_k)$  and  $\widehat{w}_k^y(p_k, q_k)$  are the estimated speeds of the river at the location  $(p_k, q_k)$  in  $x$  and  $y$  directions, respectively, which are obtained from the river state estimate  $\widehat{\xi}_k^{\text{riv}}$  and the river model presented in Section 2. The speed and

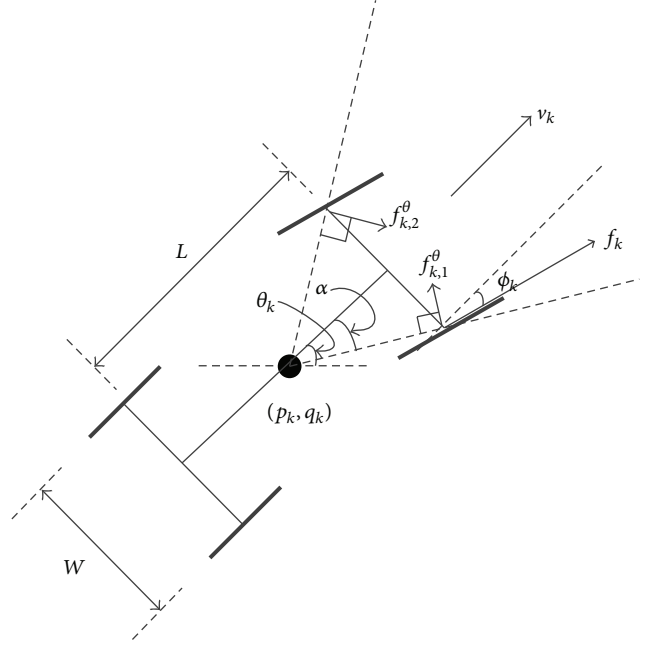


FIGURE 2: Free body diagram of an AAV.

the heading angle update equations remain the same as in the case of land. When in water (or river), the control variable  $f_k$  lies within the interval  $[-f_{\text{water}}, f_{\text{water}}]$ , where  $f_{\text{water}}$  is the maximum acceleration, and  $\phi_k$  lies within the interval  $[-\delta_{\text{water}}, \delta_{\text{water}}]$ , where  $\delta_{\text{water}}$  is the maximum steering angle. Typically, the values of  $f_{\text{water}}$  and  $\delta_{\text{water}}$  are much smaller compared to that of  $f_{\text{land}}$  and  $\delta_{\text{land}}$ .

## 5. Simulation

We implement the NBO method in MATLAB, and we use the command `fmincon` (MATLAB's optimization tool) to solve the optimization problem. For performance comparison, we also implement a greedy approach, where we optimize only the current kinematic control for the AAVs such that the following symmetric-distance-based cost is minimized:

$$\begin{aligned} J_{\text{Greedy}}(b_k) &= \sum_{i=1}^N \min_j \|\widehat{s}_{k+1}^{i,\text{pos}} - \widehat{\xi}_{k+1}^{j,\text{targ, pos}}\| \\ &+ \sum_{j=1}^N \min_i \|\widehat{s}_{k+1}^{i,\text{pos}} - \widehat{\xi}_{k+1}^{j,\text{targ, pos}}\|, \end{aligned} \quad (22)$$

where  $\widehat{s}_{k+1}^{i,\text{pos}}$  and  $\widehat{\xi}_{k+1}^{j,\text{targ, pos}}$  are the *nominal* positions (obtained by evolving the belief states with zero noise) of the  $i$ th AAV and the  $j$ th target at time  $k + 1$ , respectively. Our simulation environment is two dimensional, that is, the AAVs, the river, and the targets move in 2D. According to the river model, the speed of the river stream  $w_k$  at a location  $(p, q)$  is given by  $w_k(p, q) = C_1 [\log(d_k(p, q)) + C_2]$ , where  $d_k(p, q)$  is the depth of the river at  $(p, q)$ , and  $C_1$  and  $C_2$  are constants. Since the depth of the river is not fully observable, we estimate  $d_k(p, q)$  as follows. The elevation map of the landscape is known

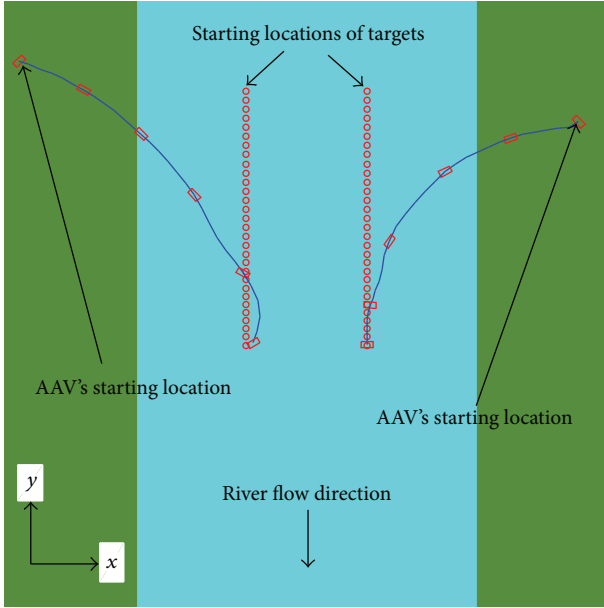


FIGURE 3: Simulation of Scenario I with NBO approach, *average rescue time* = 36 steps.

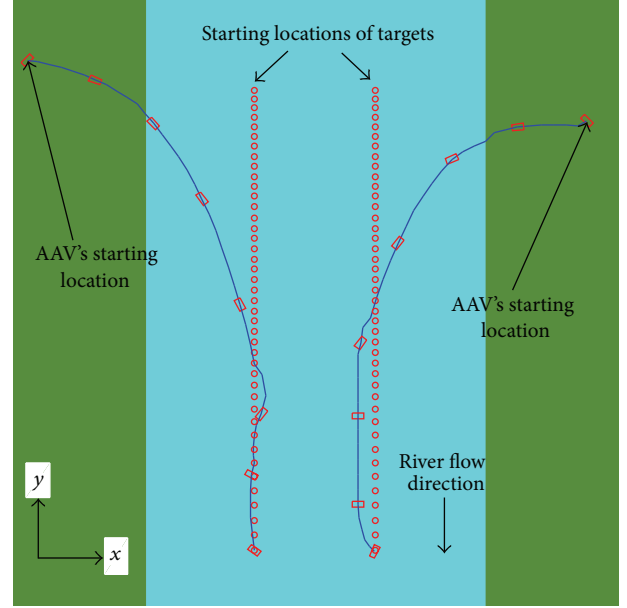


FIGURE 4: Simulation of Scenario I with greedy approach, *average rescue time* = 64 steps.

a priori, that is, if we know the depth of the river at a particular location, we can obtain the depth of the river at all locations. Therefore, we estimate the depth of the river at location  $(p, q)$ , that is,  $\hat{d}_k(p, q)$  using the estimated depth of the river at the reference point  $\hat{d}_k^{\text{ref}} (= \hat{\xi}_k^{\text{riv}})$ . Therefore, the estimated speed of the river at location  $(p, q)$  is given by  $\hat{w}_k(p, q) = C_1[\log(\hat{d}_k(p, q)) + C_2]$ . We set the length of the horizon  $H$  to 6 time steps, and the length of the time step  $T$  to 1 second. In the simulations, the flooded river flows along a valley in the landscape from the north toward the south as shown in Figure 1. Since the simulations are in 2D, the river flows toward the  $-y$  direction, and the river speed in  $x$  direction (toward the east) is zero at every location. Therefore, the estimated speeds of the river at location  $(p, q)$  in  $x$  and  $y$  directions are given by  $\hat{w}_k^x(p, q) = 0$  and  $\hat{w}_k^y(p, q) = -C_1[\log(\hat{d}_k(p, q)) + C_2]$ . Here, we model the dynamics of the target motion by the *constant velocity model* (see [23] for the definition of the variables  $F$  and  $U_{\text{targ}}$  in (5)).

In the simulations, an AAV is represented by a rectangle, and the line connecting the rectangles represents the trajectory of the AAV. We define a performance metric called *average rescue time*—the average of the rescue times of each target (the rescue time of a target is the time elapsed after the start of the simulation until it is rescued). The POMDP cost function defined in Section 3 is reflective of this performance metric. We simulate three scenarios: Scenario I, Scenario II, and Scenario III. In Scenario I, there are two AAVs, each one located on the opposite banks of the river, and two targets are moving (being drifted by the moving water) in the river, as shown in Figure 3. Figure 3 shows a snapshot of the scenario at the end of the simulation with the NBO approach, where the *average rescue time* is 36 time steps. We also simulate

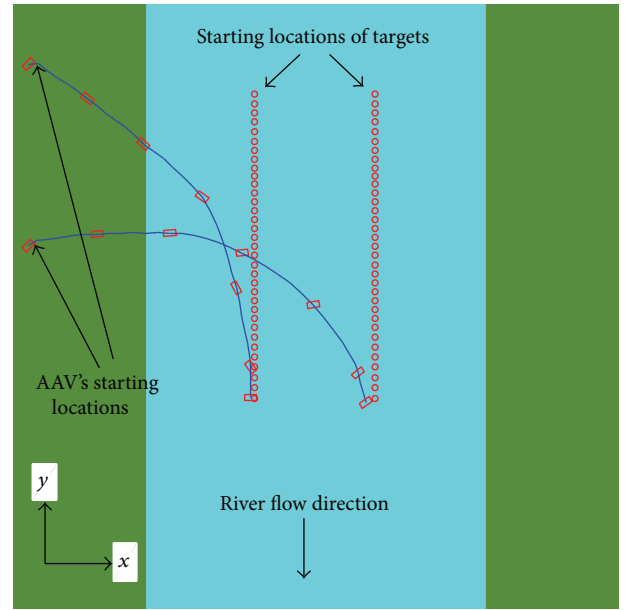


FIGURE 5: Simulation of Scenario II with NBO approach, *average rescue time* = 45 steps.

Scenario I with the greedy approach, as shown in Figure 4, where the *average rescue time* is 64 time steps. In Scenario II, there are two AAVs on the left bank of the river, and two targets are moving in the river. We simulate this scenario with both the NBO and the greedy approaches. Figure 5 shows the snapshot of the scenario with the NBO approach at the end of the simulation, where the *average rescue time* is 45 time

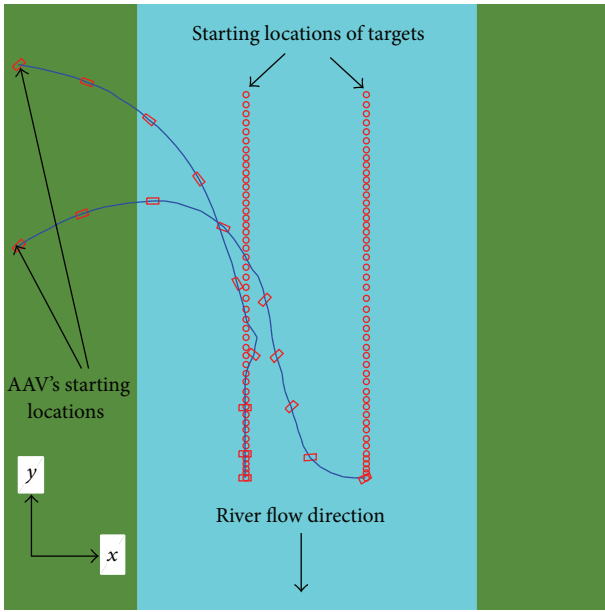


FIGURE 6: Simulation of Scenario II with the greedy approach, average rescue time = 62 steps.

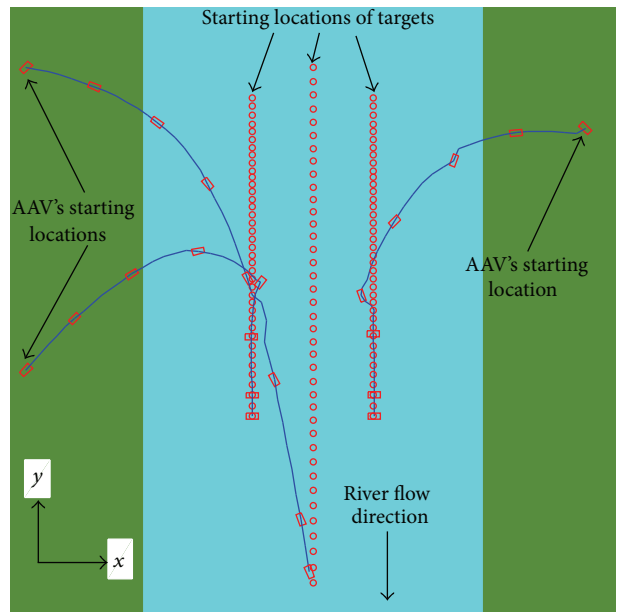


FIGURE 8: Simulation of Scenario III with the greedy approach, average rescue time = 76 steps.

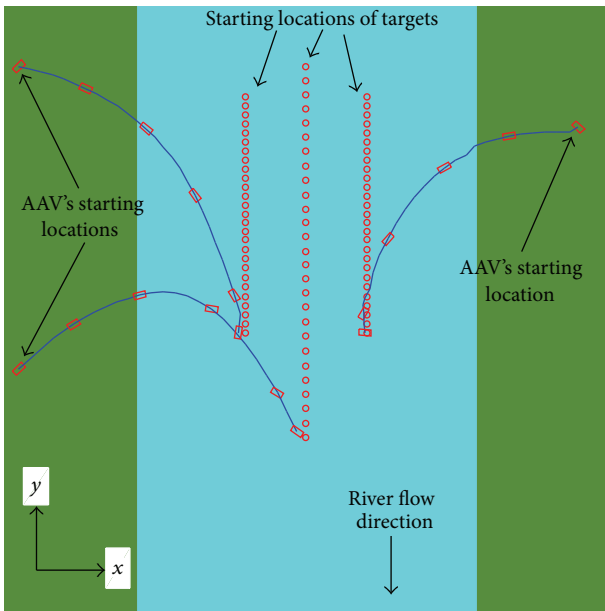


FIGURE 7: Simulation of Scenario III with NBO approach, average rescue time = 48 steps.

steps, and Figure 6 shows the simulation of the same scenario with the greedy approach, where the average rescue time is 62 time steps. In Scenario III, there are three AAVs (two on the left bank of the river and one on the right), and three targets are moving in the river. We simulate this scenario with both the NBO and the greedy approaches. Figure 7 shows the scenario with the NBO approach, where the average rescue time is 48 time steps, and Figure 8 shows the simulation of the same scenario with the greedy approach, where the

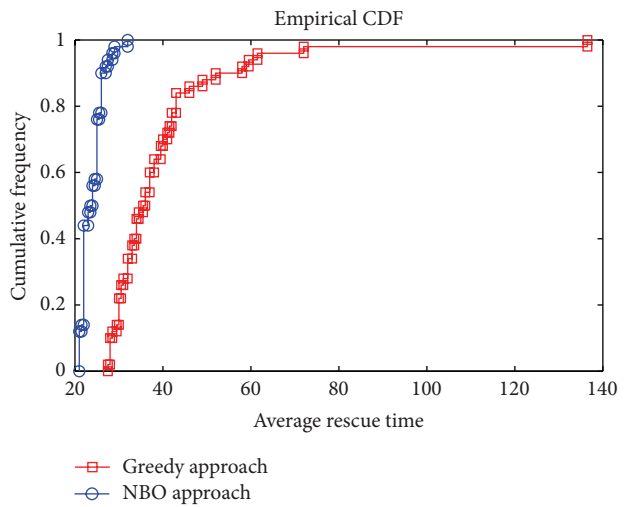


FIGURE 9: Performance comparison for Scenario I: NBO approach versus greedy approach.

average rescue time is 76 time steps. The simulation of these scenarios demonstrates that the NBO approach achieves a better coordination among the AAVs compared to the greedy approach while rescuing the targets, as evident from the average rescue times.

We compare the performance of the NBO approach with that of the greedy approach through Monte-Carlo simulations. We simulate the above scenarios with the NBO and the greedy approaches separately for 50 Monte-Carlo runs. In each scenario, we compute the average rescue time in every run for both the NBO and the greedy approaches. Figures 9, 10, and 11 show the plots of the cumulative

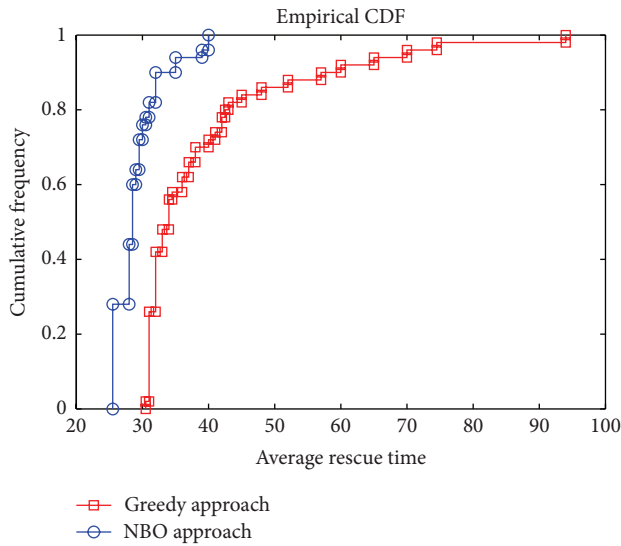


FIGURE 10: Performance comparison for Scenario II: NBO approach versus greedy approach.

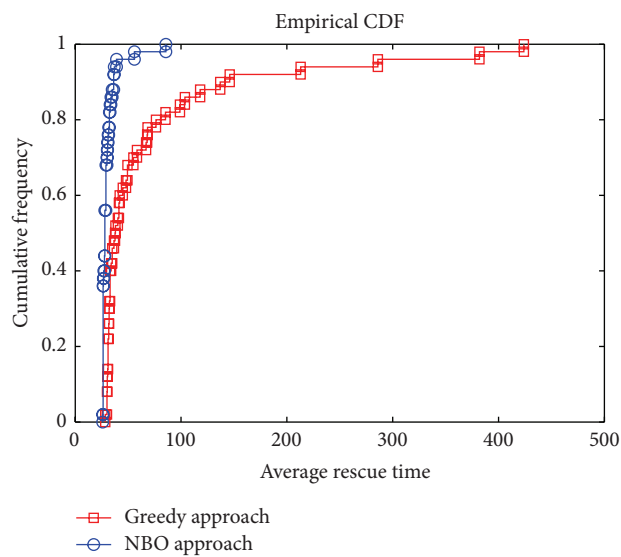


FIGURE 11: Performance comparison for Scenario III: NBO approach versus greedy approach.

frequencies of *average rescue times* for the NBO and the greedy approaches for Scenarios I, II, and III, respectively. Figures 9, 10, and 11 demonstrate that the NBO approach significantly outperforms the greedy approach.

The algorithm (NBO) runtime to compute the control commands for three AAVs (in Scenario III) in any time step in MATLAB is approximately 4 seconds on a lab computer (Intel Core i7-860 Quad-Core Processor with 8 MB Cache and 2.80 GHz speed). This runtime can be greatly reduced on a better processor and by further optimizing the code. Since the algorithm runtime is not prohibitive, it can be used in real time (i.e., for practical purposes).

## 6. Conclusions, Remarks, and Future Scope

We designed a guidance algorithm for autonomous amphibious vehicles (AAVs) to rescue moving targets in a 2D flood scenario, where the flood water flows across the scene, and the targets move in the flood water. We designed this algorithm based on the theory of *partially observable Markov decision process* (POMDP). Since a POMDP problem is intractable to solve exactly, we used an approximation method called *nominal belief-state optimization* (NBO). We simulated a few scenarios to demonstrate the coordination among the AAVs achieved by the NBO approach. We defined a performance metric called *average rescue time* to compare the performance of our approach with a greedy approach. Our results show that the NBO approach outperforms the greedy approach significantly. This was expected because unlike the greedy approach the NBO approach has a lookahead property, that is, the NBO approach trades off the short-term performance for the long-term performance. Although the greedy approach achieves coordination among the AAVs in that the AAVs eventually rescue all the targets, but the performance in terms of *average rescue time*, which is crucial in these kinds of rescue missions, is poor compared to our NBO approach. In our future work, we would like to develop methods to further improve our NBO approach (e.g., NBO with adaptive horizon). We would also like to extend our approach to a decentralized AAV guidance problem to rescue multiple targets. In this decentralized case, we will induce coordination among the AAVs to rescue multiple targets by appropriately optimizing the communication (at the network level) between the AAVs along with the kinematic controls for the AAVs.

## Acknowledgments

This work was supported in part by the Fulbright Foundation. The authors would also like to acknowledge Colorado State University's support via the Libraries Open Access Research and Scholarship Fund (OARS).

## References

- [1] M. Frejek and S. Nokleby, "Design of a small-scale autonomous amphibious vehicle," in *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE '08)*, pp. 781–786, Niagara Falls, Canada, May 2008.
- [2] E. Papadopoulos and M. Misailidis, "On differential drive robot odometry with application to path planning," in *Proceedings of the European Control Conference*, pp. 5492–5499, Kos, Greece, July 2007.
- [3] Y. Tee, Y. Tan, B. Teoh, E. Tan, and Z. Wong, "A compact design of zero-radius steering autonomous amphibious vehicle with direct differential directional drive—UTAR-AAV," in *Proceedings of IEEE International Conference on Robotics, Automation and Mechatronics (RAM '10)*, pp. 176–181, Singapore, June 2010.
- [4] Q. P. Ha, T. H. Tran, S. Scheduling, G. Dissanayake, and H. F. Durrant-Whyte, "Control issues of an autonomous vehicle," in *Proceedings of the 22nd International Symposium on Automation and Robotics in Construction (ISARC '05)*, Ferrara, Italy, September 2005.



- [5] W. Masayoshi, "Research and development of electric vehicles for clean transportation," *Journal of Environmental Sciences*, vol. 21, no. 6, pp. 745–749, 2009.
- [6] T. Brunl, *Embedded Robotics*, Springer, Berlin, Germany, 3rd edition, 2008.
- [7] T. H. Tran, Q. P. Ha, R. Grover, and S. Scheduling, "Modelling of an autonomous amphibious vehicle," in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA '04)*, Canberra, Australia, December 2004.
- [8] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots*, vol. 18, no. 1, pp. 81–102, 2005.
- [9] S. Lacroix, A. Mallet, D. Bonnafous et al., "Autonomous rover navigation on unknown terrains: functions and integration," *International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 917–942, 2002.
- [10] S. Ragi and E. K. P. Chong, "Dynamic UAV path planning for multitarget tracking," in *Proceedings of the American Control Conference (ACC '12)*, pp. 3845–3850, Montreal, Canada, June 2012.
- [11] S. A. Miller, Z. A. Harris, and E. K. P. Chong, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 724597, 17 pages, 2009.
- [12] T. H. Tran, *Modelling and control of unmanned ground vehicles [Ph.D. thesis]*, 2007.
- [13] S. A. Watson and P. N. Green, "Design considerations for micro-autonomous underwater vehicles ( $\mu$ AUVs)," in *Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics (RAM '10)*, pp. 429–434, Singapore, June 2010.
- [14] S. A. Watson and P. N. Green, "Propulsion systems for micro-autonomous underwater vehicles ( $\mu$ AUVs)," in *Proceedings of IEEE International Conference on Robotics, Automation and Mechatronics (RAM '10)*, pp. 435–440, Singapore, June 2010.
- [15] L. D. Landau, *Fluid Mechanics*, chapter IV, Pergamon Press, 2nd edition, 2000.
- [16] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1957.
- [17] C. Kreucher, A. O. Hero, K. Kastella, and D. Chang, "Efficient methods of non-myopic sensor management for multitarget tracking," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, pp. 722–727, Paradise Island, Bahamas, December 2004.
- [18] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, Mass, USA, 1996.
- [19] D. P. Bertsekas and D. A. Castañón, "Rollout algorithms for stochastic scheduling problems," *Journal of Heuristics*, vol. 5, no. 1, pp. 89–108, 1999.
- [20] E. K. P. Chong, R. L. Givan, and H. S. Chang, "A framework for simulation-based network control via hindsight optimization," in *Proceedings of the 39th IEEE Conference on Decision and Control*, pp. 1433–1438, Sydney, Australia, December 2000.
- [21] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2, Athena Scientific, Belmont, Mass, USA, 2007.
- [22] Y. Tee, B. Teoh, D. E. B. Tan, Z. Wong, C. Tan, and Y. Tan, "Design considerations of autonomous amphibious vehicle (UTAR-AAV)," in *Proceedings of IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT '10)*, pp. 13–18, Petaling Jaya, Malaysia, November 2010.
- [23] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Boston, Mass, USA, 1999.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

