Multi-Sensor Data Fusion for Helicopter Guidance

R. S. Doyle and C. J. Harris

Introduction

Helicopter crews are required to carry out a wide range of duties most of which involve operational safety. One particular flight safety problem that has relevance to a variety of different types of rotorcraft is assisting the pilot in obstacle avoidance. Obstacles, in this context, may typically include other aircraft, terrain features, buildings, bridges, overhead cables and poles. The problem is especially difficult in bad weather conditions, (such as fog, snow, and heavy rain), in the presence of heavy smoke, or dust, or at night. Also it must be noted that the dynamics of the moving obstacles (other aircraft etc) will not in general be linear or indeed even known *a priori*.

In order to avoid obstacles effectively a system must have knowledge about the obstacles' positions and likely future positions relative the system's own aircraft. Since the information being provided by the sensors will be imperfect, (i.e. it will have some uncertainty associated with it), and since the process model, which must be used to predict any future positions, will also be uncertain, the required positions must be robustly estimated. Since the dynamics of moving obstacles will be *a priori* unknown, it will be necessary to learn process models for them. Since the dynamics of the obstacles may not be be linear, the process models must be capable of reflecting non-linear behaviour. The uncertain information produced by the various sensors will be related to required knowledge about the obstacles by a sensor model, however this relationship need not be linear, and may even have to be learnt. The information sources of interest may be distributed over many platforms, therefore the architecture of the data fusion system must reflect the spatially distributed nature of the problem.

In essence then, the main aim of this research is the design an estimator which is capable of dealing with non-linear process and sensor models, which may result from learning processes, and with distributed information sources.

Estimation

For helicopter obstacle avoidance, estimators are required for both localisation (estimating the helicopter's own current position), and obstacle tracking, (estimating the tracks of dynamic obstacles in the helicopter's environment). The design of both types of estimator is based around an asynchronous predictor-corrector estimator model.

Neuro-fuzzy networks

The estimators are neuro-fuzzy estimators since the models which describe the various relationships between inputs, states, and observations can be produced using fuzzy neural networks. Fuzzy neural networks based on B-spline associative memory neural networks[1], can provide real-time learning, guaranteed learning convergence and temporal stability, and as such they are a very efficient way of forming the required process and observation models. The real-time learning property also ensures that the estimators can be adapted on-line.

Once these networks have been trained they have the added benefit of being transparent. The fuzzy logic element allows a linguistic interpretation to be made, i.e. the system can provide a linguistic representation of what it has learnt. These networks may also be linguistically initialised by experts, making the learning process shorter and easier.

In addition there are closed form solutions for the partial differentiation of these networks, which means that the jacobians of the functions represented by the trained networks may be easily computed, then in turn used in an extended Kalman filter.

Kalman filter

Currently the Kalman filter and its various adaptations are the state estimators most commonly applied to dynamic tracking.

Some assumptions are made in the formulation of the Kalman filter equations which may mean that its practical performance is suboptimal. Firstly and most importantly the Kalman filter assumes that the process being estimated is linear. A variation of the Kalman filter, called the Extended Kalman filter, attempts to deal with this restriction. It is specifically designed to deal with non-linear processes. However this method does use a linear approximation based on the first terms of the Taylor series expansion of a non-linear process.

Assumptions are made about the type of noise that is catered for in the estimation process. Two sources of noise are considered, process noise, which amounts to an unknown vector forcing function input to some part of the system, and measurement noise which corrupts the system output before it can be measured. Process noise is therefore associated with the object and measurement noise is associated with the sensor. In order to achieve optimality in a minimum variance sense the Kalman filter assumes both forms of noise to be additive Gaussian white noise with zero mean. This may not be the case in practice, for example, gyroscopic sensor noise is generally considered to be multiplicative.

Selection of sensor suite

In selecting a sensor suite, the main aim was to provide a system capable of avoiding obstacles in a wide range of hazardous conditions which would be robust in the face of individual sensor failure but would involve minimum alteration to the aircraft. In light of these considerations the following sensor suite was chosen.

- Inertial Navigation System (INS)
- Radar Altimeter (RadAlt)

- Air Data System (ADS)
- Traffic alert and Collision Avoidance System (TCAS)
- Terrain database
- Global Positioning System
- Microwave radar
- Millimetric wave radar

Estimators

To date, two estimators to perform MSDF on the output of the sensors outlined above have been designed.

Approach 1: Kalman Filter using Neurofuzzy Models

In a general Kalman filter approach to the MSDF problem, typical process models are chosen, and observation models are constructed from manufacturers' specifications and simple geometry. This first approach simply uses trained neurofuzzy networks (B-spline neural networks) for process and observation models, and uses the Kalman filter algorithm for the fusion of data. This is possible due to the fact the output of the neurofuzzy networks may be easily partially differentiated with respect to the inputs, and so the jacobians necessary for the filter equations may be directly evaluated from the network's weights. The process and observation covariance however still need to be tuned in the same manner as for a normal Kalman filter approach.

The main advantage of this approach is the fact that process and observation models can be derived from observations of actual manoeuvres rather than being based on some hypothetical model. Another advantage of this approach is that the models can continue to be adapted online. This can be very useful in the case of the observation models in particular as the sensor characteristics may vary throughout the flight, e.g. due to thermal effects (warming up), they may also vary during the life of the sensor due to ageing.

Approach 2: Approximate Bayes Filtering using Neurofuzzy Models

This second approach highlights the innacuracies of the extended Kalman filter's approximation and presents a less biased Bayesian approximation. The accuracy of this method will be scalable (in terms of accuracy) dependent on the sampling frequency of probability density functions (pdf) used in the calculations, and also on the number and types of basis functions used in the B-spline neural networks.

The first point to notice about the extended Kalman filter's approximation is that the mean value of the prediction is calculated using only the mean value of the input probability density function (assumed Gaussian).

Consider the effect on the mean of the output of a general non-linear function $g(x)$ when the input $\mathbf{x}$ is a random variable with input distribution $f_x(x)$. The first thing to note is that the mean of the output will not in general be equal to the output of the function corresponding to the mean of the input. This is illustrated in Figure 1 for a function $g(x) = 3x^2$ and a Gaussian input pdf with

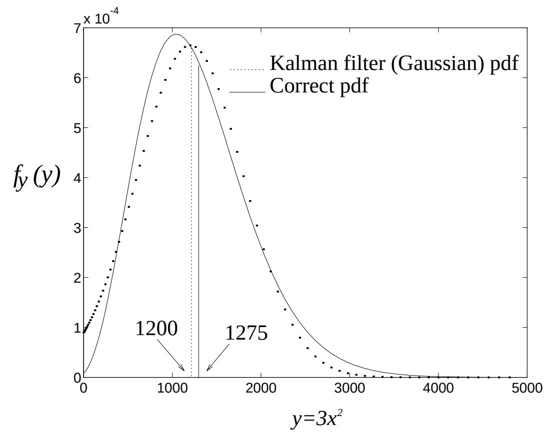

Figure 1: Comparison of actual output pdf and a pdf calculated using the extended Kalman filter

mean $\lambda_x = 20$ and standard deviation $\sigma_x = 5$. (Note in this example the actual value of the mean of the output is compared with the value calculated using an extended Kalman filter algorithm which assumes (incorrectly) that the mean of the output is equal to the output of the function corresponding to the mean of the input).

The true mean value of a function of n random variables (i.e. $g(\mathbf{x}_1, \ldots, \mathbf{x}_n)$) may be calculated as follows

$$E\{g(\mathbf{x}_1, \ldots, \mathbf{x}_n)\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(x_1, \ldots, x_n) f_x(x_1, \ldots, x_n) dx_1 \ldots dx_n \quad (1)$$

where $f_x(x_1 \ldots x_n)$ is the joint pdf of the n random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

Finding the mean value of the output distribution is only a first step; for the problem at hand some measure of the accuracy of the output is also desirable. From a probabilistic point of view the most complete measure would be the actual output pdf.

In the multi-input multi-output case the general expression for a joint output pdf given a joint input pdf, $f_x(x_1, \ldots, x_n)$ and n multi-input single output functions, $g_1(x_1, \ldots, x_n) \ldots, g_n(x_1, \ldots, x_n)$, is

$$f_y(y_1, \ldots, y_n) = \sum_i \frac{f_x(x_{1,i}, \ldots, x_{n,i})}{|J(x_{1,i}, \ldots, x_{n,i})|} \quad (2)$$

where $x_{1,i} \ldots x_{n,i}$ are the solutions to the system of equations equations

$$y_1 = g_1(x_1, \ldots, x_n) \ldots y_n = g_n(x_1, \ldots, x_n) \quad (3)$$

and

$$J(x_{1,i}, \ldots, x_{n,i}) = \begin{vmatrix} \frac{\partial g_1}{\partial x_{1,i}} & \ldots & \frac{\partial g_1}{\partial x_{n,i}} \\ \ldots & \ldots & \ldots \\ \frac{\partial g_n}{\partial x_{1,i}} & \ldots & \frac{\partial g_n}{\partial x_{n,i}} \end{vmatrix} \quad (4)$$

Equation 2 can also be used to find the joint density of $r < n$ functions. To do so, $n - r$ auxiliary equations,

$\mathbf{y}_{r+1} = \mathbf{x}_{r+1}, \ldots, \mathbf{y}_n = \mathbf{x}_n$, need to be introduced, then the joint density of the resulting system can be found using equation 2 and finally the marginal density of $f_y(y_1, \ldots, y_r)$ can be found by the following integration

$$f_y(y_1, \ldots, y_r) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_y(y_1, \ldots, y_n) dx_{r+1} \ldots dx_n \quad (5)$$

Equation 2 involves both the inverse and first partial derivatives of the functions $g_1(x_1, \ldots, x_n)$, …, $g_n(x_1, \ldots, x_n)$. If these functions are modelled by B-spline neural networks there exists simple formulas for finding partial derivatives of the networks [1]. Also there are several strategies which may be used to either provide exact solutions or approximate solutions to the inverse for non-singular inversions [1].

If the roots of the above equations cannot be found analytically then they will have to be found numerically for each point on the output pdf which may be very computationally expensive. Therefore it would seem sensible to look for other methods of approximating the output pdf.

One possible solution would be to sample the actual input pdf, compute the outputs corresponding to these samples, calculate the value of the output pdf at these points using equation 2, linearly interpolate between the calculated values for the output pdf, and finally normalise the resulting distribution, i.e. ensure

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_y(y_1, \ldots, y_n) dy_1 \ldots dy_n = 1 \quad (6)$$

The final normalisation step is necessary as the interpolated output pdf will only be an approximation to the actual output distribution and therefore in general it will not have the integration to 1 property required by pdfs. Using the same example as Figure 1, Figure 2 illustrates the result of such an interpolation strategy using 7 points

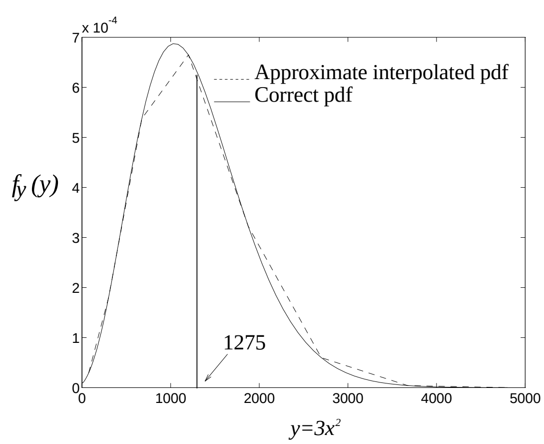

Figure 2: Comparison of actual distribution and interpolated distribution (7 points)

from the original Gaussian input pdf. It can be seen that the interpolated output pdf closely matches the actual pdf and the mean calculated using the interpolated pdf is correct to 4 decimal places.

Next Figure 3 compares the actual output pdf, the Kalman filter predicted pdf, and an approximate interpolated pdf (calculated using the above strategy) which only sampled the input distribution at 4 points. It may be difficult to decide visually whether the Gaussian pdf or the interpolated pdf matches the actual pdf more closely, however, it can be seen that the interpolated pdf does provide a closer approximation to the mean.

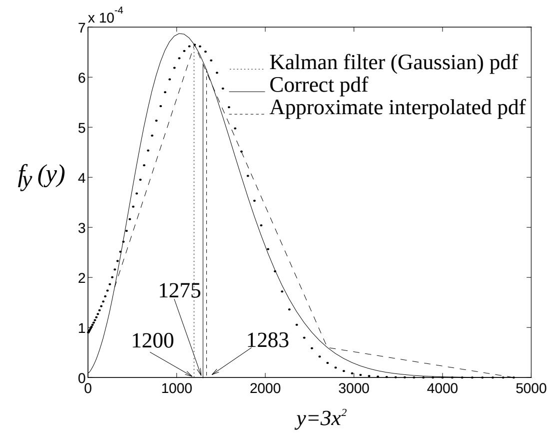

Figure 3: Comparison of actual output pdf, a pdf calculated using the extended Kalman filter, and an interpolated distribution (4 points).

So using the strategy outlined above, the predicted pdf and the observation pdf in state space may be computed. (Note the predicted pdf at this stage should be adjusted to reflect process noise, e.g. convolved with the process noise if the noise is assumed to be additive). It now remains to combine these pdfs to form the output pdf. This may be accomplished using a variation of Bayes rule, i.e.

$$f(\mathbf{x}|\mathbf{z}) = \frac{f(\mathbf{z}|\mathbf{x})\, f(\mathbf{x})}{\int_{-\infty}^{\infty} f(\mathbf{z} \mid x)\, f(x) dx} \quad (7)$$

Note if the pdfs are assumed to be independent then equation 7 is equivalent to finding the normalised intersection of the sets represented by the pdfs (where the intersection operator is taken to the the algebraic product).

This approach presents an effective non-linear estimator for systems where process and observation models must be learned. Note the accuracy of the approach is easily scalable and may be adjusted to meet the requirements of the system under consideration. This may be achieved in the selection of the basis functions for the process and observation models, and in the sampling frequency for the pdf approximations.

Conclusions

The proposed second approach fulfils the requirements for multi-sensor data fusion for helicopter guidance as outlined in the introduction, presenting an effective and scalable (in terms of accuracy) approach to the problem of non-linear estimation for systems where observations and process models must be learned. The approach is essentially Bayesian based on a standard predictor-corrector structure using B-spline neural networks for process and observation models. It exploits the strengths of probability, neural networks, and fuzzy logic. Probability is used to measure uncertainty, neural networks are used for their learning capabilities, and fuzzy logic is used to facilitate an understanding of what the neural network has learned and allow initialisation of the models based on an experts experience.

Acknowledgements

This work is being carried out with the financial support of the U.K. Department of Trade and Industry (DTI) and Lucas Aerospace. This research is part of on-going joint research project into advanced MSDF for helicopter control. The project partners are DTI, The UK Defence Research Agency (DRA) Farnborough, Westland Helicopters, and the University of Southampton.

References

1 M. Brown and C. J. Harris. *Neurofuzzy Adaptive Modelling and Control*. Systems and Control Engineering. Prentice-Hall, Hemel Hempstead, UK., first edition, 1994.