Journal of
Marine Science and Technology

Volume 9 | Issue 2 Article 1

A Fuzzy Autopilot Design Approach that Utilizes Non-Linear Consequent Terms

Robert Sutton
Department of Mechanical and Marine Engineering University of Plymouth UK

Paul J. Craven
Racal Research Limited Reading UK

A Fuzzy Autopilot Design Approach that Utilizes Non-Linear Consequent Terms

Acknowledgements

The authors wish to thank the Sea Systems Sector, DERA, Winfrith, for supplying the AUV model and for their continued support.

A FUZZY AUTOPILOT DESIGN APPROACH THAT UTILIZES NON-LINEAR CONSEQUENT TERMS

Robert Sutton* and Paul J Craven**

Keywords: underwater vehicle, fuzzy inference system, autopilot design.

ABSTRACT

This paper describes the development of a fuzzy autopilot for controlling the non-linear yaw dynamics of an autonomous underwater vehicle (AUV) model. The autopilot design is based on a new approach that uses a Gaussian fuzzy inference mechanism. For the design study, the AUV model is hosted in the MATLAB/Simulink environment. Simulation results are presented which illustrate the potential of this novel methodology.

INTRODUCTION

Previous autopilot designs [1, 2] have shown the attractiveness of fusing fuzzy logic with neural network architectures in the design process. The resulting controllers are linguistically interpretable in comparison to the black-box structure of a typical neural network. Additionally, these designs retain their learning and adaptation capabilities that allow the topographic fitting of the non-linear function representing the process input-output behaviour. Hardy [3] described the process of fitting a topographic surface to a given set of data points as

> *"given a set of discrete data on a topographic surface, reduce it to a satisfactory continuous function representing the topographic surface."*

This empirical modelling is directly analogous to the process of tuning a fuzzy inference system (FIS) to reproduce a particular output, given specific inputs. When tuning the membership functions of a FIS, a set of representative training data is collected over a number of pre-chosen variables, and is subsequently presented to the encoded FIS architecture. The parameters of the fuzzy premise (input) and consequent (output) functions within this architecture then adapt in order that the future presentation of a particular set of input data pairs produces the required output. Clearly, this training data represents known points on the topographic surface that is to be modelled; the ability of the resulting network to interpolate between these training data points is then regarded as a measure of network generalization.

In recent years, the theory of radial basis function (RBF) approximations to topographic surfaces has been established and shown to provide an excellent framework for modelling smooth non-linear functions. When considered as a network architecture, RBF approximations employ a linear combination of non-linear basis functions, each of which is defined within a particular operating region. Thus the overall network output is a linear combination of local network responses. Consequently, such RBF networks can be referred to as non-linear gain schedulers when employed in a control context.

This paper introduces a novel extension to the adaptive network-based fuzzy inference system (ANFIS) [4] regime, which employs the fuzzy concepts developed within ANFIS in conjunction with a RBF model for the inference system's consequent functions that will allow for better mapping and coverage of the output space to take place. The aim of such a fusion is to retain the linguistic interpretability of ANFIS whilst exploiting the functional non-linearity of such RBF local modelling techniques. The resulting fuzzy RBF network will be transparent in structure and easy to examine.

THE ANFIS APPROACH TO FUNCTION MODELLING

In the past the FISs employed variants of the Takagi-Sugeno (TS) [5] model, where the i_{th} fuzzy rule is of the form (for two inputs):

$$\text{If } x_1 \text{ is } A_i \text{ and } x_2 \text{ is } B_j \text{ then } f_k = f(x_1, x_2) \quad (1)$$

for:

Paper Received Dec. 11, 2000. Author for Correspondence: R. Sutton.

*Department of Mechanical and Marine Engineering University of Plymouth UK
**Racal Research Limited Reading UK

i = 1, 2, ..., m (the number of input membership functions on x_1),

j = 1, 2, ..., n (the number of input membership functions on x_2), and

k = 1, 2, ..., mn.

The f_k's are typically taken as a linear function of the input variables for a TS FIS. These rule outputs perform linear interpolation of the function $f(x)$ describing the input-output behaviour of the underlying model and are written as

$$f = \sum_{k=1}^{mn} \sum_{j=1}^{n} a_{kj} x_j + b_k \tag{2}$$

where f is a vector of *dependent* variables, the x_j's *are the independent* variables and n represents the number of independent variables. Also the coefficient a_{kj} represents the weight of the node corresponding to the (k, j) th position in the network architecture and the b_k coefficient represents the bias term of the k^{th} linguistic rule. In terms of a FIS, each linear rule can be envisaged as a moving singleton spike, its position in n-dimensional space being determined by the values of the input variables x_j.

Figure 1 provides an illustration of a FIS of this form, where one input variable is mapped onto one output variable via four fuzzy rules where y is a non-linear function and t is the operational range. Each individual linear rule is effective within a certain region. However, certain functions can prove difficult to model, and some mismatch between the model and the underlying function is not unusual, especially at the points where rules cross-over. Indeed, this input-output function could be particularly irregular with sharp gradients.

When approximating a non-linear function, one method to diminish the modelling error between local models and the actual function may be to increase the number of fuzzy rules mapping the function from input to output domains. Consequently, the width of each interval is reduced, as is the error incurred at the interval cross-over points. However, this leads to an increase in the number of parameters and inevitably lengthens the training period when adjusting these parameters to make the fuzzy model fit the desired function more closely.

Alternatively, the linear functions employed in a typical TS FIS output space can be replaced with non-linear functions that can better approximate the required input-output behaviour. Piecewise polynomials are often chosen for this purpose. This again implies that $\Re^n$ has to be divided into suitable regions for which a particular polynomial applies. Each polynomial must then be pieced together to provide continuity over the approximation, yet even polynomial function approximations are not always effective at modelling the somewhat sharp variations in many real topographic surfaces.

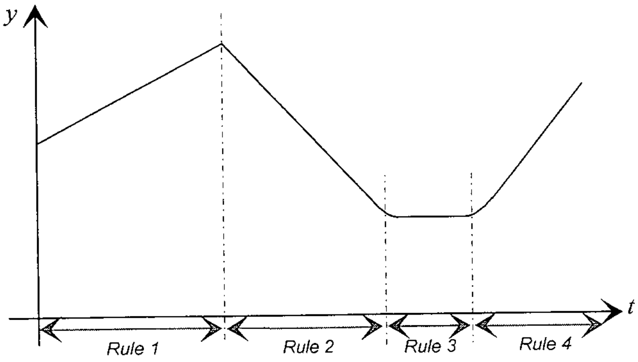

Fig. 1. Piecewise interpolation of a non-linear function using linear rule outputs.

One particular type of function that has been extensively studied takes the form

$$f_j = \sum_{k=1}^{n} \eta_k \xi_k(x) \tag{3}$$

where $\xi_k(x)$ is a non-linear function of the independent variables x_k, and the η_k's are unknown parameters. This type of model defines a non-linear relationship between the dependent variable f_j and the independent variables x_k, and defines a linear relationship between f_j and the unknown parameters η_k.. One clear advantage of employing this type of non-linear model is that the unknown parameters η_k can still be computed by a linear technique, as used in the hybrid learning rule of ANFIS.

Jang *et al.* [6] employed sigmoidal functions of the form

$$\xi_k = \frac{1}{1 + \exp[-(p_k x_1 + q_k x_2 + \cdots + v_k)]} \tag{4}$$

as their ANFIS consequents in contrast to the typically documented linear functions and where pk and qk are the weights applied to the input vector and v_k is the coefficient which pertains to the intercept term of the linear function of the input terms. This leads to a revised ANFIS structure that employs non-linear consequents of sigmoidal form and thus non-linear fuzzy rule outputs whose weights can still be found by linear algorithm approaches. The advantage of the chosen consequent functions is in their similarity to the original consequents of a typical ANFIS model, producing a non-linear structure that requires only a simple modification of the original ANFIS modelling regime.

Conversely, White and Sofge [7] state how the non-local nature of the commonly employed sigmoidal function can lead to neural networks that experience learning problems. In essence, each sigmoid does not relate to a specific region of the input space and thus incremental learning of the network parameters can yield conflicts between minimizing the network cost function and retaining the knowledge already stored within the network structure. That is, the use of the sigmoid neuron does not usually represent an overall network that can exhibit spatially localized learning properties. Various approaches have been documented to attempt to overcome these problems, such as local batch learning, very slow learning rates and distributed (uncorrelated) input sequences. However, these learning improvements were not documented or employed by Jang *et al*. [6].

Clearly in control applications the local model should represent the actual system within the desired range of influence, and therefore be a good approximation to the system locally [8]. Bossley [9] highlights that B-spline membership functions, which have been used extensively in neurofuzzy modelling by Brown and Harris [10], are not suited to locally modelling this relationship. A more suitable choice is often trapezoids, as in the work of Lin and Juang [11] or Gaussian functions, which are constant for the majority of their response but represent the system over the desired operating region. This requires that these functions be positioned in the areas of the output space $\Re_n$ which are most heavily populated by training data points.

Moreover, Poggio and Girosi [12] report the superior approximation power of neural network architectures containing RBFs over those using layers of sigmoidal functions. Indeed they provide a technical note on the different types of basis functions that can be employed.

These and other studies [13, 6, 14] have lead to the consideration of composite Gaussian functions as the consequent equations herein for non-linear rule implementation within an ANFIS type architecture.

THE APPROXIMATION PROBLEM

In order to approximate topographic surfaces and functions with modelling architectures, Poggio and Girosi [15] formulate the approximation / interpolation problem within the framework of regularization theory. They defined this problem as :

Given a set of data $S = \{(x_i, f_i)\}_{i=1}^{N}$ which is obtained by sampling an unknown function F in the presence of noise, the approximation problem is to recover the function F (or an estimate of it) from the sampled data set S.

This type of problem is referred to as ‘ill-posed’ since there exists an infinite number of possible solutions. In order to facilitate the approximation, some *a’ priori* assumptions are made about the unknown function. The function may be constrained to take a specific form $F(\eta, x)$ which is dependent on an unknown parameter vector η. Subsequently, the problem is transformed into one of regression. However, the resulting solution depends highly on the appropriateness of this *a’ priori* assumption or assumed functional form. Consequently, it is more usual to make the assumption that the unknown function $F(\eta,x)$ is smooth such that similar inputs produce similar outputs.

The regularization approach to the ‘ill-posed’ approximation problem determines the approximating function *f* that minimizes a cost function of the form:

$$H[f] = \sum_{i=1}^{N}(f(x_i) - y_i)^2 + \lambda\xi[f] \tag{5}$$

where λ is a positive constant generally known as the regularization parameter. In addition, $H[f]$ defines a cost function between the approximating function $f(x)$ and the function to be approximated y. Also an additional term is included ($\xi[f]$) to ensure the smoothness of the approximating solution. This smoothness functional incorporates a form of the second derivative of the function *f* to allow this smoothing to be incorporated. The cost function is clearly composed of two terms. The first term on the right hand side of Eqn. [5] minimizes the difference between the actual function being approximated and the approximation itself. The second term is included to encourage smoothness within the approximating function *f*, while the regularization parameter λ controls the weighting between the two components. Function smoothness is incorporated by defining a smoothness functional $\xi[f]$ in such a way that the lower values of the functional correspond to smoother functions. The regularization parameter λ can also be viewed as an indicator of the sufficiency of the given data set as examples that specify the solution $f(x)$. In particular, the limiting value $\lambda \to 0$, implies that the problem is unconstrained, with the solution $f(x)$ being completely determined by the examples. The other limiting value $\lambda \to \infty$ implies that the $a^{/}$ priori smoothness constraint is by itself sufficient to specify the solution function f(x), which also means that the data is irrelevant to the solution chosen by the smoothness constraint. In practice, the regularization parameter λ is assigned a value between these two extremes, so that both the sample data and the smoothness constraint contribute to the solution $f(x)$. Commonly the regularization parameter λ is chosen according to cross-validation techniques.

It has been shown in Poggio and Girosi [15] for a wide class of functional forms ξ that the minimization of Eqn. (5) (and thus the solution of the regularization problem) yields a solution of the form:

$$f(x) = \frac{1}{\lambda}\sum_{i=1}^{N}[y_i - f(x_i)]\xi(x, x_i) \tag{6}$$

where $\xi(x, x_i)$ is a basis function centred at point x_i. Eqn. (6) states that the solution to the regularization problem is a weighted sum of N basis functions centred at the sampled data points. Let $\eta_i = [y_i - f(x_i)] / \lambda$, then the solution to the regularization problem lies in an N-dimensional subspace of the space of smooth functions, and the set of basis functions $\{\xi(x, x_i)\}$ centred at the data points constitutes a basis for this space. If the basis function ξ is chosen from a set of rotationally and translationally invariant functions, then ξ becomes a radial symmetric function denoted by $\xi = \xi(||x - x_i||)$. Using η_i instead of $[y_i - f(x_i)] / \lambda$, the regularization solution is given in the form:

$$f(x) = \sum_{i=1}^{N}[\eta_i \xi(\|x - x_i\|) \tag{7}$$

Note that Eqn. (7) is almost identical to Eqn. (3), except that the basis functions are centred at data points x_i. Thus the method of RBFs with basis function centred at data points can be derived from regularization theory. Alternatively, regularization theory provides a firm theoretical background to the method of RBFs.

An approximating solution to the regularization solution can be formed by considering the basis functions to have moving centres. The regularization approach specifies the requirement for each training data example pair to be represented by a specific basis function. Obviously, this becomes computationally inefficient when the number of samples is large. Instead of using one function centre for each data point, a smaller number of function centres $\{\eta_i, i = 1, 2, ..., m, m \leq N)$ could be used to replace xi in Eqn. (7). It has been shown [15] theoretically that this approach constructs an approximation to the regularization solution. However, the number and location of the centres will strongly effect the outcome of the approximation. This leads to an approximating function of the form:

$$f = \sum_{\alpha=1}^{m}\mu_\alpha \xi(x, x_\alpha) \tag{8}$$

where α is the number of linearly independent functions (which are now fewer than the number of training data pairs), and the parameters x_α are called the centres of the basis functions denoted by ξ.

Each radial function is combined with a multiplicative weight η_α which determines the algebraic sign and gradient of the basis functions' slope, as shown in Fig 2 for a two input (dimensional) problem. Thus when the complete set of coefficients are substituted into the consequent functions and these functions are summed together, the overall output surface should fit the discrete data points exactly, providing logical interpolation for intermediary points. Such a surface is depicted in Fig 3.

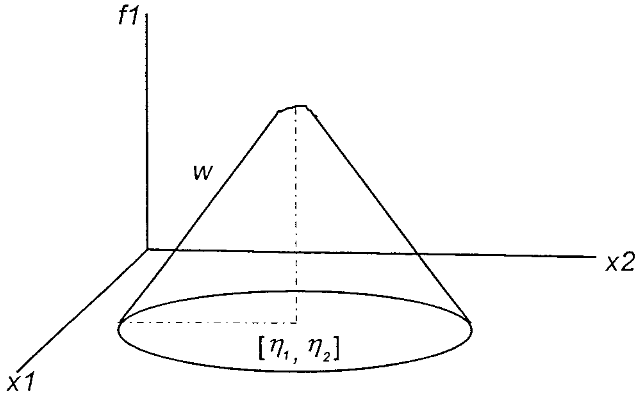

Fig. 2. A two dimensional representation of a Gaussian radial basis function.

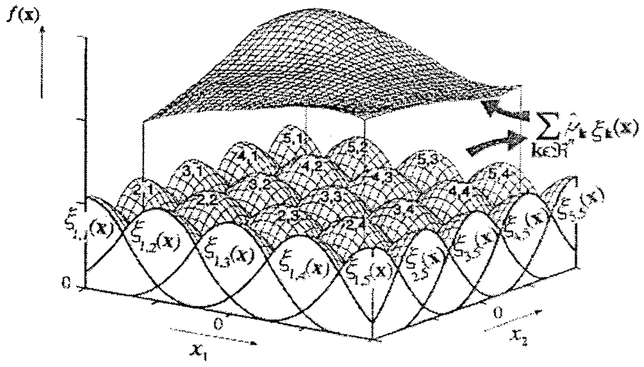

Fig. 3. Gaussian basis function modelling of a smooth surface (After Heiss and Kampl (1996).

Additionally, the shape of the overall surface at any particular point is a function of the summation of all the individual surfaces at significant points. This means that sub-surfaces (as depicted above in Fig 2) with particularly sharp gradients, which are not in close proximity to the point of interest can still have a direct influence on regions containing slower sloped sub-surfaces. In Fig 2, w represents the gradient of the particular basis function in question. This behaviour is an important feature of this method because it suggests the ability of the resulting approximation to provide smooth transition over the output space.

RADIAL BASIS FUNCTION APPROACHES TO CONTROL SYSTEM DESIGN

The current literature provides examples of RBF approaches to control system designs; of the existing literature, few are dedicated to fusing the approximation power of RBF networks with fuzzy logic. Indeed those few do not address the problem of rulebase interpretability and all employ neural network architectures consisting of three layers. As one of the main strengths of fuzzy logic is the availability of a clear rulebase for validation, the present work is driven towards producing a linguistically interpretable tuning structure fusing fuzzy logic with the locality properties of Gaussian basis functions.

McDowell *et al.* [16] implemented a RBF network as a multi-input multi-output (MIMO) bank-to-turn autopilot for a surface-to-air missile. This application was based upon Gaussian basis functions, which were trained to adapt and compensate for roll induced cross-coupling. An excellent evaluation of the resulting neural autopilot was provided in 3-dimensions (6 degrees of freedom) using a command to line of sight algorithm. Results obtained are employed in conjunction with a gain-scheduling autopilot design, in the presence of time-varying aerodynamic derivatives and control surface saturation constraints. The performance of the overall autopilot system is commended, however no verification tests appear to have been performed on the neural structure. Due to the lack of linguistic transparency of the resulting neural autopilot it is difficult to employ an expert to verify the resulting control rules.

Horikawa *et al.* [17] highlight this requirement, stating that five layered network architectures are preferable in order that the transparency of the control rules be preserved during learning. However, their consequent functions do not possess the attractive properties of the multi-dimensional radially symmetric functions used by McDowell *et al.* [16].

Choi and Hwang [18] present the application of a MIMO fuzzy RBF network for control of an autonomous submersible. The results of the fuzzy RBF network autopilot simulations appear promising, yet the network again only employs a three layered architecture and is thus difficult to verify, or document as a fuzzy rulebase.

Due to the limited availability of formal design methods and robustness verification techniques concerning fuzzy controllers, it is considered important by the present authors to have access to the tuned rule base as a means of control rule verification and analysis. Three layered representations of fuzzy RBF networks do not possess this transparency, and the applications within the literature typically ignore the important verification of the resulting structure.

NON-LINEAR CONSEQUENT FUNCTIONS OF N-DIMENSIONAL FORM

By employing composite Gaussian functions in place of the linear equations of the consequent layer of an ANFIS architecture, each fuzzy rule output becomes a non-linear function of the network inputs (Eqn. (8)). Rule outputs are then dependent on the width of the Gaussian basis function, its centre position in n-dimensional space and the multiplicative weight c_α. The overall effect of the architecture becomes that of a non-linear gain-scheduling controller, which is dependent on locally receptive fields. The i_{th} fuzzy rule is thus of the form:

$$\text{If } x_1 \text{ is } A_i \text{ and } x_2 \text{ is } B_j \text{ then } f_k = \eta_k \exp\left(\frac{-\|x_k - c_k\|^2}{2\sigma_k^2}\right) \tag{9}$$

for:

$i =$ 1, 2, ..., m (the number of input membership functions on x_1),
j = 1, 2, ..., n (the number of input membership functions on x_2), and
k = 1, 2, ..., mn.

Re-writing Eqn. (9) as the sum of composite Gaussian basis functions, yields the following system of n simultaneous linear equations for the unknown coefficients η_j:

$$\left.\begin{aligned}
f_1 &= \eta_1 \exp\left[\frac{-\|x_1 - c_1\|^2}{2(\sigma_1)^2}\right] + \cdots + \eta_n \exp\left[\frac{-\|x_1 - c_n\|^2}{2(\sigma_1)^2}\right],\\
f_2 &= \eta_1 \exp\left[\frac{-\|x_2 - c_1\|^2}{2(\sigma_1)^2}\right] + \cdots + \eta_n \exp\left[\frac{-\|x_2 - c_n\|^2}{2(\sigma_n)^2}\right],\\
f_n &= \eta_1 \exp\left[\frac{-\|x_n - c_1\|^2}{2(\sigma_1)^2}\right] + \cdots + \eta_n \exp\left[\frac{-\|x_n - c_n\|^2}{2(\sigma_n)^2}\right],
\end{aligned}\right\} \tag{10}$$

which when written in matrix notation yields once more

$$f = A\eta \tag{11}$$

where A is a matrix of Gaussian basis functions. Typically there will be fewer basis functions than available training samples, and consequently an initial estimate for the centre positions of each Gaussian is required. In this instance, the matrix A is not square and a sequential least squares estimate to the parameter vector η is often sought.

1. The Modified Fuzzy Inference Mechanism

As a result of employing this new form of conse-

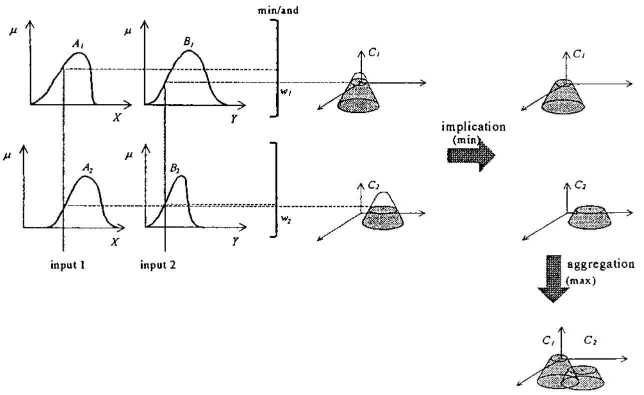

Fig. 4. The proposed composite Gaussian fuzzy inference diagram.

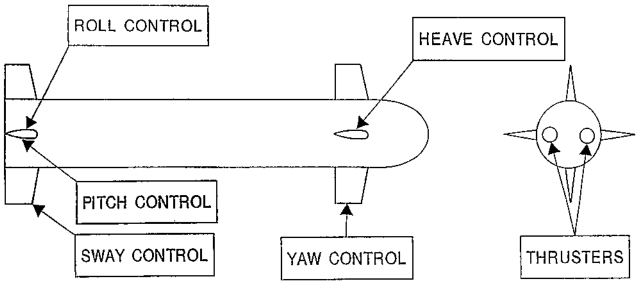

Fig. 5. The complete control authority of the AUV.

quent function, the fuzzy inference procedure changes. Figure 4 illustrates the proposed composite Gaussian fuzzy inference mechanism where μ is the degree of membership, and C_1 and C_2 are the consequential outputs. To elicit diagrammatic representation of the new output space, FISs considering two inputs and one output (with only two rules) are considered.

MODELLING THE AUTONOMOUS UNDERWATER VEHICLE DYNAMICS

Figure 5 shows the complete control authority of the autonomous underwater vehicle (AUV) model which is used as the test bed for the proposed autopilot designs. It should be noted that for this study the upper and lower canard rudders, situated at the bow of the AUV, are used to control the yaw dynamics are limited to + / - 25.2 degrees. Dimensionally, the model represents an underwater vehicle which is 7 m long and approximately 1 m in diameter and has a displacement of 3600 kg.

The equations of motion describing the dynamic behaviour of the vehicle in six degrees of freedom are implemented using a non-linear MATLAB / Simulink simulation model supplied by the Defence Evaluation and Research Agency (DERA), Sea Systems Sector, Winfrith. The model having been validated against standard DERA non-linear hydrodynamic code using tank test data and an experimentally derived set of hydrodynamic coefficients from the Southampton Oceanography Centre's AUTOSUB vehicle. In addition, the MATLAB / Simulink model structure also takes into account the dynamic behaviour of the actuators by describing them as first order lags with appropriate limiters. Further details concerning the model can be found in [2].

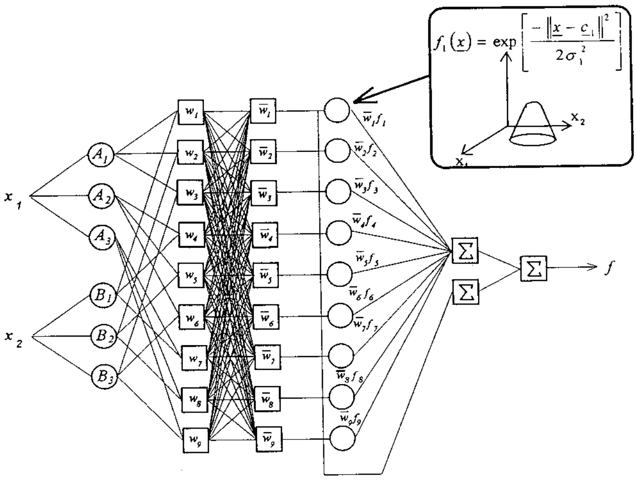

Fig. 6. The proposed Gaussian FIS for two inputs and one functional output.

RESULTS AND DISCUSSION

Based upon the proposed method of composite Gaussian fuzzy inference a 9 rule fuzzy autopilot was developed (2 inputs and one functional output). The architecture of this autopilot is shown in the schematic of Fig 6 in which each function within layer 4 is a composite Gaussian of dimension 2 (as illustrated for the first consequent function). The output domain of this new autopilot is represented graphically in Fig 3 and it is thus apparent that the two-dimensional consequent functions are hill shaped in nature. By varying the parameter set of the proposed consequent functions, any smooth function can be approximated, assuming that the centres are positioned correctly [14].

1. Applying the Hybrid Learning Rule

The linear dependence of the proposed consequent functions on their multiplicative weight coefficients elicits direct application of the hybrid learning rule for backpropagation tuning of the premise parameters and least squares tuning of the consequent weight coefficients η_k. (Full details of the hybrid learning algorithm are given in [2]. However, this assumes that the non-linear parameters (c_k and σ_k) of the Gaussian consequent functions remain fixed.

Upon completion of 300.5 epochs of tuning, the fuzzy sets of the hybrid rule tuned Gaussian autopilot were taken as depicted in Fig 7. Clearly some adaptation of the autopilots input parameter set has taken place.

The rule base of the hybrid rule tuned Gaussian autopilot was taken as Eqn. (12), where the coefficients η_k have been tuned using the least-squares algorithm and ψ_ε is the yaw error input to the architecture, ψ is the rate of change in error rate input and δ is the canard demand applied to the AUV model:

$$\text{If } \psi_\varepsilon \text{ is } N \text{ and } \dot{\psi} \text{ is } N \text{ then } \delta = 1.499 \bullet \exp\left[\frac{-(\psi_\varepsilon - 1)^2 - (\dot{\psi} - 1)^2}{2(0.4)^2}\right]$$

$$\text{If } \psi_\varepsilon \text{ is } N \text{ and } \dot{\psi} \text{ is } Z \text{ then } \delta = 0.368 \bullet \exp\left[\frac{-(\psi_\varepsilon - 0.75)^2 - (\dot{\psi} - 0.75)^2}{2(0.4)^2}\right]$$

$$\text{If } \psi_\varepsilon \text{ is } N \text{ and } \dot{\psi} \text{ is } P \text{ then } \delta = 2.091 \bullet \exp\left[\frac{-(\psi_\varepsilon - 0.5)^2 - (\dot{\psi} - 0.5)^2}{2(0.4)^2}\right]$$

$$\text{If } \psi_\varepsilon \text{ is } Z \text{ and } \dot{\psi} \text{ is } N \text{ then } \delta = 0.390 \bullet \exp\left[\frac{-(\psi_\varepsilon - 0.25)^2 - (\dot{\psi} - 0.25)^2}{2(0.4)^2}\right]$$

$$\text{If } \psi_\varepsilon \text{ is } Z \text{ and } \dot{\psi} \text{ is } Z \text{ then } \delta = 0.897 \bullet \exp\left[\frac{-(\psi_\varepsilon - 0)^2 - (\dot{\psi} - 0)^2}{2(0.4)^2}\right] \qquad (12)$$

$$\text{If } \psi_\varepsilon \text{ is } Z \text{ and } \dot{\psi} \text{ is } P \text{ then } \delta = 1.350 \bullet \exp\left[\frac{-(\psi_\varepsilon + 0.25)^2 - (\dot{\psi} + 0.25)^2}{2(0.4)^2}\right]$$

$$\text{If } \psi_\varepsilon \text{ is } P \text{ and } \dot{\psi} \text{ is } N \text{ then } \delta = 1.012 \bullet \exp\left[\frac{-(\psi_\varepsilon + 0.5)^2 - (\dot{\psi} + 0.5)^2}{2(0.4)^2}\right]$$

$$\text{If } \psi_\varepsilon \text{ is } P \text{ and } \dot{\psi} \text{ is } Z \text{ then } \delta = 0.576 \bullet \exp\left[\frac{-(\psi_\varepsilon + 0.75)^2 - (\dot{\psi} + 0.75)^2}{2(0.4)^2}\right]$$

$$\text{If } \psi_\varepsilon \text{ is } P \text{ and } \dot{\psi} \text{ is } P \text{ then } \delta = -0.601 \bullet \exp\left[\frac{-(\psi_\varepsilon + 1)^2 - (\dot{\psi} + 1)^2}{2(0.4)^2}\right]$$

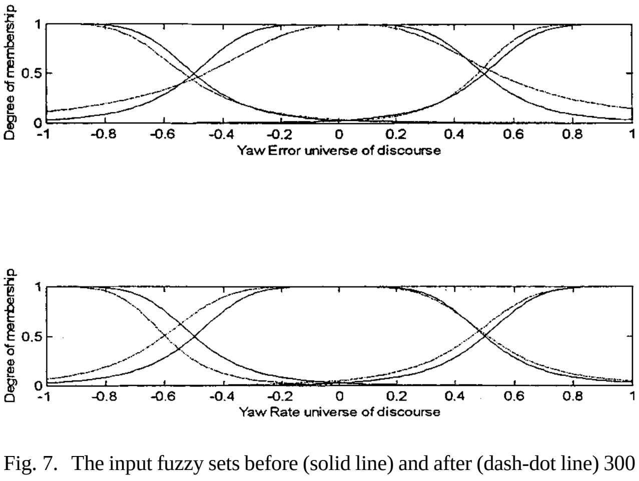

Fig. 7. The input fuzzy sets before (solid line) and after (dash-dot line) 300 epochs of tuning with the hybrid learning algorithm for the Gaussian FIS.

The course-changing response of the AUV, when employing this autopilot in the forward path is depicted in Fig 8; the corresponding low canard rudder response is given in Fig 9.

The initial positions of the non-linear consequent function centres (c_k) remained fixed throughout the tuning process; the widths of the consequents (σ_k) remain fixed also. As detailed in section 3, the rule base considers a depleted number of basis functions compared to typical RBF control structures, and therefore the correct positioning of the function centres is of paramount importance in achieving suitable approximations to the underlying models behaviour. Therefore

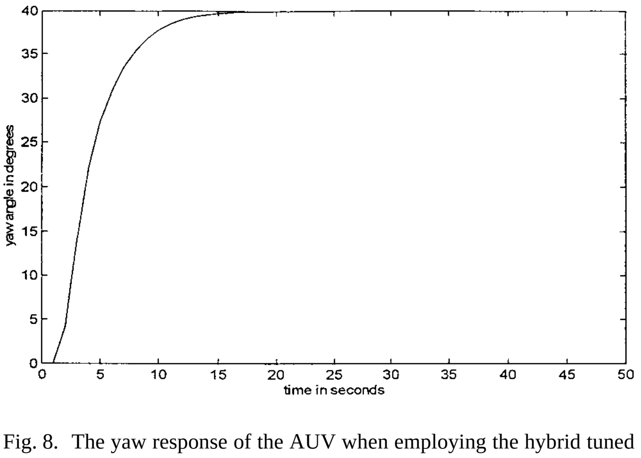

Fig. 8. The yaw response of the AUV when employing the hybrid tuned Gaussian autopilot for a 40 degree course-changing manoeuvre.

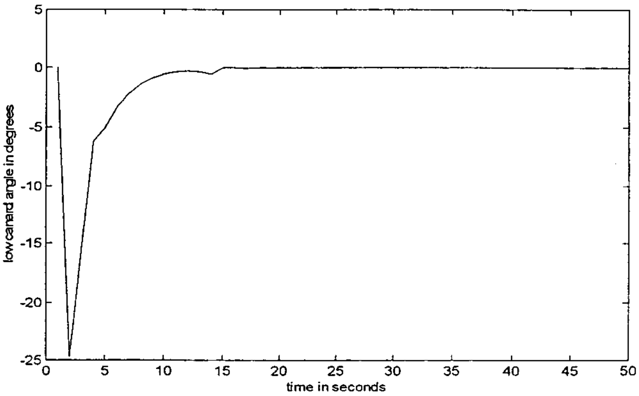

Fig. 9. The lower canard rudder response for the manoeuvre shown in Fig. 8.

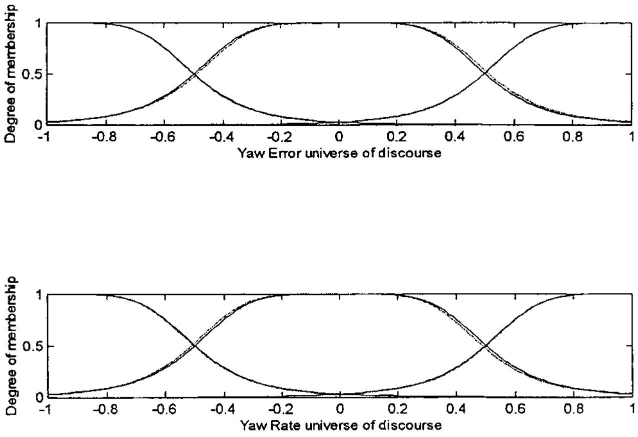

Fig. 10. The input fuzzy sets before (solid line) and after (dash-dot line) tuning with the extended hybrid learning algorithm.

to further improve the course-changing effectiveness of this autopilot it is important that the centres of the consequents be positioned correctly.

2. Extension to the Hybrid Learning Rule

To elicit full adaptation of the consequent functions it is necessary to employ a non-linear tuning algorithm. The backpropagation training element of the hybrid learning rule is clearly a convenient way to facilitate this. By extending the hybrid rule to adapt the non-linear parameters of the Gaussian consequent functions in the backward pass of each epoch, the whole non-linear parameter set can be locally tuned.

Applying this approach to the pre-tuned Gaussian autopilot, produced the premise fuzzy sets of Fig 10. The amount by which these sets have adapted to approximate the underlying model's behaviour is clearly less than in the case of the autopilot tuned using the original hybrid learning algorithm. This may be due to the fitting of the non-linear consequent functions parameters to the underlying model, which would alleviate the requirement for the non-linear parameter set of the input space to vary as significantly.

The rule base of the Gaussian autopilot after completion of 300.5 epochs of tuning via the extended hybrid learning rule was taken as follows (Eqn. (13)):

If ψ_ε is N and $\dot{\psi}$ is N then $\delta = 4.894$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon - 0.987)^2 - (\dot{\psi} - 0.998)^2}{2(0.394)^2}\right]$$

If ψ_ε is N and $\dot{\psi}$ is Z then $\delta = 1.471$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon - 0.735)^2 - (\dot{\psi} - 0.746)^2}{2(0.401)^2}\right]$$

If ψ_ε is N and $\dot{\psi}$ is P then $\delta = -0.788$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon - 0.489)^2 - (\dot{\psi} - 0.504)^2}{2(0.395)^2}\right]$$

If ψ_ε is Z and $\dot{\psi}$ is N then $\delta = -0.263$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon - 0.251)^2 - (\dot{\psi} - 0.241)^2}{2(0.401)^2}\right]$$

If ψ_ε is Z and $\dot{\psi}$ is Z then $\delta = 0.569$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon + 0.004)^2 - (\dot{\psi} + 0.003)^2}{2(0.394)^2}\right] \tag{13}$$

If ψ_ε is Z and $\dot{\psi}$ is P then $\delta = 0.395$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon + 0.261)^2 - (\dot{\psi} + 0.239)^2}{2(0.395)^2}\right]$$

If ψ_ε is P and $\dot{\psi}$ is N then $\delta = 0.397$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon + 0.485)^2 - (\dot{\psi} + 0.496)^2}{2(0.397)^2}\right]$$

If ψ_ε is P and $\dot{\psi}$ is Z then $\delta = 0.399$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon + 0.773)^2 - (\dot{\psi} + 0.770)^2}{2(0.399)^2}\right]$$

If ψ_ε is P and $\dot{\psi}$ is P then $\delta =0.403$

$$\bullet \exp\left[\frac{-(\psi_\varepsilon + 0.994)^2 - (\dot{\psi} + 1.003)^2}{2(0.403)^2}\right]$$

Where again ψ_ε, $\dot{\psi}$ and δ are as defined earlier in

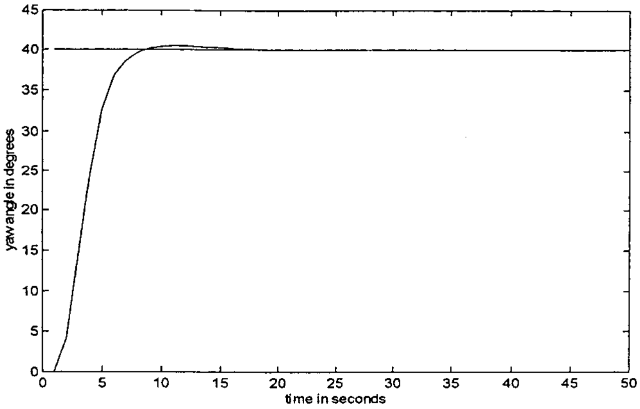

Fig. 11. The yew response of the AUV when employing the extended hybrid tuned Gaussian autopilot for a 40 degree course-changing manoeuvre.

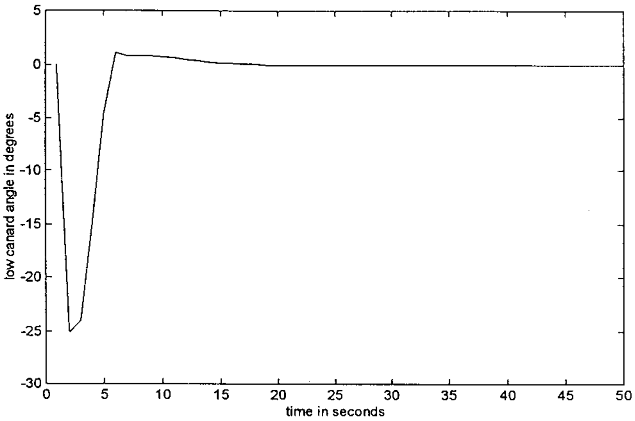

Fig. 12. The lower canard rudder response for the manoeuvre shown in Fig. 11.

the text. The application of this new algorithm produces a course-changing autopilot that incurs a fast accurate response when employed within the AUV model (Fig 11 and Fig 12).

For further comparisons, the rise times for the autopilots designed using the hybrid Gaussian and the extended hybrid Gaussian approaches are 14.31 s and 7.37s respectively. It can be seen the extended version provides an improvement in the rise time of 48%. However, as can be seen from the results, the penalty paid for the faster rise time is the introduction of 1.76% maximum overshoot in the yaw response.

CONCLUDING REMARKS

This paper has demonstrated that a novel type of fuzzy inference, namely Gaussian fuzzy inference, can be employed to develop AUV autopilot designs. The use of composite Gaussian RBF networks can provide a non-linear modelling technique which can be tuned with a linear algorithm such as least-squares. However, the work documented within the paper has highlighted the potential improvements to be gained by using a more sophisticated algorithm which made full use of the extra non-linearity introduced within the fuzzy mode.

ACKNOWLEDGEMENTS

The authors wish to thank the Sea Systems Sector, DERA, Winfrith, for supplying the AUV model and for their continued support.

REFERENCES

1. Sutton, R., Taylor, S. D. H, and Roberts, G. N. Tuning Fuzzy Ship Autopilots using Artificial Neural Networks. Trans Inst MC, Vol. 19, No. 2, pp. 94-106 (1997).
2. Sutton, R. and Craven, P. J. The ANFIS Approach Applied to AUV Autopilot Design. Neural Computing and Applications, 7, pp. 131-140 (1998).
3. Hardy, R. L. Multiquadric Equations of Topography and Other Irregular Surfaces. *Journal of Geophysical Research*, Vol. 76, No. 8, pp. 1905-1915 (1971).
4. Jang, J-S. R. Neuro-Fuzzy Modeling : Architecture, Analyses and Applications. PhD Thesis, University of California, Berkeley, USA, (1992).
5. Takagi, T. and Sugeno, M. Fuzzy Identification of Systems and its Applications to Modelling and Control. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, pp. 116-132 (1985).
6. Jang, J.-S. R., Sun, C.-T. and Mizutani, E. Neuro-Fuzzy and Soft Computing. Prentice-Hall, New Jersey, (1997).
7. White, D. A. and Sofge, D. A. Handbook of Intelligent Control. Multiscience Press Inc., Van Nostrand Reinhold, New York, (1992).
8. Hunt, J. K. and Johansen, T. A. Design and Analysis of Gain-Scheduled Control using Local Controller Networks. International Journal of Control, Vol. 66, No. 5, pp. 619-651 (1997).
9. Bossley, K. M. Neurofuzzy Modelling Approaches in System Identification. Ph.D Thesis, University of Southampton, UK (1997).
10. Brown, M. and Harris, C. J. Neurofuzzy Adaptive Modelling and Control. Prentice Hall, Hemel Hempstead (1994).
11. Lin, C.-T. and Juang, C.-F. An Adaptive Fuzzy Filter and Its Applications. IEEE Transactions on Systems, Man and Cybernetics–Part B: Cybernetics, Vol. 24, No. 4, pp. 635-656 (1997).
12. Poggio, T. and Girosi, F. Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks. Science, Vol. 247, pp. 978-982 (1990).

13. Poggio, T. and Girosi, F. A Theory of Networks for Approximation and Learning. A. I. Memo No. 1140, Center for Biological Information Processing, Whitaker College, A. I. Laboratory, Massachusetts Institute of Technology (1989).
14. Heiss, M. and Kampl, S. Multiplication-Free Radial Basis Function Network. *IEEE Transactions on Neural Networks*, Vol. 7, No. 6, pp. 1461-1464 (1996).
15. Poggio, T. and Girosi, F. Networks for Approximation and Learning. *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1481-1497 (1990).
16. McDowell, D. M., Irwin, G. W., Lightbody, G. and McConnell, G. Hybrid Neural Adaptive Control for Bank-to-Turn Missiles. *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 3, pp. 297-308 (1997).
17. Horikawa, S.-I., Furuhashi, T. and Uchikawa, Y. On Fuzzy Modelling Using Fuzzy Neural Networks with the Back-Propagation Algorithm. *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp. 801-806 (1992).
18. Choi, J.-L. and Hwang, C.-S. On the Fuzzy-Neural Controller for a Multivariable Nonlinear System. Proceedings of the 2nd Asian Conference, Seoul, South Korea, Vol. III, pp. 631-634 (1997).

使用非線性項的模糊自動導航設計方法

羅勃沙騰及保羅克萊門

模里茅斯大學機械及輪機工程學系

摘　要

本文描述一自主水下載具之非線性動力控制的模糊自動導航發展。此導航設計係採用高斯模糊推斷機制的新方法。就設計而言，AUV模式是建構在MATLAB/Simulink的環境下，模擬結果顯示此法之優越性。

關鍵字：水下載具，模糊推斷系統，自動導航設計