Q-Learning Approach to Automated Unmanned Air Vehicle (UAV) Demining

Silvia Ferrari and Greyson Daugherty

Laboratory for Intelligent Systems and Control (LISC), Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA

ABSTRACT

This paper develops a Q-learning approach to Unmanned Air Vehicle (UAV) navigation, or path planning, for sensing applications in which an infrared (IR) sensor or camera is installed onboard the UAV for the purpose of detecting and classifying multiple, stationary ground targets. The problem can be considered as a geometric sensor-path planning problem, because the geometry and position of the sensor's field of view (FOV) determines what targets can be detected and classified at any given time. The advantage of this approach over existing path planning techniques is that the optimal guidance policy is learned via the Q-function, without explicit knowledge of the system models and environmental conditions. The approach is demonstrated through a demining application in which a UAV-based IR sensor is capable of determining the optimal altitude for properly detecting and classifying targets buried in a complex region of interest.

Keywords: Aircraft, neural network, Q-learning

1. INTRODUCTION

This paper develops a Q-learning sensor planning approach to Unmanned Air Vehicle (UAV) navigation for sensing and surveillance applications. The approach is applicable to airborne sensors, such as UAVs or helicopters with onboard sensors, that are deployed over a region of interest (ROI), for the purpose of detecting and classifying hidden (e.g. buried) targets, in variable and uncertain environmental conditions, as illustrated in Fig. 1. The approach is demonstrated through an application involving an infrared (IR) sensor installed onboard a UAV, referred to as UAV-IR sensor, that flies over a two-dimensional ROI for the purpose of detecting and classifying buried targets, such as clutter, mines, or unexploded ordnance. The goal of the Q-learning algorithm is to determine the UAV guidance law or *policy* that maximizes the number of targets that are properly classified by the onboard IR sensor, without explicit knowledge of the UAV and sensor models, or of the environmental conditions. Environmental conditions, such as time-of-day, weather, and vegetation, influence the IR sensor performance and, therefore, the optimal policy. However, they may be partially unknown or changing over time and, therefore, they are learned from the Q-function, based on the immediate sensing reward.

Sensor planning is concerned with determining a policy for gathering sensor measurements to support a sensing objective, such as target classification. When the sensors are installed on mobile platforms an important part of the problem is determining the optimal sensor path or guidance policy.[2] Several approaches have been proposed for planning the path of mobile robots with on-board sensors to enable navigation and obstacle avoidance in unstructured dynamic environments, e.g.,.[3] However, these methods are not directly applicable to robotic sensors whose primary goal is to support a sensing objective, rather than to navigate a dynamic environment.[4] The reason is that they focus on how the sensor measurements can best support the robot motion, rather than focusing on the robot motions that best support the sensing objective.[4] This paper addresses the problem of planning the path and measurements of a robotic sensor, in order to support the sensing objective of properly classifying multiple targets distributed in an obstacle-populated workspace. This problem can be viewed as a *geometric* sensor-path planning problem because the sensor's field of view (FOV) determines what targets can be detected and classified at any given time.

Further author information: (Send correspondence to Silvia Ferrari)
Silvia Ferrari: E-mail: sferrari@duke.edu, Telephone: 1 919 660 5484

Unmanned Systems Technology XII, edited by Grant R. Gerhart, Douglas W. Gage, Charles M. Shoemaker,
Proc. of SPIE Vol. 7692, 76920S · © 2010 SPIE · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.850135

Figure 1. Example of airborne sensing scenario and applications.[1]

Geometric sensing problems require a description of the geometry and position of the targets, and of the sensor's FOV.[5] Viewpoint planning has been shown by several authors to be an effective approach for optimally placing or moving vision sensors based on the target geometry and sensor FOV, using weighted functions or tessellated space approaches.[6] In geometric sensor path-planning problem, the optimal sensor path and measurement sequence depend on the UAV dynamics, the IR sensor's characteristics and FOV, and on the characteristics of the ROI.[7] Probabilistic deployment is an effective approach for detecting targets in an ROI, by computing a search path based on the probability of finding a target in every unit bin of a discretized, obstacle-free ROI.[8] These existing approaches to geometric sensor-path planning, however, require prior information, such as sensor and platform models, environmental conditions, and prior sensor measurements. The approach presented in this paper allows the sensor to learn an implicit model of its own dynamics and measurement process, and of the ROI, based on the immediate sensing reward calculated by flying over a training ROI in which only the actual classification of the targets is known *a priori*. Through Q-learning, the UAV-IR sensor learns an optimal guidance policy that can be applied to new ROIs, without the need for designer intervention.

The paper is organized as follows. The Q-learning approach is reviewed in Section 2. The mathematical models used to simulate the system and learn the optimal policy are described in Section 3. The Q-learning approach to UAV-IR sensor-path planning is presented in Section 4. Then, the methodology is demonstrated in Section 5 by means of a novel simulation comprised of an integrated demining system where the IR sensor is installed onboard a UAV that obeys a six-degree-of-freedom equation of motion derived from full-scale wind tunnel data and physical models reviewed in.[9]

2. BACKGROUND ON Q-LEARNING

Approximate dynamic programming (ADP) methods, such as Q-Learning, are valuable tools for solving optimal control problems online, subject to partial or imperfect knowledge of the system state and models.[10] Optimal control problems involve a dynamic system (or process) that is either stochastic or deterministic. Although various notations are in use in the ADP literature,[11] in this paper, we will adopt the notation that is typically used in the optimal control and dynamic programming community (see[12] for a detailed description and introduction). Assuming time can be discretized and indexed by k, a deterministic dynamical system may be modeled by the difference equation,

$$x_{k+1} = f(x_k, u_k, k) \tag{1}$$

where, the state x_k at time k is an element of the *state space* $\mathcal{X}$, and the control u_k at time k is an element of the space $\mathcal{A}$ of admissible actions or decisions. If the dynamical system is stochastic, then it may be modeled as a Markov decision process (MDP).[13] An MDP is a tuple $\mathcal{M} = \{\mathcal{X}, \mathcal{A}, T, R\}$ representing a random and sequential decision process. In this case, the state space is a finite set of possible state values, denoted by $\mathcal{X} = \{s_1, \ldots, s_n\}$,

and the space $\mathcal{A} = \{a_1, \ldots, a_m\}$ is a finite set of admissible actions or decisions. T is the transition probability function, $T : \mathcal{X} \times \mathcal{A} \to P(\mathcal{X})$, which describes the MDP state transitions, such that whenever the state at time k has value $x_k = s_i$ and the decision is $u_k = a_j$, there is a probability $P(x_{k+1} = s_l \mid x_k = s_i, u_k = a_j)$ that the next state value is $x_{k+1} = s_l$. In many real-world applications of optimal control, however, the exact form of the difference equation (1) or the transition matrix T are unknown or approximate, and can only be determined online.

In optimal control problems, there exists a reward associated with the dynamic system that may be represented by the reward function, $R : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, specifies the value of the immediate reward, $r_k = R(\ x_k, u_k)$, received after executing the action decision u_k in state x_k. A policy is a mapping of state values to actions, $\pi : \mathcal{X} \to \mathcal{A}$. Let the value function $V^\pi(x_k)$ denote the expected discounted return of a policy π, defined as:

$$V^\pi(x_k) = E\left\{\sum_{i=0}^{\infty} \gamma^i r_{k+i} \mid \pi,\ x_k\right\} \tag{2}$$

where, r_{k+i} is the reward received i steps into future, and the discount factor $0 \leq \gamma < 1$ modulates the effect of future rewards on present decisions, with small values emphasizing near-term gain and larger values emphasizing later rewards. Then, an *optimal policy* π^* is one that maximizes $V^\pi(x_k)$ for all possible states $x_k \in \mathcal{X}$. The Markov property guarantees that an optimal policy exists, though it may not be unique, and, thus, it is associated with an *optimal value function* $V^*(x_k) = \max_\pi V^\pi(x_k)$. The optimal policy of an MDP, $\mathcal{M}$, is a fixed point of the Bellman's equation, which can be determined iteratively using policy iteration or value iteration algorithms [14, Chapter 5].

In value iteration, the value of a state $V(x_k)$ is the total expected discounted reward accrued by a policy starting at $x_k \in \mathcal{X}$. The Q function of a state-action pair, $Q(x_k, u_k)$, is the total expected discounted reward accrued by a policy that produces $u_k = \pi(x_k)$ [14, Chapter 5]. The Bellman equation can be formulated in terms of the aforementioned functions, such that the state-action value function is

$$Q(x_k, u_k) = E\left\{R(x_k, u_k) + \gamma V(x_{k+1})\right\} \tag{3}$$

$$V(x_{k+1}) = \max_{u_{k+1} \in \mathcal{A}} Q(x_{k+1}, u_{k+1}). \tag{4}$$

If two functions $Q(\cdot)$ and $V(\cdot)$ satisfy the above Bellman equation, then they specify an optimal *greedy* policy

$$\pi^*(x_k) = \arg\max_{u_k \in \mathcal{A}} Q(x_k, u_k) \tag{5}$$

Value-iteration algorithms use eq. (3) to iteratively determine $Q(\cdot)$ and $V(\cdot)$ and, subsequently, determine $\pi^*(\cdot)$.

Value iteration can be used to determine the optimal policy of an MDP, $\mathcal{M}$, provided the transition probability function T is known. If T is unavailable, Q-Learning can be utilized to learn an approximate state-action value function $Q(x_k, u_k)$ that is iteratively updated by the rule,

$$\begin{aligned} Q(x_k, u_k) \quad &\leftarrow \quad (1-\alpha)Q(x_k, u_k) \\ &+ \quad \alpha[r_k + \gamma \max_{u_k \in \mathcal{A}} Q(x_{k+1}, u_k)] \end{aligned} \tag{6}$$

where, α is the learning rate, and $0 < \alpha \leq 1$. In this paper, Q-Learning is used to solve a new sensor planning problem described in the next section, which consists of obtaining the optimal guidance policy for an IR sensor deployed onboard an UAV for mine detection and classification.

3. MATHEMATICAL MODELS AND PROBLEM FORMULATION

3.1 UAV-IR Dynamic Equation

The problem considered in this paper consists of learning an optimal guidance policy for an UAV with an onboard infrared (IR) sensor that flies over a minefield $\mathcal{W} \subset \mathbb{R}^2$, or region of interest (ROI), for the purpose of detecting and classifying buried landmines and unexploded ordnances (UXOs). The UAV dynamics can be modeled by

a six-degree-of-freedom equation of motion derived from Newton's second law using an inertial- and a body-reference frame.[9] The full aircraft state consists of the 12-dimensional vector $x_a = [u\ v\ w\ x_r\ y_r\ z_r\ p\ q\ r\ \phi\ \theta\ \psi]^T$, where, u, v, w, and p, q, r are the UAV velocities and angular rates in body frame, respectively, and x_r, y_r, z_r, and ϕ, θ, ψ, are the UAV translational and angular positions in inertial frame, respectively. The body state accelerations, denoted by X_b, Y_b, Z_b, L_b, M_b, and N_b are a function of the available thrust, and of the aerodynamic force and moment coefficients produced by the controls for the present aircraft state and wind field. The model estimates low-angle-of-attack Mach effects, power effects, and moments and products of inertia by using available full-scale wind tunnel data and physical characteristics, according to the methods described in.[9] The moments of inertia I_{xx}, I_{yy}, I_{zz}, and product of inertia I_{xz} are estimated using simplified mass distributions, and are held fixed at all times. Then, using the classical aircraft angles definitions and coordinate transformations described in,[9] the following UAV equation of motion can be obtained:

$$\begin{aligned}
\dot{u} &= X_b + g_x + rv - qw \\
\dot{v} &= Y_b + g_{b_y} + pw - ru \\
\dot{w} &= Z_b + g_{b_z} + qu - pv \\
\dot{x}_r &= u\cos\theta\cos\psi + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) \\
&+ w(\cos\phi\sin\theta\cos\psi - \sin\phi\sin\psi) \\
\dot{y}_r &= u\cos\theta\sin\psi + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) \\
&+ w(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\
\dot{z}_r &= -u\sin\theta + v\sin\phi\cos\theta + w\cos\phi\cos\theta \\
\dot{p} &= \frac{q}{(I_{xx}I_{zz} - I_{xz}^2)}\{I_{zz}L_b + I_{xz}N_b - p[I_{xz}(I_{yy} - I_{xx} \\
&- I_{zz})] + r[I_{xz}^2 + I_{zz}(I_{zz} - I_{yy})]\} \\
\dot{q} &= \frac{\left(M_b - pr\left(I_{xx} - I_{zz}\right) - I_{xz}\left(p^2 - r^2\right)\right)}{I_{yy}} \\
\dot{r} &= \frac{q}{I_{xx}I_{zz} - I_{xz}^2}\{I_{xz}L_b + I_{zz}N_b + r[I_{xz}(I_{yy} - I_{xx} \\
&- I_{zz})] + p[I_{xz}^2 + I_{xx}(I_{xx} - I_{yy})]\} \\
\dot{\phi} &= p + (q\sin\phi + r\cos\phi)\tan\theta \\
\dot{\theta} &= q\cos\phi - r\sin\phi \\
\dot{\psi} &= \frac{q\sin\phi + r\cos\phi}{\cos\theta}
\end{aligned} \tag{7}$$

The aircraft control inputs consist of the throttle δT, the elevator δE, the aileron δA, and rudder δR, i.e., $u_a = [\delta T\ \delta E\ \delta A\ \delta R]^T$. As shown in,[15] the UAV can be fully controlled by means of a reduced state vector $\mathbf{x}_{UAV} = [V\ \gamma\ q\ \theta\ r\ \beta\ p\ \mu]^T$, which is formulated in terms of the aircraft speed V, sideslip angle β, and path angle γ, where,

$$V = \sqrt{u^2 + v^2 + w^2} \tag{8}$$

$$\beta = \sin^{-1}(v/V) \tag{9}$$

$$\gamma = \sin^{-1}(-w/V) \tag{10}$$

and in terms of the bank angle μ, defined in.[9]

3.2 Model of IR Sensor Measurements

The field-of-view (FOV) of the onboard IR sensor is assumed to be a closed and bounded subset of an Euclidian space, $\mathcal{S} = [0,\ L_{IR}]^2 \subset \mathbb{R}^2$, with the square geometry illustrated by the grey area in Fig. 2. It can be easily shown using planar geometry that the size of the FOV is a function of the aircraft altitude $H = -z_r$,

$$L_{IR} = H\sin\theta_{IR} \tag{11}$$

where, z_r is defined positive downward by convention,[9] and θ_{IR} is the sensor's aperture angle. In this paper, it is assumed that θ_{IR} is held constant, and that the orientation of the IR sensor is fixed with respect to the UAV body frame. It follows that the position and size of the FOV are a function of time, $\mathcal{S} = \mathcal{S}(t)$, and change based on the aircraft trajectory or path. For simplicity, it is also assumed that the centroid of $\mathcal{S}(t)$ coincides with the UAV coordinates in inertial frame, $x_r(t)$ and $y_r(t)$ at any time t.

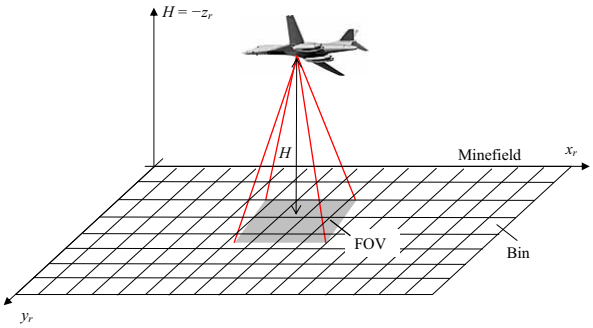


Figure 2. Problem description.

As illustrated in Fig. 2, the position and geometry of the FOV determine which regions of the minefield can be measured by the airborne IR sensor. The IR sensor measurements are influenced by its height above the ground (H), and by the environmental conditions in the minefield. A two-dimensional grid is superimposed on the minefield dividing it into unit-square cells. Soil characteristics, vegetation, and time-varying meteorological conditions, modeled according to,[16,17] are assigned to each cell, either at random or at user-specified positions. Buried targets are modeled as anti-tank mines (ATM), anti-personnel mines (APM), unexploded ordnance (UXO), and clutter objects (CLUT) that are sampled and reproduced using the Ordata Database,[18] which contains over 5,000 explosive items and 3,000 metallic and plastic objects that resemble anti-personnel mines. Each target i occupies one or more cells in the minefield depending on its size z_i, and is characterized by a depth d_i, and shape s_i (Table 1). The IR sensor mode, v_{IR}, is given by the UAV altitude (H) in km, which is discretized in a set of m possible values $\{a_1, \ldots, a_m\}$. At any given time, the space of admissible values of v_{IR} depends on the UAV speed, and is known from aircraft flight envelope. The aircraft flight envelope, denoted by $\mathcal{E}$, is the set of altitudes and velocities for which the aircraft can be trimmed (an example is shown in Fig. 4). The envelope's boundary is designed by considering the stall speed, the thrust/power required and available, compressibility effects, and the maximum allowable dynamic pressure to prevent structural damage.[9]

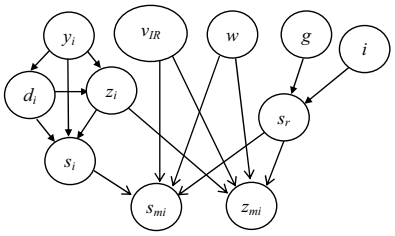


Figure 3. BN model of IR sensor (taken from[7]).

IR sensors detect anomalies in infrared radiation and, based on their height above the ground, build an image of an horizontal area, obtaining cursory measurements of shape and size for shallow-buried objects. Because they rely on temperature variations, their performance is highly influenced by illumination, weather, vegetation, and soil properties. As shown in,[19] an IR sensor can be modeled by the Bayesian network (BN) in Fig. 3, based on data and on the IR working principles and detailed studies of Agema Thermovision 900 sensors.[17] All BN nodes represent variables that influence the IR measurement process, and are defined as shown in Table 1. The IR BN model approximates the joint probability mass function (PMF) underlying the IR sensor measurements

Table 1. IR Sensor Variables and Environmental Conditions

Symbol:	Nodes:	Range:
y_i	Target classification	{mine (1), not mine (0)}
v_{IR}	IR mode (km)	$\{a_1, \ldots, a_m \mid a_h = h \cdot 0.2km\}$
E	Soil moisture (%): s_r	{dry [0, 10], wet (10, 40], saturated (> 40)}
	Soil composition: s_c	{very-sandy, sandy, high-clay, clay, silt}
	Soil uniformity: s_u	{yes, no}
	Vegetation: g	{no-vegetation, sparse, dense}
	Weather: w	{clear, overcast, raining}
	Illumination: i	{low (7-10 a.m. and 6-9 p.m.), medium (10-1 p.m.), high (1-6 p.m.)}
F_i	Depth (cm): d_i	{surface [0], shallow-buried (0, 12], buried (12, 60], deep-buried (> 60)}
	Size (cm): z_i	{small (2, 13], medium (13, 24], large (24, 40], extra-large (> 40)}
	Shape: s_i	{cylinder, box, sphere, long-slender, irregular}

in terms of the recursive factorization

$$\begin{aligned} P(v_{IR}, E, M_i, F_i, y_i) &= P(M_i \mid v_{IR}, E, F_i) P(F_i \mid y_i) \\ &\times P(y_i) P(v_{IR}) P(E), \quad \forall i \end{aligned} \tag{12}$$

Where, $F_i = \{d_i, z_i, s_i\}$ is the set the features of the i^{th} target, $M_i = \{d_{m_i}, z_{m_i}, s_{m_i}\}$ are the *measured* target features extracted from sensor images, and y_i denotes the i^{th} target classification with the range $\mathcal{Y} = \{$mine, not mine$\}$. In this paper it is assumed that the environmental conditions E_i are constant and uniform everywhere in $\mathcal{W}$, but are possibly unknown. The factors in (12) are conditional PMFs given by the BN conditional probability tables (CPTs) (see[20] for a comprehensive review of BNs). By this approach, non-Gaussian sensor models can be obtained and used for sensor planning, as shown in Section 4.

As shown in previous research,[7,19] when a sensor is installed on a mobile platform, the measurement gathering process can be modeled as a Markov decision process (MDP), under proper assumptions. Although the MDP transition probability matrix of the UAV-IR system could potentially be obtained from the nonlinear dynamic equation (7), the BN sensor model, environmental maps, and weather forecasts, it would be computationally prohibitive to determine it for every minefield, weather, UAV, and IR sensor characteristics. Therefore, the goal of this paper is to develop a Q-Learning technique that can learn the UAV-IR guidance policy in real time from the sensing reward, without explicit knowledge of the transition probability matrix. By this approach, the same guidance algorithm can be applied to different airborne sensors and minefields, without redesigning the algorithm or modeling every system component.

4. Q-LEARNING APPROACH TO UAV-IR SENSOR-PATH PLANNING

The problem of determining the optimal sensor path for searching and classifying hidden targets, known as *treasure hunt problem*, was first formulated as an MDP in.[21] An effective Q-learning technique for solving treasure hunt problems was presented in,[22] and demonstrated through the benchmark problem of the game of CLUE®. In this paper, Q-learning is applied to the new UAV-IR demining problem described in Section 3, which can be viewed as a new application example of treasure hunt. Since the sensor is installed onboard a UAV, the UAV path determines what cells can be intersected by the sensor's FOV, and measured by the IR sensor at any time.

4.1 Definition of Q-Learning State and Control Vectors

As a first step, the aircraft dynamics (7) are evaluated at N equally-spaced discrete points in time, $t_k = t_0 + k\Delta t$, $k = 0, \ldots, (N-1)$, over the interval $[t_0,\ t_f]$, where $\Delta t = (t_f - t_0)/N$ is the discretization interval. Between any two points in time, the control is assumed to be piecewise-constant and the UAV dynamics (7) are integrated by a 3^{th} order Runge-Kutta integration routine [23, pg. 77]. In order to apply the Q-Learning technique in Section 2, the MDP state x_k must be observable, and may be defined as a subset of the full system state. Thus,

based on the problem formulation in Section 3, the $x_k = [x_r(t_k)\ y_r(t_k)\ z_r(t_k)\ V(t_k)]^T$, since this subset of state variables determines the IR-sensor FOV's size and position at t_k. The FOV's size and position, in turn, determine the hidden target characteristics in cell i, denoted by the set $\zeta_i = \{d_i, z_i, s_i, y_i\}$, and the hidden environmental conditions E_i, through the subset of cells that are intersected by the FOV, with index set I_k. The environmental conditions may or may not be known depending on the scenario. The set ζ_i of hidden variables can be estimated only after the FOV has intersected cell i.

The objective of the optimal greedy guidance policy, $u_k = \pi^*(x_k)$, is to compute the next UAV position, at t_{k+1}, such that the resulting UAV path such that the IR sensing performance over time is maximized. Therefore, the control vector is defined as the next waypoint, i.e., $u_k = [x_r(t_{k+1})\ y_r(t_{k+1})\ z_r(t_{k+1})]^T$, where $z_r(k+1)$ determines the next IR sensor mode $v_{IR}(t_{k+1})$. In this approach, the Q-learning technique is said to provide an outer-loop algorithm, whose output can be followed by means of the inner-loop proportional-integral (PI) controller described in.[15]

4.2 DEFINITION OF Q-LEARNING REWARD

After the IR measurements are obtained from all the cells inside the sensor's FOV, the IR BN model (12) is used to estimate (or infer) the target classification based on the *measured* target features (M_i) extracted from sensor images, the sensor mode v_{IR} and, possibly, known environmental conditions. In this paper, BN inference is performed by junction-tree algorithm available through the Matlab® BN-Toolbox commands *jtree_inf_engine*, *enter_evidence*, and *marginal_nodes*.[24] The inference algorithm provides the posterior PMF $P(y_i, d_i, z_i, s_i \mid v_{IR}, d_{i_m}, z_{i_m}, s_{i_m}, E_i)$, and the target classification is estimated by choosing the value of highest posterior probability, i.e.:

$$\hat{y}_i = \arg\max_{y_i^k \in \mathcal{Y}_i} P(y_i \mid v_{IR}, d_{i_m}, z_{i_m}, s_{i_m}, E_i) \tag{13}$$

The estimated target classification is then accompanied by the CL, denoted by $c_i = P(\hat{y}_i \mid v_{IR}, d_{i_m}, z_{i_m}, s_{i_m}, E_i)$, which represents the confidence in the estimated value and, for a binary variable, is $0.5 < c_i \leq 1$.

Let y_i^* denote the actual classification of the target in cell i. Then, the classification error defined as,

$$e_i = |\hat{y}_i - y_i^*| \tag{14}$$

also is binary, and takes a value of 0 when the estimate is correct, and a value of 1 when the estimate is incorrect. If the estimate is correct, a higher CL is desirable, but if the estimate is incorrect, a lower CL is desirable because it indicates that the estimate is uncertain. Thus, the IR sensor performance is reflected in the classification error (14) and in the CL. Additionally, the sensor performance depends on application-specific objectives for deploying the UAV-IR. For example, in some applications it may be of interest to minimize the number of false alarms, whereas in others it may be of interest to find cells without targets, in order to determine a safe path through $\mathcal{W}$. In this paper, the application's objectives are characterized by a discrete risk function defined as,

$$\rho_i = \begin{cases} w_1 & \text{if} \quad \hat{y}_i = 1,\ y_i^* = 0 \text{ (false alarm)} \\ w_2 & \text{if} \quad \hat{y}_i = 0,\ y_i^* = 1 \text{ (misclassification)} \\ w_3 & \text{if} \quad \hat{y}_i = 1,\ y_i^* = 1 \text{ (mine detection)} \\ w_4 & \text{if} \quad \hat{y}_i = 0,\ y_i^* = 0 \text{ (void-cell detection)} \end{cases} \tag{15}$$

where, $w_1, \ldots, w_4$ are user-defined positive constants that weigh the relative importance of the four cases listed in (15). If in a mission a false alarm poses a much greater risk than a misclassification, then $w_1 >> w_2$. If, in addition, it is of secondary importance to correctly classify mines, then $w_1 >> w_3 >> w_2, w_1$, and so on.

Then, the immediate reward from cell i can then be defined as a tradeoff between the measurement value and error,

$$r_i = W_v[(1 - e_i)]c_i\rho_i - W_e(e_i c_i \rho_i) \tag{16}$$

where W_v and W_e are user-defined positive constants that represent the desired tradeoff between the measurement value of obtaining correct classifications of mines or void cells, and the measurement error of incorrectly classifying

mines or false alarms. At every time t_k, the IR sensor obtains measurements from a set of cells in its FOV, $\mathcal{S}(t_k)$ and, thus, the total value of the immediate reward is,

$$r_k = R(x_k, u_k) = \sum_{i \in \mathcal{S}(t_k)} W_v[(1 - e_i)]c_i\rho_i - W_e(e_i c_i \rho_i) \tag{17}$$

and can be computed from the IR sensor measurements, and the actual target classification y_i^*.

In this paper, the Q function is approximated by a feedforward sigmoidal neural network (NN),

$$Q(x_k, u_k) = W_2\Phi(W_1\ [x_k^T\ u_k^T]^T + b_1) + b_2 \tag{18}$$

by means of the update rule (6) as the UAV explores the state and control spaces. The s-dimensional operator Φ represents one hidden layer of s sigmoidal functions of the form $\sigma(n) \equiv 1/(1 + e^{-n})$. The NN weights $W_1 \in \mathbb{R}^{s\times(n+m)}$, $W_2 \in \mathbb{R}^{1\times s}$, $b_1 \in \mathbb{R}^s$, and $b_2 \in \mathbb{R}^2$, are determined by the resilient backpropagation algorithm ('trainrp'[25]). A training set for (18) is formed according to the Q-learning approach. As a first step, the Cartesian product of the state and control spaces $\mathcal{X} \times \mathcal{A}$ is discretized. With the state and control definitions in Section 4.1, this is achieved by discretizing the flight envelope (e.g., see crosses in Fig. 4). As a second step, the rule in (6) is applied iteratively over the discrete time t_k, while exploring $\mathcal{W}$ (already discretized into cells) by flying the UAV at every feasible pair of altitudes and velocities.

After the reward (17) is evaluated for every pair of state and control values explored by the UAV-IR, the data can be used to learn the Q function using (18). Then, the optimal policy, $u_k = \pi^*(x_k)$, is determined by maximizing the learned Q function using the greedy rule in (5). The effectiveness of this approach is demonstrated in the next section, using the system models described in Section 3.

5. NUMERICAL SIMULATIONS AND RESULTS

The UAV is simulated by integrating the ODE in (7), using the Matlab® program FLIGHT, developed in.[9] This program simulates the aircraft flight, and determines the trim conditions for any nominal altitude and velocity that lie inside the aircraft flight envelope $\mathcal{E}$, illustrated in Fig. 4. In this paper, it is assumed that the UAV maintains steady-level longitudinal flight, and that the Q-guidance algorithm must determine the optimal nominal altitude H^* and velocity V^*, based on the UAV-IR sensing reward obtained by flying over a minefield $\mathcal{W}$ with unknown geometric and environmental conditions, and known targets.

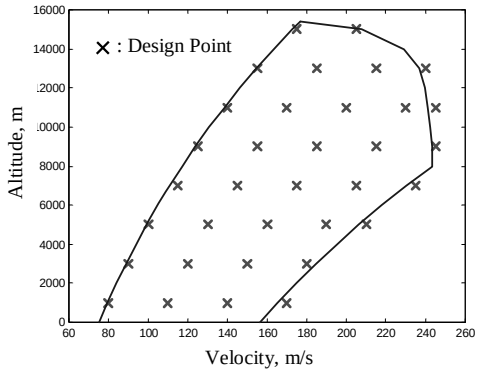


Figure 4. Aircraft Flight Envelope, taken from.[15]

The UAV-IR sensor, and the minefield $\mathcal{W}$ are simulated numerically, using the mathematical models described in Section 3. In the simulation, as soon as the FOV of the IR sensor, $\mathcal{F}$, intersects a cell containing a target, measurements are reproduced and deteriorated based on the target features, the sensor's mode and working principles, and the environmental conditions in the cell.[19] As an example, the set of cells measured by the UAV-IR sensor at a sample moment in time, and an altitude $a_8 = 1.6$ km is shown by purple bins in Fig. 5.

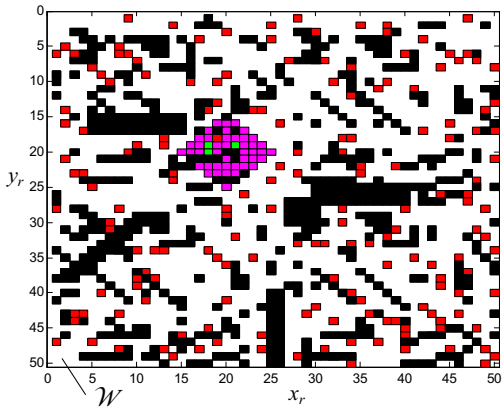


Figure 5. Instantaneous UAV-IR sensor's FOV at $a_8 = 1.6$ km.

Red bins represent undetected targets (mines or clutter), green bins represent detected targets, and black bins represent obstacles.

In this paper, the environmental conditions in Table 1 are assumed uniform in $\mathcal{W}$ and, therefore, the UAV-IR can learn the Q-function by flying over $\mathcal{W}$ at various altitudes and velocities in $\mathcal{E}$. At higher altitudes, the IR measurements are typically less accurate, but more cells are measured because the FOV is larger. At lower altitudes, the FOV is smaller, translating to less cells being measured by the sensor, but the IR measurements are more accurate. As an example, the cells measured by the UAV-IR sensor along a path at $a_8 = 1.6$ km, and during the time interval $[t_0,\ t_f]$, are shown by the purple bins in Fig. 6, using the same color legend used in Fig. 5. For comparison, the cells measured along the same path at $a_{32} = 6.4$ km is shown in Fig. 7.

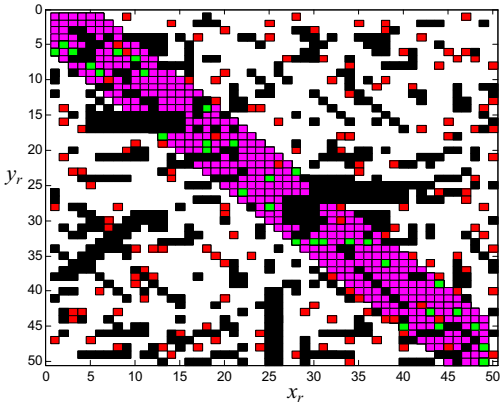


Figure 6. Cells measured by UAV-IR during $[t_0,\ t_f]$, at $a_8 = 1.6$ km.

As shown in Fig. 8, the percentages of mines (red bins) and clutter-targets (yellow bins) that are correctly classified is higher at a_8, whereas the percentage of targets that were undetected (blue bins, labeled by "U") is higher at a_{32}. The CL (%) of the classifications, plotted on each target detected in Fig. 8, also tends to be higher at lower altitudes. While the UAV-IR explores the state space, its measurements and the actual classification of the targets are used to compute the reward in (17), and to learn the Q-function from (6). In this paper, the weights in the risk function (15) are chosen as $w_2 = w_3 = 1$ in order to represent a high risk for incorrectly classifying mines, and as $w_1 = w_4 = 0.1$ in order to represent a low risk for incorrectly classifying cells that contain clutter or are empty. As a result, the Q-function learned by UAV-IR, and approximated by the NN in (18), is the highly nonlinear function that is plotted in Fig. 9 with respect to v_{IR}.

Once learning is completed, the greedy policy in (5) is computed by maximizing the learned Q-function using the Matlab® Optimization Toolbox.[26] This policy provides the optimal sensor's altitude at time $t_{(k+1)}$ as a function of the present aircraft state (altitude and velocity) at t_k. In this study, the optimal altitude is found to

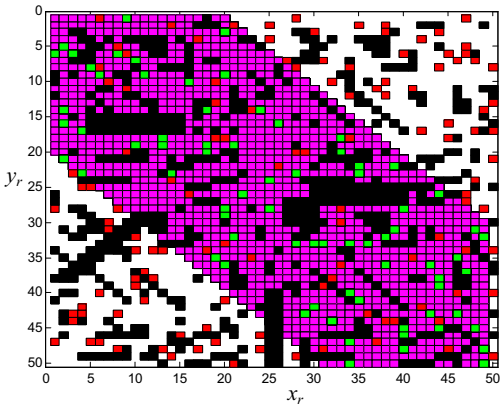
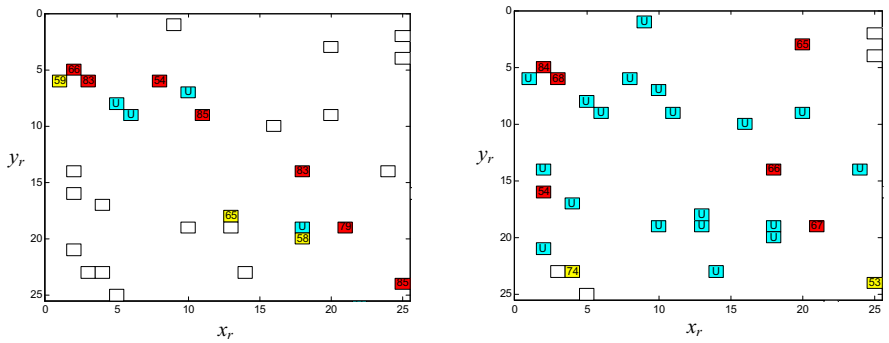


Figure 7. Cells measured by UAV-IR during $[t_0,\ t_f]$, at $a_{32} = 6.4$ km.



(a) (b)

Figure 8. Classification results for the UAV-IR at $a_8 = 1.6$ (a), and at $a_{32} = 6.4$ km (b), during a sample time interval.

be $H^* = a_{25} = 5.0$ km. Thus, as verified by the sensing performance comparison in Table 2, when the UAV-IR sensor flies at this altitude, it obtains the highest number of correct mine detections, as specified by through the risk function (15).

Table 2. Performance Comparison

UAV-IR Mode	Total Bins	Mine Detections (CL)	Clutter Detections (CL)	Undetected Targets
a_8	515	15 (74%)	5 (77%)	50
a_{25}	1295	19 (69%)	8 (71%)	73
a_{32}	1551	17 (67%)	9 (67%)	81

6. CONCLUSIONS AND FUTURE WORK

This paper presents a novel Q-learning approach to geometric sensor path planning that is applicable to Unmanned Air Vehicle (UAV) navigation for sensing and surveillance applications. The goal of the Q-learning algorithm is to determine the UAV guidance policy that maximizes the number of targets that are properly classified by the onboard IR sensor, without explicit knowledge of the UAV and sensor models, or of the environmental conditions. By this approach, the UAV-IR sensor learns the Q-function based solely on the actual classification of known targets buried in the ROI. Through the Q-function, the sensor learns about the system models implicitly, and obtains a greedy policy by minimizing the Q-function with respect to the control. The approach is demonstrated through an application involving an IR sensor installed onboard a UAV that flies over

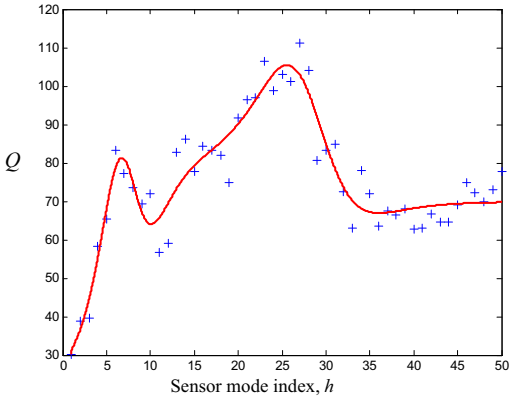


Figure 9. Q-function learned by UAV-IR as a function of v_{IR}.

a two-dimensional ROI for the purpose of detecting and classifying buried targets, such as clutter, mines, or unexploded ordnance.

REFERENCES

[1] Tolic, D., Fierro, R., and Ferrari, S., "Cooperative multi-target tracking via hybrid modeling and geometric optimization," in [*Proc. of the 19th Mediterranean Conference on Control and Automation*], 440–445 (2009).

[2] Acar, E. U., "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *International Journal of Robotic Research* **22** (2003).

[3] Lazanas, A. and Latombe, J. C., "Motion planning with uncertainty - a landmark approach," *Artificial Intelligence* **76**, 287–317 (1995).

[4] Choset, H., "Coverage for robotics: A survey of recent results," *Annals of Mathematics and Artificial Intelligence* **31**(1-4), 113–126 (2001).

[5] Hager, G. D. and Mintz, M., "Computational methods for task-directed sensor data fusion and sensor planning," *International Journal of Robotics Research* **10**, 285–313 (1991).

[6] Chen, S. Y. and Li, Y. F., "Vision sensor planning for 3cd model acquisition," *IEEE Transactions on Systems, Man, and Cybernetics - Part B* **35**(5), 894–904 (2005).

[7] Cai, C. and Ferrari, S., "Information-driven sensor path planning by approximate cell decomposition," *IEEE Transactions on Systems, Man, and Cybernetics - Part B* **39** (2009).

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

[9] Stengel, R. F., [*Flight Dynamics*], Princeton University Press, Princeton, NJ (2004).

[10] Ferrari, S. and Stengel, R., "Model-based adaptive critic designs," in [*Learning and Approximate Dynamic Programming*], Si, J., Barto, A., and Powell, W., eds., John Wiley and Sons (2004).

[11] Si, J., Barto, A., and Powell, W., "Learning and approximate dynamic programming," *Proceedings of the IEEE* **80**(9), 1384–1399 (1992).

[12] Bertsekas, D. P., [*Dynamic Programming and Optimal Control, Vols. I and II*], Athena Scientific, Belmont, MA (1995).

[13] Bertsekas, D. P. and Tsitsiklis, J. N., [*Introduction to Probability*], Athena Scientific, Belmont, MA (2002).

[14] Russell, S. and Norvig, P., [*Artificial Intelligence: A Modern Approach*], Prentice Hall, NJ (2003).

[15] Ferrari, S. and Stengel, R., "Classical/neural synthesis of nonlinear control systems," *Journal of Guidance, Control, and Dynamics* **25**(3), 442–448 (2002).

[16] MacDonald, J., [*Alternatives for Landmine Detection*], Rand Publications (2003).
[17] Dam, R. V., Borchers, B., Hendrickx, J., and Hong, S., "Soil effects on thermal signatures of buried nonmetallic landmines," in [*Detection and remediation technologies for mines and minelike targets VIII, Proc. of the SPIE*], **5089**, 1210–1218 (2003).
[18] Explosive, Ordnance, Disposal, (EOD), and Technicians, [*ORDATA Online*], Avaliable: http://maic.jmu.edu/ordata/mission.asp (2006).
[19] Ferrari, S. and Vaghi, A., "Demining sensor modeling and feature-level fusion by bayesian networks," *IEEE Sensors* **6**, 471–483 (2006).
[20] Jensen, F., [*Bayesian Networks and Decision Graphs*], Springer-Verlag (2001).
[21] Ferrari, S. and Cai, C., "Information-driven search strategies in the board game of CLUE®," *IEEE Transactions on Systems, Man, and Cybernetics - Part B* **39**(2) (2009).
[22] Cai, C. and Ferrari, S., "A Q-learning approach to developing an automated neural computer player for the board game of CLUE®," in [*International Joint Conference on Neural Networks*], 2347–2353 (2008).
[23] Stengel, R. F., [*Optimal Control and Estimation*], Dover Publications, Inc. (1986).
[24] Murphy, K., [*How To Use Bayes Net Toolbox*], [Online]. Avaliable: http://www.ai.mit.edu/ murphyk/Software/BNT/bnt.html (2004).
[25] Mathworks, [*MATLAB Neural Network Toolbox*], [Online]. Avaliable: http://www.mathworks.com (2006). function: trainbr.
[26] Mathworks, [*Matlab Optimization Toolbox*], [Online]. Avaliable: http://www.mathworks.com (2004). function: fminmax.