JMR

Journal of Maritime Research, Vol. VIII. No. 1, pp. 49-64, 2011

ISSN: 1697-4840

EXPERT GUIDANCE SYSTEM FOR UNMANNED AERIAL VEHICLES BASED ON ARTIFICAL NEURAL NETWORKS

F. Fernández[1,2], **E. Besada**[1,3], **D. Sanchez**[1,4] and **J.A. López-Orozco**[1,5]

Received 10 February 2011; in revised form 16 February 2011; accepted 17 March 2011

ABSTRACT

This article proposes an expert guidance system for Unmanned Aerial Vehicles (UAVs) for marine rescue missions. The difficulty of the problem, due to the time constraints that the mission has to fulfil are lightened by the use of Artificial Neuronal Networks, taking advantage of their high adaptability, low memory requirements, real time response capability, and extrapolation properties. We use them to implement two different types of behaviours for the two main phases of the task: in prediction mode they are responsible of calculating the displacement that the castaways suffer due to the sea and wind currents and in sensing mode they are in charge of guiding the UAV while it tracks already found shipwrecked and search for new ones. To illustrate the successful behaviour of the expert system embedded in a simulator, some results are shown in the final section.

Key words: Castaway, Shipwreck, Artificial Neural Network, Unmanned Aerial Vehicle, Search and rescue.

INTRODUCTION

The technological advances in unmanned vehicles are being exploited by a growing number of projects of the research areas of control, cooperation and artificial intelligence (ASF, BERK, CALT-Mur, MAGIC, MICA, MIT). The employment of these

[1] Universidad Complutense de Madrid, Av. Complutense s/n, 28040 Madrid, Spain. [2] Phd student, Email: francisco.fernandez.ramirez@gmail.com, Tel. 34913944436, Fax. 34913944687. [3] Assistant Professor, Email: evabes@dacya.ucm.es, Tel. 34913944740, Fax. 34913944687. [4] Phd Student, Universidad Complutense de Madrid, Email: davisanc@fis.ucm.es, Tel.34913944375, Fax. 34913944687. [5] Associated Professor, Email: jalo@dacya.ucm.es, Tel. 34913944436, Fax. 34913944687.

vehicles is extending to multiple civil and military fields too, such as surveillance tasks (Houden, 2008) or reconnaissance missions (Tian, 2006; Besada-Portas 2010).

The demanding time constraints of sea rescue tasks can benefit of the use of these types of vehicles: when a ship wrecks, the time elapsed between the shipwreck, its detection by a rescue centre, and the departure and arrival of a rescue vessel, can't be too long. Finding the castaways quickly is primordial but hard: as time passes the shipwrecked are spread by the sea winds and currents along a wide area, rendering their location difficult. The complementary capacities of Unmanned Air Vehicles (UAVs) and Unmanned Surface Vessels (USVs) can facilitate the rescue task: quickly UAVs can be sent to locate the castaways and precise USVs to perform their rescuing. This paper focuses on the UAV search and tracking side of the sea rescue task.

The capabilities of the UAVs to perform rescue or tracking tasks in designated areas have already been used in other problems (Kamrani and Ayani, 2009; Rubio et al., 2004). In our research, these capabilities are developed by an expert system that is in charge of obtaining the high level commands that will properly guide the UAV towards and inside the rescue area. We assume that the UAV already incorporates a low level stabilization and control system that interprets the high level commands. That is, the high level expert system guides the vehicle, while the low level stabilization and control system drives it. Finally, the time restrictions of the task and the resource constraints of the onboard UAV CPU have to be considered too in the expert system design.

In order to achieve all the necessary requirements, we have designed a guidance expert system based on Artificial Neural Networks (ANNs), which are parallel computing structures for *modelling* and *learning* nonlinear complex behaviours (Haykin, 1999; Patterson, 1996). Their high adaptability, low memory requirements, real time response capability, and easy integrability are also appealing. Our expert system incorporates two types of ANNs: ones to predict the position of the castaways before they are located by the UAV and other to guide the UAV after finding the first shipwrecked. We also learn their behaviours: the prediction ANN parameters are adapted to the sea rescue environment while the sensing ANN behaviour extrapolates the knowledge of an expert to different situations. So, our ANN based guidance expert system adapts to the rescue task. Besides it doesn't require much memory or CPU resources.

The rest of this paper is organized as follows. Section 2 describes and formalizes the sea rescue problem. Section 3 presents the UAV expert system, starting with their differing elements and ending with the whole system. Finally, section 4 shows the results of using the designed expert system in two different simulated rescue tasks.

PROBLEM DESCRIPTION

Searching and tracking shipwrecked people or items with an UAV is a difficult task due to the dynamics and uncertain behaviour associated to the different elements of the system. On one hand, the shipwrecked elements are stochastically moved by the sea and wind currents. On the other, the UAV only collects noisy measurements of the shipwrecked positions that are inside the camera field of view or that have been beaconed by the UAV when first detected. Then, while no shipwrecked are observed, the UAV needs to find them using a predictive model. Once shipwrecked are found and beaconed, it can track them while searching the rest. In both cases, the search is carried out by moving the UAV, whose position is deterministically controlled by high level commands. Hence, the rescue task consists on selecting the commands that let the UAV efficiently find the shipwrecked. In the remaining parts of this section we present the notation and the model of the problem used throughout the paper.

Notation

In this paper, a capital italic letter (V) represents a unidimensional variable, a boldface capital italic letter ($\boldsymbol{V}$)- a multidimensional one, and a lowercase roman letter (f) a function. Sub-indexes are used to distinguish variables: t associates the variable to the t-th timestep and i – to any of its possible realizations. Super-indexes are used to distinguish the elements of multidimensional variables: x and y refer them to Cartesian coordinates, r and θ- to polar coordinates. For example, $M_{t,i}$ represents the i-th variable labelled M at time step t and $M^x_{t,i}$ stands for its corresponding x coordinate. Finally, dir(ΔX,ΔY) is the function that calculates the orientation of the vector [ΔX, ΔY] and IsTrue(*BooleanExpression*) the indicator function that returns 1 when the Boolean expression is true and 0 otherwise.

General problem Formulation

To model the behaviour of the problem, we assume that the number of elements needing rescue is fixed and equal to N, and consider the following variables: U_t to represent the position of the UAV; A_t, the high level control command applied to the UAV; $M_{t,i}$, the real position of the i-th shipwrecked; $D_{t,i}$, the moment the i-th shipwrecked element was first detected (i.e.: never, just, previously); and $S_{t,i}$, the measurement obtained by the UAV for the i-th detected shipwrecked. Moreover, $U_t = [U^x_t, U^y_t, U^\theta_t]$, $M_{t,i} = [M^x_{t,i}, M^y_{t,i}]$ and $S_{t,i} = [S^x_{t,i}, S^y_{t,i}]$. Finally, the control command $A_t = [A^x_t, A^y_t]$ indicates the next waypoint that the UAV has to be driven to by the low level onboard UAV controller.

The relationships among all these variables are schematized in figure 1, where variables in circles belong to the hidden state space, variables in squares are observations, and an arrow $V \rightarrow W$ means that the value of W depends on the value of V. In other words, figure 1 represents that U_{t+1}= f(U_t, A_{t+1}), $M_{t+1,i}$ = g($M_{t,i}$), $D_{t+1,i}$ = h($D_{t,i}$, $M_{t+1,i}$, U_{t+1}), and $S_{t+1,i}$ = q($D_{t+1,i}$, $M_{t+1,i}$). Function f deterministically models the UAV evolution, and so it depends on the UAV characteristics. Function g stochastically models the shipwrecked evolution, and so the sea wind and currents that move the shipwrecked elements are included in its definition. The remaining two functions are related with the beacon and camera measurement systems. Function h models the evolution of $D_{t,i}$: from *never* detected to *just* detected when the camera first observes the object falling inside its field of view, and from *just* detected to *previously* detected in the following time step. When the shipwrecked is just detected, function q behaves as the noisy camera measurement model, and hereafter, as the noisy beacon measurement model.

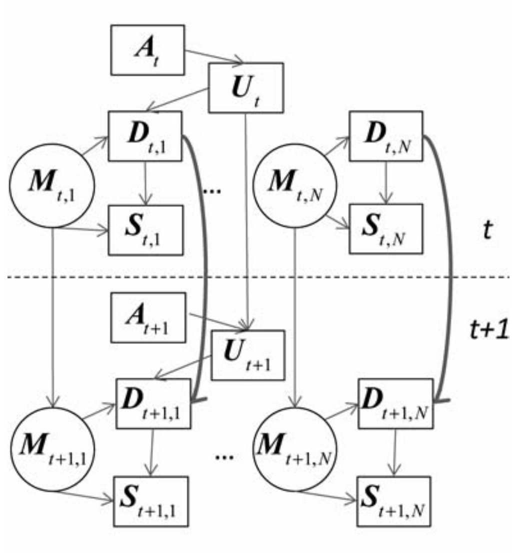


Figure 1. Dependencies of the problem variables.

At each time step, the high level command A_t applied to the UAV must increment the chances of finding new never detected shipwrecked. Then, A_t is calculated in closed loop to be able to take into account the current observations and past history.

The way to proceed to obtain A_t depends on the available information. Before the first shipwrecked is located by the UAV camera system, the UAV can only use the predicted $M_{t,i}$ and the UAV position U_t to decide where to go. Once the position of any item is detected, i.e. when the UAV starts collecting $S_{t,i}$, the high level command A_t can directly depend on the shipwrecked measured locations. In short: before finding castaways, $A_t = \mathrm{r}(U_{t-1}, \{M_{t,i}|i = 1{:}N\})$; and afterwards, $A_t = \mathrm{c}\ (U_{t-1}, \{S_{k,l}|D_{k,l} \neq \text{never}, k \leq t\})$, where the functions r and c represent the high level controllers for the prediction and sensing working modes respectively.

The objective of our research is to find efficient implementations for both high level controllers that let the UAV robustly respond to the evolution of $M_{t,i}$ or $S_{t,i}$.

UAV CONTROLLERS BASED ON ANNs

In the following two sections, we present the controllers used in each of the working modes. They fulfil the efficiency and robustness requirements by means of employing properly trained ANNs. Afterward, the whole system, that also includes a manager responsible of selecting the correct controller and handling exceptions, is described.

Prediction mode

Before the first castaway is located by the UAV, the UAV can only be driven towards the shipwrecked positions $M_{t,i}$. However, as $M_{t,i}$ belongs to the system state and not to the observations, $M_{t,i}$ has to be calculated too. That is, when the controller is working in this mode, not only does the expert system have to implement the controller A_t = r(U_{t-1}, {$M_{t,i}|i$ = 1:N}), but also a module, called predictor hereafter, responsible for obtaining all $M_{t,i}$ with the selected $M_{t+1,i}$ = g($M_{t,i}$) and the initial location $M_{0,i}$ where the vessel sunk. Figure 2 shows the connection between both subsystems.

Since the efficiency of the complete prediction controller (which obtains $M_{t,i}$ and A_t) depends on both functions g and r, special care should be taken when selecting them.

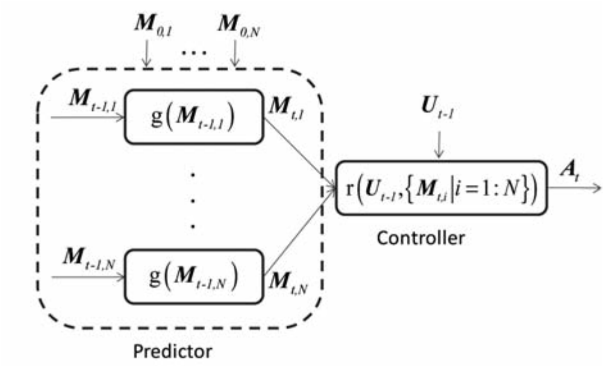


Figure 2. Complete prediction controller.

Prediction model $M_{t+1,i}$=g($M_{t,i}$)

The prediction g models usually available in rescue centres, such as the Mercator Ocean (MERCATOR) or the Spanish Project ESEOO (Álvarez, 2005), are too complex and slow for the UAV CPU. Online predictions based on wind and current maps, generated with numerical models such as HIRLAM (High Resolution Limited Area Model, Gómez and Carretero, 2005) or CEPPM (Medium Term Prediction European Center), are not fast enough either because they have to generate dense maps to obtain the wind and current values at every $M_{t,i}$, and predict $M_{t+1,i}$ based on the previous $M_{t,i}$. Therefore we opt to implement a prediction model g with an ANN trained offline at the rescue centre before the UAV starts its mission.

To develop our predictor, we implement the function g as the incremental model presented in equation (1), where function p is a feedforward ANN.

$$\boldsymbol{M}_{t+1,i} = \boldsymbol{M}_{t,i} + \mathrm{p}\left(\boldsymbol{M}_{t,i}\right) \tag{1}$$

This incremental implementation of g makes $M_{t,i} = [M^x_{t,i}, M^y_{t,i}]$ the natural input for the ANN, and $M_{t+1,i} - M_{t,i} = [M^x_{t+1,i} - M^x_{t,i}, M^y_{t+1,i} - M^y_{t,i}] = [\Delta M^x_{t+1,i}, \Delta M^y_{t+1,i}] = \Delta M_{t+1,i}$ its output. Then, since $\Delta M_{t+1,i}$ = p($M_{t,i}$), the ANN is forced to learn the displacement caused to the shipwrecked by the local conditions on each point of the environment.

The pairs of input $\boldsymbol{M}_{t,i}$ – output $\Delta\boldsymbol{M}_{t+1,i}$ used to train the ANN are generated with the numerical predictor g used in the rescue centre. As function g returns $\boldsymbol{M}_{t+1,i}$ and the ANN output is $\Delta\boldsymbol{M}_{t+1,i}$, this last value has to be calculated to generate the training data pairs. The training data are used to update the weights of the ANN with the Bayesian Regulation Backpropagation algorithm. This algorithm calculates the ANN parameters, using a method that combines the squared errors and ANN weights in such a way that a well generalizing ANN is usually obtained (MacKay D., 1992). The training phase takes into account the error between the $\Delta M_{t+1,i}$ obtained with the ANN and the $\Delta M_{t+1,i}$ obtained after subtracting $M_{t,i}$ from the output $M_{t+1,i}$ of the numerical predictor g. Figure 3 summarizes the complete training task.

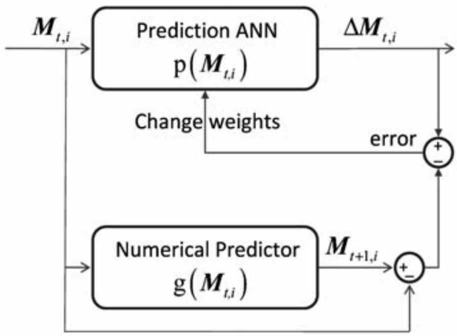


Figure 3. Training step of the prediction network.

The numerical predictor g used in our experiments to train the ANN p predicts the next shipwrecked positions using a grid map of forces caused by the sea winds and currents. The trained ANN generalizes the effects of those winds and currents over the shipwrecked on the points defined by the grid map.

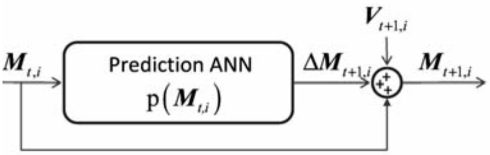


Figure 4. Onboard predictor $M_{t+1,i} = g(M_{t,i})$

The onboard predictor g, represented in figure 4, obtains $M_{t+1,i}$ from $M_{t,i}$ using the trained ANN p to generate the displacement $\Delta M_{t+1,i}$ and an additive Gaussian random variable $V_{t+1,i}$, with zero mean and covariance Q, that models little disturbances not included in the numerical predictor.

The simplicity of the onboard g depends on the properties of the ANN. To speed up these operations, we successfully use a feedforward ANN with two inputs $[M^x_{t,i}, M^y_{t,i}]$, two outputs $[\Delta M^x_{t+1,i}, \Delta M^y_{t+1,i}]$, and three layers with only two neurons in the input layer, four in the hidden and two in the output. Its training time is small too: a convergent ANN is usually available in only 5 minutes using a Pentium Core Duo. Therefore, the offline training step might be carried out while the UAV gets ready for its mission.

Finally, it is worth mentioning that there are other types of ANNs, such as the recurrent ones, that are directly used to predict a sequence (at different time steps) of outputs given the initial conditions (Hontoria et. al, 2001). Our prediction ANN is different (it learns the displacement between two successive points of the sequence), facilitates the learning task, and usually reduces the accumulated error at the end of the sequence.

Controller $A_t = r(U_{t-1}, \{M_{t,i} | i = 1{:}N\})$

The function that obtains the high level command A_t based on the predicted positions of the shipwrecked elements $M_{t,i}$ needs to conduct the UAV towards them as quickly as possible to let the UAV visual system find the first shipwrecked.

An efficient way to achieve an appropriated behaviour consists on generating the high level command that makes the UAV move towards the mean value of $M_{t,i}$. So, the controller function r implements the following equation:

$$\boldsymbol{A}_t = \frac{1}{N}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{M}_{t,i} \qquad (2)$$

Note that with this way of proceeding the UAV doesn't necessary arrives to the mean value of $M_{t,i}$ in the following time step, because the high level command only identifies the next waypoint that the UAV has to visit. Besides, the trajectory followed by the UAV to reach the waypoint depends on the UAV and low level controller properties.

In spite of the simplicity of this controller, the UAV can usually intercept the mean predicted trajectory of the shipwrecked by redirecting the UAV while approaching the castaways. However, when the shipwrecked predicted positions are not correct (due to a significant discrepancy on the real and predicted environmental conditions), the UAV can only verify that there are no castaways in the vicinity of the mean expected area and notify it to the rescue centre.

Sensing Mode

When the vessel wrecks near the rescue centre or sea winds and currents do not disperse the shipwrecked elements far from the wreckage zone, the UAV may find them quickly. However, when the shipwrecked have been dispersed before finding the first, the UAV has to track it and look for the remaining.

In order to facilitate the tracking task, the UAV puts a beacon in each shipwrecked the first time it detects them. So, after the UAV visual system first observes the location $S_{t,i}$ of any element, the UAV keeps obtaining its new locations $S_{t,i}$ from its designated beacon. This way of proceeding also favours the search and rescue tasks: the UAV can move freely to search unobserved elements while tracking the already observed that fall outside its field of view, and the vessels in charge of rescuing the shipwrecked can be sent towards the designated beacons.

The searching task requires a function that obtains A_t taking into account the possible errors of the predictors and the shipwrecked dispersion. The first requirement is fulfilled using (in the controller c of the sensing mode) the available measurements of the shipwrecked $\{S_{k,l} | D_{k,l} \neq \text{never}, k \leq t\}$ instead of the predicted position $M_{t,i}$ (used in the prediction controller r). For achieving the second, the sensing controller c is developed over a feedforward ANN s, which is trained with the behaviours proposed by an expert for different situations.

Sensing ANN controller $A_t = \mathrm{c}(U_{t-1}, \{S_{k,l}|D_{k,l} \neq \text{never}, k \leq t\})$

Implementing a sensing controller that uses the available measurements $\{S_{k,l}|D_{k,l} \neq \text{never}, k \leq t\})$ requires a function c with an increasing number of inputs. To fix the number of inputs and compact the available information, we select the following four parameters (after checking other possibilities also based on $\{S_{k,l}|D_{k,l} \neq \text{never}, k \leq t\})$ as the best sensing ANN inputs:

- $I_{t,1}$, the orientation of the mean direction of the previously observed shipwrecked. This variable lets the ANN know the global tendency of the already located shipwrecked elements.
- $I_{t,2}$, the distance of the UAV to the mean location of the previously observed shipwrecked. This variable lets the ANN know how far the UAV might travel while it still observes castaways and changes the searching radio of the ANN.
- $I_{t,3}$, the orientation of the mean direction of the shipwrecked that have only been observed twice (because we need two elements to determine a direction). This variable lets the ANN know if the unobserved elements are dispersing and consider new searching directions.
- $I_{t,4}$, the percentage of already located shipwrecked. This variable adapts the erratic behavior and searching radio of the ANN.

The relationships between the four ANN inputs $[I_{t,1}, I_{t,2}, I_{t,3}, I_{t,4}]$ and the available measurements $\{S_{k,l}|D_{k,l} \neq \text{never}, k \leq t\}$ are presented in the following equations:

$$I_{t,1} = \mathrm{dir}\left(\sum_{i|D_{t-1,i}=\text{previously}} \left(S^x_{t,i} - S^x_{t-1,i}\right), \sum_{i|D_{t-1,i}=\text{previously}} \left(S^y_{t,i} - S^y_{t-1,i}\right) \right) \tag{3}$$

$$I_{t,2} = \sqrt{\left(\frac{\sum_{i|D_{t-1,i}=\text{previously}} S^x_{t,i}}{\sum_{i=1:N} \mathrm{IsTrue}\left(D_{t-1,i} = \text{previously}\right)} - U^x_t \right)^2 + \left(\frac{\sum_{i|D_{t-1,i}=\text{previously}} S^y_{t,i}}{\sum_{i=1:N} \mathrm{IsTrue}\left(D_{t-1,i} = \text{previously}\right)} - U^y_t \right)^2} \tag{4}$$

$$I_{t,3} = \mathrm{dir}\left(\sum_{i|D_{t-1,i}=\text{just}} \left(S^x_{t,i} - S^x_{t-1,i}\right), \sum_{i|D_{t-1,i}=\text{just}} \left(S^y_{t,i} - S^y_{t-1,i}\right) \right) \tag{5}$$

$$I_{t,4} = \frac{\sum_{i=1:N} \mathrm{IsTrue}\left(D_{t-1,i} \neq \text{never}\right)}{N} \tag{6}$$

Note that to obtain $I_{t,1}$ and $I_{t,3}$, we don't divide the summations between N, because vectors $[\Delta X, \Delta Y]$ and $\left[\frac{\Delta X}{N}, \frac{\Delta Y}{N}\right]$ have the same orientation. Besides, the summations are over $D_{t-1,i}$ to ensure that there are at least two observations for the new located elements and more than two for the previously located ones.

The sensing ANN output cannot be directly A_t since it stores the next waypoint, in global Cartesian coordinates, that the UAV has to reach, and the ANN inputs $[I_{t,1}, I_{t,2}, I_{t,3}]$ are distances and orientations, which only provide relative information. Therefore, we choose as sensing ANN output $O_t = [O_t^r, O_t^\theta]$, a high level command that stores the displacement and orientation that the UAV has to follow to reach the waypoint A_t. The relationships between A_t and O_t are defined by the next expressions:

$$A_t^x = U_t^x + O_t^r \cos O_t^\theta \tag{7}$$

$$A_t^y = U_t^y + O_t^r \sin O_t^\theta \tag{8}$$

Figure 5 presents the complete sensing controller $A_t = \mathrm{c}(U_{t-1}, \{S_{k,l}|D_{k,l} \neq \text{never}, k \leq t\})$, that consists of the sensing ANN $O_t = [I_{t,1}, I_{t,2}, I_{t,3}, I_{t,4}]$, and the input and output translation processes (Equations (3-6) and (7-8)). The simplicity of the translation operations doesn't overload the complete sensing controller. Moreover, the use of a sensing ANN, whose inputs and outputs are relative coordinates, allows the complete sensing controller to extend the behaviour learnt through the information gathered on one point, to the rest of the space. Finally, the selected ANN is a feedforward neural network with four inputs $[I_{t,1}, I_{t,2}, I_{t,3}, I_{t,4}]$, two outputs $[O_t^r, O_t^\theta]$, and three layers with only four neurons in the input layer, eight in the hidden and two in the output.

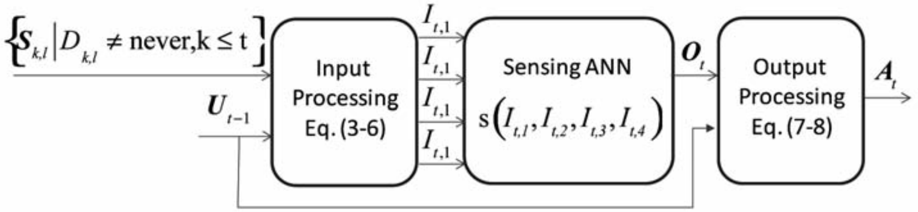


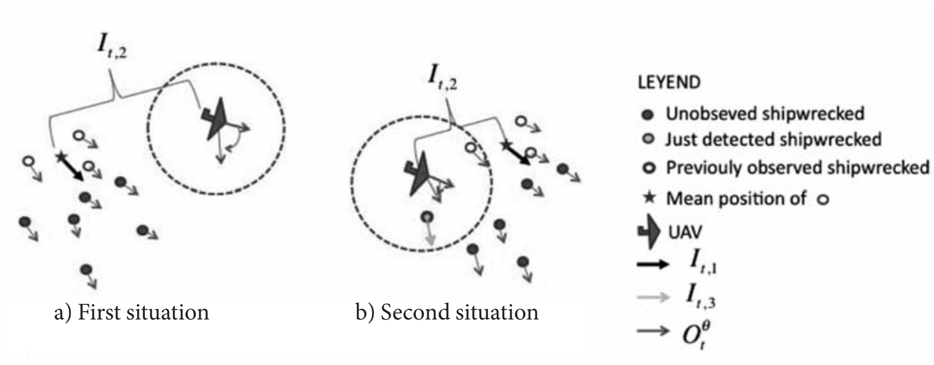
Figure 5. Complete sensing controller $A_t = \mathrm{c}(U_{t-1}, \{S_{k,l}|D_{k,l} \neq \text{never}, k \leq t\})$

Training

The pairs of inputs $[I_{t,1}, I_{t,2}, I_{t,3}, I_{t,4}]$ - outputs $[O_t^r, O_t^\theta]$ used to train the sensing ANN s are generated according to a set of rules defined for different situations. The next two situations illustrate the followed process:

— When none of the beaconed shipwrecked falls inside the UAV field of view, the UAV should return to the observation area. The O_t^θ and O_t^r that will drive the UAV towards it, can be obtained based on $I_{t,2}$ and $I_{t,1}$. Figure 6.a) illustrates this situation.

— When the UAV is flying according to the direction of the already observed shipwrecked and detects a new one that is moving with a different orientation, the UAV should correct its orientation to search new unobserved ones in the surrounding area of the just observed element. The O_t^θ that will drive the UAV towards it, can be obtained based on $I_{t,1}$ and $I_{t,2}$. Figure 6.b) illustrates this situation.



a) First situation b) Second situation

Figure 6. Training situations of the sensing ANN.

For each situation, we create a rule based on the parameters that govern it. With the rule, we generate pairs of inputs-outputs that are used to train the sensing ANN. The training step, performed only once for each UAV type, uses the UAV model $U_{t+1} = f(U_t, A_t)$.

The whole system

The onboard UAV controller is implemented as an expert system made up by the two controllers (see previous sections), alongside a manager in charge of the following:

1) Deciding in which working mode the expert system is, according to the information provided by the UAV onboard sensors. In particular, before the UAV vision system finds the first shipwrecked the expert system must work in prediction mode; once the first element is detected, it switches to sensing mode.
2) Controlling the number of already observed shipwrecked.
3) Managing exceptions related with the lapse until a new element is detected. The behaviour depends on the working mode. When the UAV is in prediction mode and no other element is observed for a period of time longer than originally expected, the UAV must send an alarm to the rescue centre requesting orders. When the UAV is in sensing mode and cannot find any new element during a designated period of time, the manager directly modifies the sensing ANN input parameter $I_{t,2}$, incrementing its value accordingly to the time that has passed since the last new observation, with the purpose of exploring areas farther away from the mean position of the already located shipwrecked.

The steps carried out after the rescue centre receives an alarm are the following:

1) As soon as the alarm is received, the rescue centre runs its numerical predictor $M_{t+1,i} = \mathrm{g}(M_{t,i})$ to generate the data pairs $M_{t,i}$ and $\Delta M_{t+1,i} = M_{t+1,i} - M_{t,i}$ used to train the prediction ANN $\Delta M_{t+1,i} = \mathrm{p}(M_{t,i})$.
2) Next, the trained prediction ANN $\Delta M_{t+1,i} = \mathrm{p}(M_{t,i})$ is loaded into the UAV expert system and then, the mission on prediction mode starts.
3) While flying in prediction mode, the expert system runs the complete prediction controller, that includes the prediction ANNs and prediction controller, to obtain the high level commands A_t that drive the UAV towards the mean of the predicted values of the shipwrecked positions.
4) Once the first shipwrecked is observed, the expert system stops the complete prediction controller and starts running the complete sensing controller, that includes the sensing ANN $O_t = \mathrm{s}(I_{t,1}, I_{t,2}, I_{t,3}, I_{t,4})$, and the input and output processing steps. When the UAV does not observe any new shipwrecked for a long period of time, it also modifies directly the input $I_{t,2}$ of the sensing ANN.

The whole expert system and these main steps are presented in figure 7, which does not include all the variables and connections to make it visually simpler. The missing information can easily be inferred from the previous figures and equations.

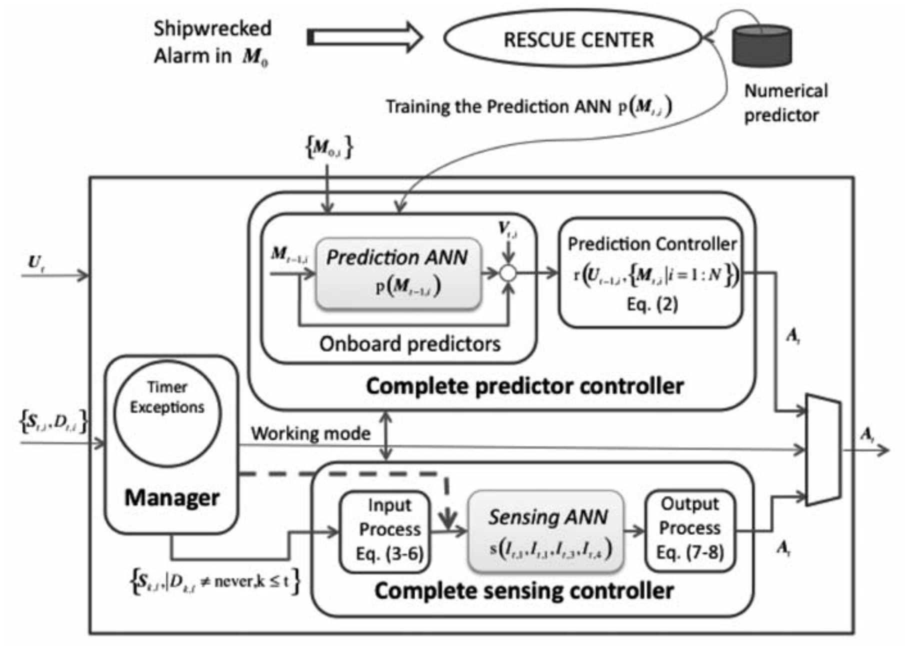


Figure 7. Expert system.

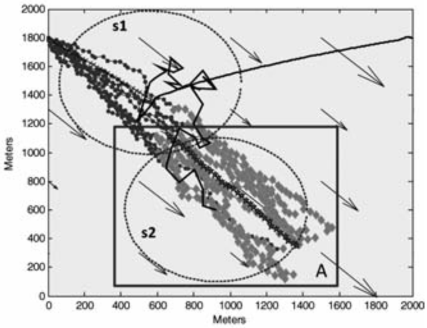
RESULTS

Next, we show the behaviour of the whole expert system in two simulated sea rescue tasks that differ in the distance that exists between the rescue centre and the wrecked area, and therefore, in the spread of the shipwrecked when the UAV finds the first element. In both cases, the total number of castaways N=10.

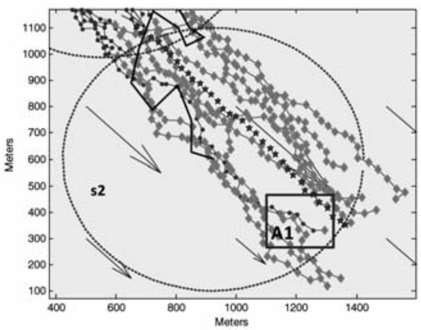
To run the experiments, we include our expert system in a MATLAB simulator that is also responsible of calculating at every time step the 'real' castaways and UAV positions. The shipwrecked movements, simulated during the two phases with the rescue centre predictor instead of the onboard ANN based predictor, are used to obtain the shipwrecked measurements ($S_{t,i}$ and $D_{t,i}$). The UAV positions (U_t) are obtained with a complex model that defines the UAV dynamics and includes an onboard low level controller in charge of stabilizing and driving the UAV towards the high level command positions $A_{t,i}$ obtained by the expert system. Besides, the simulator also randomly generates the initial positions of the shipwrecked in a small area around the inital wrecked position $M_{0,i}$. Therefore, the simulator closes the control loop: from the point of view of the expert system it applies its output ($A_{t,i}$) to the UAV to generates its inputs (U_t, $S_{t,i}$ and $D_{t,i}$) considering the UAV and shipwrecked simulated positions.

The results of the two experiments are presented in figures 8 and 9 using distinct glyphs for different elements and phases, whose meaning is shown at the legend at the bottom of figure 8. The simulated positions of the shipwrecked before each of them is first observed are presented with a dark grey circle and afterwards with a light grey diamond. Besides, to identify which point belongs to each castaway, we join them with a line in order to observe their trajectory too. The mean of the estimated positions of the shipwrecked at each time step before the first is observed is shown with a dark *x*. The mean of the observed positions of the detected shipwrecked after the first observation is presented with a dark star. When the shapes are not distinguishable, the differences on the grey levels among all these symbols can be used to identify these elements. The UAV trajectory is presented with a black line and the field of view at the positions when a new unobserved castaway is detected with a dashed circle identified as s#. The arrows represent the mean direction of the sea wind and currents. The first graphic inside each figure represents the whole experiment while the others show a zoomed region of the experiment (marked in the first figure with a square). Finally, note that as we draw the trajectories of the simulated shipwrecked and UAV, the represented unobserved simulated elements that fall inside the UAV field of view at a given time t don't necessary correspond to any of the simulated position they have at t. In other words, an unobserved castaway (circle) that is inside a dashed circle doesn't become observed (diamond) unless it was really inside the dashed circle at the correct time step.

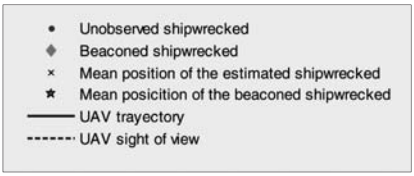
In the first experiment the ship wrecks at the upper left corner of figure 8.a) and the rescue centre is placed only 2000 m apart in the upper right corner. The rescue centre



a) Global view



b) A zone. Sensing mode

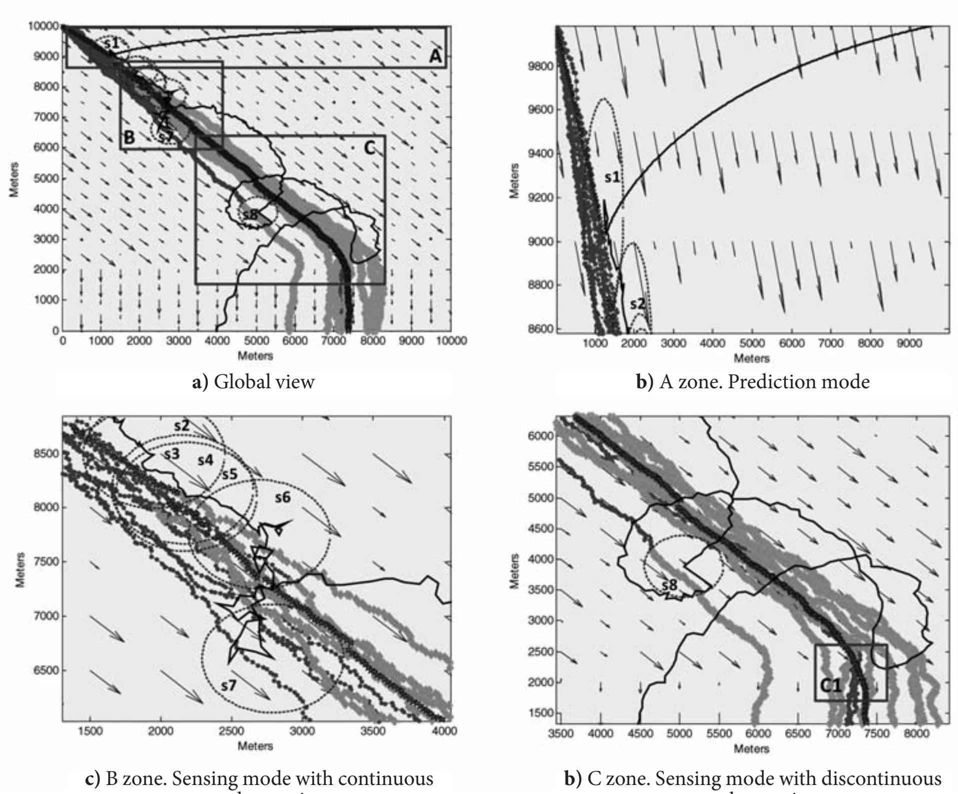


c) Legend

Figure 8. First experiment.

receives the message, trains the onboard prediction ANN for the current sea state, and the UAV starts flying on prediction mode towards the mean estimation obtained by the expert system, while the castaways are adrift by the simulator. Once the UAV has reached the online predicted spot, before the castaways have been significantly spread, it detects simultaneously all but one shipwrecked when its field of view is **s1**. Then, the UAV starts flying on sensing mode, tracking the 9 observed shipwrecked (light diamonds) while it searches the remaining (dark circles) following a zigzag trajectory close to the mean of the observed castways (dark stars). This behavior, shown in figures 8.a) and 8.b), continues until the UAV field of view becomes s2 and the UAV observes the last item in the A1 region. Then the UAV finishes its mission.

The setup of the second experiment, presented in figure 9, differs from the first on the distance of 10000 meters between the ship wrecked position and rescue centre. Figure 9.b) shows the prediction phase and how this bigger distance lets the sea winds and currents increment the simulated shipwrecked dispersion a lot before the UAV arrives at the online predicted spot and detects the first shipwrecked with the **s1** field of view. Figure 9.c) shows the part of the sensing mode behavior where the following 6 castaways are observed when the UAV fields of view are **s2**-**s7**, and so their dark circle glyph becomes a light diamond. In this area the UAV trajectory is close to the mean trajectory of the found castaways (dark stars) because the lapse between two new observations is small. Figure 9.a) and 9.d) show how the behavior changes after observing the 7th castaway, because no new observations are obtained for long and the expert system extends the UAV searching zone incrementing $I_{t,2}$ accordingly with the time without new observations. This allows locating elements that are further away from the mean trajectory of the detected ones. Figure 9.d) shows when the last castaway, moved away from the main



a) Global view

b) A zone. Prediction mode

c) B zone. Sensing mode with continuous new observation.

b) C zone. Sensing mode with discontinuous new observations

Figure 9. Second experiment.

group, is found at s8. At this moment, the searching radio is reduced to locate closer castaways. As time passes without new findings the searching radio increases again. The remaining two shipwrecked are not found in the represented experiment. Finally, although in region C1 the UAV and unobserved trajectories are really close, the UAV does not find them because their position at the same time step is not.

CONCLUSIONS

In this paper, we present a new expert system, based on neural networks, to guide a UAV that has to search and locate castaways on a wide area after a shipwreck. The expert system is designed to work in real time on board of any UAV, using the prediction or the measurements of the castaways provided by the UAV. So far, it works successfully with little information about the castaways position and really simple user-defined behaviors. We plan to expand both in the near future, including statistical techniques to tack the observed castaways, incrementing the types of inputs of the sensing ANN and training it with more complex behaviors.

REFERENCES

Alvarez Fanjul, E. (2005): Establecimiento de un Sistema Español de Oceanografía Operacional. *Revista de Puertos del Estado*. March 2, 57-64.

ASF, http://www.aa.washington.edu/research/asfl [Accessed 18 December 2010]

BERK, http://vehicle.me.berkeley.edu [Accessed 18 December 2010]

Besada-Portas, E. de la Torre, L. de la Cruz, J. M. de Andrés-Toro, B. (2010): Evolutionary Trajectory Planner for Multiple UAVs in Realistic Scenarios. *IEEE Transactions on Robotics* 26 (4), 619 – 634.

CALT-Mur, http://www.cds.caltech.edu/~murray/projects/darpa01-mica [Accessed 18 December 2010]

Farzad Kamrani and Rassul Ayani (2009): UAV Path Planning in Search Operations, Aerial Vehicles, Thanh Mung Lam (Ed.), Available from: http://www.intechopen.com/articles/show/title/uav_path_planning_in_search_operations [Accessed 10 December 2010]

Gómez Lahoz, M. and Carretero Albiach, J. C. (2005): Wave forecasting at the Spanish coasts. *Journal of Atmospheric & Ocean Science*, 1741-7546, Volume 10 (4), 389-405.

Haykin, S. (1999): Neural Networks: A comprehensible Fundation. Second edition, Prentice-Hall New Jersey, USA.

Hontoria L., Aguilera J., Riesco J. and Zufiria P. (2001): Recurrent Neural Supervised Models for Generating Solar Radiation Synthetic Series. *Journal of Intelligent and Robotic Systems*, 31 (1-3)

Howden D. and Hendtlass T. (2008): Collective intelligence and bush fire spotting, in *Proc. Genet. Evol. Comput. Conf.*, pp. 41–48.

Mackay D.J.C. (1992): Bayesian interpolation, *Journal Neural Computation,* Volume 4 (3), pp. 415 - 447

MAGIC, http://magicc.et.byu.edu/cmsmadesimple [Accessed 18 December 2010]

MERCATOR, http://www.mercator.eu.org/html/mercator/index_en.html [Accessed 18 December 2010]

MICA, http://www.darpa.mil/ixo/mica.asp [Accessed 18 December 2010]

MIT-How, http://www.mit.edu/people/jhow [Accessed 18 December 2010]

Patterson D.W. (1996): Artificial Neural Networks. Theory and Applications, *Prentice Hall.*

Rubio J.C., Vagners,J, and Rysdykz R. (2004): Adaptive Path Planning for Autonomous UAV Oceanic Search Missions, *AIAA 1st Intelligent Systems Technical Conference*, 20 - 22 September, Chicago, Illinois, pp 1-10.

Tian J., Shen L., and Zheng Y. (2006): Genetic algorithm based approach for multi-UAV cooperative reconnaissance mission planning problem, in *Lecture Notes in Computer Science*, vol. 4203. Berlin, Germany: Springer-Verlag, pp. 101–110.

SISTEMA EXPERTO PARA GUIADO DE VEHÍCULOS AÉREOS NO TRIPULADOS BASADO EN REDES NEURONALES ARTIFICIALES

RESUMEN

En las tareas de rescate marítimo es crucial alcanzar la zona de la catástrofe en el mínimo tiempo posible, porque según éste aumenta suele crecer la dispersión de los naufragos y la dificultad de su búsqueda. Por lo tanto, el uso combinado de vehículos aéreos y marítimos no tripulados (UAVs y USVs) en este tipo de tareas suele resultar ventajoso, ya que el tiempo de llegada de los primeros es habitualmente mucho menor que el de los segundos. Teniendo en cuenta las capacidades de ambos tipos de vehículos, una distribución conveniente de la tarea de rescate consiste en asignarle a los UAVs las labores de localización y seguimiento de los naufragos, y a los USVs las de rescate de los naufragos localizados. Este artículo se centra en las partes de la tarea de rescate directamente relacionadas con el UAV.

Con este objetivo, se ha diseñado un sistema experto que genera las órdenes de alto nivel que indican al UAV hacia donde debe dirigirse para localizar los náufragos. Este sistema tiene que ser incorporado en un UAV, motivo por el que es conveniente minimizar su coste computacional y de memoria. Por esta razón, se han utilizado como núcleo del sistema experto un conjunto de redes neuronales, ya que además son fáciles de implementar e integrar en la tecnología existente. Por último, sus capacidades de respuesta en tiempo real y su alta adaptabilidad a diferentes situaciones, hacen que resulten elementos apropiados para resolver nuestro problema.

El sistema experto finalmente diseñado consta de dos tipos de redes neuronales: unas encargadas de predecir la posición de los náufragos antes de que estos sean localizados y otras de guiar su búsqueda una vez que el primer elemento ha sido encontrado. El primer tipo de red forma parte del subsistema que funciona durante la fase en la que el UAV sigue, de acuerdo con una ley de control muy sencilla, las posiciones predichas por este tipo de red. El segundo constituye la parte fundamental del subsistema que funciona durante la fase de sensorización y búsqueda. La figura 10 muestra un esquema de todo el sistema, en el que las operaciones primordiales son realizadas por las redes neuronales, explotándose así su eficiencia intrínseca.

Finalmente, queremos hacer notar que los parámetros de las redes neuronales utilizadas son obtenidos de dos procesos de entrenamiento diferentes de forma que las redes neuronales de predicción adaptan su comportamiento al estado de los vientos y corrientes de la zona de naufragio y que las redes neuronales de sensorización lo hacen al comportamiento sugerido por un experto para diferentes situaciones de rescate.