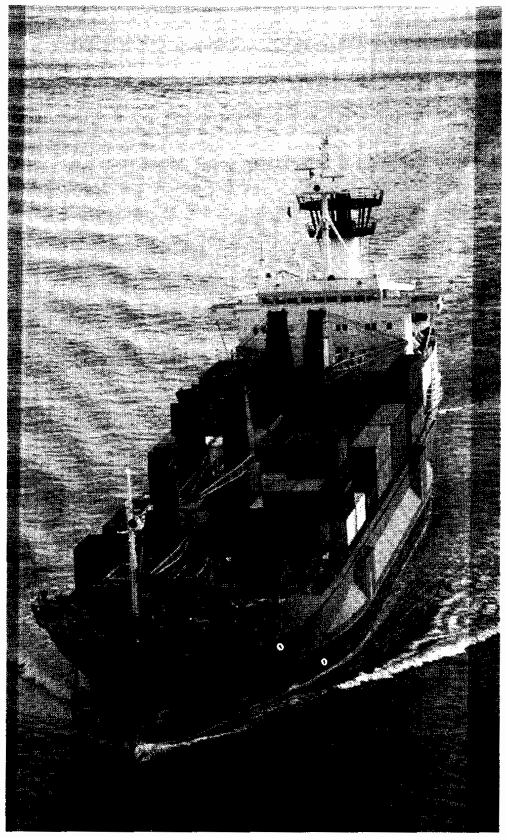
# Fuzzy Model Reference Learning Control for Cargo Ship Steering

Jeffery R. Layne and Kevin M. Passino

While fuzzy control has emerged as an alternative to some conventional control schemes since it has shown success in some application areas (e.g., in train control and camera auto-focusing), there are several drawbacks to this approach: a) the design of fuzzy controllers is usually performed in an *ad hoc* manner where it is hard to justify the choice of some controller parameters (e.g., the membership functions), and b) the fuzzy controller constructed for the nominal plant may later perform inadequately if significant and unpredictable plant parameter variations occur. While the "fuzzy model reference learning controller" (FMRLC) introduced in [1], [2] and other adaptive fuzzy control approaches seek to address these issues, they primarily focus on improving existing learning control approaches or introducing new ones. In this article we provide a comparative analysis of the FMRLC and conventional "model reference adaptive control" (MRAC) for a cargo ship steering application. Our main objective is to make an initial assessment of what advantages (if any) a fuzzy learning control approach has over conventional adaptive control approaches. For the cargo ship steering application, our simulation results show that the FMRLC has several *potential* advantages over MRAC including a) improved convergence rates, b) use of less control energy, c) enhanced disturbance rejection properties, and d) lack of dependence on a mathematical model. Using our comparative analysis we discuss how the well developed concepts in conventional adaptive control can be used to evaluate fuzzy learning control techniques.



## Learning Control Systems

In recent years, to improve fuel efficiency and reduce wear on ship components, autopilot systems have been developed and implemented for controlling the directional heading of ships. Generally, the autopilots utilize simple control schemes such as PID control. Often, however, the capability for manual adjustments of the parameters of the controller is added to compensate for disturbances acting upon the ship such as wind and currents. Once suitable controller parameters are found manually, the controller will generally work well for small variations in the operating conditions; for large variations the parameters of the autopilot must be continually modified. Such continual adjustments are necessary because the dynamics of a ship vary with speed, trim, and loading. Also, it is useful to change the autopilot control law parameters when the ship is exposed to large disturbances which result from changes in the wind, waves, current, and water depth. Manual adjustment of the controller parameters is often a burden on the crew. Moreover, poor adjustment may result from human error. As a result, it is of great interest to have a method for automatically adjusting or modifying the underlying controller.

In this article, we investigate the use of a "learning control system" to maintain adequate performance of a cargo ship autopilot when there are process disturbances or variations as men-

*J. Layne was with the Department of Electrical Engineering, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210. He is now with Wright Laboratories, WL/AAAS-3, Wright Patterson AFB, OH 45433-6543. K. Passino is with the Department of Electrical Engineering, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210. Email: passino@ee.eng.ohio-state.edu.* *The work of J. Layne was supported by the U.S. Air Force Palace Knight Program. The work of K. Passino was partially supported by an Engineering Foundation Research Initiation Grant and by National Science Foundation Grant IRI-9210332.*


0272-1708/93/$03.00©1993IEEE

tioned above. In general, a "learning system" possesses the capability to improve its performance over time by interaction with its environment. A "learning control system" is designed so that its "learning controller" has the ability to improve the performance of the closed loop system by generating command inputs to the plant and utilizing feedback information from the plant. The learning control algorithm considered here is based on a direct fuzzy controller. In general, a "fuzzy controller" utilizes a fuzzy system to capture a human expert's knowledge about how to control a process for use in a computer algorithm. Often, the human expert's knowledge must be known *a priori* for fuzzy controller design. However, the learning control algorithm presented here automatically generates the fuzzy controller's knowledge base on-line as new information on how to control the ship is gathered. For instance, the FMRLC can automatically synthesize a fuzzy controller for the cargo ship and later tune it if there are significant disturbances/process variations.

The FMRLC algorithm employs a reference model (a model of how you would like the system to behave) to provide closed loop performance feedback for generating and modifying a fuzzy controller's knowledge base. Consequently, this algorithm is referred to as a "fuzzy model reference learning controller" (FMRLC). The FMRLC algorithm, which was first introduced in [1]-[3], grew from research performed on the linguistic self-organizing controller (SOC) presented in [4] by Procyk and Mamdani and ideas in conventional "model reference adaptive control" (MRAC) [5], [6]. Since the basic architecture and functionality of the FMRLC is consistent with the prevailing definition of a "learning controller," the term "learning" is used rather than "adaptive" (for a more detailed discussion on this issue see [3], [7]). A significant amount of research has been done on the SOC. For instance, the linguistic SOC has been used in robotics applications [8], [9], motor and temperature control [10], blood pressure control [11], and in satellite control [12]-[14]. Other relevant literature that focuses on adaptation of a direct fuzzy controller includes the work in [15] where an adaptive fuzzy system is developed for a continuous casting plant, and the approach in [16] where a fuzzy system adapts itself to driver characteristics for an automotive speed control device. The use of fuzzy systems for estimation/identification [17]-[23] is relevant, especially if indirect adaptive [5], [6] fuzzy control techniques such as those in [24]-[26] are used. Since the introduction of the FMRLC in [3] some other relevant new adaptive/learning techniques have been developed [27]-[30]. While we describe here the application of the FMRLC to cargo ship steering, it has also been used for a robotics problem, a rocket velocity control problem, and a cart-pendulum system [1], [2] and it has been been used to improve the performance of anti-skid brakes in adverse road conditions [31], [32].

We provide a comparative analysis of FMRLC and several types of MRAC for a ship steering application. In particular, for the ship steering application described in [5], [6] we develop a FMRLC and a gradient-based and Lyapunov-based MRAC, provide some simulation studies, and compare their performance. We find that for our simulations, the FMRLC learns to control the ship faster, does so with less control energy, and accomodates for a wind disturbance better than the two types of MRAC considered. Moreover, the development of the FMRLC, although somewhat *ad hoc*, does not depend on the form of the mathematical model of the cargo ship as the MRAC approaches do.

Next, we interpret the results of our comparative analysis of FMRLC and MRAC. First, we emphasize that one must be careful not to over-generalize the results of our simulation-based analysis for this application. While the results look somewhat promising, we have not performed a) a mathematical analysis of the stability and convergence properties of the FMRLC, b) an analysis of *persistency of excitation* (which relates to how well the knowledge base of the fuzzy controller is "filled in"), c) an analysis of the robustness properties of the FMRLC, or d) an analysis of the computational properties of the FMRLC. Such conventional approaches to the analysis of adaptive systems are rarely used for fuzzy learning control (one exception lies in [29]) but they do provide useful directions for future work in the study of fuzzy learning control systems.

In the next section we provide a detailed overview of the FMRLC algorithm. Following this, we describe the cargo ship steering problem, design an appropriate FMRLC, and develop several conventional MRAC designs. Next, we provide the results from several simulations to compare the performance of the FMRLC to the MRAC designs. Then we discuss the advantages and disadvantages of FMRLC from a control-theoretic perspective in order to lay a foundation for future work in modeling and analysis of fuzzy learning control systems. Finally, some concluding remarks are given.

## Fuzzy Model Reference Learning Control

The FMRLC, which is shown in Fig. 1, utilizes a learning mechanism that a) observes data from a fuzzy control system (i.e., $\underline{y}_r(kT)$ and $\underline{y}(kT)$), b) characterizes its current performance, and c) automatically synthesizes and/or adjusts the fuzzy controller so that some prespecified performance objectives are met. These performance objectives are characterized via the *reference model* shown in Fig. 1. In an analogous manner to conventional MRAC where conventional controllers are adjusted, the learning mechanism seeks to adjust the fuzzy controller so that the closed-loop system (the map from $\underline{y}_r(kT)$ to $\underline{y}(kT)$) acts like a prespecified reference model (the map from $\underline{y}_r(kT)$ to $\underline{y}_m(kT)$). Next we describe each component of the FMRLC in more detail.

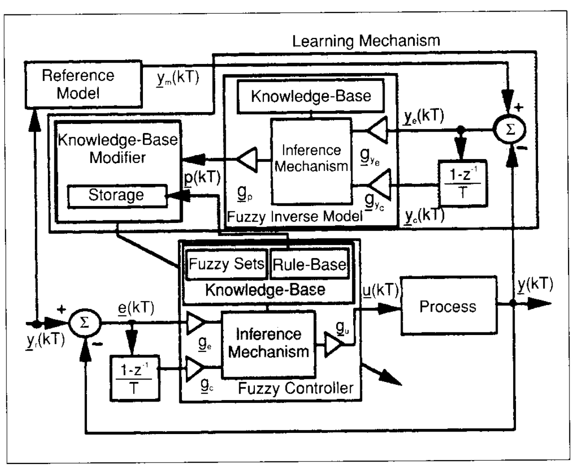



*Fig. 1. Architecture for the FMRLC.*

## Fuzzy Controller

The process in Fig. 1 is assumed to have $r$ inputs denoted by the $r$ - dimensional vector $\underline{u}(kT) = [u_1(kT) \ldots u_r(kT)]^t$ ($T$ is the sample period) and $s$ outputs denoted by the $s$ - dimensional vector $\underline{y}(kT) = [y_1(kT) \ldots y_s(kT)]^t$. Most often the inputs to the fuzzy controller are generated via some function of the plant output $\underline{y}(kT)$ and reference input $\underline{y}_r(kT)$. Fig. 1 shows a special case of such a map that was found useful in many applications. The inputs to the fuzzy controller are the error $\underline{e}(kT) = [e_1(kT) \ldots e_s(kT)]^t$ and change in error $\underline{c}(kT) = [c_1(kT) \ldots c_s(kT)]^t$ defined as

$$\underline{e}(kT) = \underline{y}_r(kT) - \underline{y}(kT)$$

$$\underline{c}(kT) = \frac{e(kT) - e(kT - T)}{T}$$

respectively, where

$$\underline{y}_r(kT) = [y_{r_1}(kT) \ldots y_{r_s}(kT)]^t$$

denotes the desired process output.

In fuzzy control theory, the range of values for a given controller input or output is often called the "universe of discourse" [33]. Often, for greater flexibility in fuzzy controller implementation, the universes of discourse for each process input are "normalized" to the interval [-1, +1] by means of constant scaling factors. For our fuzzy controller design, the gains $g_e$, $g_c$, and $g_u$ were employed to normalize the universe of discourse for the error $\underline{e}(kT)$, change in error $\underline{c}(kT)$, and controller output $\underline{u}(kT)$, respectively (e.g., $g_e = [g_{e_1}, \ldots, g_{e_s}]^t$ so that $g_{e_i}e_i(kT)$ is an input to the fuzzy controller). The gains $g_e$ are chosen so that the range of values of $g_{e_i}e_i(kT)$ lie on [-1, 1] and $g_u$ is chosen by using the allowed range of inputs to the plant in a similar way. The gains $g_c$ are determined by experimenting with various inputs to the system to determine the normal range of values that $c(kT)$ will take on; then $g_c$ is chosen so that this range of values is scaled to [-1, 1].

We utilize $r$ multiple-input single-ouput (MISO) fuzzy controllers, one for each process input $u_n$ (equivalent to using one MIMO controller). The knowledge base for the fuzzy controller associated with the $n$th process input is generated from IF-THEN control rules of the form:

**If** $\tilde{e}_1$ is $\tilde{E}_1^j$ **and ... and** $\tilde{e}_s$ is $\tilde{E}_s^k$ **and** $\tilde{c}_1$ is $\tilde{C}_1^l$ **and ... and** $\tilde{c}_s$ **is** $\tilde{C}_s^m$ **Then** $\tilde{u}_n$ is $\tilde{U}_n^{j,\ldots,k,l,\ldots,m}$

where $\tilde{e}_a$ and $\tilde{c}_a$ denote the *linguistic variables* associated with controller inputs $e_a$ and $c_a$, respectively, $\tilde{u}_n$ denotes the linguistic variable associated with the controller output $u_n$, $\tilde{E}_a^b$ and $\tilde{C}_a^b$ denote the $b$th *linguistic value* associate associated with $\tilde{e}_a$ and $\tilde{c}_a$, respectively, and $\tilde{U}_n^{j,\ldots,k,l,\ldots,m}$ denotes the *consequent linguistic value* associated with $\tilde{u}_n$. Hence, as an example, one fuzzy control rule could be

**If** *error* is *positive-large* **and** *change-in-error* is *negative-small* **Then** *plant-input is positive-big*

(in this case $\tilde{e}_1$ = *"error,"* $\tilde{E}_1^4$ = *"positive-large"*, etc.). A set of such rules forms the "rule-base" which characterizes how to control a dynamical system.

The above control rule may be quantified by utilizing fuzzy set theory to obtain a fuzzy implication of the form:

**If** $E_1^j$ **and ... and** $E_s^k$ **and** $E_1^l$ **and ... and** $C_s^m$ **Then** $U_n^{j,\ldots,k,l,\ldots,m}$,

where $E_a^b$, $C_a^b$ and $U_n^{j,\ldots,k,l,\ldots,m}$ denote the fuzzy sets that quantify the *linguistic statements* "$\tilde{e}_a$ is $\tilde{E}_a^b$," "$\tilde{c}_s$ is $\tilde{C}_s^m$," and "$\tilde{u}_n$ is $\tilde{U}_n^{j,\ldots,k,l,\ldots,m}$," respectively. For the example above, we may use fuzzy sets on the $e_i(t)$ normalized universes of discourse as shown in Fig. 2. Assume that we use the same fuzzy sets on the $c_i(t)$ normalized universes of discourse. The membership functions on the output universe of discourse are assumed to be unknown; they are what the FMRLC will automatically synthesize. In fact, we will initialize the fuzzy controller knowledge base with 121 rules (for our examples we utilize the fuzzy sets shown in Fig. 2 and use all possible combinations of rules) where all the right-hand-side membership functions are triangular with base widths of 0.4 and centers at zero (to model that the fuzzy controller initially knows nothing about how to control the plant; of course one can often make a reasonable best guess at how to specify a fuzzy controller that is "more knowledgable"). For example, if $s = 1$ then all rules in our controller will take on the form

**if** $E_1^j$ **and** $C_1^l$ **Then** $U_n^{j,l}$

where the membership functions for $E_1^j$ and $C_1^l$ are shown in Fig. 2 and $U_n^{j,l}$ is a fuzzy set with triangular membership function with base width 0.4 centered at zero. In conventional direct fuzzy controller development the designer specifies a set of such control rules where $U_n^{j,l}$ are also specified *a priori*; for the FMRLC, the system will automatically specify and/or modify the fuzzy sets $U_n^{j,l}$ to improve/maintain performance. Finally, we note that we use Zadeh's compositional rule of inference and the standard center-of-gravity (COG) defuzzification technique [33].

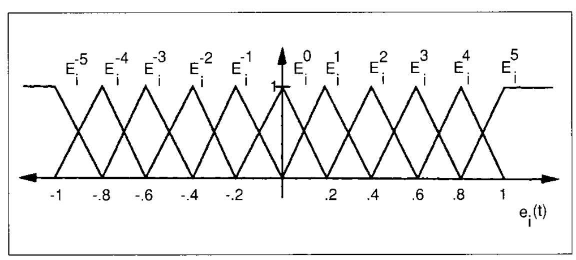



*Fig. 2. Fuzzy sets on a universe of discourse.*

## Reference Model

The reference model provides a capability for quantifying the desired performance. In general, the reference model may be any type of dynamical system (linear or nonlinear, time-invariant or time-varying, discrete or continuous time, etc.). The performance of the overall system is computed with respect to the reference model by generating an error signal

$$\underline{y}_e(kT) = [y_{e_1} \ \dots \ y_{e_s}]^t$$

where

$$\underline{y}_e(kT) = \underline{y}_m(kT) - \underline{y}(kT).$$

Given that the reference model characterizes design criteria such as rise time and overshoot and the input to the reference model is the reference input $\underline{y}_r(kT)$, the desired performance of the controlled process is met if the learning mechanism forces $\underline{y}_e(kT)$ to remain very small for all time; hence, the error $\underline{y}_e(kT)$ provides a characterization of the extent to which the desired performance is met at time $t = kT$. If the performance is met ($\underline{y}_e(kT) \approx 0$) then the learning mechanism will not make significant modifications to the fuzzy controller. On the other hand if $\underline{y}_e(kT)$ is big, the desired performance is not achieved and the learning mechanism must adjust the fuzzy controller.

### Learning Mechanism

As previously mentioned, the learning mechanism performs the function of modifying the knowledge base of a direct fuzzy controller so that the closed loop system behaves like the reference model. These knowledge base modifications are made by observing data from the controlled process, the reference model, and the fuzzy controller. The learning mechanism consists of two parts: a fuzzy inverse model and a knowledge base modifier. The fuzzy inverse model performs the function of mapping $\underline{y}_e(kT)$ (representing the deviation from the desired behavior), to changes in the process inputs $\underline{p} = [p_1 \ \dots \ p_r]^t$ that are necessary to force $\underline{y}_e(kT)$ to zero. The knowledge base modifier performs the function of modifying the fuzzy controller's knowledge base to affect the needed changes in the process inputs. More details of this process are discussed next.

Using the fact that most often a control engineer will know how to roughly characterize the inverse model of the plant, the authors in [1] introduce the idea of using a fuzzy system to map $\underline{y}_e(kT)$ and possibly functions of $\underline{y}_e(kT)$ (or process operating conditions), to the necessary changes in the process inputs $\underline{p}(kT)$. This map is called the *fuzzy inverse model* since information about the plant inverse dynamics is used in its specification. Note that similar to the fuzzy controller, the fuzzy inverse model shown in Fig. 1 contains normalizing scaling factors, namely $g_{y_e}$, $g_{y_c}$, and $g_p$, for each universe of discourse. Given that $g_{y_e} y_{e_i}$ and $g_{y_{c_i}} y_{c_i}$ are inputs to the fuzzy inverse model, the knowledge base for the fuzzy inverse model associated with the $n$th process input is generated from fuzzy implications of the form:

**If** $Y_{e_1}^k$ and ... and $Y_{e_s}^k$ and $Y_{c_1}^l$ and ... and $Y_{c_s}^m$ **Then** $P_n^{j,\dots,k,l,\dots,m}$

where $\boldsymbol{Y}_{\boldsymbol{e}_a}^{\boldsymbol{b}}$ and $Y_{c_a}^b$ denote the $b$th fuzzy set for the error $y_{e_a}$ and change in error $y_{c_a}$, respectively, associated with the $a$th process output and $P_n^{j,\dots,k,l,\dots,m}$ denotes the consequent fuzzy set for this rule describing the necessary change in the $n$th process input. As with the fuzzy controller, we utilize membership functions for the normalized input universes of discourse as shown in Fig. 2, triangular membership functions for the output universes of discourse, Zadeh's compositional rule of inference, and COG defuzzification. Given the information about the necessary changes in the input as expressed by the vector $\underline{p}(kT)$, the *knowledge base modifier* changes the knowledge base of the fuzzy controller so that the previously applied control action will be modified by the amount $\underline{p}(kT)$. Therefore, consider the previously computed control action $\underline{u}(kT - T)$, which contributed to the present good/bad system performance. Note that $\underline{e}(kT - T)$ and $\underline{c}(kT - T)$ would have been the process error and change in error, respectively, at that time. By modifying the fuzzy controller's knowledge base we may force the fuzzy controller to produce a desired output $\underline{u}(kT - T) + \underline{p}(kT)$. Assume that only symmetric membership functions are defined for the fuzzy controller's output so that $c_n^{j,\dots,k,l,\dots,m}$ denotes the center value of the membership function associated with the fuzzy set $U_n^{j,\dots,k,l,\dots,m}$ (initially, $c_n^{j,\dots,k,l,\dots,m}(0) = 0$). Knowledge base modification is performed by shifting centers of the membership functions of the fuzzy sets $U_n^{j,\dots,k,l,\dots,m}$ which are associated with the fuzzy implications that contributed to the previous control action $\underline{u}(kT - T)$ (initially possibly shifting them away from having centers at zero). This modification involves shifting these membership functions by an amount specified by $\underline{p}(kT) = [p_1(kT) \ \dots \ p_r(kT)]^t$ so that

$$c_n^{j,\dots,k,l,\dots,m}(kT) = c_n^{j,\dots,k,l,\dots,m}(kT - T) + p_n(kT). \qquad (1)$$

The degree of contribution for a particular fuzzy implication whose fuzzy relation is denoted $R_n^{j,\dots,k,l,\dots,m}$ is determined by its "activation level", defined

$$\delta_n^{j,\dots,k,l,\dots,m}(t) = \min\{\mu_{E_1^j}(e_1(t)),\dots,\mu_{E_s^k}(e_s(t))$$

$$\mu_{E_1^l}(c_1(t)),\dots,\mu_{C_s^m}(c_s(t))\} \qquad (2)$$

where $\mu_A$ denotes the membership function of the fuzzy set $A$. Only those rules whose activation level $\delta_n^{j,\dots,k,l,\dots,m}(kT - T) > 0$ are modified; all others remain unchanged. It is important to note that our rule-base modification procedure implements a form of *local* learning and hence utilizes memory. In other words, different parts of the rule-base are "filled in" based on different operating conditions for the system, and when one area of the rule-base is updated, other rules are not affected. Hence, the controller adapts to new situations and also remembers how it has adapted to past situations. This justifies the use of the term "learning" rather than "adaptive" (for more details on this point see [1], [2], [7]).

For example, assume that all the normalizing gains for both the direct fuzzy controller and the fuzzy inverse model are unity and that the fuzzy inverse model produces an output $p_n(kT) = 0.5$ indicating that the value of the output to the plant at time $kT - T$ should have been $u(kT - T) + 0.5$ to improve performance (i.e., to force $y_{e_1} \approx 0$). Next, suppose that $e_1(kT - T) = 0.75$ and $c_1(kT - T) = -0.2$. Then rules

**If** $E_1^3$ **and** $C_1^{-1}$ **Then** $U_n^{3,-1}$, and

**If** $E_1^4$ **and** $C_1^{-1}$ **Then** $U_n^{4,-1}$

are the only rules with activation levels greater than zero ($\delta_n^{3,-1}$=0.25 and $\delta_n^{4,-1}$=0.75) so these rules will be the only ones that have their consequent fuzzy sets ($U_n^{3,-1}$, $U_n^{4,-1}$) modified (See Fig. 2). To modify these fuzzy sets we simply shift their centers according to (1).

### Design Procedure

Selection of the normalizing gains can impact the overall performance so we provide a gain selection procedure in the following. Note that although it is often not highlighted, most learning/adaptive control approaches assume that you are given an initial controller structure and parameters (e.g., initial controller gains must be chosen in adaptive control approaches). In what follows we provide a procedure to pick such initial parameters for the FMRLC.

1. Select the controller gains $g_{y_e}$ associated with the desired output change $\underline{y}_e(kT)$ such that each universe of discourse is mapped to the interval [-1, 1].

2. Choose the controller gain $g_p$ to be the same as for the fuzzy controller output gain $g_u$. This will allow the elements of $\underline{p}(kT)$ to take on values as large as the largest possible inputs.

3. Assign the numerical value 0 to the scaling factors associated with the changes in the desired output changes (i.e., all elements of $g_{y_c}$ are set equal to 0).

4. Apply a step input to the process which is of a magnitude that may be typical for the process during normal operation. Observe the process response and the reference model response.

5. Three cases:

a) If there exist unacceptable oscillations in a given process output response about the reference model response, then increase the associated element of $g_{y_c}$. Go to step 4.

b) If a given process output response is unable to "keep up" with the reference model response, then decrease the associated element of $g_{y_c}$. Go to step 4.

c) If the process response is acceptable with respect to the reference model response, then the controller design is completed.

For the application presented in this paper, the above gain selection procedure has proven very successful. However, given that the procedure is a result of practical experience with the FMRLC rather than strict mathematical analysis, it is possible that it will not work for all processes. For some applications (although none of the ones studied in [1], [3], [32]), the procedure may result in an unstable process. In such situations, it may be necessary to modify other controller parameters such as the controller sampling period $T$ or the number of fuzzy controller rules. Clearly, the stability analysis of the FMRLC is an important research direction.

## Learning and Adaptive Autopilots for a Cargo Ship

### Problem Statement

Generally, ship dynamics are obtained by applying Newton's laws of motion to the ship. For very large ships, the motion in the vertical plane may be neglected since the "bobbing" or "bouncing" effects of the ship are small for large vessels. The motion of the ship is generally described by a coordinate system which is fixed to the ship [5], [6]. See Fig. 3.

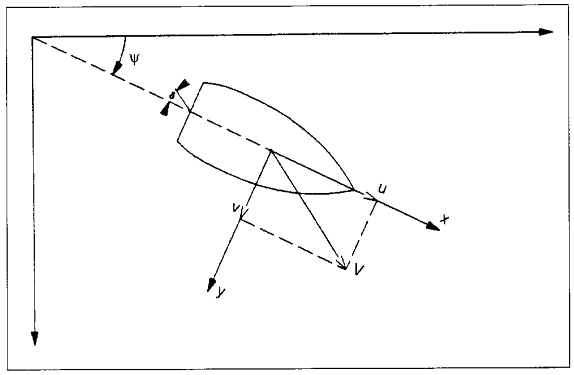



*Fig. 3. Cargo ship.*

A simple model which describes the dynamical behavior of the ship may be expressed by the following differential equation:

$$\dddot{\psi}(t) + \left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\ddot{\psi}(t) + \left(\frac{1}{\tau_1\ \tau_2}\right)\dot{\psi}(t) = \frac{K}{\tau_1\ \tau_2}(\tau_3\ \dot{\delta}(t) + \delta(t)) \tag{3}$$

where $\psi$ is the heading of the ship and $\delta$ is the rudder angle. Assuming zero initial conditions, (3) can be written

$$\frac{\phi(s)}{\delta s} = \frac{K(s\tau_3 + 1)}{s(s\tau_1 + 1)(s\tau_2 + 1)} \tag{4}$$

where $K$, $\tau_1$, $\tau_2$, and $\tau_3$ are parameters which are a function of the ship's constant forward velocity $u$ and its length $l$ as expressed below:

$$K = K_0\left(\frac{u}{l}\right) \tag{5}$$

$$\tau_i = \tau_{i0}\left(\frac{l}{u}\right),\quad i = 1, 2, 3. \tag{6}$$

where we assume that for a cargo ship $K_0$ = -3.86, $\tau_{10}$ = 5.66, $\tau_{20}$ = 0.38, $\tau_{30}$ = 0.89, and $l$ = 161 (m) [6]. Also, we will assume that the ship is traveling in the $x$ direction at a velocity of 5 m/s.

In normal steering, a ship often makes only small deviations from a straight line path. Therefore, the model in (3) is obtained by linearizing the equations of motion around the zero rudder angle ($\delta$ = 0). As a result, the rudder angle should not exceed approximately 5°, otherwise the model will be inaccurate. For our purposes, we need a model which is suited for rudder angles which are larger than 5°; hence, we use the model proposed by Bech and Smitt in [34]. This extended model is given by

$$\dddot{\psi}(t) + \left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\ddot{\psi}(t) + \left(\frac{1}{\tau_1\ \tau_2}\right)H(\dot{\psi}(t)) = \frac{K}{\tau_1\ \tau_2}(\tau_3\ \dot{\delta}(t) + \delta(t)) \tag{7}$$

where $H(\dot{\psi})$ is a nonlinear function of $\dot{\psi}(t)$. The function $H(\dot{\psi})$ can be found from the relationship between $\delta$ and $\dot{\psi}$ in steady state such that $\dddot{\psi} = \ddot{\psi} = \dot{\delta} = 0$. An experiment known as the "spiral test" has shown that $H(\dot{\psi})$ can be approximated by

$$H(\dot{\psi}) \approx a\,\dot{\psi}^3 + b\,\dot{\psi} \tag{8}$$

where $a$ and $b$ are real valued constants such that $a$ is always positive. For our simulations we choose the values of both $a$ and $b$ to be 1.

### FMRLC Design

In this section we present a FMRLC algorithm for controlling the directional heading of a cargo ship. The inputs to the fuzzy controller are the heading error and change in heading error expressed as

$$e(kT) = \psi_r(kT) - \psi(kT)$$

and

$$c(kT) = \frac{e(kT) - e(kT - T)}{T}$$

respectively, where $\psi_r(kT)$ is the desired ship heading ($T = 50$ ms). The controller output is the rudder angle, $\delta(kT)$, of the ship. Here we assume that the dynamics of the actuator which is used to position the rudder are much faster than the ship dynamics so that they may be neglected. In this fuzzy controller design, 11 fuzzy sets are defined for each controller input such that the membership functions are triangular shaped and evenly distributed on the appropriate universes of discourse (as shown in Fig. 2). The normalizing controller gains for the error, change error, and the controller output are chosen to be $g_e = 1/\pi$, $g_c = 100$, and $g_u = 8\pi/18$, respectively, according to the design procedure above. The fuzzy sets for the fuzzy controller output are also assumed to be triangular shaped with a width of 0.4, and centered at zero on the normalized universe of discourse. The reference model was chosen so as to represent somewhat realistic performance requirements as

$$\ddot{\psi}(t) + 0.1\ \dot{\psi}_m(t) + 0.0025\ \psi_m(t) = 0.0025\ \psi_r(t) \tag{9}$$

where $\psi_m(t)$ specifies the desired system performance for the ship heading $\psi(t)$.

The input to the fuzzy inverse model includes the error and change in error between the reference model and the ship heading expressed as

$$\psi_e(kT) = \psi_m(kT) - \psi(kT)$$

and

$$\psi_c(kT) = \frac{\psi_e(kT) - \psi_e(kT - T)}{T}$$

respectively. For these inputs, 11 fuzzy sets are defined with triangular shaped membership functions which are evenly distributed on the appropriate universes of discourse (as shown in Fig. 2). The normalizing controller gains associated with $\psi_e(kT)$, $\psi_c(kT)$, and $p(kT)$ are chosen to be

$$g_{\psi_e} = \frac{1}{n},\ g_{\psi_c} = 5, \text{ and } g_p = \frac{8\pi}{18}$$

respectively, according to the design procedure above. For a cargo ship, an increase in the rudder angle $\delta(kT)$, will generally result in a decrease in the ship heading angle. The knowledge base array shown in Table I was employed for the fuzzy inverse model for the cargo ship (for more details on how to choose the fuzzy inverse model for other plants see [1], [2], [32]). In Table I, $\Psi_e^j$ denotes the $j$th fuzzy set associated with the error signal $\psi_e$ and $\Psi_c^k$ denotes the $k$th fuzzy set associated with the change in error signal $\psi_c$. The entries of the table represent the center values of triangular membership functions with base widths 0.4 for fuzzy sets $P_i^{j,k}$ on the normalized universe of discourse.

It is important to note that in designing the FMRLC, the only ways in which the design procedure relied on the plant model was in the choice of the normalizing gains and in the specification of the fuzzy inverse model. Although the design procedure for the FMRLC is somewhat *ad hoc* it does have the advantage that it neither relies on the explicit mathematical model of the process nor on the form of such a model.

### Model Reference Adaptive Control

In this section we present two MRAC designs which utilize an underlying proportional derivative (PD) control law for the direct controller. The PD control law is used to obtain a fair comparison with FMRLC algorithm where error and change in error are employed as controller inputs. We will consider both the gradient method and the Lyapunov stability method for MRAC design.

*Gradient Approach.* The controller parameter adjustment mechanism for the gradient approach to MRAC can be implemented via the *MIT rule*. For this, the cost function

$$J(\underline{\theta}) = \frac{1}{2}\psi_e^2(t)$$

where

$$\psi_e(t) = \psi_m(t) - \psi(t)$$

is used and

$$\frac{d\theta}{dt} = -\gamma\frac{\partial J}{\partial \underline{\theta}}$$

so that

$$\frac{d\theta}{dt} = -\gamma\,\psi_e(t)\frac{\partial \psi_e(t)}{\partial \underline{\theta}}$$

**Table I**
**Knowledge Base Array Table Employed**
**in the Fuzzy Inverse Model for a Cargo Ship**

| $P_l^{j,k}$ | | $\Psi_c^k$ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | -5 | -4 | -3 | -2 | -1 | +0 | +1 | +2 | +3 | +4 | +5 |
| $\Psi_e^j$ | -5 | +1.0 | +1.0 | +1.0 | +1.0 | +1.0 | +1.0 | +0.8 | +0.6 | +0.4 | +0.2 | 0.0 |
| | -4 | +1.0 | +1.0 | +1.0 | +1.0 | +1.0 | +0.8 | +0.6 | +0.4 | +0.2 | 0.0 | -0.2 |
| | -3 | +1.0 | +1.0 | +1.0 | +1.0 | +0.8 | +0.6 | +0.4 | +0.2 | 0.0 | -0.2 | -0.4 |
| | -2 | +1.0 | +1.0 | +1.0 | +0.8 | +0.6 | +0.4 | +0.2 | 0.0 | -0.2 | -0.4 | -0.6 |
| | -1 | +1.0 | +1.0 | +0.8 | +0.6 | +0.4 | +0.2 | 0.0 | -0.2 | -0.4 | -0.6 | -0.8 |
| | 0 | +1.0 | +0.8 | +0.6 | +0.4 | +0.2 | 0.0 | -0.2 | -0.4 | -0.6 | -0.8 | -1.0 |
| | +1 | +0.8 | +0.6 | +0.4 | +0.2 | 0.0 | -0.2 | -0.4 | -0.6 | -0.8 | -1.0 | -1.0 |
| | +2 | +0.6 | +0.4 | +0.2 | 0.0 | -0.2 | -0.4 | -0.6 | -0.8 | -1.0 | -1.0 | -1.0 |
| | +3 | +0.4 | +0.2 | 0.0 | -0.2 | -0.4 | -0.6 | -0.8 | -1.0 | -1.0 | -1.0 | -1.0 |
| | +4 | +0.2 | 0.0 | -0.2 | -0.4 | -0.6 | -0.8 | -1.0 | -1.0 | -1.0 | -1.0 | -1.0 |
| | +5 | 0.0 | -0.2 | -0.4 | -0.6 | -0.8 | -1.0 | -1.0 | -1.0 | -1.0 | -1.0 | -1.0 |

which is commonly referred to as the M.I.T. rule.

For developing the M.I.T. rule for the ship we assume that the ship may be modeled by a second order linear differential equation. This model is obtained by eliminating the process pole resulting from $\tau_2$ in (3) since its associated dynamics are significantly faster than those resulting from $\tau_1$. Also, for small heading variations the rudder angle derivative $\dot{\delta}$ is likely to be small and may be neglected. Therefore we obtain the following reduced order model for the ship

$$\ddot{\psi}(t) + \left(\frac{1}{\tau_1}\right)\dot{\psi}(t) = \left(\frac{K}{\tau_1}\right)\delta(t) \tag{10}$$

The PD-type control law which will be employed for this process may be expressed by

$$\delta(t) = k_p\,(\psi_r(t) - \psi(t)) - k_d\,\dot{\psi}(t) \tag{11}$$

$k_p$ and $k_d$ are the proportional and derivative gains, respectively, and $\psi_r(t)$ is the desired process output. Substituting (10) into (11) we obtain

$$\ddot{\psi}(t) + \left(\frac{1 + K\,k_d}{\tau_1}\right)\dot{\psi}(t) + \left(\frac{K\,k_p}{\tau_1}\right)\psi(t) = \left(\frac{K\,k_p}{\tau_t}\right)\psi_r(t). \tag{12}$$

It follows from (12) that

$$\psi(t) = \frac{\dfrac{K\,k_p}{\tau_1}}{p^2 + \left(\dfrac{1 + K\;k_d}{\tau_1}\right)p + \left(\dfrac{K\,k_p}{\tau_1}\right)}\,\psi_r(t) \tag{13}$$

where $p$ is the differential operator. The reference model for this process is chosen to be

$$\psi_m(t) = \frac{\omega_n}{p^2 + \zeta\omega_n p + \omega_n^2}\,\psi_r(t). \tag{14}$$

where to be consistent with the FMRLC design we choose $\zeta = 1$ and $\omega_n = 0.05$. Combining (14) and (13) and finding the partial derivatives with respect to the proportional gain $k_p$ and the derivative gain $k_d$ we find that

$$\frac{\partial\psi_e}{\partial k_p} = \left(\frac{\dfrac{K}{\tau_1}}{p^2 + \left(\dfrac{1 + K\,k_d}{\tau_1}\right)p + \left(\dfrac{K\,k_p}{\tau_1}\right)}\right)(\psi - \psi_r) \tag{15}$$

and

$$\frac{\partial\psi_e}{\partial k_d} = \left(\frac{\dfrac{K}{\tau_1}p}{p^2 + \left(\dfrac{1 + K\,k_d}{\tau_1}\right)p + \left(\dfrac{K\,k_p}{\tau_1}\right)}\right)\psi. \tag{16}$$

In general, (15) and (16) cannot be used because the controller parameters $k_p$ and $k_d$ are not known. Observe that for the "optimal values" of $k_p$ and $k_d$ we have

$$p^2 + \left(\frac{1 + K\,k_d}{\tau_1}\right)p + \left(\frac{K\,k_p}{\tau_1}\right) = p^2 + 2\zeta\omega_n\,p + \omega_n^2. \tag{17}$$

Furthermore, the term $K/\tau_1$ may be absorbed into the adaptation gain $\gamma$. However, this requires that the sign of $K/\tau_1$ be known since, in general, $\gamma$ should be positive to ensure that the controller updates are made in the direction of the negative gradient. For a forward moving cargo ship the sign of $K/\tau_1$ happens to be negative which implies that the term $\gamma$ with $K/\tau_1$ absorbed into it must be negative to achieve the appropriate negative gradient. After making the above approximations we obtain the following differential equations for updating the PD controller gains:

$$\frac{d\,k_p}{dt} = -\gamma_1\left(\frac{1}{p^2 + 2\zeta\omega_n p + \omega_n^2}(\psi - \psi_r)\right)\psi_e \tag{18}$$

$$\frac{d\,k_d}{dt} = -\gamma_2\left(\frac{1}{p^2 + 2\zeta\omega_n p + \omega_n^2}\psi\right)\psi_e \tag{19}$$

where $\gamma_1$ and $\gamma_2$ are negative real numbers. After many simulations, the best values that we could find for the $\gamma_i$ are $\gamma_1 = -0.005$ and $\gamma_2 = -0.1$.

*Lyapunov Approach*. Examples of Lyapunov based MRAC designs are illustrated by Narendra and Annaswamy in [5] and by Amerongen and Cate in [35]; here we utilize the approach in [5] to design our Lyapunov-based MRAC. Recall from the Problem Statement section that the ship dynamics may be approximated by a second order linear time-invariant differential equation given by (10). Once again, we use the PD control law defined in (11). The dynamical equation which describes the compensated system is

$$\dot{\underline{\psi}} = A_c\underline{\psi} + B_c\psi_r$$

where $\underline{\psi} = [\psi\ \dot{\psi}]^T$ and

$$A_c = \begin{bmatrix} 0 & 1 \\ -\dfrac{K\,k_p}{\tau_1} & -\dfrac{(1 + K\,k_d)}{\tau_1} \end{bmatrix} \tag{20}$$

$$B_c = \begin{bmatrix} 0 \\ \dfrac{K\,k_p}{\tau_1} \end{bmatrix} \tag{21}$$

The reference model is given by

$$\dot{\underline{\psi}} = A_m\underline{\psi}_m + B_m\underline{\psi}_r \tag{22}$$

where $\underline{\psi}_m = [\psi_m\ \dot{\psi}_m]^T$ and

$$A_m = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \tag{23}$$

$$B_m = \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \tag{24}$$

and where to be consistent with the FMRLC design we choose $\zeta = 1$ and $\omega_n = 0.05$.

The dynamical equation which describes the error ($\psi_e(t) = \psi_m(t) - \psi(t)$) may be expressed by

$$\dot{\underline{\psi}}_e = A_m(t)\underline{\psi}_e + (A_m(t) - A_c(t))\underline{\psi} + (B_m(t) - B_c(t))\underline{\psi}_r \tag{25}$$

The equilibrium point $\psi_e = 0$ in (25) is asymptotically stable if we choose the adaptation laws to be

$$\dot{A}_c(t) = \gamma P\underline{\psi}_e\underline{\psi}^T \tag{26}$$

$$\dot{B}_c(t) = \gamma P\underline{\psi}_e\underline{\psi}_r^T \tag{27}$$

where $P \in \Re^{n\times n}$ is a symmetric positive definite matrix which is a solution of the Lyapunov equation $A_m^T P + PA_m = -Q < 0$. Assuming that $Q$ is a $2 \times 2$ identity matrix and solving for $P$ we find that

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} 25.0015 & 200.000 \\ 200.000 & 2005.00 \end{bmatrix} \tag{28}$$

Solving for $\dot{k}_p$ and $\dot{k}_d$ in (26) and (27), respectively, the adaptation law in (26) and (27) may be implemented as

$$\dot{k}_p = -\gamma_1(p_{21}\psi_e + p_{22}\dot{\psi}_e)(\psi - \psi_r) \tag{29}$$

$$\dot{k}_d = -\gamma_2(p_{21}\psi_3 + p_{22}\dot{\psi}_e)\dot{\psi}. \tag{30}$$

Of course, (29) and (30) assume that the plant parameters and disturbance are varying slowly. In obtaining (29) and (30) the term $K/\tau_1$ was absorbed into the adaptation gains $\gamma_1$ and $\gamma_2$. Recall that for the cargo ship $K/\tau_1$ happens to be a negative quantity. Therefore, both $\gamma_1$ and $\gamma_2$ must be negative to compensate for this fact. Again we found that $\gamma_1 = -0.005$ and $\gamma_2 = -0.1$ were suitable. See Narendra and Annaswamy [5] for more details about obtaining (29) and (30).

## Comparative Analysis of FMRLC and MRAC: Simulation Results

For the simulations for all three adaptive control methods presented above (FMRLC, gradient MRAC, and Lyapunov MRAC) we use the nonlinear process model given in (7) to emulate using the "true" ship dynamics. Fig. 4 shows the results for the FMRLC controller. Recall that initially the right-hand-sides of the control rules have membership functions with centers all at zero (i.e., initially, the controller knows little about how to control the plant). Note in Fig. 4 that the FMRLC algorithm was quite successful in generating the appropriate control rules for a good process response since the reference model and the ship heading track almost perfectly. In fact the maximum deviation between the two signals was observed to be less than 1°. As a result, the system exhibits a fast transient response with no overshoot. Also note that the rate of convergence for the FMRLC

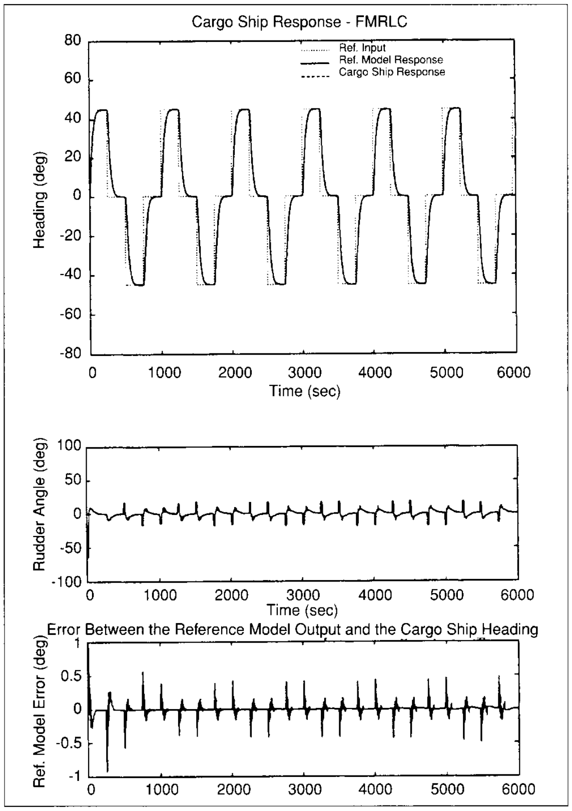



*Fig. 4. Simulation results for the FMRLC algorithm when employed for a cargo ship.*

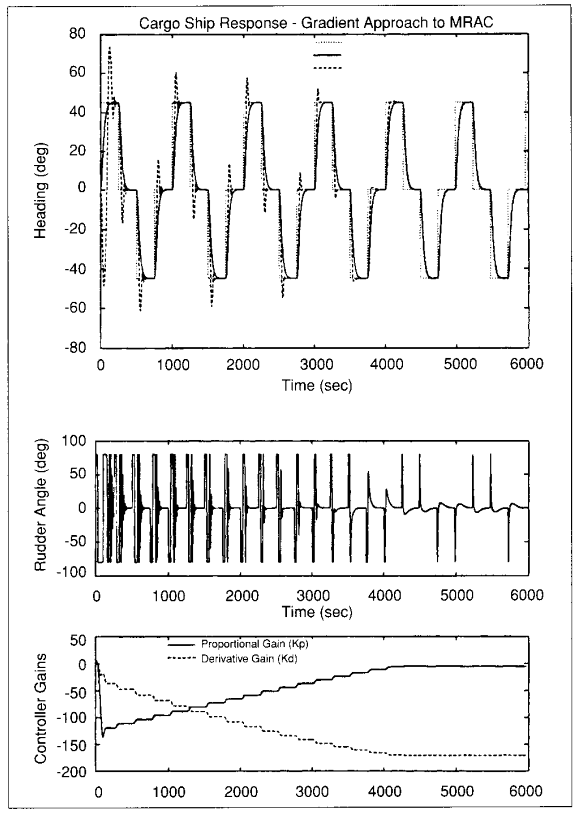



*Fig. 5. Simulation results for the gradient approach to MRAC when employed for a cargo ship.*

algorithm was very fast. In the last plot of Fig. 4 two large spikes of a magnitude of about 1° occur after the first two step input changes. However, as time progresses the spikes resulting from subsequent step input changes are reduced to less than 0.5° as a result of learning.

Compare the results for the FMRLC with those obtained for the gradient and the Lyapunov approach to MRAC shown in Figs. 5 and 6, respectively. The controller gains $k_p$ and $k_d$ for both MRAC algorithms where initially chosen to be 5. This choice of initial controller gains happens to be an unstable case for the linear second order process model (in the case where the adaptation mechanism is disconnected). However, we felt this to be a fair comparison since the fuzzy controller is initially chosen so that it would put in zero degrees of rudder no matter what the controller input values were. We would have chosen both controller gains to be 0; but, this choice resulted in a very slow convergence rate for the MRAC. It is easy to show that the compensated second order process model (described by $A_c$ and $B_c$) is equal to the reference model in (22) if the controller gains are chosen to be $k_p$ = -3.8 and $k_d$ = -143.7. In Table II, we summarize the final values of the process gains shown in Figs. 5 and 6. Although the process gains for both algorithms converged to values relatively close to the optimal values, they do not match exactly. This reason for this may be explained by a few simple facts:

- Both MRAC algorithms are designed to minimize the error signal $\varphi_e$. This does not necessarily mean that the process parameters will also converge.
- The reduced second order linear process model, for which the controller designs and the "optimal" process gains are based, is not a completely accurate characterization of the third order nonlinear process model used in simulation.

For both the gradient approach and the Lyapunov approach, the system response converged to track the reference model. However, the convergence rate of both algorithms was significantly slower than the FMRLC method.

Another significant advantage of the FMRLC algorithm may be seen in the amount of input energy which was spent at the system input to obtain accurate tracking with the reference model. Due to the fact that for our simulations the magnitude of the rudder angle is generally larger for both MRAC approaches than for the FMRLC algorithm, we may suspect that the input energy for the FMRLC is significantly less. The rudder angle plots shown in Figs. 4, 5, and 6 for each of the adaptive control algorithms represent 1200 sampled data points. We may obtain a measure of the input energy if we think of these data points as a vector $\delta = [(0)\ \delta(T)\ \delta(2T) \ldots \delta(1199T)]^T$, where the energy is the square of the 2-norm for this vector (i.e., $energy = \delta^T\delta$). Upon performing this for the data shown in Figs. 4, 5, and 6 in radians rather than degrees, we obtain the result shown in Table III. As

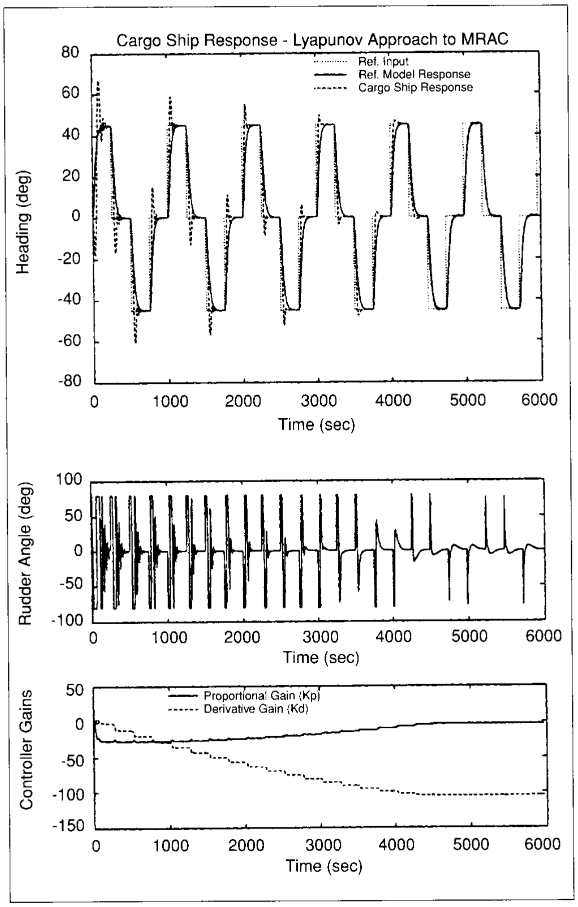



*Fig. 6. Simulation results for the Lyapunov approach to MRAC when employed for a cargo ship.*

expected, the process input energy for the FMRLC was significantly less than that obtained for both MRAC approaches.

The final set of experiments performed for this process were designed to illustrate the ability of the learning and adaptive controllers to compensate for disturbances at the process input. Fig. 7 illustrates the results obtained for this simulation. The disturbance added at the rudder was chosen to be be a sinusoid with a frequency of one cycle per minute and a magnitude of 2° with a bias of 1°. The effect of this disturbance is similar to that of a gusting wind acting upon the ship since wind effects are generally modeled as a rudder disturbance. To provide a fair

**Table II**
**Final Values of Controller Gains $k_p$ and $k_d$ in the Simulation Results for the Gradient and Lyapunov Approach to MRAC**

| | MRAC Approach | |
|---|---|---|
| Controller Parameter | Gradient | Lyapunov |
| $k_p(6000)$ | -4.7752 | -3.0974 |
| $k_d(6000)$ | -171.8580 | -105.0242 |

**Table III**
**Comparison of the Process Input Energy for the FMRLC, the Gradient MRAC, and the Lyapunov MRAC when Employed as an Autopilot for a Cargo Ship**

| Process Input Energy ( $\|\delta\|^2$ ) | | |
|---|---|---|
| FMRLC | Gradient MRAC | Lyapunov MRAC |
| 17.3368 | 458.6324 | 425.7801 |

comparison with the FMRLC algorithm, we initially loaded the PD controllers in both MRAC algorithms with the controller gains shown in Table II, which were previously found by each method. However, the centers of the right-hand-sides of the membership functions for the knowledge base of the fuzzy controller in the FMRLC algorithm were initialized with all zeros as before (hence, we are giving the MRAC an advantage). Notice that the FMRLC algorithm was nearly able to completely cancel the effects of the disturbance input. However, the gradient and the Lyapunov approaches to MRAC where not nearly so successful.

### Control Engineering Perspective

We now summarize and more carefully analyze the conclusions from our simulation studies. Our goal is to provide an objective control-theoretic assessment that will identify research directions focusing on a careful engineering evaluation of the FMRLC. The results in the previous section indicate the following advantages of the FMRLC:

- Fast convergence compared with MRAC.
- Minimal amount of control energy needed as compared to MRAC.
- Good disturbance rejection properties compared to MRAC.
- The FMRLC design is independent of the particular form of the mathematical model of the underlying process (in the MRAC designs we need an explicit mathematical model of a particular form).

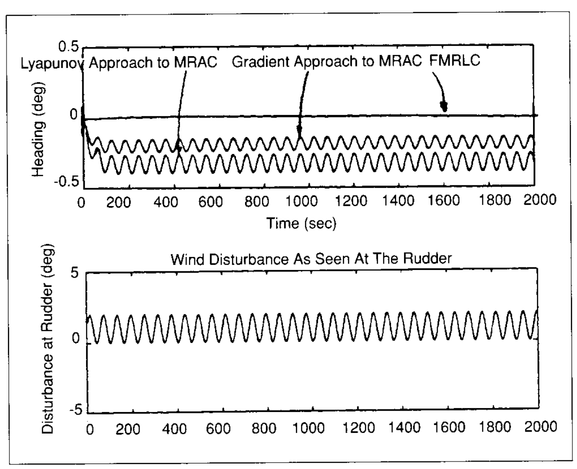



*Fig. 7. Simulation results compare disturbance rejection for the FMRLC, the gradient approach to MRAC, and the Lyapunov approach to MRAC.*

Overall, the FMRLC provides a method to synthesize (i.e., automatically design) and tune the knowledge base for a direct fuzzy controller. As the direct fuzzy controller is a nonlinear controller, some of the above advantages may be attributed to the fact that the underlying controller that is tuned inherently has more significant functional capabilities as compared to the PD controllers used in the MRAC designs.

While our application may indicate that FMRLC is a promising alternative to conventional MRAC, we must emphasize that:

• We have compared the FMRLC to only two types of MRACs, for only one application, for a limited class of reference inputs, and only in simulation. There are a wide variety of other adaptive control approaches that deserve consideration (e.g., those of the self-tuning regulator type [6], [5]).

• There are no guarantees of stability or convergence; hence, we can simply pick a different reference input and the system may be unstable (we have not found this in other simulations, but it is possible).

• There seem to be no investigations into persistency of excitation issues for the FMRLC or any other fuzzy learning control technique. Persistency of excitation is related to the learning controller's ability to always generate an appropriate plant input and its ability to *generalize* the results of what it has learned earlier and apply this to new situations. In this context, for the ship we ask the following questions:

– What if we need to turn the ship in a different direction — will the rule base be "filled in" for this direction?

– Or will it have to learn for each new direction?

– If it learns for the new directions, will it forget how to control for the old ones?

• In terms of control energy we may have just gotten lucky for this application and for the chosen reference input. There seem to be no analytical results that guarantee that the FMRLC or any other fuzzy learning control technique minimizes the use of control energy for a wide class of plants.

• This is a very limited investigation of the disturbance rejection properties (i.e., only one type of wind disturbance is considered). As of yet there seem to be no results on mathematical robustness analysis for a wide class of plants for the FMRLC or any other fuzzy learning control technique.

• The design approach for the FMRLC, although it did not depend on a mathematical model, it is somewhat *ad hoc*. Will there be fundamental limitations on the FMRLC imposed by nonminimum phase systems? Certainly there will be limitations for classes of nonlinear systems. What will these limitations be? It is important to note that the use of a mathematical model helps to show what these limitations will be (hence it cannot *always* be considered an advantage that many fuzzy control techniques do not depend on the specification of the mathematical model). Also note that due to our avoidance of using a mathematical model of the plant, we have also ignored the important "model matching problem" in adaptive control.

• There may be gains in performance, but are these gains being made by paying a high price in computational complexity for the FMRLC? The FMRLC is somewhat computationally intensive as are many neural and fuzzy learning control approaches but we have neither performed a careful study of the computational properties of the MRAC versus the FMRLC nor investigated techniques to simplify the FMRLC computations.

All things considered, the major conclusions one can draw from this work are: a) that the FMRLC looks promising and deserves further attention — especially analysis in a control-theoretic framework, and b) the conventional control-theoretic viewpoint is quite useful for the study of this class of intelligent controllers.

## Laying Foundations for Comparative Analysis

We have provided a detailed description of the FMRLC algorithm, and developed a FMRLC and two MRACs for a ship steering application. Then we conducted some simulation studies to evaluate the performance of the FMRLC as compared to a gradient-based and Lyapunov-based MRAC design. Moreover, we discuss the results from a control-theoretic perspective to provide an objective assessment of the FMRLC and to indicate future research directions.

We want to emphasize the importance of laying foundations for comparative analyses of conventional and "intelligent" control techniques. Many concepts and results from conventional control (e.g., stability and stability analysis) can provide for a more careful engineering evaluation of intelligent control techniques and provide for productive research directions. At the same time, the intelligent control techniques have much to offer conventional control by infusing new concepts, approaches to control, and new design methodologies. In this article we have made a small move in the direction of bridging the gap between fuzzy learning control and conventional adaptive control; it is hoped that this will be beneficial to both fields.

## References


[1] J. Layne and K. Passino, "Fuzzy model reference learning control," in *Proc. 1st IEEE Conf. Control Applications,* Dayton, OH, Sept. 1992, pp. 686–691.

[2] J. Layne and K. Passino, "Fuzzy model reference learning control," submitted for publication.

[3] J. Layne, "Fuzzy model reference learning control," Master's thesis, Dept. of Elec. Eng., The Ohio State Univ., Mar. 3 1992.

[4] T. Procyk and E. Mamdani, "A linguistic self-organizing process controller," *Automatica,* vol. 15, no. 1, pp. 15–30, 1979.

[5] K. Narendra and A. Annaswamy, *Stable Adaptive Systems.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[6] K. Åström and B. Wittenmark, *Adaptive Control.* Reading, MA: Addison-Wesley Pub. Co., 1989.

[7] J. Farrell and W. Baker, "Learning control systems," in *An Introduction to Intelligent and Autonomous Control Systems*, P. Antsaklis and K. Passino, Eds. Norwell, MA: Kluwer, 1993.

[8] E. Scharf and N. Mandic, "The application of a fuzzy controller to the control of a multi-degree-of-freedom robot arm," in *Industrial Applications of Fuzzy Control.* Amsterdam, The Netherlands: M. Sugeno, Ed., 1985, pp. 41–62.

[9] R. Tanscheit and E. Scharf, "Experiments with the use of a rule-based self-organising controller for robotics applications," *Fuzzy Sets Syst.,* vol. 26, pp. 195–214, 1988.

[10] S. Shao, "Fuzzy self-organizing controller and its application for dynamic processes," *Fuzzy Sets Syst.,* vol. 26, pp. 151–164, 1988.

[11] S. Isaka, A. Sebald, A. Karimi, N. Smith, and M. Quinn, "On the design and performance evaluation of adaptive fuzzy controllers," in *Proc. 1988 IEEE Conf. Decision and Control,* Austin, TX, Dec. 1988, pp. 1068–1069.

[12] S. Daley and K.F. Gill, "Comparison of a fuzzy logic controller with a P+D control law," *J. Dynamical Syst., Meas., Control*, vol. 111, pp. 128–137, June 1989.

[13] S. Daley and K.F. Gill, "Altitude control of a spacecraft using an extended self-organizing fuzzy logic controller," *Proc. I. Mech. E.*, vol. 201, no. 2, pp. 97–106, 1987.

[14] S. Daley and K.F. Gill, "A design study of a self-organizing fuzzy logic controller," *Proc. I. Mech. E.*, vol. 200, pp. 59–69, 1986.

[15] G. Bartolini, G. Casalino, F. Davoli, R.M.M. Mastretta, and E. Morten, "Development of performance adaptive fuzzy controllers with applications to continuous casting plants," *Industrial Application of Fuzzy Control*, pp. 73–86, 1985.

[16] H. Takahashi, "Automatic speed control device using self-tuning fuzzy logic," in *Proc. 1988 IEEE Workshop on Automotive Applications of Electronics*, Dearborn, MI, Oct. 1988, pp. 65–71.

[17] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116–132, 1985.

[18] L. Wang and J. Mendel, "Generating fuzzy rules by learning from examples," in *Proc. 1991 IEEE Int. Symp. Intell. Control*, Arlington, VA, Aug. 1991, pp. 263–268.

[19] R.M. Tong, "Some properties of fuzzy feedback systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 10, pp. 327–330, June 1980.

[20] A. Cumani, "On a possibilistic approach to the analysis of fuzzy feedback systems," *IEEE Trans. Syst., Man, Cybern.*, pp. 417–422, May/June 1982.

[21] E. Czogala and W. Pedrycz, "On identification in fuzzy systems and its applications in control problems," *Fuzzy Sets Syst.*, vol. 6, pp. 73–83, 1981.

[22] E. Czogala and W. Pedrycz, "Control problems in fuzzy systems," *Fuzzy Sets Syst.*, vol. 7, pp. 257–273, 1982.

[23] C. Batur, A. Srinivasan, and C. Chan, "Automatic rule based model generation for uncertain complex dynamical systems," in *Proc. 1991 IEEE Int. Symp. Intelligent Control*, pp. 275–279, 1991.

[24] F.V.D. Rhee, H.V.N. Lemke, and J. Dijkman, "Knowledge based fuzzy control of systems," *IEEE Trans. Auto. Control*, vol. 35, pp. 148–155, Feb. 1990.

[25] P. Graham and R. Newell, "Fuzzy adaptive control of a first-order process," *Fuzzy Sets Syst.*, vol. 31, pp. 47–65, 1989.

[26] P. Graham and R. Newell, "Fuzzy identification and control of a liquid level rig," *Fuzzy Sets Syst.*, vol. 8, pp. 255–273, 1988.

[27] *Proc. IEEE Int. Conf. Fuzzy Systems*, San Diego, CA, 1992.

[28] R. Langari and H. Berenji, "Fuzzy logic in control engineering," in *Handbook of Intelligent Control*, D. White and D. Sofge, Eds. New York, NY: Van Nostrand Reinhold, 1992, pp. 93–140.

[29] L. Wang, "Stable adaptive fuzzy control of nonlinear systems," in *Proc. IEEE Conf. Decision and Control*, Dec. 1992, Tuscon, AZ, pp. 2511-2516.

[30] H. Berenji, "Fuzzy and neural control," in *An Introduction to Intelligent and Autonomous Control Systems*, P. Antsaklis and K. Passino, Eds. Norwell, MA: Kluwer, 1993.

[31] J. Layne, K. Passino, and S. Yurkovich, "Fuzzy learning control for anti-skid braking systems," in *Proc. IEEE Conf. Decision and Control*, Tucson, AZ, Dec. 1992, pp. 2523–2528.

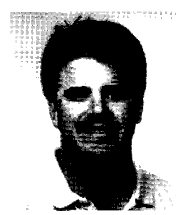
[32] J. Layne, K. Passino, and S. Yurkovich, "Fuzzy learning control for anti-skid braking systems," *IEEE Trans. Control Syst. Technol.*, vol. 1, pp. 122-129, June 1993.

[33] C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller-part I," *IEEE Trans. Syst., Man., Cybern.*, vol. 20, pp. 404–418, Mar./Apr. 1990.

[34] M. Bech and L. Smitt, "Analogue simulation of ship maneuvers," Tech. Rep., Hydro-og Aerodynmisk Laboratorium, Lyngby, Denmark, 1969.

[35] J. Amerongen and A. Cate, "Model reference adaptive autopilots for ships," *Automatica*, vol. 11, pp. 441–449, 1975.

**Jeffrey R. Layne** was born in Ashland, KY, in 1966. He received the B.S. degree from Wright State University, Dayton, OH, in 1990, and the M.S. degree from the Ohio State University in 1992, both in electrical engineering. He has worked at Armco Steel Corporation in Middletown, OH, and is currently with Wright Laboratory in Dayton, working on research in the area of integrated avionic systems. His research interests include fuzzy systems, intelligent systems, sensor integration, data fusion, stochastic systems, and Kalman filters.



**Kevin M. Passino** received the Ph.D. degree in electrical engineering from the University of Notre Dame in 1989. He has worked in the Control Systems Group at Magnavox Electronic Systems Co. in Ft. Wayne, IN, and at McDonnell Aircraft Co., St. Louis, MO, on research in intelligent flight control. He spent a year at Notre Dame as a Visiting Assistant Professor and is currently an Assistant Professor in the Department of Electrical Engineering at The Ohio State University. He serves as the Chair of Student Activities for the IEEE Control Systems Society and is on the Editorial Board of the *International Journal for Engineering Applications of Artificial Intelligence*. He is an Associate Editor for the *IEEE Transactions on Automatic Control*. He is co-editor (with P.J. Antsaklis) of the book ( *An Introduction to Intelligent and Autonomous Control* (Kluwer). His research interests include intelligent and autonomous control, fuzzy and expert systems and control, discrete event and hybrid systems, stability theory, and failure detection and identification systems.