INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. **13**, 203–218 (1999)

AUTOMATIC STEERING OF SHIPS USING NEURAL NETWORKS

M. A. UNAR* AND D. J. MURRAY-SMITH

Centre for Systems and Control and Department of Electronics and Electrical Engineering, Rankine Building, University of Glasgow, Glasgow G12 8LT, U.K.

SUMMARY

Ship steering control system design presents challenges because the dynamic properties of the vessel itself vary significantly. The use of an artificial neural network as a controller which incorporates the properties of a series of conventional controllers designed for different operating conditions could provide an alternative to adaptive control or gain scheduling in this application. Local model network methods could also provide a basis for efficient modelling of the vessel over a range of operating conditions. The paper describes an investigation of radial basis function networks for ship steering control and of local model networks for representation of ship dynamics. Performance is demonstrated by a series of simulation studies. Copyright © 1999 John Wiley & Sons, Ltd.

Key Words: ship steering control; ship modelling; radial basis function network; local model network

1. INTRODUCTION

Automatic steering of ships has its origin near the beginning of this century, following the invention of the gyrocompass. Minorsky's[1] work on automatic ship steering was one of the principal contributions to the early literature in the general field of automatic control. In the same year, Sperry[2] introduced the first automatic steering control system for ships. These early autopilots were purely mechanical in construction and they provided a very simple steering action, the rudder demand being proportional to the heading error. To prevent oscillatory behaviour, a low gain was selected which rendered the device useful only in the course-keeping mode, where there was no significant desire for a high degree of accuracy in the response. When proportional-integral-derivative (PID) controllers became commercially available, they greatly improved the performance and until the 1980s almost all makes of autopilots were based on these controllers. A disadvantage of a PID controller is that it can provide optimal performance only at the operating point it is designed for. The ship parameters vary significantly with operating conditions such as with the forward speed of the vessel. Under these varying operating conditions,

* Correspondence to: M. A. Unar, Centre for Systems and Control and Department of Electronics and Electrical Engineering, Rankine Building, University of Glasgow, Glasgow G12 8LT, U.K. E-mail: unar@elec.gla.ac.uk

CCC 0890–6327/99/040203–16$17.50

it is tedious and difficult to determine properly the fixed parameters of the controller that result in good performance. Furthermore, the PID autopilots can cause difficulties when the ship makes large manoeuvres involving non-linear dynamic behaviour.

To avoid these problems of fixed structure PID autopilots, adaptive autopilots were introduced in the 1970s and have remained a major area of research.[3–13] It is because of their significant benefits such as improved fuel economy, increased speed of the vessel, and reduced manual settings to compensate for changes in operating and environmental conditions that they are attractive for such applications. Despite these potential benefits, adaptive control systems have some disadvantages. These include the following:

1. The design and analysis of non-linear adaptive systems is difficult and in comparison with neural networks leading to relatively expensive solutions in computational terms.[14]
2. Some forms of adaptive control systems do not have long-term memory and therefore do not retain the optimal controller parameters corresponding to different configurations of the plant.[15]
3. They need a significant amount of *a priori* information for successful application.[16]
4. There is some concern about potential instabilities associated with adaptive system behaviour.[17]

These and other disadvantages of adaptive control systems provide motivation for the use of artificial neural networks (ANNs). Artificial neural networks have the ability to handle variations of plant dynamics without the element of unpredictability that may cause concern when adaptive control is considered for safety critical applications. Witt *et al.*[18] report that a neuro controller can improve the profit margin of the vessel and contribute to the safety of the vessel by: (i) reducing manning levels required on the bridge (ii) achieving a fuel saving by allowing the vessel to stay on course with little deviation and (iii) providing accurate steering in an environment of increased traffic density and close proximity of obstacles.

The investigation of neural networks for ship steering control is in an early stage and only a few papers have appeared so far.[19–23] Almost all of these papers have made use of a multilayer perceptron (MLP) architecture and have trained neural networks by using the supervised learning of ANNs. This paper is based on a similar approach but it is different in two important respects: (a) we investigate the applicability of radial basis function (RBF) networks for developing ANN controllers at different speeds and (b) we also investigate the potential of local model networks (LMNs) for modelling the ship dynamics. Local model networks, or operating regime-based models, have an architecture which relates closely to ANNs. Such networks have already proved of value in other applications involving the modelling of systems in which the dynamic characteristics can vary significantly with the system operating conditions.[24,25]

The paper is organized as follows. Section 2 gives a general overview of ship steering control systems. Section 3 describes radial basis function networks and local model networks are introduced in Section 4. Simulation studies are presented in Section 5 in terms of two specific examples. Section 6 presents further discussion and conclusions.

2. SHIP STEERING CONTROL

Generally speaking, a ship steering control system is a single input–single output (SISO) control system, as shown in Figure 1, where Ψ_r is the reference heading, Ψ_d is the desired heading, Ψ is the

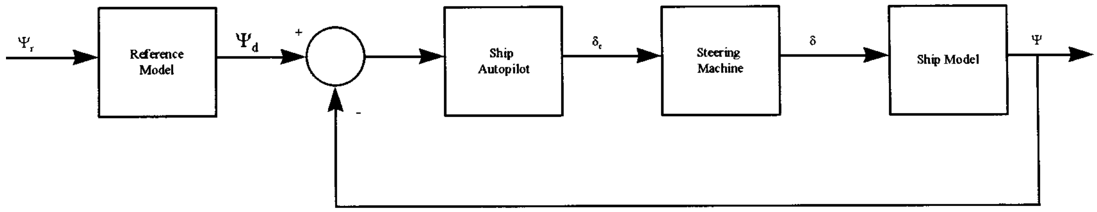

Figure 1. Ship steering control system

actual heading, δ_c is the commanded rudder angle and δ is the actual rudder angle (all in degrees). The four blocks of the figure are described in the sub-sections below. Additional background information about ship control systems may be found in appropriate texts, such as that by Fossen.[26]

2.1. Reference model

The dynamics of the reference model should be matched to the dynamics of the ship regardless of the magnitude of the demanded change of reference heading angle. A reference model which is too sluggish cannot produce an optimal performance since the ship cannot reach the required heading in the minimum time. On the other hand, we should not use a reference model which is too fast compared with the ship response characteristics because this may cause rudder actuator saturation and performance degradation. A reference model may be regarded as a prefilter which ensures that difficulties associated with large step changes of reference are avoided.[26]

Generally, a second-order reference model is used. Such a model can be described mathematically as follows:[27]

$$\frac{\Psi_d}{\Psi_r} = \frac{K_m}{T_m s^2 + s + K_m} \tag{1}$$

where T_m and K_m are the design parameters that describe the closed-loop behaviour of the system.

A second-order model may not be sufficient to generate smooth accelerations. In such situations, a third-order model of the following form can be used:

$$\frac{\Psi_d}{\Psi_r} = \frac{c_m}{s^3 + a_m s^2 + b_m s + c_m} \tag{2}$$

where a_m, b_m and c_m are constants. The method of computing these parameters can be found elsewhere.[17]

2.2. Steering machine

The function of a steering machine is to move the rudder angle to a desired heading when demanded by the control system or by the helmsmen. A simplified model of the steering machine is shown in Figure 2. This model of the steering machine was proposed by Van Amerongen.[27]

Generally, the rudder limiter and rudder rate limiters in Figure 2 are typically in the ranges: $\delta_{max} = \pm 35°$, and $\dot{\delta}_{max} = \pm 2$ to $\pm 7°$ s.

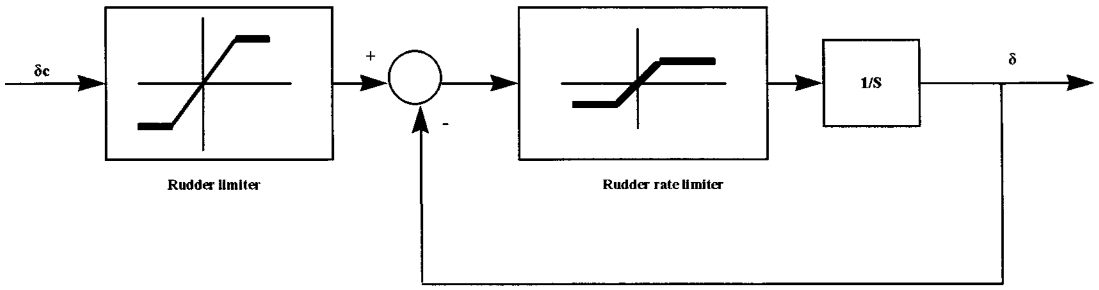

Figure 2. Simplified diagram of steering machine

2.3. Ship autopilot

An autopilot is a ship's steering controller, which acts within the overall control system to manipulate the rudder to decrease the error between the reference heading angle and the actual heading angle. Ship autopilots can be designed to perform two entirely different functions: course changing and course keeping. In course changing, the autopilot should provide good manoeuvrability, whereas in course keeping the ship should stay on a set course. Particular consideration has been given to the course changing problem in this paper.

2.4. Ship models

The equations describing the horizontal motion of a ship are well established. These equations can be derived by using Newton's laws expressing conservation of linear and angular momentum, and are given by[26,28,29]

$$
\begin{aligned}
m(\dot{u} - vr - x_G r^2) &= X \\
m(\dot{v} + ur + x_G \dot{r}) &= Y \\
I_z \dot{r} + m x_G(\dot{v} + ru) &= N
\end{aligned}
\tag{3}
$$

where m is the mass of the ship, I_z is the moment of inertia about the z-axis, x_G is the x co-ordinate of the centre of gravity, and u, v, and r denote surge, sway and yaw velocity, respectively. X and Y are the components of the hydrodynamic forces on the x and y-axis and N is the z-component of the hydrodynamic moments. The main difficulty in modelling ship dynamics is to find suitable expressions for X, Y and N. These are complicated functions of the ship motion. Various functional forms of these have been suggested in the literature. For example, Abkowitz[29] suggested the following functional form for X, Y and N:

$$
\begin{aligned}
X &= X(u, v, r, \delta, \dot{u}, \dot{v}, \dot{r}) \\
Y &= Y(u, v, r, \delta, \dot{u}, \dot{v}, \dot{r}) \\
N &= N(u, v, r, \delta, \dot{u}, \dot{v}, \dot{r})
\end{aligned}
\tag{4}
$$

and approximated the functions with Taylor series expansions about the steady-state conditions $u = u_0$, $v = r = \delta = \dot{u} = \dot{v} = \dot{r} = 0$. The derivatives of X, Y and N are called hydrodynamic derivatives.

Linearizing the equations of motion (3) about $v = r = 0$, $u = u_0$, and normalizing gives

$$\begin{bmatrix} m' - Y'_{\dot v} & L(m'x'_G - Y'_{\dot r}) \\ m'x'_G - N'_{\dot v} & L(I'_z - N'_{\dot r}) \end{bmatrix} \begin{bmatrix} \dot v \\ \dot r \end{bmatrix} = \begin{bmatrix} \frac{U}{L} Y'_v & U(Y'_r - m') \\ \frac{U}{L} N'_v & U(N'_r - m'x'_G) \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} + \begin{bmatrix} \frac{U^2}{L} Y'_\delta \\ \frac{U^2}{L} N'_\delta \end{bmatrix} \tag{5}$$

where L is the length of the ship, and U is its forward speed. In the above equations $Y'_v = \partial Y/\partial v$, $N'_v = \partial N/\partial v$ and so on for the other coefficients. The above equations have been normalized by using the Prime system of SNAME.[30] In this system, the length unit is taken as the length of the ship, the unit of time is L/U, and the mass unit is $1/2m\rho^3$ where ρ is the mass density of water.

From the above equations, we can easily obtain the well-known Nomoto's second-order model[31] that gives a relationship between r and δ as

$$\ddot r + \left(\frac{T_1 + T_2}{T_1 T_2}\right)\dot r + \frac{1}{T_1 T_2} r = \frac{K}{T_1 T_2}(T_3 \dot\delta + \delta) \tag{6}$$

A first-order approximation of (6) (known as Nomoto first-order model) can be obtained by letting the effective time constant be $T = T_1 + T_2 - T_3$ giving

$$\dot r + \frac{1}{T} r = \frac{K}{T}\delta \tag{7}$$

Nomoto's first-order and second-order models have been used extensively by control engineers for analysis and design of ship autopilots.[7,8,10,11,31–34]

Despite their popularity, the Nomoto models are only valid for small rudder angles and low frequencies of rudder action. Norrbin[35] suggests that if we need a steering equation also valid for large rudder angles, we have to substitute a non-linear characteristic, H_N, for r in equation (7):

$$H_N = \alpha_3 r^3 + \alpha_2 r^2 + \alpha_1 r + \alpha_0 \tag{8}$$

where α_i $(i = 0, 1, 2, 3)$ are called Norrbin's coefficients. For ships with a symmetrical hull, $\alpha_2 = \alpha_0 \approx 0$, and thus

$$H_N = \alpha_3 r^3 + \alpha_1 r \tag{9}$$

substituting H_N for r in (7), the corresponding Norrbin model can be obtained as

$$\dot r + \frac{\alpha_3}{T} r^3 + \frac{\alpha_1}{T} r = \frac{K}{T}\delta \tag{10}$$

3. RADIAL BASIS FUNCTION NETWORKS

The RBF network is a powerful feedforward neural network architecture. This type of network was originally introduced by Hardy[36] and the corresponding theory was developed by Powell.[37] These networks were originally applied to the real multivariable interpolation problem and were first formulated as neural networks by Broomhead and Lowe.[38] In earlier schemes of RBF networks, the number of basis/radial basis functions was necessarily equal to the number of data points. This was a serious limitation because, in many applications, the number of data points is very large. Later, this limitation was overcome, and now the number of radial basis functions is generally much smaller than the number of data points. During the past seven years, these networks have proved to be an attractive area of research within the neural network community and thus have found many applications in areas such as image processing, speech recognition, adaptive equalization, and signal processing. They are also gaining popularity in the field of systems and control. In this paper, we use these networks for a ship steering control system. The reasons of using RBF networks for the application are many and include the following:

(1) To the best of our knowledge, these networks have not been investigated for a ship steering control system by any author in the past. In view of their successful applications in other fields, it is appropriate to explore the use of these networks for this application.
(2) As mentioned earlier, only MLP networks have been investigated previously for this application. There is no straightforward rule for choice of the number of hidden layers and the number of hidden layer neurons in an MLP network for a particular application. There is no such problem in RBF networks.
(3) In theory, both MLP as well as RBF networks are capable of approximating any continuous non-linear mapping. However, Poggio and Girossi[39] emphasize that the property of approximating functions arbitrarily well is not sufficient for characterizing good approximation schemes, as many schemes have this property. Those authors proposed that the key property is not that of arbitrary approximation, but the property of *best approximation*. An approximation scheme is said to have this property if in the set of approximating functions there is one which has the minimum distance from the given function. The first main result of their paper is that MLP networks do not have the best approximation property. Secondly, they prove that RBF networks do have the best approximation property. This result is very significant and provides theoretical support for favouring RBF networks.
(4) RBF networks are generally faster than MLP networks for a given application.

An RBF network consists of three entirely different layers. The first layer, or the input layer, consists of a number of units clamped to the input vector. The hidden layer is composed of units, each having an overall response function, usually a Gaussian as defined below

$$g_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{\sigma_i^2}\right) \tag{11}$$

where $\mathbf{x}$ is the input vector, $\mathbf{c}_i$ is the centre of the kth RBF and σ_i^2 is its variance. The centres can be either fixed before the training phase or learned through the training of the network. The third layer computes the output function for each class as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{M} W_i \cdot g_i(\mathbf{x}) \tag{12}$$

where M is the number of RBFs and W_i is the weight of each RBF. Several approaches to training RBF networks are available in the literature. Most of these can be divided into two stages. The first stage involves the determination of an appropriate set of RBF centres and widths and the second stage deals with the determination of the connection weights from the hidden layer to the output layer. Indeed, the selection of the RBF centres is the most crucial problem in designing the RBF network. These should be located according to the demands of the system to be modelled. Several different algorithms are available in the literature for the selection of appropriate RBF centres. In this paper, we use the orthogonal least-squares (OLS) method developed by Chen *et al.*[40] A brief description of the method is given below:

Let us view the RBF network (12) as a special case of the linear regression model:

$$d(t) = \sum_{i=1}^{M} p_i(t) q_i + e(t) \tag{13}$$

where $d(t)$ is the desired output, the q_i are the model parameters, the $p_i(t)$ are the regressors and $e(t)$ is the error signal. In matrix notation, the above equation can be written as

$$\mathbf{D} = \mathbf{PQ} + \mathbf{E} \tag{14}$$

where

$$\begin{aligned}
\mathbf{D} &= [d(1) \dots d(N)]^{\mathrm{T}} \\
\mathbf{P} &= [\mathbf{p}_1 \dots \mathbf{p}_M]^{\mathrm{T}}, \quad \mathbf{p}_i = [p_i(1) \dots p_i(N)]^{\mathrm{T}}, \quad 1 \leqslant i \leqslant M \\
\mathbf{Q} &= [q_1 \dots q_M]^{\mathrm{T}} \\
\mathbf{E} &= [e(1) \dots e(N)]^{\mathrm{T}}.
\end{aligned}$$

The regressor vectors $\mathbf{p}_i$ form a set of basis vectors, and the least-squares solution of equation (14) satisfies the condition that the matrix product $\mathbf{PQ}$ should be the projection of $\mathbf{D}$ onto the space spanned by these vectors. The OLS method involves the transformation of the regressor vectors into a corresponding set of orthogonal basis vectors denoted by $\mathbf{w}_i$, $i = 1, 2, \dots, M$. For example, the standard Gram–Schmidt orthogonalization procedure may be used to perform this transformation, as shown by

$$\begin{aligned}
\mathbf{w}_1 &= \mathbf{p}_1 \\
a_{ik} &= \frac{\mathbf{w}_j^{\mathrm{T}} \mathbf{p}^{\mathrm{T}}}{\mathbf{w}_j^{\mathrm{T}} \mathbf{w}_k} \\
\mathbf{w}_k &= \mathbf{p}_k - \sum_{j=1}^{k-1} a_{jk} \mathbf{p}_i
\end{aligned} \tag{15}$$

where $1 \leqslant j < k$ and $k = 2, \dots, M$.

In the context of a neural network, the OLS learning procedure chooses the RBF centres $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_M$ as a subset of the training data vectors $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N$, where $M < N$. The centres are determined one by one in a well-defined manner, until a network having adequate performance is constructed. At each step of the procedure, the increment of the explained variance of the desired response is maximized. In this way, the OLS learning procedure generally produces an RBF network having a hidden layer which is smaller than that of an RBF network with randomly selected centres.

4. LOCAL MODEL NETWORKS

Local model networks (LMNs) were developed by Johansen and Foss[41,42] and also by Murray-Smith[43–45] and may be regarded as a special form of RBF network. An LMN is a set of models weighted by some activation function (see Figure 3). The same input signal is fed to each model

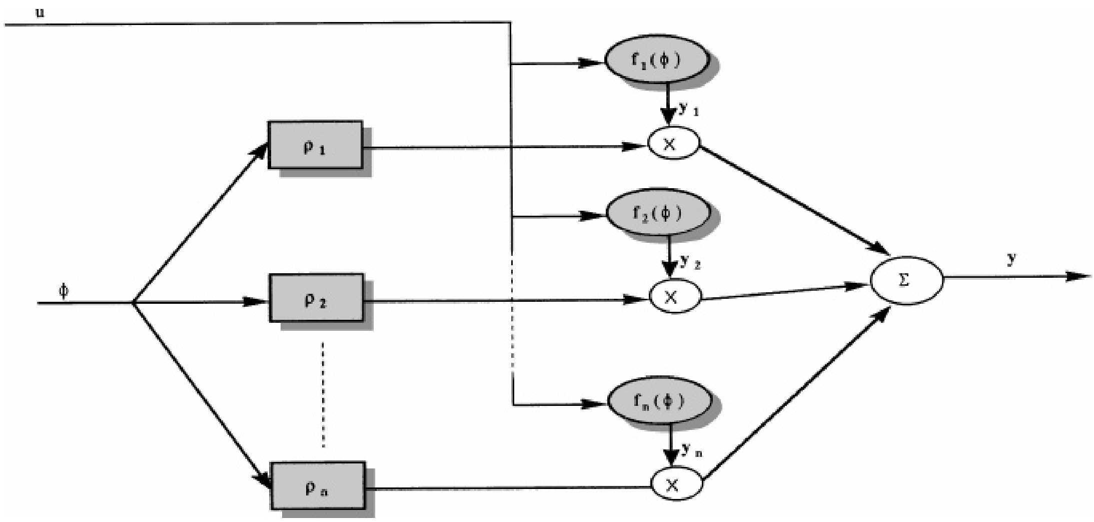

Figure 3. General architecture of LMN

and outputs are weighted according to some variable or variables, ϕ, to give the model network output as

$$y(t) = \sum_{i=1}^{n} y_i(t)\rho_i(\phi) \tag{16}$$

where $\rho_i(\phi)$ is the validity function (basis function) of the ith model, n is the number of models, and $y_i(t)$ is the output of the ith local model $f_i(\phi)$. The weighting or activation of each local model is calculated using an activation function which is a function of the scheduling variable. The scheduling variable could be a system state variable, an input variable or some other system parameter. It is also feasible to schedule on more than one variable and to establish a multi-dimensional local model network.[45] Although any function with a locally limited activation might be applied as an activation function, Gaussian functions are applied most widely. Other popular validity functions include B-splines[46,47] and Kernal functions.[48] In this paper we restrict ourselves to the Gaussian function. For modelling tasks the validity functions should form a partition of unity for the input space, i.e. at any point in the input space, the sum of all basis function activations should be unity. This is a necessary requirement for the network to globally approximate systems as complex as the basis function's local models. Werntges[49] discusses the advantages of normalization in RBF networks, promoting the advantages of a partition of unity produced by normalization. The basis functions can be normalized as follows:

$$\rho_i(\phi) = \frac{\rho_i(\phi)}{\sum_{j=1}^{n} \rho_j(\phi)} \tag{17}$$

The individual component (local) models f_i of an LMN can be of any form; they can be non-linear or linear, have a state–space or input–output description, or be discrete or continuous time. They can be of different character, using physical models of the system for operating conditions where they are available, and parametric models for conditions when there is no physical description available. They can also be ANN models such as MLP or RBF networks. The individual local models are smoothly interpolated by the validity functions ρ_i to produce the overall model.

The learning process in local model networks can be divided into two parts:

1. Find the optimal number, position and shape of the validity functions, i.e. define the structure of the network.
2. Find the optimal set of parameters for the local models, i.e. define the parameters of the network. These parameters could be the complete set of coefficients for a linear model, numerical parameters of a non-linear model, or even switches which alter the local model structure. The parameters are usually optimized using a least-squares output error criterion. The details of the criterion can be found in.[45,50,51]

The potential advantages of LMNs are summarized below:

1. An LMN has a transparent structure which allows a direct analysis of local model properties
2. An LMN architecture is less sensitive to the curse of the dimensionality than other local representations, such as RBF networks.
3. Non-linear models based on LMNs are able to capture the non-linear effects and provide accuracy over a wide operational range.
4. The LMN framework allows the integration of *a priori* knowledge to define the model structure for a particular problem. This leads to more interpretable models which can be more reliably identified from a limited amount of observed data.

4.1. LMNs for ship dynamics modelling

As mentioned earlier, ship dynamics change with the forward speed of the vessel. If we derive various linear (e.g. Nomoto models of equation (7)) or non-linear models (e.g. Norbbin Model of equation (10)) at different forward speeds of the ship, then an LMN network can easily be developed that could represent the ship model. The derivation of linear or non-linear models at a particular forward speed is well established and such models are already available in the literature for most commercial ships. Moreover, it is relatively easy to develop ship models at a particular operating condition from real data collected from a scale model of a ship or from ship sea trials data. These individual models, developed at different operating conditions, can be interpolated smoothly to form an LMN that could represent the ship in question for a wide range of operating conditions.

The training of LMNs for modelling ship dynamics is not difficult. In this paper, we use physically oriented models which are already available in the literature, as the local models. This means that the problem of parameter estimation and optimization is automatically solved.

In general, there is no straightforward rule for choice of the optimal number of local models for a particular application. This number is usually decided on the basis of the range of scheduling variable ϕ. For example, if we desire to develop a ship model for a range of forward speeds from 5 to 10 $\mathrm{m\,s^{-1}}$, then we can choose several local models derived separately at these speeds, and possibly at some other speed(s) within this range. However, our experience shows that only two local models will be sufficient for the above range of forward speeds. This will be illustrated with the help of simulation studies presented in the next section.

The selection of centres is a crucial problem in RBF networks when a Gaussian function is used as a validity function. This is not a problem in LMNs. These can be selected at the operating point about which the local model is developed. For example, if a local model network consists of two local models derived at 5 and 10 $\mathrm{m\,s^{-1}}$, respectively, then Gaussian functions centred at these speeds can be used as validity functions ρ_i. The width of the Gaussian function can be found in a number of ways, for example, by using the following formula:[52]

$$\sigma_i = k_\sigma \sum_{j=1}^{n} |c_i - c_{i,j}| \tag{18}$$

where c_i is the current centre and $c_{i,j}$ is the jth nearest neighbour to c_i. The scaling factor k_σ defines the degree of overlap between the validity functions.

5. SIMULATION STUDIES

In this section, two examples are presented which demonstrate the potential of RBF networks as well as LMNs for the application.

The approach which has been adopted for the training of RBF networks, involves the training of a single artificial neural network to represent a series of conventional controllers for different operating conditions. The resulting network thus captures, in a non-linear fashion, the essential characteristics of all of the conventional controllers. The reason for using a series of conventional controllers is obvious. A conventional controller can yield optimal performance only at a given operating condition. Its performance cannot be optimal at other operating conditions. If we use several conventional controllers designed at different operating conditions as supervisors during the training phase, the resulting neural network will be able to perform well for the range of operating conditions over which it is trained, and probably slightly outside that range due to the generalization property.

Example 5.1

This example involves simulation results carried out from a model of ROV Zeefakkel. This is a small ship of length 45 m. The motion of the ship can be described by the Norrbin model of equation (10) with the following set of parameters[27] at a forward speed of 5 $\mathrm{ms^{-1}}$: $T = 31$ s, $K = 0{\cdot}5\,\mathrm{s^{-1}}$, $\alpha_1 = 1$ and $\alpha_3 = 0{\cdot}4\ \mathrm{s^2}$. The RBF controller for this ship was developed as follows:

To generate data for training, two PID controllers were designed at 5 and 10 $\mathrm{m\,s^{-1}}$ by using the feedback linearization laws of Fossen and Paulsen.[12] According to these laws, if we represent a ship by equation (10), then the rudder angle can be computed as

$$\delta = \mathrm{m}a_\Psi + \frac{\alpha_1}{K}\dot{\Psi} + \frac{\alpha_3}{K}\dot{\Psi}^3 \tag{19}$$

where $m = T/K$ and the commanded acceleration a_Ψ is given by

$$a_\Psi = \ddot{\Psi}_{\mathrm{d}} - 3\lambda(\dot{\Psi} - \dot{\Psi}_{\mathrm{d}}) - 3\lambda^2(\Psi - \Psi_{\mathrm{d}}) - \lambda^3 \int_0^t (\Psi(\tau) - \Psi_{\mathrm{d}}(\tau))\,\mathrm{d}\tau \tag{20}$$

where $\lambda > 0$ is a constant. An RBF network was then trained using these controllers as supervisors. The inputs to the network were the following: (i) $\ddot{\Psi}_{\mathrm{d}}$; (ii) $(\dot{\Psi} - \dot{\Psi}_{\mathrm{d}})$; (iii) $(\Psi - \Psi_{\mathrm{d}})$;

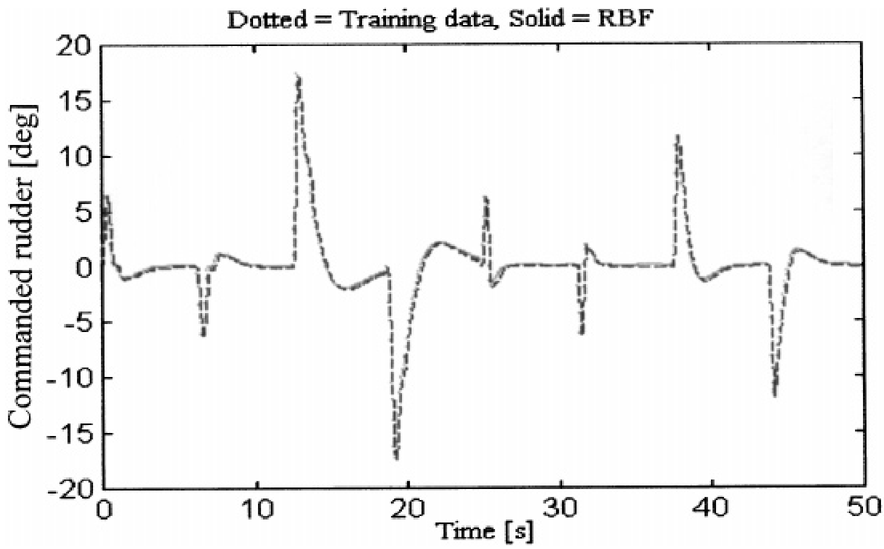

Figure 4. Matching of data with the trained network

(iv) $\int(\Psi - \Psi_d)$; and (v) the speed vector U. It was assumed that the rudder angle should not exceed $\pm 35^0$ and the maximum rudder rate should not be more than $\pm 7^\circ \mathrm{s}^{-1}$. The matching of data is shown in Figure 4 which shows that the RBF controller mimics the dynamics of the feedback linearization controllers very efficiently.

To develop the LMN, we used the structure of Figure 3 with two local models. These models were the two Norrbin models derived at 5 and 10 $\mathrm{m\,s}^{-1}$. The normalized Gaussian functions with centres at the above speeds were chosen as the activation functions. Extensive simulation results revealed that the LMN could provide the same behaviour as is achieved by the conventional non-linear ship model. Some results are illustrated in Figure 5, where the conventional model is the Norrbin model of equation (10).

Figure 6(a) shows responses for the system with the RBF controller based on data sets obtained at 5 and 10 $\mathrm{m\,s}^{-1}$ and operating now at a forward speed of 5 $\mathrm{m\,s}^{-1}$. It also shows the corresponding responses for a PID controller optimized for a forward speed of 7 $\mathrm{m/s}^{-1}$ but operating now in simulated conditions corresponding to 5 $\mathrm{m\,s}^{-1}$. There is a significant difference between these responses. The responses of the system with the same PID controller operating at a ship speed of 7 $\mathrm{m\,s}^{-1}$ coincide with the responses for the case with the RBF controller (Figure 6(b)).

It should be noted that for the purpose of this comparison, dimensionalized parameters have been used in the ship model.

Example 5.2

In this example we consider a 210 000 dwt tanker of length 310 m. The main parameters of the tanker are:[53] $K = -0{\cdot}0105\ 1\,\mathrm{s}^{-1}$, $T_1 = 1058$ s, $T_2 = 37{\cdot}8$ s and $T_3 = 84{\cdot}68$ s, $U = 4{\cdot}1\ \mathrm{m\,s}^{-1}$. An RBF controller was developed by generating the training data from two PID controllers that

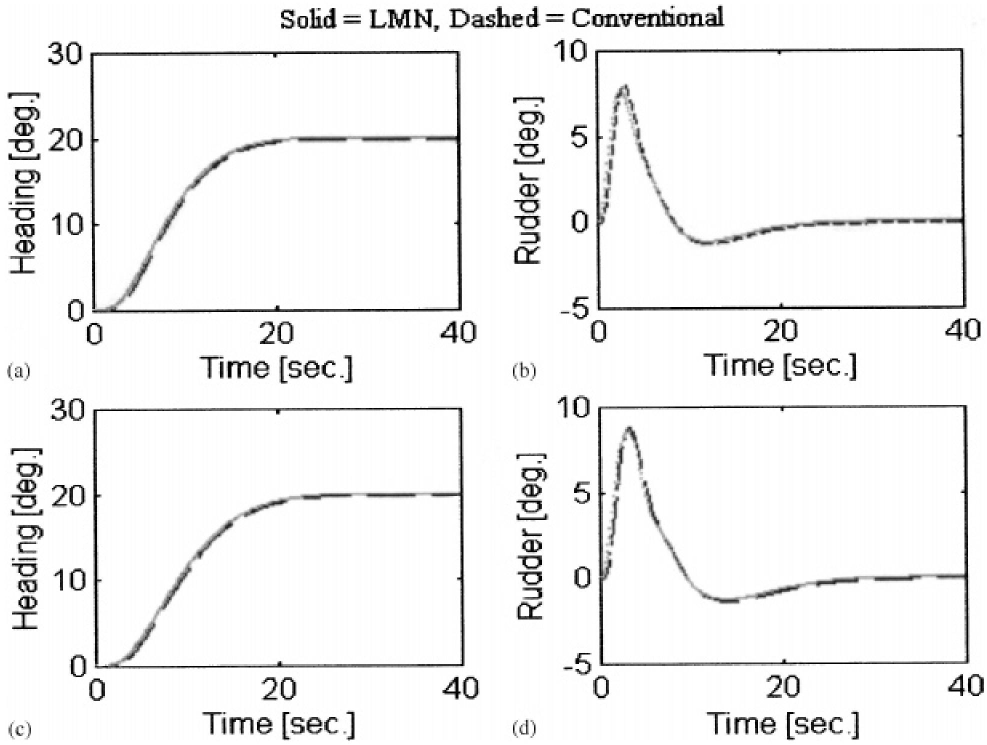

Figure 5. (a) Heading response at 10 m s^{-1} (b) the corresponding rudder response (c) Heading response at 7 m s^{-1} (d) the corresponding rudder response

were separately designed for speeds of 4·1 and 8·2 m s^{-1}. An LMN was also developed to represent the ship dynamics. In this case, two Nomoto first-order models (equation (7)) were considered as local models f_1 and f_2 (see Figure 3) at 4·1 and 8·2 m s^{-1}. Some results are presented in Figures 7 and 8. In these figures, the conventional model is the Nomoto model of equation (7). The success of the RBF controller as well as the LMN is quite obvious from these figures.

6. DISCUSSION AND CONCLUSIONS

Ship steering control systems and associated autopilots provide a number of design challenges. Most systems currently in use involve PID-based controllers which have well-known limitations because of the wide range of dynamical behaviour which can be exhibited by the vessel. Other approaches based on adaptive control techniques can give important benefits in terms of performance, but can also suffer from disadvantages such as potential instabilities.

This paper presents preliminary results concerning the use of artificial neural networks to obtain a controller which incorporates the properties of a series of conventional controllers. The form of network used is the radial basis function network which has been used successfully in

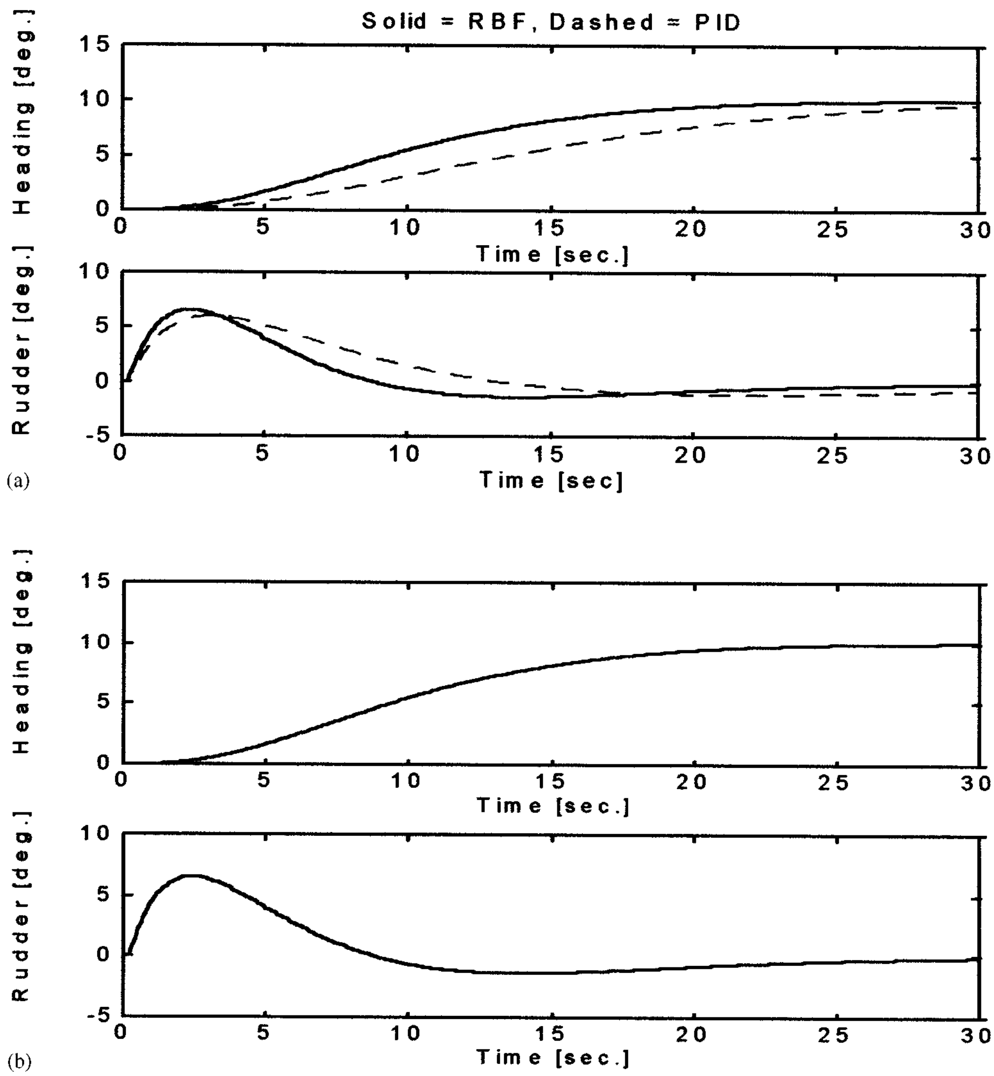

Figure 6. (a) Comparison of RBF and PID controller at a speed of 5 m s^{-1} (b) Performance of RBF and PID controller at a speed of 7 m s^{-1}

other control system applications and has favourable characteristics in terms of the best approximation property.

The paper also describes an investigation of local model networks for characterization of the dynamical properties of a ship for a range of forward speeds. Local model networks trained from simulation data have been used successfully to represent ship dynamics for a range of operating conditions and these representations have been incorporated into computer simulation models used for optimization of the radial basis function controllers.

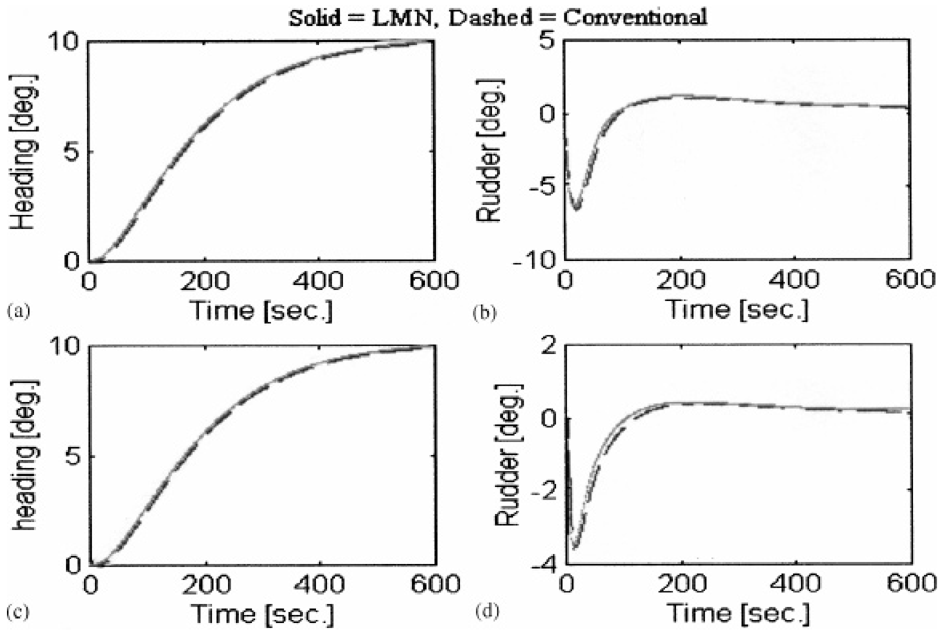

Figure 7. (a) Heading response at $5\,\mathrm{m\,s^{-1}}$ (b) the corresponding rudder response (c) Heading response at $10\,\mathrm{m\,s^{-1}}$ (d) the corresponding rudder response

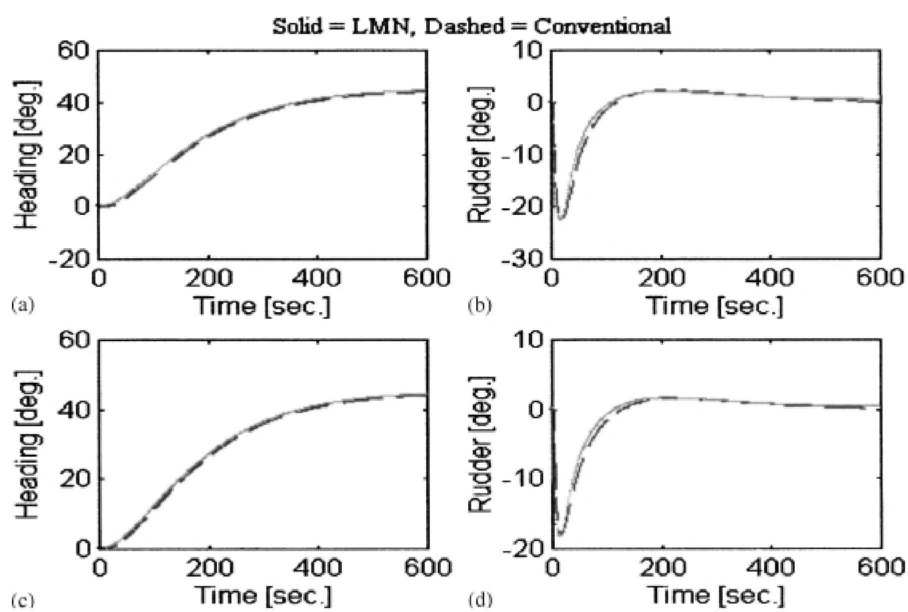

Figure 8. (a) Heading response at $7\,\mathrm{m\,s^{-1}}$ (b) the corresponding rudder response (c) Heading response at $9\,\mathrm{m\,s^{-1}}$ (d) the corresponding rudder response

In an implementation on a real ship the local model network could be trained using experimental data from ship trials with a conventional ship steering controller. The radial basis function controller intended to cover the complete operating envelope of the ship could also be trained from input–output data from a series of trials for different operating conditions, using conventional controllers optimally tuned for each case. The resulting artificial neural network controller should then incorporate the properties of all the conventional controllers from which it has been derived. Simulation studies have shown that the approach can provide performance improvements compared with a single conventional controller operating over a range of conditions.

Further research is currently underway to investigate the use of this approach for variation of ship dynamics with depth of water as well as forward speed and to compare the performance of controllers based on neural networks with gain scheduled systems.

REFERENCES

1. Minorsky, N., 'Directional stability of automatically steered bodies', *J. Naval Eng.*, **XXXIV**, 280–309 (1922).
2. Sperry, E., 'Automatic steering', *Trans. SNAME*, 53–57 (1922).
3. Hondered, G. and J. E. W. Winkelman, 'An adaptive autopilot for ships', *3rd Ship Control Symp., Ministry of Defence, Foxhill, Bath, Somerset, UK*, 1972, pp. 1–11.
4. Oldenburgh, J., 'Experiments with a new adaptive autopilot intended for controlled turns as well as for course keeping', *Proc. 4th Ship Control Symp., The Hague, Netherland*, 1975.
5. Ohtsu, K. and G. Kitagama, 'An advanced ship autopilot system by a stochastic model', *Proc. 5th Ship Control System Symp., Annapolis, MD, U.S.A.*, 1978.
6. Sugimoto, A. and T. Kojima, 'A new autopilot system with condition adaptivity', *Proc. 5th Control System Symp., Annapolis, MD, U.S.A.*, 1978.
7. Källström, C. G., K. J. Åström, N. E. Thorell, J. Eriksson and L. Sten, 'Adaptive autopilots for tankers', *Automatica*, **15**, 241–254 (1979).
8. Åström, K. J., 'Why use adaptive techniques for steering large tankers', *Int. J. Control*, **32**(4), 689–708 (1980).
9. Van Amerongen, J., 'Adaptive steering of ships – a model reference approach', *Automatica*, **20**, 3–14 (1984).
10. Arie, T., M. Itoh, A. Senoh, N. Takahashi, S. Fujii and N. Mizuno, 'An adaptive steering system for a ship', *IEEE Control Systems Mag.*, 3–7 (1986).
11. Katebi, M. R. and J. C. Byrne, 'LQG adaptive ship autopilot', *Trans. Inst. MC*, **10**(4), 187–197 (1988).
12. Fossen, T. I. and M. J. Paulsen, 'Adaptive feedback linearization applied to steering of ships', *Modelling, Identification Control*, **14**(4), (1993).
13. Garcia, R. F. and F. J. Castelo, 'Adaptive PID controller applied to marine DP control using frequency analysis', *Proc. 3rd IFAC Workshop on Control Applications in Marine Systems, Trondheim, Norway*, 1995, pp. 356–361.
14. Pao, Y. H., *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, MA, 1989.
15. Narendra, K. S. and K. Parthasarathy, 'Identification and control of dynamical systems using neural networks', *IEEE Control Systems Mag.* 11–18 (1992).
16. Tulunay, E., 'Introduction to neural networks and their application to process control', in E. Gelenbe (ed.), *Neural Networks: Advances and Applications*, Elsevier, Amsterdam, 1991, pp. 241–273.
17. Simensen, R. and D. J. Murray-Smith, 'Ship steering control by neural networks trained using feedback linearization control laws', *Preprints IFAC/IMACS Int. Workshop on Artificial Intelligence in Real Time Control*, Bled. Slovenia, 1995, pp. 269–274.
18. Witt, N. A. J., R. Sutton and K. M. Miller, 'A track keeping neural network controller for ship guidance', *Proc. 3rd IFAC Workshop on Control Applications in Marine Systems, Trondheim, Norway*, 1995, pp. 385–392.
19. Richter, R. and R. S. Burns, 'An artificial neural network autopilot for ship guidance', *Conf. United Kingdom Simulation Soc. (UKSS'93), Edinburgh*, 1993, pp. 168–172.
20. Balasuriya, B. A. A. P. and P. R. P. Hoole, 'Feedforward neural network controller for ship steering', *Proc. 3rd IFAC Workshop on Control Applications in Marine Systems, Trondheim, Norway*, 1995, pp. 400–404.
21. Burns, R. S., 'The use of artificial neural networks for the intelligent optimal control of surface ships', *IEEE J. Oceanic Engng*, **20**(1), 65–72 (1995).
22. Unar, M. A. and D. J. Murray-Smith, "Artificial neural networks for ship steering control systems', *Proc. 3rd Int. Conf. on Engineering Applications of Neural Networks (EANN'97), Stockholm, Sweden*, 1997, pp. 87–90.
23. Zhang, Y., P. Sen and G. E. Hearn, 'An on-line trained adaptive neural controller', *IEEE Control Systems Mag.*, 67–75 (1995).

24. Murray-Smith, R. and T. A. Johansen (ed.), *Multiple Model Approach to Modelling and Control*, Taylor & Francis, London, 1997.
25. Hunt, K. J. and T. A. Johansen, 'Design and analysis of gain scheduled control using local controller networks', *Int. J. Control*, **66**(5), 619–651 (1997).
26. Fossen, T. I., *Guidance and Control of Ocean Vehicles*, Wiley, New York, 1994.
27. Van Amerongen, J., 'Adaptive steering of ships – a model reference approach to improved manoeuvring and economical course keeping', *Ph.D. Thesis*, Delft University of Technology, 1982.
28. Crane, C. L., H. Eda and A. C. Landsburg, 'Controllability', in Lewis, E.V. (ed.), *Principles of Naval Architects*, Vol. III, SNAME, Jersey City, NJ, 1989.
29. Abkowitz, M. A., 'Lectures on ship hydrodynamics – steering and manoeuvrability'. *Report No. Hy-5, Hydro- and Aerodynamics Laboratory*, Lyngby, Denmark, 1964.
30. SNAME. The society of Naval Architects and Marine Engineers. Nomenclature for treating the motion of a submerged body through a fluid, *Tech. Res. Bull.*, No. 1–5 (1955).
31. Nomoto, K., T. Taguchi, K. Honda and S. Hirano, 'On steering qualities of ships', *Int. Ship Building Progr.*, **4**(35), 354–370 (1957).
32. Koyama, T., 'Improvement of course stability and control of a subsidiary automatic control', *J. Mech. Engng. Sci.* **14**(7) (Supplementary Issue), 132–141 (1972).
33. Lauvdal, T. and T. I. Fossen, 'A globally stable adaptive ship autopilot with wave filter using only yaw angle measurements', *Proc. 3rd IFAC Workshop on Control Applications in Marine Systems (CAMS'95), Trondheim, Norway*, 1995, pp. 262–269.
34. Holzhüter, T. and R. Schultze, 'Operating experience with a high precision track controller for commercial ships', *Control Eng. Practice*, **4**(3), 343–350 (1996).
35. Norrbin, N. H., 'On the design and analysis of the zig–zag test on base of quasi linear frequency response', *Technical Report B104-3, The Swedish State Shipbuilding Experimental Tank (SSPA), Gothenburg, Sweden*, 1963.
36. Hardy, R. L., 'Multiquadric equations of topography and other irregular surfaces', *J. Geophys. Res.*, **76**(8), 1905–1915 (1971).
37. Powell, M. J. D., 'Radial basis functions for multivariable interpolation: a review', *Proc. IMA Conf. Algorithms for Approximation of Functions and Data, Shrivenham, UK*, 1985.
38. Broomhead, D. S. and D. Lowe, 'Multivariable function interpolation and adaptive networks', *Complex Systems*, **2**, 321–355 (1988).
39. Poggio, T. and F. Girossi, 'Networks for approximation and learning', *Proc. IEEE*, **78**(9), 1481–1497 (1990).
40. Chen, S., C. F. N. Cowan and P. M. Grant, 'Orthogonal least squares algorithm for radial basis function networks', *IEEE Trans. Neural Networks*, **2**(2), 302–309 (1991).
41. Johansen, T. A. and B. A. Foss, 'A NARMAX model representation for adaptive control based on local models', *Modelling, Identification Control*, **13**(1), 25–39 (1992).
42. Johansen, T. A. and B. A. Foss, 'Constructing NARMAX models using ARMAX models, *Int. J. Control*, **58**, 1125–1153 (1993).
43. Murray-Smith, R., 'A fractal radial basis function network for modelling', *Int. Conf. on Automation, Robotics and Computer Vision*, Singapore, Vol. 1, 1992, 2.6.1–2.6.5.
44. Murray-Smith, R., Neumerkel, D. and Sbarbaro-Hofer, D., 'Neural networks for modelling and control of a nonlinear dynamic system', *IEEE Symp. on Intelligent Control*, Glasgow, 1992, pp. 404–409.
45. Murray-Smith, R., '*A local model approach to non-linear modelling*', *Ph.D. Thesis*, University of Strathclyde, Glasgow, Scotland, 1994.
46. Friedmann, J. H., 'Multivariable adaptive regression Splines', *Ann. Statist.* **19**, 1–141 (1991).
47. Kaveli, T., 'ASMOD – an algorithm for adaptive spline modelling of observation data', *Int. J. Control* **58**, 947–967 (1993).
48. Hlavàckovà, K, 'An upper estimate of the error of approximation of continuous multivariable functions by KBF networks', *Proc. 3rd European Symp. on Artificial Neural Networks*, Brussels, 1995, pp. 333–340.
49. Werntges, H. W., 'Partition of unity improves neural functional approximation', *Proc. IEEE Int. Conf. on Neural Networks*, Vol. 2, San Francisco, CA, 1993, pp. 914–918.
50. Johansen, T. A. and B. A. Foss, 'Identification of non-linear system structure and parameters using regime decomposition', *Automatica*, **31**(2), 321–326 (1995).
51. Gawthrop, P. J., 'Continuous time local model networks', in Zbikowski, R. and K. J. Hunt (eds), *Neural Adaptive Control Technology*, World Scientific, Singapore, 1996, pp. 41–70.
52. Gollee, H., K. J. Hunt, N. de N. Donaldson and C. J. Jarvis, 'Modelling of electrically simulated muscle', in Murray-Smith, R. and T. A. Johansen (eds), *Multiple Modelling Approaches to Modelling and Control*, Chapter 3, Taylor & Francis, London, 1997.
53. Åström, K. J. and C. G. Källström, 'Identification of ship dynamics', *Automatica*, **12**, 9–22 (1976).

