Particle Swarm Optimization of Fuzzy Model Reference Learning Controller for Tanker Ship Steering

[1]C. K. Loo, [2]Nikos, E. Mastorakis
[1]Faculty of Engineering and Technology
Multimedia University
Melaka, Malaysia
[2]Department of Electrical Engineering and Computer Science
Hellenic Naval Academy
Piraues, GREECE

http://www1.mmu.edu.my/~ckloo
http://wseas.org/mastorakis/

Abstract: - This paper discussed the implementation of Particle Swarm Optimization (PSO) to optimize a Fuzzy Model Reference Learning Controller (FMRLC) for tanker ship. FMRLC is developed by synthesizing several basic ideas from fuzzy set and control theory. It can achieve the heading regulation of tanker ship exposed to plant changes and disturbances by adjusting the rules in a direct fuzzy controller so that the overall system behaves like a "reference model". However, the tuning of the fuzzy inverse model scaling gains is considered to be difficult and tedious due to the high nonlinearity of the ship dynamic model and the external disturbances. It is shown that PSO can provide a very promising technique for the design of FMRLC for its simplicity and ease of use. Moreover, Centroidal Voronoi Tessellation (CVT) is implemented to select the starting positions of the particles strategically. The promising results from the experiment provide direct evidence for the feasibility and effectiveness of PSO for the optimization of FMRL controller for tanker ship heading regulation.

Key-Words: - Particle Swarm optimization, autopilots, nonlinear optimization, tanker ship steering, fuzzy model reference learning controller.

1 Introduction

An autopilot is a ship's steering controller, which automatically manipulates the rudder to decrease the error between the reference heading angle and the actual heading angle. To improve fuel efficiency and reduce wear on ship components, autopilot systems have been developed and implemented for controlling the directional heading of ships. Often, the autopilots utilize simple control schemes such as PID control. However, manual adjustments of the PID parameters are required to compensate for disturbances acting upon the ship such as wind and currents. Once the PID parameters are fine-tuned 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, for example, speed, trim, and loading. In addition, it is useful to change the autopilot control law parameters when the ship is exposed to large disturbances resulting 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 interest to have a method for automatically adjusting or modifying the underlying controller. The use of a Fuzzy Model Reference Learning Controller (FMRLC) to maintain adequate performance of a tanker ship autopilot when there are process disturbances and variations has been investigated in [11]. However, the three FMRLC design parameters (controller gains) are heuristically determined by trial and errors. In this paper, a Particle Swarm Optimization (PSO) of Fuzzy Model Reference Learning Controller (FMRLC) [11] is proposed to achieve the heading regulation of tanker ship exposed to plant changes and disturbances. PSO has been shown to be a promising approach for solving both unconstrained and constrained optimization problems [2][3][4][5]. Recently, several heuristics have been developed to improve the performance

and set up suitable parameters for the PSO algorithm [2][3]. Some theoretical work to analyze the trajectory of particles has been carried out. A constriction factor has been proposed by Clerc and Kennedy [2] to ensure convergence. Trelea [3] reported on the trajectory analysis using dynamic systems theory. In addition, random starting configurations always lead to additional variability in PSO. The performance of PSO can be improved by strategically selecting the starting positions of the particles. It has been suggested the use of generators from Centroidal Voronoi Tessellation (CVT) as the starting points for the swarm can ensure the broad coverage of the search space and thus the solution space is fully explored for the optimal solution [6]. The same population initialization strategy, CVT will be applied in this paper.

Section II presents the steering dynamics and control of a tanker ship model. In Section III, the Fuzzy Model Reference Learning Controller (FMRLC) is explained. Section IV describes the Trelea's PSO model. Centrodail Vornoi Tesselation algorithm is briefly described in Section V. Section VI discusses experimental results of the optimized FMRLC controller. Conclusion is presented in Section VII.

2 Tanker Ship Steering

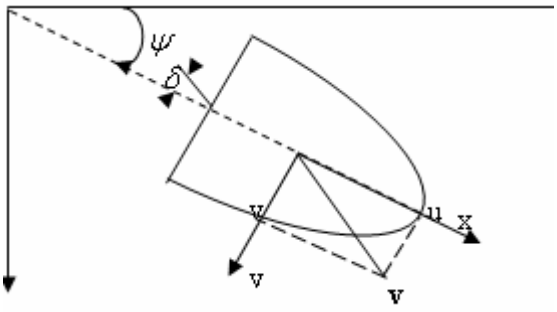

Figure 1. Tanker Ship

Assuming the "bobbing" or "bouncing" effects of the ship is neglected for large vessels, the motion of the ship is described by a coordinate system which is fixed to the ship [10]. Based on Fig. 4, a simple model which describes the dynamical behavior of the ship can be expressed by the following differential equation:

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

where ψ is the heading of the ship and δ is the rudder angle. Assuming zero initial conditions, (1) can be written

$$\frac{\phi(s)}{\delta s}=\frac{K(s\tau_3+1)}{s(s\tau_1+1)(s\tau_2+1)} \quad (2)$$

where K,τ_1,τ_2 and τ_3 are parameters which are a function of the ship's constant forward velocity u and its length l as expressed below:

$$K=K_0\left(\frac{u}{l}\right) \quad (3)$$

$$\tau_i=\tau_{i0}\left(\frac{l}{u}\right),\quad i=1,2,3. \quad (4)$$

We assumes that for a tanker ship under "ballast" conditions (a very heavy ship), $K_0=5.88,\ \tau_{10}=-16.91,\ \tau_{20}=0.45,\ \tau_{30}=1.43,$ and $l=350$ meters [8]. For "full" conditions, $K_0=0.83,\ \tau_{10}=-2.88,\ \tau_{20}=0.38,\ \tau_{30}=1.07$.

The nominal speed of the tanker ship traveling in the x direction is $u=5m/s$. The maximum deviation of the rudder angle is assumed to be ± 80 degrees. This is only valid if the ship make small deviations from a straight line path, $(\delta<5^\circ)$. For $\delta>5^\circ$, an extended model given as follows should be used [8].

$$\dddot{\psi}(t)+\left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\ddot{\psi}(t)+\left(\frac{1}{\tau_1}\frac{1}{\tau_2}\right)H(\dot{\psi}(t)) = \frac{K}{\tau_1\tau_2}\left(\tau_3\dot{\delta}(t)+\delta(t)\right) \quad (5)$$

where $H(\dot{\psi}(t))$ is a nonlinear function of $\dot{\psi}(t)$. The function is approximated as equation (19) under steady state condition, $\dddot{\psi}=\ddot{\psi}=\dot{\delta}=0.$

$$H(\dot{\psi})=a\dot{\psi}^3+b\dot{\psi} \quad (6)$$

where a and b are real valued constants such that a is always positive. They are chosen to be 1 in this paper. For optimization purpose, the following cost function which represents the propulsive energy losses due to steering is defined.

$$J = \frac{1}{T}\left[\sum_{t=1}^{T}\left(e(t)^2 + \lambda\delta(t)^2\right)\right] \quad (7)$$

The λ value is set to 0.01 in the simulation.

3 Fuzzy Model Reference Model Learning Controller (FMRLC)

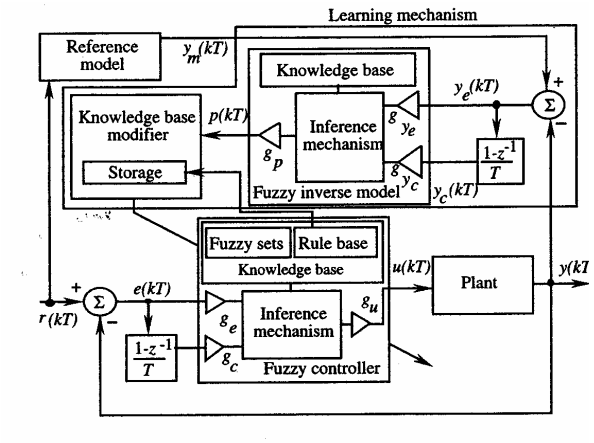

Figure 2. Fuzzy Model Reference Learning Controller.

The functional block diagram for the FMRLC is shown in Fig. 2. It has four main parts: the plant, the fuzzy controller to be tuned, the reference model, and the learning mechanism (an adaptation mechanism). The FMRLC uses the learning mechanism to observe numerical data from a fuzzy control system (i.e., $r(kT)$ and $y(kT)$ where T is the sampling period). Using these values, the fuzzy control system's performance can be characterized and fuzzy controller can be synthesized and adjusted so that the given performance objectives are met. These performance objectives (closed-loop specifications) are characterized via the reference model shown Fig. 2. The learning mechanism seeks to adjust the fuzzy controller so that the closed-loop system (the map from $r(kT)$ to $y(kT)$) acts like the given reference model (the map from $r(kT)$ to $y_m(kT)$). Basically, the upper level fuzzy control system loop operates to make $y(kT)$ track $r(kT)$ by manipulating $u(kT)$, while the upper-level adaptation control loop seeks to make the output of the plant $y(kT)$ track the output of the reference model $y_m(kT)$ by manipulating the fuzzy controller parameters. The fuzzy controller inputs are defined to be (13)

$$e(kT) = \psi_r(kT) - \psi(kT)$$
$$c(kT) = \frac{e(kT) - e(kT-T)}{T} \quad (8)$$

where $\psi_r(kT)$ is the desired ship heading ($T = 10$ seconds). The controller output is the rudder angle, $\delta(kT)$, of the ship. Eleven uniformly spaced triangular membership functions for each controller input on normalized universe of discourse are used. The output membership functions are assumed to be symmetric and triangular shaped with a base width 0.4 (on a normalized universes of discourse), and centered via the tuning of the fuzzy controller for the nominal plant. The reference model is chosen to be

$$\frac{\psi_m(s)}{\psi_r(s)} = \frac{1}{150s+1} \quad (9)$$

where $\psi_m(t)$ specifies the desired system performance for the ship heading $\psi(t)$. The fuzzy inverse model inputs are chosen to be

$$\psi_e(kT) = \psi_m(kT) - \psi(kT)$$
$$\psi_c(kT) = \frac{\psi_e(kT) - \psi_e(kT-T)}{T} \quad (10)$$

We use eleven fuzzy sets defined with symmetric and triangular shaped membership functions, which are evenly distributed on the appropriate universes of discourse. For fuzzy inverse model design, note that for a tanker ship, an increase in the rudder angle $\delta(kT)$ will generally result in a *decrease* in the ship heading angle. This is the information about the inverse dynamics of the plant that we use in the fuzzy inverse model rules. Specifically, we will use rules of the form

$$\text{If } \tilde{\psi}_e \text{ is } \tilde{\Psi}_e^i \text{ and } \tilde{\psi}_c \text{ is } \tilde{\Psi}_c^j \text{ Then } \tilde{p} \text{ is } \tilde{P}^m \quad (11)$$

The output membership function centers are $c_{i,j}$ (i^{th} membership function for $\tilde{\psi}_e$ and j^{th} membership function for $\tilde{\psi}_c$). The fuzzy rule base is shown in Table 1 for the fuzzy inverse model. In Table 1, Ψ_e^i denotes the i^{th} fuzzy set associated with the error signal ψ_e, and Ψ_c^j denotes the j^{th} fuzzy set

Table 1. Rule base for the tanker ship fuzzy inverse model

$c_{i,j}$		Ψ_c^j										
		−5	−4	−3	−2	−1	0	1	2	3	4	5
Ψ_e^i	−5	1	1	1	1	1	1	.8	.6	.4	.2	0
	−4	1	1	1	1	1	.8	.6	.4	.2	0	−.2
	−3	1	1	1	1	.8	.6	.4	.2	0	−.2	−.4
	−2	1	1	1	.8	.6	.4	.2	0	−.2	−.4	−.6
	−1	1	1	.8	.6	.4	.2	0	−.2	−.4	−.6	−.8
	0	1	.8	.6	.4	.2	0	−.2	−.4	−.6	−.8	−1
	1	.8	.6	.4	.2	0	−.2	−.4	−.6	−.8	−1	−1
	2	.6	.4	.2	0	−.2	−.4	−.6	−.8	−1	−1	−1
	3	.4	.2	0	−.2	−.4	−.6	−.8	−1	−1	−1	−1
	4	.2	0	−.2	−.4	−.6	−.8	−1	−1	−1	−1	−1
	5	0	−.2	−.4	−.6	−.8	−1	−1	−1	−1	−1	−1

associated with the change in error signal ψ_c. The entries of the table represent the center values of symmetric triangular membership functions $c_{i,j}$ with base widths 0.4 for output fuzzy sets P^m on the normalized universe of discourse. In this paper, we utilize minimum to represent the premise and implication, and center of gravity (COG) for defuzzification. Rule base modification is performed by shifting centers b_m of the membership functions of the output linguistic value that are associated with the fuzzy controller rules that contributed to the previous control action $u(kT-T)$. This is a two-step process.

- Find all the rules in the fuzzy controller whose premise certainty $\mu_i\left(e(kT-T), c(kT-T)\right) > 0$ and call this the "active set" of rules at time $kT-T$.
- For all rules in the active set, use

$$b_m(kT) = b_m(kT-T) + p(kT) \qquad (12)$$

to modify the output membership function centers. Rules that are not in the active set do not have their output membership functions modified. The $p(kT)$ is calculated based on COG of the output fuzzy sets P^m.

The fuzzy inverse model scaling gains, $g_{\psi_e}, g_{\psi_c}, g_u$ are optimized using the Particle Swarm Optimization approach whereas the fuzzy controller output gain $g_p = 0.4$ is chosen.

4 Particle Swarm Optimization (PSO)

The particle swarm optimization (PSO) is a parallel evolutionary computation technique, originally developed by Kennedy and Eberhart [1][2]. The PSO algorithm is initialized with a population of random candidate solutions, conceptualized as particles. Each particle is assigned a randomized velocity and is iteratively moved through the problem space. Each particle is attracted towards the location of the best fitness so far by the particle itself and by the location of the best fitness achieved so far across the population. The algorithm is described in the following:

(i) Initialize a population of random positions and velocities in the n-dimensional problem space.

(ii) Evaluate the fitness value for each particle.

(iii) Each particle's fitness evaluation is compared with the current particle's pbest. If current value is better than pbest, update the pbest with the current value in n-dimensional space.

(iv) Compare fitness evaluation with the population's global best fitness, gbest, then reset gbest to the current particle's array index and value.

(v) The velocities and positions of the particle are updated as follows:

$$\begin{aligned} \boldsymbol{v}_i(t+1) &= w\boldsymbol{v}_i(t) + c_1 rand_1\left(\boldsymbol{pbest}_i - \boldsymbol{x}_i(t)\right) \\ &+ c_2 rand_2\left(\boldsymbol{gbest} - \boldsymbol{x}_i(t)\right) \qquad (13) \\ \boldsymbol{x}_i(t+1) &= \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1) \end{aligned}$$

(vi) Loop to (ii) until a stopping criterion is met.

The vector $\boldsymbol{x_i} = \left[x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in}\right]^T$ stands for the position of the i^{th} particle, $\boldsymbol{v_i} = \left[v_{i1}, v_{i2}, v_{i3}, \ldots, v_{in}\right]^T$ represents the velocity of the ith particle. The variable w is the inertia weight, c_1 and c_2 are positive constants; $rand_1$ and $rand_2$ are uniform random numbers in the range [0; 1], generated anew for each dimension,

$i = 1, \ldots, n$ of the particle i. The inertia weight w represents the momentum of the particles. The constants c_1 and c_2 stands for the controlling factors of the "cognition" and "social" parts that attract each particle toward pbest and gbest. The "cognition" factor governs the independent behavior of the particle itself whereas the "social" factor determines the collective behavior of the particles. In the paper [2], Clerc and Kennedy introduced a constriction coefficient k for swarm system stability and convergence. The velocity equation is updated according to

$$\begin{aligned} &\boldsymbol{v}_i(t+1) \\ &= k\left[\boldsymbol{v}_i(t) + c_1 rand_1\left(\boldsymbol{pbest}_i - \boldsymbol{x}_i(t)\right) + c_2 rand_2\left(\boldsymbol{gbest} - \boldsymbol{x}_i(t)\right)\right] \end{aligned} \tag{14}$$

where $k = \dfrac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}$ with $\varphi = c_1 + c_2 > 4$ and k is a function of c_1 and c_2. By default, φ is set to $4.1\left(c_1 = c_2 = 2.05\right)$, and the construction coefficient k is 0.729. This is equivalent to $w = 0.729, c_1 = c_2 = 1.494$ in equation (16). In this paper, the PSO is named as Clerc model. Trelea [3] also presented convergence analysis for a one-dimensional particle. The parameter settings reported in his simulations were $w = 0.85, c_1 = c_2 = 1.7$. It is named as Trelea model in this paper.

5 Centroidal Voronoi Tessalations

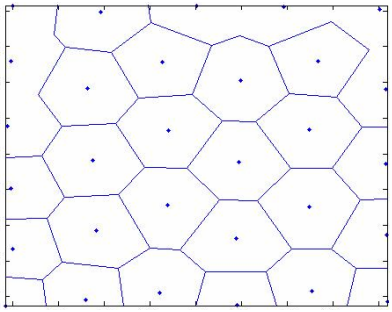

Figure 3. Centroidal Voronoi Tesselations for 30 points

A group of points in the search space is designed to be the set of solution generators. Particle initialization in PSO can be thought as a process to allocate the solution generators in the search space. The space is partitioned into compartments with particle as their centroid. The particles should be initialized so that they are distributed as evenly as possible throughout the space to ensure broad coverage of the search spaces. The standard method of particle initialization in PSO fails to accomplish this goal, especially in high-dimensional spaces [6]. Centroidal Voronoi Tessellation (CVT) is a way to partition a space into compartments [6]. It has been shown that CVT can initialize the PSO particles evenly in the solution space and lead to improved performance [6]. Two of the most well-known algorithms for computing CVTs are Mac Queen's method [6] and Lloyd's method [7]. Lloyd's algorithm is deterministic and requires only a few iterations, but each one is computationally expensive. In this paper, Lloyd's algorithm is chosen to compute CVT using the source codes available from the authors [7]. The details of the algorithm are explained in the literature [7]. Figure 3 shows the CVTs for 30 points of generators to be used as initial population of particles in PSO for the experiment to be discussed in this paper.

6 Experimental Results

The nonlinear process model given in equation (9) is used to emulate the "real" ship dynamics. The PSO models will perform the FMRL controller optimization for the three design parameters $g_{\psi_e}, g_{\psi_c}, g_u$ based on the cost function defined in equation (7). The PSO algorithms were implemented in Matlab [10]. During the numerical experiments, the Trelea PSO model was run with an initial population of particles generated by CVTs shown in Figure 3. All the running trials were carried out with a population of 30 particles and 200 generations. The results of the Trelea PSO model is shown in Figure 4. From the results obtained, the PSO model presents the ability to optimize the FMRL controller of tanker ship steering. Table 2 summarizes the results.

Table 2 Summary of results

PSO model (Trelea)	g_u	g_{ψ_e}	g_{ψ_c}	**gbest (J)**
	2.8939	1.1180	150.95	10.411

The optimized FMRL controller is utilized for the tanker ship steering regulation under different operating conditions such as wind disturbances, changes of speed, existence of sensor noise and ship weight changes.

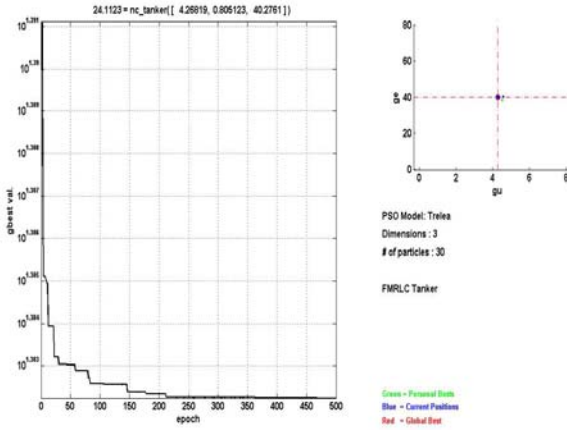

Figure 4. Result for the FMRLC controller optimization of tanker ship

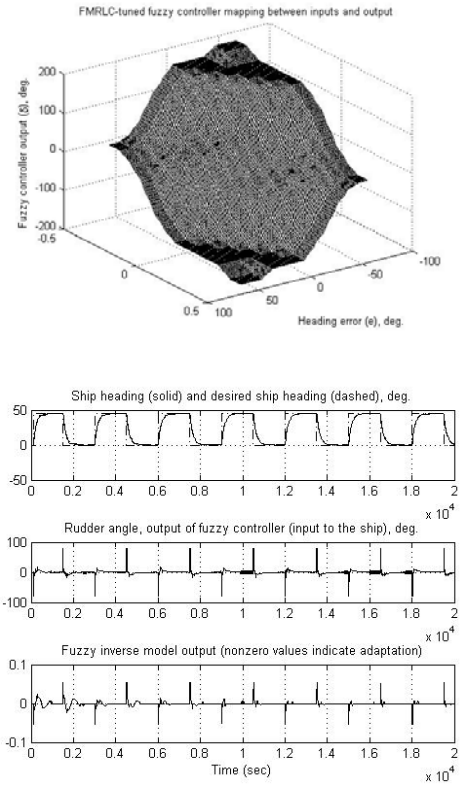

Figure 5. Controller response surface and closed-loop response resulting from using the optimized FMRL controller for tanker ship steering under nominal condition.

Nominal conditions

The first simulation study concerns the tanker ship steering control under nominal condition, where we have ballast conditions (heavy weight), no wind, no sensor noise, and a nominal speed of 5 m/s. The close-loop response is shown in Fig. 5. The optimized scale gains $g_{\psi_e}, g_{\psi_c}, g_u$ by PSO allow the fuzzy controller to start with a good guess of the controller with very little tuning occurs and get good tracking performance. The learned input-output of controller surface is shown in Fig 5.

Effects of Wind Disturbances

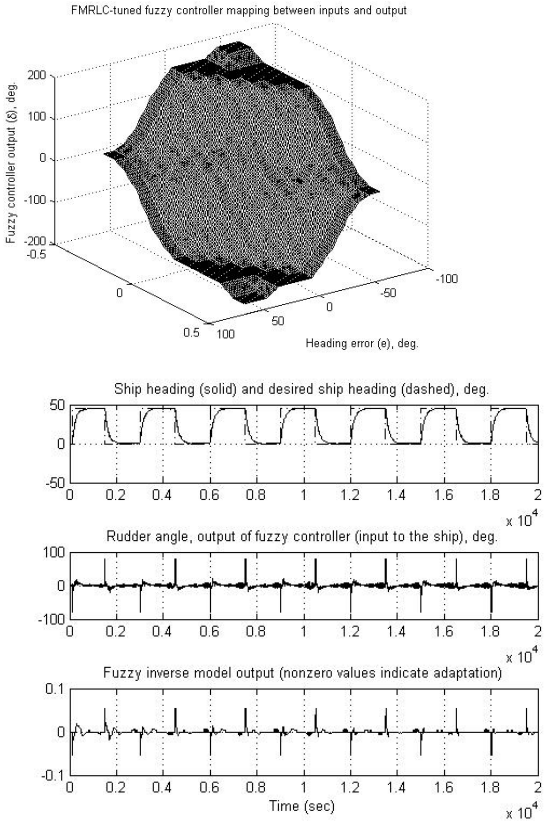

Figure 6. Controller response surface and closed-loop response resulting from using the optimized FMRL controller for tanker ship steering under wind disturbances.

The wind disturbance on effects on the ship is studied in this experiment. An additional sinusoidal disturbance effect is added to the rudder angle of the tanker ship. In Fig. 6, we

found that the FMRLC controller is able to compensate the wind disturbance effects to achieve very good regulation of the ship heading. The FMRLC controller learns how to shape the mapping to reduce the effects of the wind disturbances by on-line adaptation rule, even though the precise characteristics of the wind disturbance are not known. The final FMRL controller response surface is shown in Fig. 6.

Effects of Speed Change

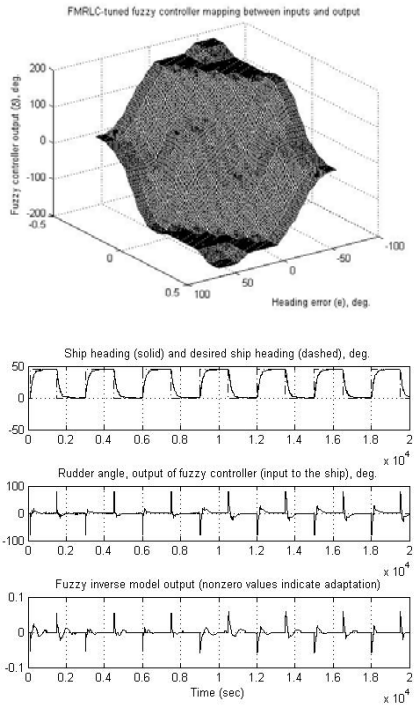

Figure 7. Controller response surface and closed-loop response resulting from using the optimized FMRL controller for tanker ship steering under speed change.

In this experiment, we consider the effect of speed change on the tanker ship steering control. For the first 9000 seconds, the ship operates at a speed of $u = 5 m/s$. The speed is changed abruptly at $t = 9000$ seconds. The close-loop response and the controller response surface is depicted in Fig 7. The same response as for nominal conditions are observed for the first 9000 seconds. Then, it adapts to the speed change after $t = 9000$ seconds.

Effects of Weight Changes

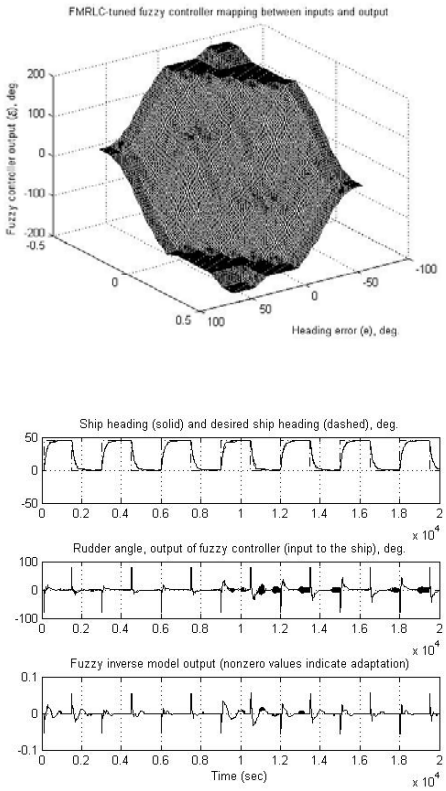

Figure 8. Controller response surface and closed-loop response resulting from using the optimized FMRL controller for tanker ship steering under weight change.

We consider the case of how the ship steers when at $t = 9000$ seconds, we switch from ballast to full conditions. Fig. shows the close-loop response. The FMRL controller is able to compensate for the weight change, and the final tuned controller response surface under full conditions (Fig. 8) is different from the response surface under ballast condition. The difference in the controller mapping is shown in Fig. 8. The input-output map changes when the controller adapts from a ballast condition to a full one.

Effects of Sensor Noise

The close-loop response shown in Fig. 8 indicates the additive sensor noise uniformly distributed on $[-0.01, 0.01]$ has little effect on the response. The FMRL controller is robust to the sensor noise effects.

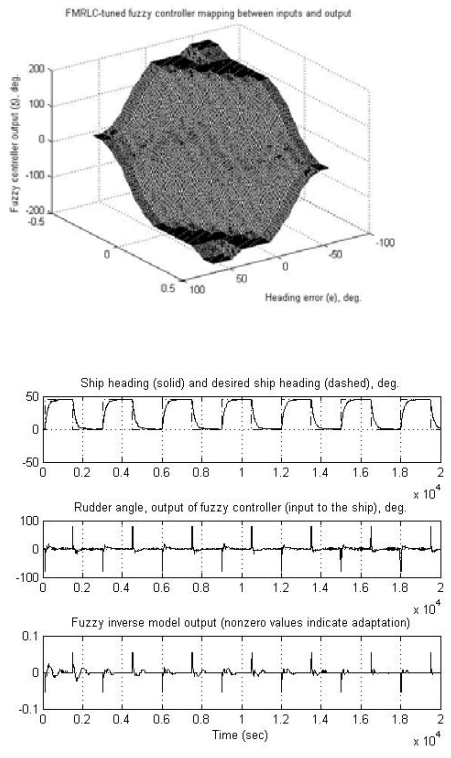

Figure 8. Controller response surface and closed-loop response resulting from using the optimized FMRL controller for tanker ship steering under sensor noise.

7 Conclusion

This paper described the study of PSO optimization of Fuzzy Model Reference Learning Controller (FMRLC) for the tanker ship steering system. Despite the highly nonlinear characteristics of tanker ship dynamics, the optimized FMRLC controller has shown its robustness and ability to regulate steering angle under wind disturbance, speed change, sensor noise effect and sudden change of ship load. It is shown that the fuzzy controller is continually updated in response ship parameters variations or wind disturbances. The promising results in this paper clearly indicate that PSO can be an effective tool to optimize the FMRLC and therefore enhance the ease of use and utility value of FMRLC for navigation control of tanker ship and other naval engineering applications.

References:

[1] C. G. Kallstrom, K. K. Astrom, N. E. Thocell, J. Eriksson, and L. Sten, "Adaptive Autopilots for Tanker," Automatica, p. 241-254, May 1979.

[2] M. Clerc and J. Kennedy, "The particle swarm – Explosion, stability, and convergence in a multidimensional complex space," IEEE Trans. Evol. Comput., vol. 6, no. 1, pp. 58-73, Feb 2002.

[3] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," Inf. Process. Lett., vol. 85, no. 6, pp. 317-325, 2003.

[4] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method for constrained optimization problems," in Intelligent Technologies – Theory and Applications: New Trends in Intelligent Technologies, vol. 76, P. Sincak, J. Vascak, V. Kvasnicka, and J. Pospicha, Eds. Amsterdam, The Netherlands: IOS Press, 2002, pp. 214-220.

[5] G. Toscano-Pulido and C. a. Coello, "A constraint handling mechanism for particle swarm optimization," in Proc. IEEE CEC, Portland, OR, Jun. 2004, vol. 2, pp. 1396-1403.

[6] Richard, Mark and Ventura, Dan, "Choosing a starting configuration for particle swarm optimization," Proceedings of the Joint Conference on Neural Networks, pages 68-75, 2005.

[7] Q. Du, M. Faber, M. Gunzburger, "Centroidal Voronoi Tessellations: Applications and algorithms," SIAM Review, vol. 41, 1999, pp. 637-676.

[8] K. Astrom and B. Wittenmark. Apdative Control. Reading, MA:Addison-Wesley Pub. Co., 1989.

[9] M. Bech and L. Smitt, "Analogue simulation of ship maneuver," Tech Rep., Hydro-og Aerodymisk Laboratorium, Lyngby, Denmark, 1969.

[10] B. Birge, "PSOt – a particle swarm optimization toolbox for use with Matlab," in Proceedings of the IEEE Swarm Intelligence Symposium, IN, pp. 182-186, 2003.

[11] Passino, K M. Biomimicry for Optimization, Control and Automation. Springer-Verlag London 2005.