Tracking Control Based on Improved Fuzzy Gauss Radial Basis Function Neural Network for Omni-Directional Intelligent Wheelchair

Jinhui Fan 1,2, Songmin Jia 1 and Wei Lu 3

1 College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing, China
2 College of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang, China
3 Shijiazhuang Medical College, Shijiazhuang, China
Email: jh_fan@163.com

Abstract. In conventional fuzzy neural network based on sigmoid function, the number of nodes at the corresponding intermediate layer grows larger with the increase of fuzzy rule numbers. In order to obtain real time performance of the control system, an improved fuzzy Gaussian radial basis function neural network controller for omni-directional intelligent wheelchair was proposed. The structure and the design method of the improved controller were described in detail. The optimized fuzzy neural network reduced the numbers of intermediate layer, thus the computation real time ability could be guaranteed. Furthermore, the stability of the controller was proved through Lyapunov stability theory. Finally, the proposed optimized controller was applied to an omni-directional intelligent wheelchair, and the simulation results of tracking control verified the real time, effectiveness and feasibility of the proposed method.

1. Introduction
On condition that the model of a mobile robot is already known, the trajectory tracking controller can be designed and described basing on its accurate model [1]. In fact, the unknown and uncertain factors exist during the trajectory tracking, such as the frictions, and the change of robot’s gravity centre.

In order to solve the above problems, optimal control, fuzzy control and neural network are widely applied in the field of mobile robot control [2-4]. Fuzzy control and neural network are applied to linear, nonlinear and uncertainty systems in complex environments, since they are independent of the accurate mathematics models [5]. In fuzzy control method, the design experiences are required for the controller designer. It is difficult for an inexperienced designer to initialize these parameters properly. These years, the combination of fuzzy control and neural network has become attractive increasingly [6-9]. Fuzzy Gaussian Radial Basis Function (FGRBF) neural network has the advantages of above algorithms. In this paper, an improved FGRBF neural network is proposed and it reduces the numbers of intermediate layer, simplifies the algorithm complexity, and guaranteed the computation real-time ability. Finally, the proposed algorithm is applied to an omni-directional intelligent wheelchair, and the simulation results verified the real-time, effectiveness and feasibility of the proposed method.

2. Problem Statement
The dynamic model of an omni-directional intelligent wheelchair system is based on reference [10]. The rigid body model in the body-fixed reference frame can be expressed as
\[ M \dot{v} + C(v) \dot{v} + D(v) \dot{v} + g(q) + \tau_d = \tau \]
\[ \dot{q} = J(q) \dot{v} \]

where \( v = [v_x, v_y, \omega]^T \), and \( q = [x, y, \phi]^T \). Here, \( v \) represents the position and attitude vector in the body-fixed reference coordinate, \( q \) denotes the position and attitude vector in the earth-fixed reference coordinate, and \( \tau \) denotes the control forces and moments generated by motors in the body-fixed reference coordinate. Here \( J(q) \) denotes a transition matrix, by which the velocity vector in body-fixed reference coordinate can be transformed into the earth-fixed reference coordinate. \( M \) is a symmetric, positive definite inertia matrix including the mass of the platform and its user, \( C(v) \) is the centripetal and Coriolis matrix, \( D(v) \) is the friction matrix, \( g(q) \) is the gravitational vector of forces and moments, and \( \tau_d \) represents the forces and moments of nonlinear random disturbances. Here, we consider \( g(q) = 0 \) since the movement of omni-directional platform is two-dimensional planar motion. Therefore, the equation of an omni-directional platform can be transformed into the earth-fixed reference coordinate as

\[ M_q(q) \ddot{q} + C_q(q,v) \dot{q} + D_q(q,v) \dot{q} + \tau_{dq} = J^{-1}(q) \tau \]

where \( M_q(q) = J^{-1}(q)MJ^{-1}(q) \cdot C_q(q,v) = J^{-1}(q)[C(q,q) - MJ^{-1}(q)J(q)]J^{-1}(q) \cdot D_q(q,v) = J^{-1}(q)D(q,q)J^{-1}(q) \), and \( \tau_{dq} = J^{-1}(q) \tau_d \). Here, \( M_q(q) \) is symmetric and positive definite, and \( C_q(q,v) \) satisfies the skew symmetric as \( x^T [\hat{C}_f(q) - 2C_c(q,v)] x = 0 \). The friction matrix \( D_q(q,v) \) is positive definite in figure 1.

In equation (2), the friction \( D_q(q,v) \) and the forces and moments of disturbances \( \tau_d \) are relevant to some random nonlinear factors. The friction coefficients between wheels and ground are difficult to obtain. An appropriate controller is necessary for nonlinear system with uncertainties and disturbances.

3. Optimization of Fuzzy Gauss Radial Basis Function Neural Networks

3.1. Simplified Fuzzy Rules

The simplified fuzzy rules can be seen as special Sugeno’s fuzzy rules[11,12]. The \( i \)-th control rule is

\[ R_i : \text{If } x_1 = A_{i1} \text{ and... and } x_n = A_{in} \text{ then } u_1 = B_{i1} \text{ and... and } u_p = B_{ip} \]

where \( R_i \) is the \( i \)-th fuzzy control rule, \( x_1, \ldots, x_n \) are the fuzzy controller inputs, \( u_1, \ldots, u_p \) are the outputs of it, \( A_p \) denotes the fuzzy set in the antecedent associated with the \( j \)-th input at the \( i \)-th control rule, and \( B_j \) presents a constant associated with the \( j \)-th in the conclusion at the \( i \)-th control rule. Considering equation (3), the \( j \)-th output is written as equation (4), here \( h_i \) is the confidence as equation (5)

\[ u_j^* = \frac{\sum_{i=1}^{n} h_i B_{ij}}{\sum_{i=1}^{n} h_i}, j = 1, 2, \ldots, p \]

\[ h_i = \mu_{A_{i1}}(x_1) \cdot \mu_{A_{i2}}(x_2) \ldots \mu_{A_{in}}(x_n) \]

and “\( \cdot \)” is the algebraic product. \( \mu_{A_n}(x_n) \) denotes the confidence of each fuzzy rule.

3.2. Membership Functions and Neural Network Structure

For a conventional multilayer neural network, the sigmoid function is usually defined as

\[ f(x) = (1 + e^{-x})^{-1} \]
where function \( f(x) \) and two sigmoid functions are with the ranges on \([0,1]\) and on \([-1,1]\) must be designed. But if sigmoid functions were applied, the node number at the intermediate layer will grow larger with the increase of fuzzy rule numbers. Sigmoid functions can be replaced by

\[
 f(x) = \exp\left[\ln(0.5) \cdot (w_c x_i - w_d)^2 + w_i^2\right] 
\]

(7)

where the connection weight \( w_c \) is the centre value of a Gaussian function, and the connection weight \( w_d \) presents the reciprocal value of deviation from the centre \( w_c \) to the Gaussian function on the standard support set has 0.5. The connection weight \( w_i \) stands for the fuzzy consequent part which is equivalent to \( B^v \) of equation (3). The structure of the fuzzy neural network can be designed as figure 2. If \( w_c, w_d \) and \( w_i \) are adjusted, the membership function is also adjusted.

![Figure 1. The kinematic model of an omni-directional intelligent wheelchair](image)

![Figure 2. The neural network structure based on optimized FGRBF](image)

According to the input and output conditions, the controller can be designed as a system with two inputs and one output. In figure 2, layers from 1 to 5 correspond to the antecedent part of the fuzzy control rules, and layers from 6 to 8 correspond to the conclusion part. The conclusion part means the outputs of the controller, here \( x = [x_1, x_2] \), and \( u = [u_1] \). The antecedent part of the fuzzy control rule is 3, thus the control rule number is \( r = 3^3 = 9 \). In figure 2, the middle layer is simplified, i.e. the middle layer for computing \( \sum_{i=1}^{5} h_i \) is pruned. The 8-th layer can be defined as
The middle layer nodes are reduced. The problem that the number of nodes at the intermediate layer grows larger with the increase of fuzzy rule numbers can be solved.

3.3. Training Method of Improved FGRBF

The improved FGRBF is designed with BP (Back-propagation). \( \delta_{j}^{M} \) in the \( j \)-th unit at output layer \( M \) and \( \delta_{j}^{k} \) in the \( j \)-th unit at any intermediate layer \( k \), are given by equation (9) and equation (10).

\[
\delta_{j}^{M} = f'(i_{j}^{M}) \sum_{i=1}^{N} (y_{a} - y_{i}) \frac{\partial y_{j}}{\partial u_{j}} \\
\delta_{j}^{k} = f'(i_{j}^{k}) \sum_{i=1}^{N} (\delta_{j}^{k+1} w_{ji}^{k+1}) \left( \prod_{i=1}^{N} w_{ji}^{k+1} o_{i} \right) 
\]

where \( f'(\cdot) \) denotes the differential of \( f(\cdot) \). For the \( j \)-th node at the \( M \)-th layer, \( i_{j}^{M} \) is the input of this node, \( y_{a} \) is the expected output of this node, and \( y_{i} \) is the actual output of this node. In equation (9), \( \frac{\partial y_{j}}{\partial u_{j}} \) denotes the partial derivative of \( y_{j} \) in relation to \( u_{j} \), \( w_{ji}^{k+1} \) is the connection weight between the \( j \)-th node at the \( k \)-th layer and the \( l \)-th node at the \((k+1)\)-th layer. The weight can be calculated by

\[
w_{ji}^{k-1,k}(t+1) = w_{ji}^{k-1,k}(t) + \eta \delta_{j}^{k} o_{i} + \alpha \Delta w_{ji}^{k-1,k}(t)
\]

where \( t \) denotes the \( t \)-th update of connection weight, \( \eta \) is the learning rate as a small positive constant, and \( \alpha \) is also a small positive constant called stabilizing factor, and \( \Delta w_{ji}^{k-1,k}(t) \) is an increment of the connection weight at the \( t \)-th step during the network training process. Equation (11) can be used to adjust the parameters of the connection weight \( w_{c}^{i} \), \( w_{d}^{i} \) and \( w_{b}^{i} \) autonomously.

4. Controller Design and Stability Proof

4.1. Controller Design Based on Improved FGRBF

The main goal is to design an improved FGRBF neural network controller in trajectory tracking process. The omni-directional platform is of multi-input and multi-output system. The inputs include \( v_{ud}^{i} , v_{vd}^{i} \) and \( \omega_{d}^{j} \). The outputs are \( v_{i} \), \( v_{c} \) and \( \omega \). Some assumptions are provided for controller design.

**Assumption 1:** During motion process of the omni-directional platform, the desired position and attitude are known bounded functions, satisfying the condition as

\[
\left[ q_{d}^{i} \dot{q}_{d}^{i} \dot{q}_{d}^{i} \right] \leq q_{a}
\]

where \( q_{d} \) denotes the desired position and attitude of the omni-directional platform in the earth-fixed reference coordinate, and we assume \( \dot{q}_{d} \) and \( \dot{q}_{d} \), the time derivatives of \( q_{d} \), can be obtained from a trajectory planner. Here, \( q_{a} \) is a positive constant. The filtered errors can be written as

\[
s = \hat{q} + \lambda \ddot{q}
\]

where \( \hat{q} = q_{d} - q \), \( \dot{q} = \dot{q}_{d} - \dot{q} \), and \( \lambda \) is a positive constant. \( \ddot{q} \) is the vector of the virtual reference trajectory in the earth-fixed reference coordinate, and (14) is satisfied as

\[
\dot{\ddot{q}} = J(q) \ddot{v}_{r}
\]
Here $\dot{q} = J^{-1}(q)\left(\ddot{q} - \mathbf{J}'(q)J^{-1}(q)\dot{q}\right)$, (13) is written as (15), and its deviation is expressed as (16)

$$s = \dot{q} - \dot{\dot{q}}$$
$$s = \dot{q} - \dot{\dot{q}} \Rightarrow \dot{q} = \dot{\dot{q}} + \lambda \dot{q}$$

Considering equation (11) and (15), equation (2) is rewritten as equation (17), and with equation (10) and (17), we can obtain equation (18)

$$M\ddot{q} + C\dot{q} + D\dot{q} = J^T(\tau - \tau_d)$$
(17)
$$M\ddot{q} + C\dot{q} + D\ddot{q} = J^T(M\dot{v} + Cv + Dv)$$
(18)

Considering equation (16) and (18), taking derivative of $s$, dynamics in terms of $s$ is written as

$$M_s \ddot{s} + C_s \dot{s} + D_s \dot{s} = J^T(M \dot{v} + Cv + Dv + \tau_d - \tau) - (C_s + D_s)s$$
(19)

Function $f(\dot{\psi}, \nu, v, q)$ is defined as the output of the FGRBF neural network. It is defined as equation (20). Nonlinear function is defined as equation (21)

$$f(\dot{\psi}, \nu, v, q) = W^T\Phi + \varepsilon$$
(20)
$$f(\dot{\psi}, \nu, v, q) = M\dot{v} + Cv + Dv$$
(21)

here, $W, \Phi$ are the same as defined above.

**Assumption 2:** For the designed FGRBF, the weight $W$ is bounded as $\|W\| \leq W_{\max}$ with $W_{\max} > 0$, where $\|W\| = \text{tr}(W^T W) = \sum_{i,j} W_{ij}^2$, $\|\cdot\|$ denotes the Frobenius norm, and $\text{tr}(\cdot)$ is the trace operation.

**Assumption 3:** The FGRBF neural network approximation error $\varepsilon$ is bounded as $\|\varepsilon\| \leq \varepsilon_N$ ($\varepsilon_N$ is a positive constant).

Because the FGRBF neural network is used to approximate the platform dynamics with high precision, the system error will be small enough. Therefore, equation (22) and $f(\dot{\psi}, \nu, v, q)$ is the approximation of the platform dynamics. The FGRBF input is defined as equation (23)

$$f(\dot{\psi}, \nu, v, q) = f(\dot{\psi}, \nu, v, q)$$
(22)
$$\tau = f(\dot{\psi}, \nu, v, q) + J^T K_s s + \alpha$$
(23)

where $K_s = K_s^T > 0$, and it denotes control gain matrix. $\alpha$ denotes a robustness control term, we define

$$\hat{f}(\dot{\psi}, \nu, v, q) = \hat{W}^T \Phi$$
(24)

here, $\hat{W}$ is the approximation of weight between hidden and output layer, and derivative of $\hat{W}$ is

$$\dot{\hat{W}} = \Gamma^T \Phi(J^{-1} s)^T - \zeta T^r \|\hat{W}\|$$
(25)

where $\Gamma$ is a positive constant matrix with $\Gamma = \Gamma^T > 0$, and $\zeta$ is a constant with $\zeta > 0$. Taking a control input as equation (23), the closed-loop system described in equation (19) can be written as

$$M_s \ddot{s} + C_s \dot{s} + D_s \dot{s} = J^T(M \dot{v} + Cv + Dv + \tau_d - \tau) - (C_s + D_s)s$$
(26)

where $f(\dot{\psi}, \nu, v, q)$ can be written as equation (27), and the weight estimated error is equation (27), and the weight estimated error is equation (28)
4.2. Stability Proof of Designed Control System

**Theorem 1:** If Assumptions 1-3 are satisfied, the input satisfies equation (23), the adaptive control law of FGRBF neural network weight is equation (25), and the robustness term is defined as

\[ \alpha = (\varepsilon_d + d_p) J^{-1} s \]

the filtered tracking errors \( \hat{s}(t) \) and weight estimated errors \( \hat{\theta} \) are uniformly ultimately bounded stable.

**Proof:** The Lyapunov function is equation (30), and using equation (13),(17), we have equation (31).

\[ V = \frac{1}{2} s^T M_s s + \frac{1}{2} \text{tr} \{ W^T \Gamma^{-1} \hat{W} \} \]

\[ V = \frac{1}{2} \left( s^T M_s s + s^T M_s \hat{s} + s^T \hat{M}_s s \right) + \text{tr} \{ \hat{W}^T \Gamma^{-1} \hat{W} \} \]

Here, \( s^T M_s s = s^T M_s \hat{s} \), equation (19), (23), (26), (27), and \( \chi^T [M_s(q) - 2C_s(q, v)] X = 0 \), equation (31) is as

\[ V = s^T (M_s + C_s) s + \text{tr} \{ \hat{W}^T \Gamma^{-1} \hat{W} \} = -s^T \left( D_v + K_s \right) s + \left[ J^{-1} s \right]^T \left( \hat{W} \phi + \varepsilon + r_s - \alpha \right) + \text{tr} \{ \hat{W}^T \Gamma^{-1} \hat{W} \} \]

Due to equation (29), equation (32) is rewritten as (33), equation (34) and (35) are obtained as (34)

\[ V \leq -s^T (D_v + K_s) s + \varepsilon \| W \| \| W - \hat{W} \| + \left( J^{-1} s \right)^T \left[ \hat{W} \phi + \varepsilon + r_s - \alpha \right] \]

\[ V \leq -s^T (D_v + K_s) s + \varepsilon \| W \| \| W - \hat{W} \| + \left( J^{-1} s \right)^T \left[ \hat{W} \phi + \varepsilon + r_s - \alpha \right] \]

\[ = -s^T (D_v + K_s) s + \varepsilon \| W \| \| W - \hat{W} \| + \left( J^{-1} s \right)^T \left[ \hat{W} \phi + \varepsilon + r_s - \alpha \right] \]

\[ = -s^T (D_v + K_s) s + \varepsilon \| W \| \| W - \hat{W} \| + \left( J^{-1} s \right)^T \left[ \hat{W} \phi + \varepsilon + r_s - \alpha \right] \]

where \( D_{q,min} \) and \( K_{d,min} \) are the minimum eigenvalues, \( D_v \) and \( K_s \), \( \| \hat{W} \| \) denotes the Frobenius norm of \( \hat{W} \), and \( W_{max} \) is the maximum eigenvalue of \( W \). Therefore, we can obtain equation (36) as

\[ \left( D_{q,min} + K_{d,min} \right) \| \varepsilon \| + \varepsilon \| \hat{W} \| \| W_{max} - \| W \| \| + \| W \| = -\varepsilon \left( D_{q,min} + K_{d,min} \right) \| \hat{W} \| \| W_{max} - \| W \| \| + \| W \| = \frac{\varepsilon W_{max}}{4} + \left( D_{q,min} + K_{d,min} \right) \| \hat{W} \| \| W_{max} - \| W \| \| + \| W \| \]

If system is stable, equation (35) must be negative definite, equation (36) must be positive definite.

\[ \| \hat{W} \| > \frac{\varepsilon W_{max}}{4} \quad \text{or} \quad \| \hat{W} \| > W_{max} \]

From equation (37), it implies that \( V \) decreases as \( \| \varepsilon \| \) and \( \| \hat{W} \| \) converges toward the boundary and remains in its vicinity. By choosing arbitrarily large \( K_s \), small error can be obtained.

**Theorem 2:** If Assumption 1 is satisfied, while the platform dynamics FGRBF neural network approximation errors is \( \varepsilon = 0 \), the external disturbances becomes zero, the control input satisfies (38), and the adaptive control law of FGRBF neural network weight vector is defined as equation (39)

\[ \tau = \hat{f}(\dot{q}, v, q) + J^T K_s s \]

\[ \hat{\theta} = \Gamma \phi(J^{-1} s)^T \]

then the filtered tracking errors \( \hat{s}(t) \to 0 \), and the weight estimated errors \( \hat{\theta} \) of the closed-loop system are uniformly ultimately bounded stable. Theorem 2 can be concluded using the similar method.

**Theorem 3:** If Assumptions 1-2 are satisfied, the control input satisfies equation (38), and the adaptive control law of FGRBF weight satisfies equation (25), then tracking errors \( \hat{s}(t) \) and weight
estimated errors \( \hat{W} \) of the closed-loop system are uniformly ultimately bounded stable. Referencing to the proof of Theorem 1, time derivative of Lyapunov function is negative as long as equation (40) is satisfied.

\[
\| \hat{W} \| > \frac{\sum W_{m}^{2}/4 + (e_{\theta} + d_{\theta})}{(D_{\psi m} + K_{\psi m})} \quad \text{or} \quad \| \hat{W} \| > \frac{\sum W_{m}^{2}/4 + (e_{\theta} + d_{\theta})}{s}
\] (40)

By comparing equation (37) with equation (40), it is easy to find the steady-state tracking error decreased through adding the robustness control term or increasing control gain matrix.

5. Simulation Results and Conclusions

In figure 1, \( L = 0.4 \text{m} \), and \( \theta = 3 \). The initial iteration number \( T_{i} = 300 \). The learning rates \( \eta_{s} = 0.002 \) and \( \eta_{w} = 0.001 \). The linear and angular velocities \( v \leq 0.2 \text{m/s} \) and \( \omega \leq 0.5 \text{rad/s} \). The iteration termination error \( E = 0.01 \). The tracking time \( T_{s} = 350 \text{s} \). An elliptical trajectory is \( x^{2}/2 + y^{2}/1^{2} = 1 \) as figure 3(a).

(a)

(b)
Figure 3. Simulation results of elliptical trajectory, tracking errors of PD and Improved FGRBF controller

The initial condition of the improved FGRBF neural network is the same as the above simulation. The disturbance is decreased to zero, and the tracking errors of x and y directions can be obtained as shown in figure 3(c). The proposed controller is verified under zero disturbances.

In this paper, an improved FGRBF neural network controller is proposed for an omni-directional intelligent wheelchair. The proposed of the FGRBF neural network is used to approximate the actual omni-directional platform model. The control scheme can obtain significant trajectory tracking performance without the explicit prior mathematical model, and guarantees the control bounded in the closed-loop system simultaneously. The adaptive controller is robust and adaptive for nonlinear and uncertain system, which is verified by the simulation results through being trained and applied online.

6. Acknowledgments
The research work is financially supported by National Natural Science Foundation (61175087, 61105033) of China, General Aviation Platform Foundation of Hebei University of Science and Technology.

7. References
[1] Watanabe K, Tang J, Nakamura M, et al. A fuzzy-Gaussian neural network and its application to mobile robot control [J]. IEEE transactions on control systems technology, 1996, 4(2): 193-199.
[2] Sanchez M A, Castillo O, Castro J R. Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems[J]. Expert Systems with Applications, 2015, 42 (14): 5904 -14.
[3] Melin P, Astudillo L, Castillo O, et al. Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm [J]. Expert Systems with Applications, 2013, 40 (8): 3185- 95.
[4] Bingül Z, Karahan O. A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control [J]. Expert Systems with Applications, 2011, 38 (1): 1017-31.
[5] Mohareri O, Dhaouadi R, Rad A B. Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks [J]. Neurocomputing, 2012, 88: 54-66.

[6] Chen Y, Zheng W X, Xue A. A new result on stability analysis for stochastic neutral systems [J]. Automatica, 2010, 46 (12): 2100-04.

[7] Yen V T, Nan W Y, Van Cuong P. Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators [J]. Neural Computing and Applications, 2018: 1-14.

[8] Watanabe K, Tang J, Nakamura M, et al. Mobile robot control using fuzzy-Gaussian neural networks [C]// Intelligent Robots and Systems’93, IROS’93. Proceedings of the 1993 IEEE/RSJ International Conference on. IEEE, 1993, 2: 919-925.

[9] Du D, Li X, Fei M, et al. Self-learning fuzzy control strategy of two-layer networked learning control systems based on improved RBF neural network [C]// Control Conference (CCC), 2011 30th Chinese. IEEE, 2011: 4735-40.

[10] S. Zhang S, Shen Y, Sun Z. Mobile Robot Local Path Planning Based on Improved TS Fuzzy Neural Network [J]. 2015.

[11] Ying H. General SISO Takagi-Sugeno fuzzy systems with linear rule consequent are universal approximators [J]. IEEE Transactions on fuzzy systems, 1998, 6 (4): 582-587.

[12] Mon Y J, Lin C M. Supervisory fuzzy gaussian neural network design for mobile robot path control [J]. International Journal of Fuzzy Systems, 2013, 15 (2): 142-148.