2013 IFAC Intelligent Autonomous Vehicles Symposium (Preprints)
The International Federation of Automatic Control
June 26-28, 2013. Gold Coast, Australia.

Feasibility Study of POMDP in Autonomous Amphibious Vehicle Guidance

Shankarachary Ragi *, **ChingSeong Tan** **, and **Edwin K. P. Chong** *

* *Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA (Tel: 970-491-0648; e-mail: {shankar.ragi,edwin.chong}@clostate.edu).*
** *Faculty of Engineering, Multimedia University, Malaysia (e-mail: cstan@mmu.edu.my)*

Abstract: We develop a path-planning method, based on the theory of *partially observable Markov decision process* (POMDP), to guide an autonomous amphibious vehicle (AAV) to reach/rescue a victim in a flood situation. 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.

Keywords: Intelligent Control; Path Planning; Autonomous Mobile Robots.

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 [Frejek and Nokleby, 2008, Papadopoulos and Misailidis, 2007, Tee et al., 2010a]. In this paper, we present a guidance framework to control an AAV to rescue a victim (henceforth called target) in a flood situation, where the flood water (henceforth called river) flows along a valley. A target is said to be rescued when the AAV is within the circular region of radius $d_{\text{dist-thresh}}$ on the 2-D plane around the target. In general, AAVs are equipped with various advanced sensors such as polarized stereo vision, laser scanning, and SONAR [Masayoshi, 2009, Brunl, 2008]. The sensors on-board the AAV generate the (noisy) measurements corresponding to the target and the river.

Guidance control methods [Frejek and Nokleby, 2008, Tran et al., 2004, Manduchi et al., 2004, Lacroix et al., 2002] for AAVs are normally based on a standard three-layered system architecture that requires human-machine interactions. There are several other autonomous control methods in the literature for AAVs and underwater vehicles, e.g., Tran [2007], Watson and Green [2010a,b]. The practicability of the autonomous AAV could not be further exploited unless there is a sustainable vehicle guidance framework that looks for long-term performance, high-level sensor management, and effective communication with humans if necessary. This paper presents an AAV guidance framework based on the theory of partially observable Markov decision process (POMDP). The POMDP approach has a look-ahead property, which trades off short-term for long-term performance. To the best knowledge of the authors, there is no sustainable solution for vehicle guidance in AAV applications. The main theme of this paper is to develop a path-planning method for an AAV to rescue a stationary or a moving target.

2. PROBLEM SPECIFICATION

The AAV guidance problem is specified as follows. In this paper, we consider the following two scenarios: 1) Scenario I—an AAV and a stationary target are located on land on the opposite sides of a flooded river, 2) Scenario II—an AAV is located on land and a target is floating on the flooded river, where the target is being drifted down by the flood water. The scenarios are depicted in Figure 1, where the target is shown as a human (flood victim). An AAV is located at $s_0 = (x_0, y_0)$ at time $k = 0$ on the 2-D plane and is equipped with sensors that observe the target and the river. In this problem, the AAV floats while moving on water. 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 some 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). Typically, a river flows slowly near the coastlines (where the river is shallow), and flows quickly far from the coastlines (i.e., at 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 Landau and Lifshitz [2000] 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 $w_k(p, q) = C_1[\log(d_k(p, q)) + C_2]$, 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 Landau and Lifshitz [2000] for more details). The sensors on-board the AAV generate noisy observations of the target's location and the depth

⋆ This work was supported by Fulbright Foundation.

of the river directly beneath the vehicle, i.e., the sensors generate the observations of the depth of the river only when the AAV is in the river. In both the scenarios above, the objective is to minimize a distance-based cost (defined later), which guides the AAV to reach the target.

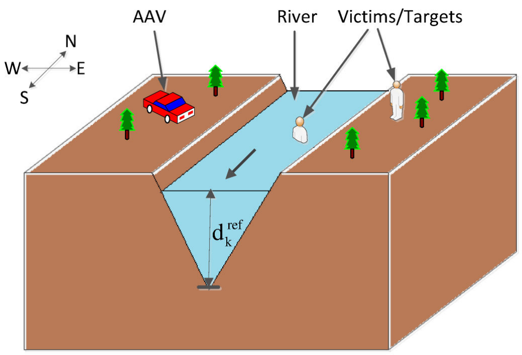

Fig. 1. Flood Scenario

3. PROBLEM FORMULATION

We pose the AAV guidance problem as a partially observable Markov decision process (POMDP). A POMDP is a mathematical framework useful for solving resource control problems (e.g., the AAV guidance problem). To pose the AAV guidance problem as a POMDP, we need to define the POMDP ingredients in terms of our guidance problem as follows:

States. Let x_k represent the state of the system at time k. The state of the system includes vehicle (AAV) state s_k, river state (depth of the river at a reference location) d_k^{ref}, target state χ_k, and track states $(\xi_k^{\text{riv}}, P_k^{\text{riv}}, \xi_k^{\text{targ}}, \mathbf{P}_k^{\text{targ}})$, i.e., $x_k = (s_k, d_k^{\text{ref}}, \chi_k, \xi_k^{\text{riv}}, P_k^{\text{riv}}, \xi_k^{\text{targ}}, \mathbf{P}_k^{\text{targ}})$. The vehicle state s_k includes the location and the velocity of the vehicle at time k. The river state d_k^{ref} is the depth of the river at some reference point (here the reference point is the lowest point in the elevation map, i.e., some location at the bottom of the valley in the landscape). Here we assume that the flow direction of the river is the same everywhere and is known a priori. The target state χ_k includes the location and the velocity of the target at time k. The track states represent the state of the tracking algorithm, where ξ_k^{riv} and P_k^{riv} are the mean and the variance, standard in Kalman filter equations, corresponding to the river state, and ξ_k^{targ} is the mean vector and $\mathbf{P}_k^{\text{targ}}$ is the covariance matrix corresponding to the target state.

Observations and Observation Law. The vehicle state and the track states are assumed to be fully observable. The river state and the target state are only partially observable. The observation of the river state from the sensors is given by

$$z_k^r = \begin{cases} d_k^{\text{ref}} + n_k & \text{if the AAV is on the water,} \\ \text{no measurement} & \text{otherwise,} \end{cases} \quad (1)$$

where $n_k \sim \mathcal{N}(0, R_k)$, and R_k is the measurement variance. The sensors on the AAV generate the measurement of the river-state only when the AAV is in the river. In practice, the sensors on the AAV measure the depth of the river exactly below the AAV. We wrote the observation model (1) as if the sensors are generating the observations of the depth of the river at the reference point because 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 observations of the target state are given by

$$z_k^\chi = \begin{cases} \mathbf{H}_{\text{targ}}\chi_k + e_k & \text{if there is line-of-sight,} \\ \text{no measurement} & \text{otherwise,} \end{cases}$$

where $\mathbf{H}_{\text{targ}}$ is the target-state observation model, and $e_k \sim \mathcal{N}(0, \mathbf{S}_k)$, where $\mathbf{S}_k$ is the measurement covariance matrix. The line-of-sight between the target and the AAV is blocked sometimes, e.g., whenever the target sinks in the water.

Actions. The actions include the controllable aspects of the system. Let u_k be the action vector at time k, which includes the control commands for the AAV. More precisely, the action vector includes the forward acceleration a_k (controls the speed) and the steering angle ϕ_k (controls the heading direction), i.e., $u_k = (a_k, \phi_k)$.

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 $s_{k+1} = \psi(s_k, u_k)$, where ψ (defined later) represents the kinematic model for the AAV. The river state evolves according to the following equation: $d_{k+1}^{\text{ref}} = d_k^{\text{ref}} + o_k$, where $o_k \sim \mathcal{N}(0, Q_k^{\text{riv}})$, where Q_k^{riv} is the process variance corresponding to the river state evolution. The target state evolves according to

$$\chi_{k+1} = \mathbf{F}_{\text{targ}}\chi_k + b_k, \quad b_k \sim \mathcal{N}(0, \mathbf{Q}_k^{\text{targ}}), \quad (2)$$

where $\mathbf{F}_{\text{targ}}$ represents the target motion model, and $\mathbf{Q}_k^{\text{targ}}$ 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 on-board the AAV.

Cost. The cost function represents the cost of taking an action at the current state. The cost function is given by $C(x_k, u_k) = \mathbf{1}\{\mathrm{E}\left[||s_{k+1}^{\text{pos}} - \xi_{k+1}^{\text{targ,pos}}||\,\middle|\, x_k, u_k\right] > d_{\text{dist-thresh}}\}$, where s_{k+1}^{pos} represents the 2-D position coordinates of the AAV and $\xi_{k+1}^{\text{targ,pos}}$ represents the estimated 2-D position coordinates of the target at time $k+1$, $||\cdot||$ is the Euclidean norm (everywhere in this paper), $\mathrm{E}[\cdot|x_k, u_k]$ is the conditional expectation given the current state x_k and the current action u_k, and $\mathbf{1}\{\cdot\}$ is the indicator function which equals 1 when the expected distance between the AAV and the target location estimate at time $k+1$ is greater than some threshold distance $d_{\text{dist-thresh}}$ and 0 otherwise.

Belief State. The belief state b_k is the posterior distribution of the state at time k. Since the vehicle state is assumed to be fully observable, the belief-state corresponding to the vehicle state is given by $b_k^s(s) = \delta(s - s_k)$. The belief-states corresponding to the river and the target are the posterior distributions of d_k^{ref} and χ_k respectively given the history of observations and actions.

4. OBJECTIVE AND OPTIMAL POLICY

Given the POMDP formulation, the goal is to find the action sequence $(u_0, u_1, \ldots, u_{H-1})$ such that the expected cumulative cost over a time horizon H is minimized. The expected cumulative cost is given by $J_H = \mathrm{E}\left[\sum_{k=0}^{H-1} C(x_k, u_k)\right]$. We can write the expected cumulative cost in terms of the belief states [Ragi and Chong, 2012, 2013] as follows: $J_H(b_0) = \mathrm{E}\left[\sum_{k=0}^{H-1} c(b_k, u_k) \,\middle|\, b_0\right]$, where $c(b_k, u_k) = \int C(x, u_k) b_k(x)\,\mathrm{d}x$ and b_0 is the belief state at time $k = 0$. From Bellman's principle of optimality [Bellman, 1957], the optimal objective function value is given by $J_H^*(b_0) = \min_u \left\{c(b_0, u) + \mathrm{E}\left[J_{H-1}^*(b_1) \,\middle|\, b_0, u\right]\right\}$, where b_1 is the random next belief state, J_{H-1}^* is the optimal cumulative cost over the horizon $H-1$, $k = 1, 2, \ldots, H-1$, and $\mathrm{E}[\cdot|b_0, u]$ is the conditional expectation given the current belief state b_0 and an action u taken at time $k = 0$. We define the Q-value of taking an action u given the current belief state b_0 as follows: $Q_H(b_0, u) = c(b_0, u) + \mathrm{E}\left[J_{H-1}^*(b_1) \,\middle|\, b_0, u\right]$. The optimal policy (from Bellman's principle) at time $k = 0$ can be written as $\pi_0^*(b_0) = \arg\min_u Q_H(b_0, u)$.

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) [Kreucher et al., 2004], parametric approximation [Bertsekas and Tsitsiklis, 1996], policy rollout [Bertsekas and Castanon, 1999], hindsight optimization [Chong et al., 2000], and foresight optimization [Bertsekas, 2007]. In this paper, we use one such approximation method called *nominal belief-state optimization* (NBO), which was introduced in Miller et al. [2009] along with other approximations and techniques specific to guidance problems. The following subsection provides a brief description of the NBO method.

4.1 NBO Approximation Method

Since the planning algorithm should be implementable in real-time, we cannot have a computationally expensive approximation method to solve the POMDP. Therefore, we choose the NBO method [Miller et al., 2009, Ragi and Chong, 2012, 2013], which is computationally least burdensome among the available approximation methods. According to the NBO method, we can approximate the objective function as follows: $J_H(b_0) \approx \sum_{k=0}^{H-1} c(\hat{b}_k, u_k)$, where $\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_{H-1}$ is a *nominal* belief-state sequence and the optimization is over an action sequence $u_0, u_1, \ldots, u_{H-1}$. We obtain the *nominal* belief-states by evolving the current belief-state with exactly zero-noise sequence over the horizon H. The belief-states corresponding to the river state and the target state are approximated as follows: $b_k^{\mathrm{riv}}(d) = \mathcal{N}\left(d - \xi_k^{\mathrm{riv}}, \mathbf{P}_k^{\mathrm{riv}}\right)$ and $b_k^{\mathrm{targ}}(\chi) = \mathcal{N}\left(\chi - \xi_k^{\mathrm{targ}}, \mathbf{P}_k^{\mathrm{targ}}\right)$ where $(\xi_k^{\mathrm{riv}}, \mathbf{P}_k^{\mathrm{riv}}, \xi_k^{\mathrm{targ}}, \mathbf{P}_k^{\mathrm{targ}})$ are the track states corresponding to the river and the target respectively, which evolve according to the Kalman filter equations. Therefore, the objective function from the NBO method is given by: $J_{\mathrm{NBO}}(b_0) = \sum_{k=0}^{H-1} \mathbf{1}\{||s_{k+1}^{\mathrm{pos}} - \hat{\xi}_{k+1}^{\mathrm{targ}}|| > d_{\text{dist-thresh}}\}$, where s_{k+1}^{pos} is the position of the AAV and $\mathcal{N}(\hat{\xi}_{k+1}^{\mathrm{targ}}, \hat{\mathbf{P}}_{k+1}^{\mathrm{targ}})$ is the nominal belief-state of the target at time $k+1$, which is obtained by evolving the track states $(\hat{\xi}_{k+1}^{\mathrm{targ}}, \hat{\mathbf{P}}_{k+1}^{\mathrm{targ}})$ via Kalman filter equations with exactly zero-noise sequence. 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 and 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 the AAV to receive a benefit by moving toward the 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 $H = 12$ in our simulations), and append the cost function with an appropriate *expected cost-to-go* (ECTG). The following is a distance-based ECTG: $J_H^{\text{dist-ECTG}} = ||s_H^{\mathrm{pos}} - \hat{\xi}_H^{\mathrm{targ}}||$, where s_H^{pos} is the 2-D position of the AAV and $\hat{\xi}_H^{\mathrm{targ}}$ is the *nominal* track-state component (target-location estimate) at time $k = H$. Therefore, the objective function from the NBO method is given by

$$J_{\mathrm{NBO}}(b_0) = \sum_{k=0}^{H-1} \mathbf{1}\{||s_{k+1}^{\mathrm{pos}} - \hat{\xi}_{k+1}^{\mathrm{targ}}|| > d_{\text{dist-thresh}}\} + J_H^{\text{dist-ECTG}},$$

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

4.2 AAV Kinematics

The kinematic equations of the AAV vary depending on whether the AAV is in river or on land. When the AAV is on river, we take into account the speed of the river to write the kinematic equations. The steering and thrust generation of the vehicle is modeled based on the work done by Papadopoulos and Misailidis [2007], Tee et al. [2010b], 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 wheel. The following subsections describe the kinematics of AAV on land and in water.

Kinematics on Land This subsection provides the definition of ψ, 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 2-D plane, v_k represents the speed of the vehicle along the heading direction, θ_k represent the heading angle of the vehicle at time k. Let $u_k = (a_k, \phi_k)$ represent the action vector of the vehicle, where a_k represents the acceleration along the direction of the front wheels and ϕ_k represents the steering angle of front-wheels. The (simplified) schematic of the basic four-wheeled vehicle is shown in Figure 2, where the heading direction of the vehicle is controlled by steering the direction of the front wheels. The control variable a_k lies within the interval $[-a_{\mathrm{land}}, a_{\mathrm{land}}]$, where a_{land} (or $-a_{\mathrm{land}}$) is the maximum acceleration (or deceleration), and the control variable ϕ_k lies within the interval $[-\delta_{\mathrm{land}}, \delta_{\mathrm{land}}]$, where δ_{land} is the maximum steering angle. The function ψ can be specified by a set of non-linear kinematic equations, as follows: $p_{k+1} = p_k + v_k T\cos(\theta_k)$, $q_{k+1} = q_k + v_k T\sin(\theta_k)$, $v_{k+1} = v_k + \mathbf{a}_k T\cos(\phi_k)$, and $\theta_{k+1} = \theta_k - (2\mathbf{a}_k T^2 L/(W^2 + L^2))\sin(\phi_{\mathbf{k}})$, 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 above heading angle update equation is derived as follows: $\theta_{k+1} = \theta_k + \frac{T^2}{\sqrt{L^2+W^2}}$

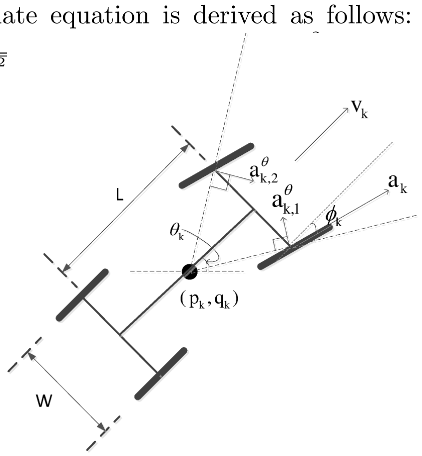

Fig. 2. Free body diagram of the AAV

Kinematics on Water This subsection provides the definition of ψ when the vehicle is in water. The kinematic equations of the AAV motion are as follows: $p_{k+1} = p_k + v_k T\cos(\theta_k) + \hat{w}^x_k(p_k, q_k)T$, $q_{k+1} = q_k + v_k T\sin(\theta_k) + \hat{w}^y_k(p_k, q_k)T$, where $\hat{w}^x_k(p_k, q_k)$ and $\hat{w}^y_k(p_k, q_k)$ are the estimated speeds of the river at the location (p_k, q_k) in x and y directions respectively. The speed and the heading angle update equations remain the same as in the case of land. When in water, the control variable a_k lies within the interval $[-a_{\text{water}}, a_{\text{water}}]$, where a_{water} is the maximum acceleration, and ϕ_k lies within the interval $[-\delta_{\text{water}}, \delta_{\text{water}}]$, where δ_{water} is the maximum steering angle. Typically, the values of a_{water} and δ_{water} are much smaller compared to that of a_{land} and δ_{land}.

5. SIMULATIONS

We implement the NBO method in MATLAB, and we use the command *fmincon* (MATLAB's optimization tool) to solve the optimization problem. *Fmincon* attempts to find a constrained minimum of a scalar function of several variables starting at an initial estimate. We choose the initial estimate by picking an action sequence over the horizon H uniformly at random from all possible action sequences. A detailed description of *fmincon* can be found in Fmincon Documentation [2013]. The algorithm (NBO) run-time to compute the control commands for the AAV in any time-step in MATLAB is approximately 950 milliseconds on a lab computer (Intel Core i7-860 Quad-Core Processor with 8MB Cache and 2.80 GHz speed). The algorithm run-time can be greatly reduced on a better processor and also by optimizing the code further. The algorithm run-time is relatively small; this demonstrates that the NBO method can be implemented in real-time.

For performance comparison, we also implement a greedy approach, where we optimize only the current action such that the distance of the AAV from the target in the next time-step is minimized. Our simulation environment is two dimensional; therefore the AAV, the river, and the target move on the 2-D plane. In our river model, the speed of the river stream w_k at a location (p, q) on the 2-D plane is given by $w_k = C_1[\log(d_k(p, q)) + C_2]$, 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 know a priori, which means that if we know the depth of the river at a particular location, we can obtain the depth of the river everywhere in the landscape. Therefore, we estimate the depth of the river at location (p, q), i.e., $\hat{d}_k(p, q)$ using the estimated depth of the river at the reference point $\hat{d}^{\text{ref}}_k (= \xi^{\text{riv}}_k)$. 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 12 time-steps, and the length of the times-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 2-D, 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 $\hat{w}^x_k(p, q) = 0$ and $\hat{w}^y_k(p, q) = -C_1[\log(\hat{d}_k(p, q)) + C_2]$.

In the simulations, the AAV is represented by a rectangle, and the line connecting the rectangles represents the trajectory of the AAV. We simulate two scenarios: Scenario I and Scenario II. In Scenario I, at the start of the simulation, an AAV and a target are located on land on opposite sides of the flooded channel. In this scenario, the target is stationary, i.e., the variables in (2) are give by $\mathbf{F}_{\text{targ}} = \mathbf{I}$ and $\mathbf{Q}_{\text{targ}} = \mathbf{0}$. Therefore, the state-transition law corresponding to the target is $\chi_{k+1} = \chi_k$. The simulation of Scenario I is shown in Figure 3, which shows the snap-shot of the scenario at the end of the simulation. Figure 3 shows the trajectories of the AAV obtained from both the NBO approach and the greedy approach. In this simulation, the NBO approach takes 125 time-steps, whereas the greedy approach takes 144 time-steps.

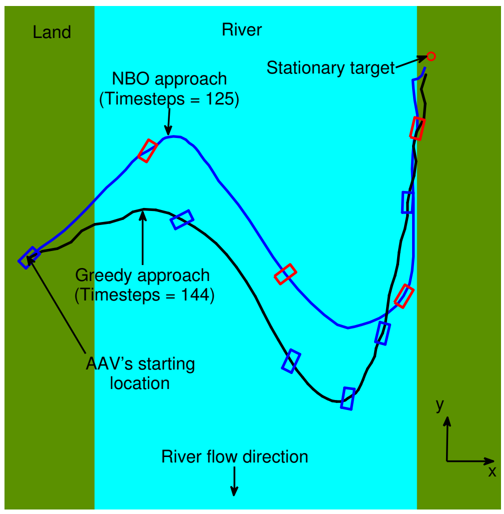

Fig. 3. AAV rescuing a stationary target

In Scenario II, the AAV is initially located on land and the target is located in water, which is being drifted toward the south by the river. Here, we model the dynamics of the target motion by the *constant velocity model* (see Blackman and Popoli [1999] for the definition of the variables $\mathbf{F}_{\text{targ}}$ and $\mathbf{Q}_{\text{targ}}$ in (2)). In the simulations, the velocity of the target at a location (p, q) at any time is equal to the velocity of the river at (p, q) at that time. Occasionally, the sensors on the AAV do not generate target observations when the line-of-sight is lost. When the target observation is not available, we implement only the prediction step of the Kalman filter (corresponding to the target-state) and we do not implement the update step (the step where we update the prediction using the observations). The simulation of this scenario, with greedy approach is shown in Figure 4, and with the NBO approach is shown in Figure 5. For this scenario, the greedy approach takes 75 time-steps, whereas the NBO approach takes 61 time-steps to rescue the target.

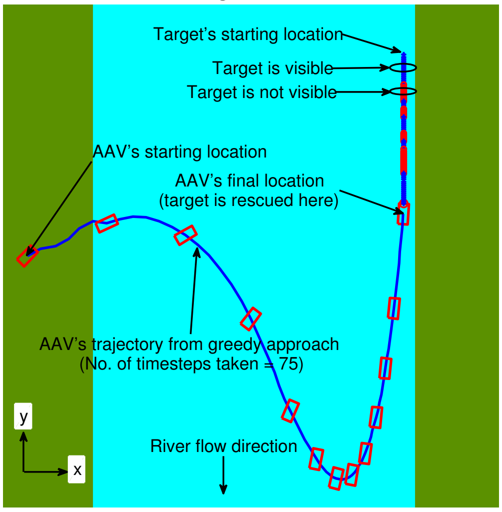

Fig. 4. AAV rescuing a moving target via greedy approach

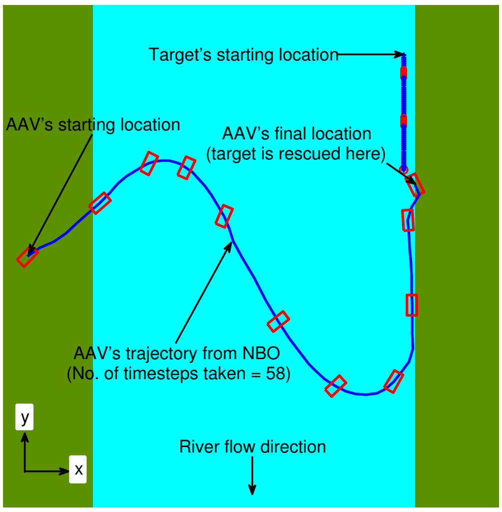

Fig. 5. AAV rescuing a moving target via NBO approach

We run the above scenarios, i.e., stationary target scenario and moving target scenario, for 200 Monte Carlo runs (using the NBO approach and the greedy approach); and in each run we measure the number of time-steps the AAV takes to reach the target. The cumulative frequency of the number of time-steps for the stationary target scenario is shown in Figure 6, and for the moving target scenario is shown in Figure 7. Figures 6 and 7 demonstrate that the NBO approach significantly outperforms the greedy approach.

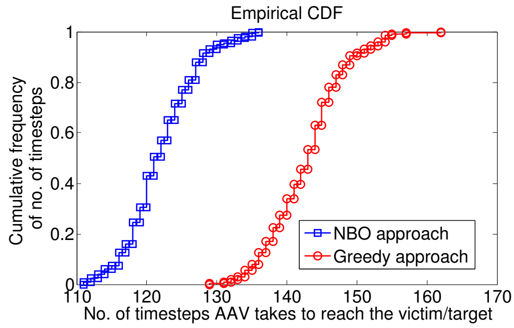

Fig. 6. Performance comparison for the stationary target scenario: NBO approach vs. greedy approach

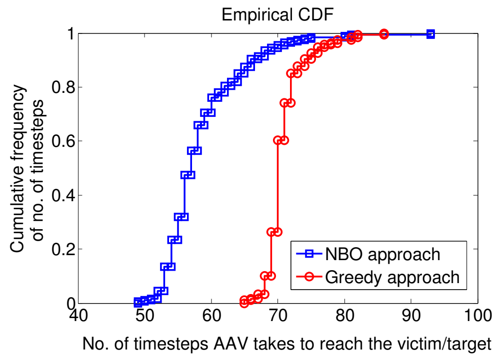

Fig. 7. Performance comparison for the moving target scenario: NBO approach vs. greedy approach

Now, we simulate a scenario, where the target moves and changes its direction of motion abruptly. Since, we assumed a stochastic linear model on the motion of the target, i.e., *constant velocity model*, and the estimates of the target location are obtained from Kalman filter equations, the AAV should still be able to track and rescue the target. Figure 8 corroborates our above claim, where the AAV successfully rescues the target, which changes its direction of motion abruptly during the simulation.

6. CONCLUSIONS AND REMARKS

In this study, we presented a mathematical framework to develop a guidance control method for an autonomous amphibious vehicle (AAV) to rescue a flood victim (also called target). We cast the AAV guidance problem as a *partially observable Markov decision process* (POMDP). Since a POMDP is hard to solve exactly, we used an approximation method called *nominal belief-state optimization* (NBO). The motion of an AAV is controlled using the control commands forward acceleration and steering angle. We implemented the NBO method in MATLAB, where the simulations are performed for stationary and moving target scenarios. In both the scenarios, we demonstrated (through Monte-Carlo simulations) that the NBO approach outperforms a greedy approach. In these scenarios, since the target's location is known with some uncertainty at every time-step, we cannot guarantee that the target can be rescued in a finite number of time-steps. However,

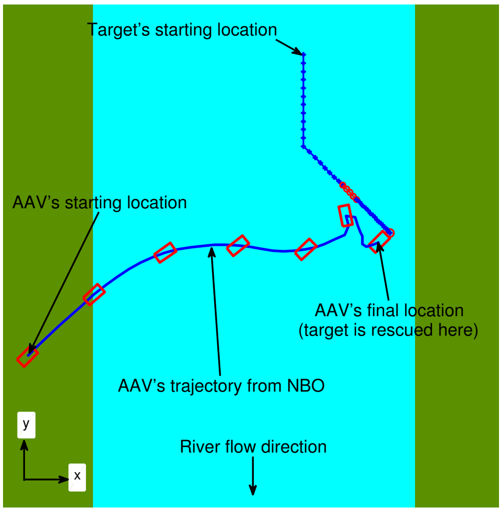

Fig. 8. AAV rescuing a moving target; the target changes its direction of motion abruptly

it turns-out that in each of our Monte-Carlo runs, in both the stationary and moving target scenarios, the target gets rescued in a finite number of time-steps. Finally, we simulated a scenario with a moving target, where the target changes its direction of motion abruptly; the results shows that the AAV successfully rescues the target. Since we assumed a stochastic model to represent the dynamics of the target motion, i.e., *constant velocity* model, the AAV was able to track the target and eventually rescue it even when the target changed its direction of motion abruptly.

REFERENCES

R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, USA, 1957.

D. P. Bertsekas. *Dynamic Programming and Optimal Control vol. 2*. Athena Scientific, Belmont, Massachusetts, USA, 2007.

D. P. Bertsekas and D. A. Castanon. Rollout algorithms for stochastic scheduling problems. *J. Heuristics*, 5:89–108, 1999.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Massachusetts, USA, 1996.

S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Boston, Massachusetts, USA, 1999.

T Brunl. *Embedded Robotics, Third Edition*. Springer, Germany, 2008.

E. K. P. Chong, R. L. Givan, and H. S. Chang. A framework for simulation-based network control via hindsight optimization. In *Proc. 39th IEEE Conference on Decision and Control*, pages 1433–1438, Sydney, Australia, December 2000.

M. Frejek and S. Nokleby. Design of a small-scale autonomous amphibious vehicle. In *Proc. Canadian Conference on Electrical and Computer Engineering (CCECE), 2008*, pages 781–786, Niagara Falls, Canada, May 2008.

C. Kreucher, A. O. Hero, K. Kastella, and D. Chang. Efficient methods of non-myopic sensor management for multitarget tracking. In *Proc. 43rd IEEE Conference on Decision and Control*, pages 722–727, Paradise Island, Bahamas, December 2004.

S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains: Functions and integration. *International Journal of Robotics Research*, 21:2002, 2002.

L. D. Landau and E. M. Lifshitz. *Fluid Mechanics, Second Edition*, chapter IV, pages 172–174. Pergamon Press, 2000.

R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robots*, 18: 81–102, 2004.

W. Masayoshi. Research and development of electric vehicles for clean transportation. *J. Environ. Sci.*, 21(6): 745–749, 2009.

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*, 2009, 2009.

E. Papadopoulos and M. Misailidis. On differential drive robot odometry with application to path planning. In *Proc. European Control Conference, 2007*, pages 5492–5499, Kos, Greece, July 2007.

S. Ragi and E. K. P. Chong. Dynamic UAV path planning for multitargte tracking. In *Proc. American Control Conference (ACC), 2012*, pages 3845–3850, Montreal, QC, Canada, June 2012.

S. Ragi and E. K. P. Chong. UAV path planning in a dynamic environment via partially observable Markov decision process. *IEEE Transactions on Aerospace and Electronic Systems*, to appear.

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 *Proc. IEEE Conference on Robotics Automation and Mechatronics (RAM), 2010*, pages 176–181, Singapore, July 2010a.

Y. Tee, B. Teoh, D. E. B. Tan, Z. Wong, C. Tan, and Y. Tan. Design considerations of autonomous amphibious vehicle (utar-aav). In *Proc. IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2010*, pages 13–18, Petaling Jaya, Malaysia, November 2010b.

T. H. Tran. Modelling and control of unmanned ground vehicles. *Ph.D. Thesis*, 2007.

T. H. Tran, Q. P. Ha, R. Grover, and S. Scheding. Modelling of an autonomous amphibious vehicle. In *Proc. Australasian Conference on Robotics and Automation (ACRA), 2004*, Canberra, Australia, December 2004.

S. A. Watson and P. N. Green. Design considerations for micro-autonomous underwater vehicles (AUVs). In *Proc. IEEE Conference on Robotics Automation and Mechatronics (RAM), 2010*, pages 429–434, Singapore, June 2010a.

S. A. Watson and P. N. Green. Propulsion systems for micro-autonomous underwater vehicles (AUVs). In *Proc. IEEE Conference on Robotics Automation and Mechatronics (RAM), 2010*, pages 435–440, Singapore, June 2010b.

Fmincon documentation. "`http://www.mathworks.com/help/optim/ug/fmincon.html`".