Using Neural Networks to predict aircraft trajectories

Yann Le Fablec
Ecole Nationale de l'Aviation Civile
7 Avenue Edouard Belin
31055 Toulouse Cedex France

Jean Marc Alliot
Centre d'Etudes de la Navigation Aérienne
7 Avenue Edouard Belin
31055 Toulouse Cedex France

Abstract ***This paper deals with the problem of predicting an aircraft trajectory in the vertical plane. A method depending on a small number of starting parameters is introduced and then used on a wide range of cases. The chosen method is based on neural networks. Neural networks are trained using a set of real trajectories and then used to forecast new ones. Two prediction methods have been developed: the first is able to take real points into account as the aircraft flies to improve precision. The second one predicts trajectories even when the aircraft is not flying. After depicting those prediction methods, the results are compared with other forecasting functions. Neural networks give better results because they only rely on precisely known parameters.***

Keywords: trajectory prediction, neural networks, aircraft.

1 Problem overview

Trajectory prediction is one of the most important problems to solve regarding Air Traffic Control. All control systems (conflict detection, monitoring, resolution, etc) heavily depend on the quality of the prediction. If horizontal prediction is quite accurate in 2D (i.e (x,y) plane), because aircraft can follow very precisely their route, predictions in 2D+1 ((x,y,t)) as well as in the vertical plane are still poor. In fact, heading can be forecasted precisely but speed prediction suffers a lack of precision. For the last years, many different technics have been tested:

- using flight equations;
- creating models of aircraft ([1], [2] or [3]) in order to simulate flights. These models are either based on simplified aerodynamic equations or tables giving speed corresponding to the current altitude
- using non-parametric methods ([4]).

The main problem with the first two methods is that they need some parameters that are not easy to get; vertical speed depends on various parameters such as the aircraft take off weight; thrust, drag and lift are functions of the aircraft type, of flight parameters given to the Flight Management Systems etc. These informations are usually unavailable to the ground control system. Moreover, such models should be developed for each type of aircraft. The third method does not take the aircraft type into account: it is just curve fitting. But, if this method is excellent for approximating functions in the middle of the curve, its precision is poor on the edges, where we look for the best prediction.

Our goal is to forecast aircraft trajectories from a reduced set of known points using neural networks. Given a set of initial radar plots of an aircraft (see figure 1), how can its future trajectory be forecasted? The method should not depend on parameters like weight, wind, operator's flight procedure or flight plans and only take into account the available information: the aircraft type and its RFL[1].

2 Principles

For a unique type of aircraft, we will use a set of recorded trajectories as a learning base for the neural network, and then evaluate its performance on a test set. The neural network used is made of neurons modeled as in [5]. It is feed-forward with a single hidden layer and the training procedure is the classic batch back-propagation algorithm as described in [6]. Time is discretized so that trajectories are represented with points sampled every 10 s. A vertical trajectory will then be a set of altitudes $z_0, z_1, z_2, \ldots$ where z_0 stands for the altitude at $t = 0$ s and z_i corresponds to $t = 10 * i$ sec-

[1]Requested Flight Level stands for the FL an aircraft wants to reach. 1 FL=100 ft

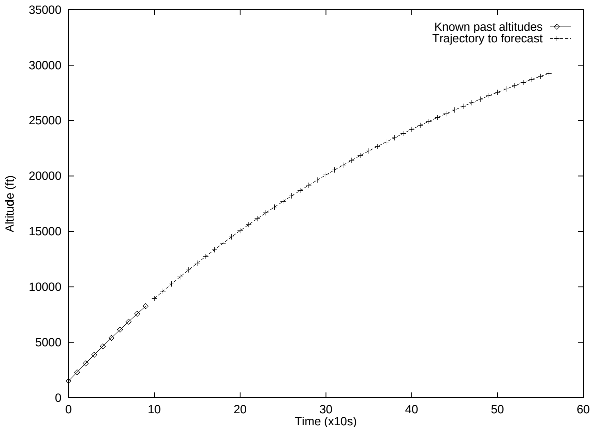

Figure 1: Example of prediction based on a known part of a climbing trajectory

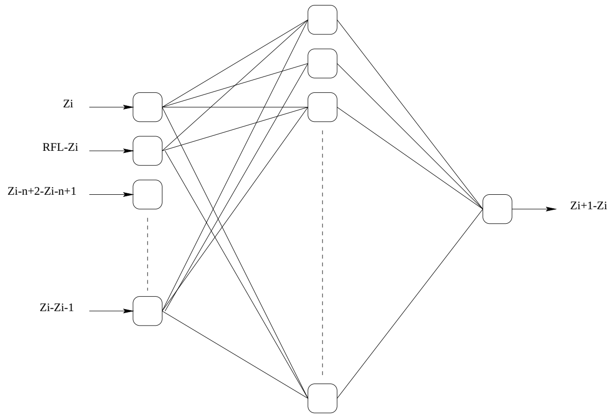

Figure 2: Network used for prediction

onds. The patterns that could be passed to the network would be composed of:

$$\underbrace{[z_i]}_{\text{current altitude}} \quad , \underbrace{[z_{i-n+1} - z_{i-n}, \ldots, z_i - z_{i-1}]}_{n \text{ past vertical speeds}}$$

as the input part while the output is

$$\underbrace{[z_{i+1} - z_i]}_{\text{speed to predict}}$$

It is important to note that the current altitude z_i is an essential input because an aircraft does not climb the same way at 1000 ft and at 30000 ft.
But, as an aircraft does not climb the same way when it is 5000 ft away from its final altitude and when it is only 500 ft away, we must also provide a fundamental information: the RFL. This data is given in the quantity $RFL - z_i$ where z_i represents the current altitude. The RFL is not directly provided because the aircraft has to intercept this flight level smoothly. So giving that difference helps it to know when to decrease the vertical speed. The neural networks architecture chosen is given on figure 2 and patterns used are composed of:

$$[z_i], [RFL - z_i], [z_{i-n+1} - z_{i-n}, \ldots, z_i - z_{i-1}]$$

as the input data and the output is still

$$[z_{i+1} - z_i]$$

2.1 Standard method

To forecast positions of the aircraft after the training phase:

- we consider $[z_i], [RFL - z_i], [z_{i-n+1} - z_{i-n}, \ldots, z_i - z_{i-1}]$ as the current pattern which is used by the trained network to predict $\hat{z}_{i+1} - z_i$. This gives the next forecasted altitude $\hat{z}_{i+1}$ corresponding to $t = 10 * (i+1)$ seconds

- we create next input pattern with this new position: $[\hat{z}_{i+1}], [RFL - \hat{z}_{i+1}], [z_{i-n+2} - z_{i-n+1}, \ldots, z_i - z_{i-1}, \hat{z}_{i+1} - z_i]$

- we go back to the first step using this pattern to forecast altitude $\hat{z}_{i+2}$. This process is repeated until climb is completed.

Then, a complete trajectory can be constructed using its first n known points with this standard (S) method. Datas are sampled each 10 seconds and the value of n is set to 10: each pattern will be composed of 10*10=100 s of flight.

2.2 Sliding windows

For practical applications, using sliding windows is very useful: an algorithm that would forecast positions in a too far future is not efficient, as prediction can be changed when modifications occur. The only way to do this is to include the real points in the patterns in order to anticipate further positions with a slight delay δt. The method is then:

- first, use n known speeds to make a prediction at $\delta t = 10 * \delta i$. Using $z_{i-n+1}, \ldots, z_i$ that represent the sliding window, the altitude $z_{i+\delta i}$ can be forecasted using the S method
- then, in order to predict $z_{i+\delta i+1}$ the next known point z_{i+1} is included and $z_{i-n+2}, \ldots, z_{i+1}$ is used as the starting data (i.e the sliding window is moved one step forward to forecast the next point)
- these steps are iterated until the aircraft has reached its RFL.

This method using sliding windows will be called the SW prediction method. As it is reactive, real initial points must be provided in order to get the prediction. If we want to get rid of these, we can try to forecast these starting altitudes with another neural network. This would give a trajectory prediction before the aircraft actually flies and then allow us to use it for avoidance systems for example.

2.3 Two networks method

Another forecasting method can be built thanks to another network learnt with the following patterns:

$$[z_0], [RFL - z_0] \rightarrow [z_1 - z_0, \ldots, z_n - z_{n-1}]$$

The starting altitude z_0 and the remaining altitude to reach the RFL are given. The n first initial speeds have to be predicted. After the training phase, a complete trajectory can be predicted using two networks (see figure 3). The other network will use the initial climbing altitudes provided by the first one and then forecast the rest of the trajectory until the aircraft reaches its RFL. The prediction method can not use sliding windows because we do not have real points at our disposal. Then the S method will be used instead. As two networks are used, the complete prediction method is called the TN method.

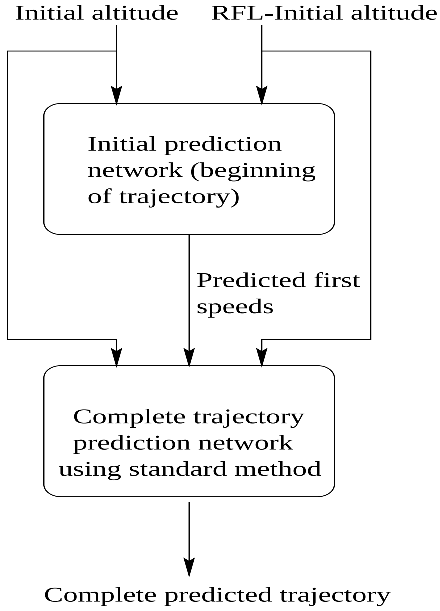

Figure 3: Complete prediction with the TN method

Table 1: Mean, Max errors and standard deviation for the SW prediction of learnt and non-learnt trajectories

δ_t	Mean err.	Max err.	Std deviation
1 min	10 ft	267 ft	9 ft
2 min	166 ft	2298 ft	88 ft
3 min	266 ft	3165 ft	145 ft
4 min	332 ft	3416 ft	186 ft
5 min	448 ft	3575 ft	212 ft

3 Results

The first method using sliding windows has been applied with a network trained with 142 trajectories from the learning base and used for prediction with these 142 trajectories and with 50 non-learnt flights. Table 1 shows the corresponding results while figures 4 and 5 give examples of prediction.

The smaller δt, the best the prediction. Indeed, if this parameter is small the method is more reactive and therefore is able to adapt rapidly to the changes occurring in the trajectory. Even if the sliding window is not used (last line in the table) prediction is not bad. Furthermore, networks can adapt to non-learnt trajectories.

Obviously neural networks are able to predict aircraft trajectories with the S and SW methods. In order to test the S and TN methods, the same set of learnt and non-learnt trajectories is used for prediction. The corre-

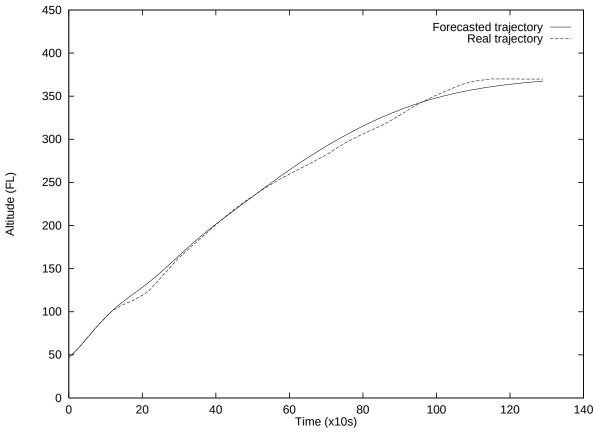

Figure 4: Example of a real trajectory and the corresponding prediction

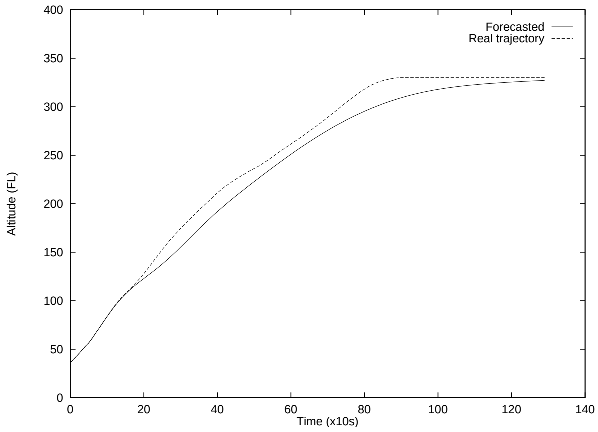

Figure 5: Real trajectory not included in the learning base and corresponding prediction

Table 2: Errors obtained with S and TN methods

Method	Mean err.	Max err.	Std dev.
S	582 ft	3612 ft	348 ft
TN	617 ft	3744 ft	340 ft

sponding results are shown in table 2.

The average error only increases by 35 ft when comparing TN to S method. It represents only 6% of the total mean error so the TN method is acceptable. The TN method can predict an aircraft trajectory with very few datas: the initial altitude, the RFL and the aircraft type are the only parameters needed. Moreover, if the real trajectories are available while the aircraft flies, it is possible to improve precision by using past known vertical speeds and then give a reactive prediction with the SW method.

Next section compares these results to those obtained by other prediction schemes.

4 Comparisons

Let us first compare the previous algorithms with non-parametric methods: the results obtained in [4] are shown in table 3. These methods use the same kind of input data (altitudes or vertical speeds) but can only work with sliding windows. Neural networks are much more efficient in predicting trajectories as average errors are twice to four times smaller than with non-parametric methods. Furthermore, these methods can not give a prediction before aircraft actually fly.

Table 3: Mean, Max prediction errors obtained with non-parametric methods

δ_t	Mean err.	Max err.
1 min 40 s	500 ft	1900 ft
3 min 20 s	900 ft	5100 ft

We can also compare our results with methods using models of aircraft. Several classical prediction functions such as Cat/Mask, Cat/Petric, Strange/Petric and Strange/Cautra are in use. They have been studied in the THAALES (Trajectory prediction Handling Aircraft and Airspace models Linked in an Evaluation Software) project as described in [2]. They include two main components:

- aircraft models. Performances are given by a table or by an algorithm based on simplified aerodynamics equations;
- an algorithm which makes the prediction. It uses the model and the flight plan to fly the aircraft and then forecast its trajectory.

Table 4 shows the results obtained by these methods applied to the prediction of climbing aircraft. The trajectories used are from the same databases as the one used for neural networks. Moreover the experiments

Table 4: Mean, Max prediction errors obtained with models of aircraft

Prediction Func.	CM	CP	SP	SC
Mean Error (ft)	1539	1278	1269	1600
Max Error (ft)	5046	3947	6376	4109
Std deviation (ft)	980	898	902	1059

were conducted on an identical aircraft type which corresponds to an Airbus A320 class (EA32 type).

Again, neural networks give better results than these prediction functions. There are several explainations, including:

- lots of parameters are used by classical prediction functions while some of them are not accurately known (take off weight or weather conditions for example)
- standard speed profiles are used to fly the aircraft but these are different for each company. Moreover, they are not always available
- some of the models (Cautra) are not precise enough to get a good prediction

5 Conclusion

Neural networks are more efficient than existing non-parametric methods. Moreover, they outperform the technics currently used in the operational systems. Their application is simple and requires very few data. They can be used before and while aircraft fly. Experiments on trajectories in real time in an operational context are now the next step to complete.

References

[1] Christian Leclaire. *THAALES, Strange Trajectory Prediction, version 3.* CENA, 1994

[2] Christian Leclaire. *Technical reports on trajectory prediction technics and their evaluation.* Centre d'Etudes de la Navigation Aérienne, 1997.

[3] Thierry Mandon. *CAT: Computation of Aircraft Trajectories, version 2.* CENA, 1993.

[4] Christophe Bontemps. *Prévision stochastique de trajectoires : procédures paramétriques et non-paramétriques, Mémoire de DEA Informatique Fondamentale et Parallélisme.* ENAC, 1997.

[5] W.S. McCulloch and W.A. Pitts *A logical calculus of the ideas immanent in nervous activity.* Bulletin of mathematics and biophysics, vol. 5, 1943.

[6] Simon Haykin. *Neural networks, a comprehensive foundation.* IEEE Press, 1994.