Adaptive PID Control Based on RBF NN For Quadrotor

Hadi Hasanpour¹, Mohsen Heydari Beni², Masood Askari³

1. Department of Electrical Engineering, Khomeinishahr Branch, Islamic Azad University, Khomeinishahr/Isfahan, Iran.
2. Mechanical Engineering Department, Shahrekord University, Shahrekord, Iran.
3. Academic Center for Education Culture & Research IUT Branch.

Corresponding Author email: hadihasanpor@live.com

Abstract: An adaptive PID decoupling control strategy based on Radial Basis Function (RBF) neural network (NN) is presented in this paper. Quadrotor UAV is a kind of under actuated system with multiple inputs and strong coupling, manual optimization of PID control parameter for the quadrotor aircraft is time consuming, and it is difficult to achieve good control effect. Based on the theory of optimization in groups, the parameters such as proportion, integration and differentiation of PID controller are tuned on-line using the self-learning ability of RBFNN, and the corresponding decoupling control law is achieved by conventional PID control algorithm. Simulation results show that the dynamic decoupling and completely static decoupling are obtained, the closed loop system has the advantages of higher speed response and stronger robustness.

Keywords: Adaptive PID Control, RBF Neural Network, Quadrotor, Quadrotor Model Dynamic

INTRODUCTION

Quadrotor is a non-linear under-actuated system with strong coupling and multiple variables. At present, PID technology applies widely in Quadrotor control system for its characteristics of simple structure, mature technology, and easy realization of engineering in practice [1].

As the most UAV, the Quadrotor helicopter is an under actuated system who possesses six degrees of freedom versus four control inputs only. To solve the Quadrotor UAV tracking control problem many techniques have been proposed as in [2-7], where the control objective is to control the three cartesian positions and the yaw motion by using the four controls inputs and the outputs that remain.

Conventional PID control has advantages of simple structure, adjustable parameters, etc., but it is excessively dependent on accurate model in complex quadrotor system with large disturbance, highly non linearity, and uncertainty, so the expected control effects cannot be realized in general. Neural networks can solve non-linear and time varying problems effectively. For recent years, there have been many control approaches based on neural networks, but many of them have disadvantages of slow learning rate and local minimum. RBFNN is an effective multi layer feed forward network. Because of linear relationship of output weights existing in structure of RBFNN, local optimization wouldn’t appear. The ability of solving nonlinearity about this network has been verified [8]. The adjustment for the node centers of hidden layer and the estimation of connection weights are two difficulties in establishment of the RBFNN. Adopting optimization algorithms reasonably to conquering these difficulties is the key to improve the performance of this network. Many efficient learning algorithms have been proposed and applied widely along with in-depth investigations [9-12].

In order to improve control performance, the adaptive PID decoupling control based on RBFNN is proposed in this paper. The parameters of PID controller are tuned and modified automatically on-line using self-learning ability of RBFNN.

This paper proposed to optimize the initial values of parameters in RBF neural network with Gradient Descent algorithm by taking the characteristic that parameter initial value of PID controller in RBF neural network has large influence on controller performance. Finally, simulation experiment was conducted to the control plan proposed in this paper, Simulation and test results verify that quadrotor PID control in RBF neural network plan based is better than traditional PID control in aspects of control performance, reaction speed, and adaptive ability.
**Quadrotor Dynamic**

The conventional structure of quadrotor consist of four propellers in cross configuration where the pairs of rotors (1,3) and (2,4), turn in opposite directions in order to prevent the device from turning on that Figure 1.

Vertical flight ascending (descending) is obtained by the increase (decrease) of the Thrust forces to getting a difference from the weight of the Quadrotor. Pitching motion in a Quadrotor is the result of the speed difference between front and rear rotors, hence a rotation around the Y axis coupled with a translation along the X axis are produced. The same analogy is applied to obtain the roll motion, but this time by changing the side motors speed. The yaw motion is obtained while increasing (decreasing) speed of motors (1.3) compared with that motors (2.4). Unlike pitch and roll motions, yaw rotation is the result of reactive torques produced by the rotation of the rotors.

![Model of a Typical Quadrotor](image)

Figure 1: Model of a Typical Quadrotor

Kinematics of a 6 degree of freedom rigid body is given by:

\[
\dot{\xi} = J_0 v \tag{1}
\]

Where, \( \dot{\xi} \) is the generalized velocity vector with respect to earth frame and \( v \) is the generalized velocity vector with respect to body frame, \( J_0 \) is the generalized matrix that transfers velocities of the body frame to earth frame.

Dynamics of the quadrotor is given by the following Newton-Euler equation [13].

\[
\begin{bmatrix}
    m & 0_{3 \times 3} \\
    0_{3 \times 3} & I
\end{bmatrix}
\begin{bmatrix}
    \dot{\psi} \\
    \omega^B \times (mV^B)
\end{bmatrix}
+ \begin{bmatrix}
    \omega^B \times (\psi^B) \\
    \omega^B \times (I \omega^B)
\end{bmatrix}
= [F^B] \tag{2}
\]

Where \( m \) is the mass of the quadrotor, \( I \) is the inertia tensor, \( I_{3 \times 3} \) is the \( 3 \times 3 \) unity matrix, \( \omega^B \) and \( V^B \) are the angular and linear velocities of the quadrotor with respect to the body frame and \( F^B \) and \( \tau^B \) are the force and torque vectors. Following two assumptions are made to simplify the body dynamics [14, 15]. Centre of mass of the quadrotor coincide with the origin of the body fixed frame. Axes of the body frame coincide with the body principal axes of inertia.

With the above assumptions the accelerations in the body frame can be derived as follows.

\[
\begin{align*}
\dot{u} &= (v r - w q) + g \sin \theta \\
\dot{v} &= (w p - u r) - g \cos \theta \sin \phi \\
\dot{w} &= (u q - v p) - g \cos \theta \sin \phi + \frac{U_1}{m} \\
\dot{p} &= \frac{l_{yy} - l_{zz}}{l_{xx}} qr - \frac{l_{rp}}{l_{xx}} q\Omega + \frac{U_2}{l_{xx}} \\
\dot{q} &= \frac{l_{zz} - l_{xx}}{l_{yy}} pr - \frac{l_{rp}}{l_{yy}} p\Omega + \frac{U_3}{l_{yy}} \\
\dot{r} &= \frac{l_{xx} - l_{yy}}{l_{zz}} pq + \frac{U_4}{l_{zz}} 
\end{align*}
\tag{3}
\]

Where \( \Omega \) is the algebraic sum of the propeller speeds and \( \Omega \) is defined as:

\[
\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \tag{4}
\]
Where $\Omega_1$ (rad s$^{-1}$) is the front propeller speed, $\Omega_2$ (rad s$^{-1}$) is the right propeller speed, $\Omega_3$ (rad s$^{-1}$) is the rear propeller speed. $\Omega_4$ (rad s$^{-1}$) is the left propeller speed.

Equation (5) shows the relationship between the basic movements and the thrusts and torques generated by each propeller.

$$
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix} =
\begin{bmatrix}
T_1 + T_2 + T_3 + T_4 \\
-T_2 + T_4 \\
-T_1 + T_3 \\
-T_1 + T_2 + T_3 + T_4
\end{bmatrix}
$$

(5)

It is desirable to control the position of the quadrotor with respect to the earth frame and orientation with respect to the body frame. Therefore a hybrid frame is defined containing position in the earth frame and orientation in the body frame [14, 16, 17]. Velocity vector of the hybrid frame is given as:

$$
\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ p \\ q \\ r \end{bmatrix}
$$

(6)

Using blade element theory and momentum theory it can be deduced that thrust and torque of a propeller are proportional to the square of the propeller speed. The constant of proportionality, thrust coefficient and the torque coefficient for the propellers were estimated [17-19]. The relationship between propeller speeds and the motion is given in equation (8), where $b$ and $d$ are thrust coefficient and the torque coefficient.

$$
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix} =
\begin{bmatrix}
b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\
-b(\Omega_1^2 + \Omega_4^2) \\
-b(\Omega_1^2 + \Omega_3^2) \\
-d(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)
\end{bmatrix}
$$

(8)

**Control Modelling**

It is possible to determine the quadrotor position by double integrating its accelerations (linear and angular). To do this operation, just the internal state and the four motor voltages must be managed. This process is also known as direct kinematics and direct dynamics [20, 21].

The goal of the quadrotor stabilization is to find those value of the motors voltage which maintains the helicopter in a certain position required in the task. This process is also known as inverse kinematics and inverse dynamics. Unlike the direct ones, the inverses operations are not always possible and not always unique. For these reasons their consideration is much more complicated.

The quadrotor dynamics must be simplified to provide an easy inverse model which can be implemented in the control algorithms. Following assumptions were used to simplify the dynamic model of the quadrotor [16, 22, 23]. The cross coupling effects of angular speeds (Coriolis centripetal effect) and gyroscopic effects are negligible. In hovering conditions the accelerations in the body frame are approximately equal to the accelerations in the earth frame.
With these assumptions quadrotor dynamics can be simplified as follows.

\[
\begin{align*}
\ddot{z} &= -g + (\cos \theta \cos \phi) \frac{U_1}{m} \\
\ddot{\phi} &= \frac{U_2}{I_{xx}} \\
\ddot{\theta} &= \frac{U_3}{I_{yy}} \\
\ddot{\psi} &= \frac{U_4}{I_{zz}}
\end{align*}
\] (9)

The role of the controller is to adjust the speed of four propellers such that a desired quadrotor orientation is achieved [13, 16, 24]. The relationship between the basic movements of the quadrotor and the propeller speeds can be derived by equation (10). Where \( l \) is the distance between opposite rotors

\[
\begin{align*}
\Omega_1^2 &= \frac{1}{4b} U_1 - \frac{1}{2bl} U_3 - \frac{1}{4d} U_4 \\
\Omega_2^2 &= \frac{1}{4b} U_1 - \frac{1}{2bl} U_2 + \frac{1}{4d} U_4 \\
\Omega_3^2 &= \frac{1}{4b} U_1 + \frac{1}{2bl} U_3 - \frac{1}{4d} U_4 \\
\Omega_4^2 &= \frac{1}{4b} U_1 + \frac{1}{2bl} U_3 + \frac{1}{4d} U_4
\end{align*}
\] (10)

From the altitude equation defined in equation (9):

\[ \ddot{z} = \frac{U_1}{m} (\cos \phi \cos \theta) - g \] (11)

The control input \( U_1 \) can be derived as:

\[ U_1 = \frac{m}{\cos \phi \cos \theta} (P_z + g) \] (12)

The necessary condition of equation (12) is \( \cos \phi \cos \theta \neq 0 \). \( P_z \) is selected as a PID controller:

\[ P_z = K_{zp} (z_d - z) + K_{zl} \sum_{i=1}^{k} t_s (z_{di} - z_i) + K_{zd} (\dot{z}_d - \dot{z}) \] (13)

Where \( K_{zp}, K_{zl} \), and \( K_{zd} \) are the proportional, integral and differential coefficients, respectively; \( k \) represents the kth iteration; and \( t_s \) is the time step.

Taking \( U_1 \) into position equation of x-direction and y-direction defined in equation (7), and adopting the yaw angle approximate to 0, the dynamics can be obtained as:

\[ \begin{align*}
P_x &= (P_z + g) \tan \theta \\
P_y &= -(P_z + g) \tan \phi
\end{align*} \] (14)

The PID controller of x-direction and y-direction can be designed as:

\[ \begin{align*}
P_x &= K_{xp} (x_d - x) + K_{xl} \sum_{i=1}^{k} t_s (x_{di} - x_i) + K_{xd} (\dot{x}_d - \dot{x}) \\
P_y &= K_{yp} (y_d - y) + K_{yl} \sum_{i=1}^{k} t_s (y_{di} - y_i) + K_{yd} (\dot{y}_d - \dot{y})
\end{align*} \] (16)

Reversely solving equations (14) and (15), the desired attitude angles \( \phi_d \) and \( \theta_d \) can be derived as:

\[ \phi_d = -\arctan \left( \frac{P_y}{P_z + g} \right) \] (18)
\[ \theta_d = \arctan \left( \frac{P_x}{P_z + g} \right) \]  

(19)

**Control Strategy**

The controller is nested by inner loop and outer loop, as shown in Figure 2.

![Figure 2: Structure of controller](image)

The intelligent fuzzy logic PID control approach is used in the inner loop in order to achieve attitude control, while the PID control algorithm is implemented in the outer loop for the altitude and position control. It can be known from the flight characteristic of quadrotor that the position movement in outer loop is determined by the attitude angle in inner loop. Attitude controller outputs \( U_i \) according to desired altitude \((z_d)\) and current altitude \((z)\). Position controller combines the desired position \((x_d, y_d)\) and current position \((x, y)\), outputting desired roll angle \((\phi_d)\) and desired pitch angle \((\theta_d)\) into attitude controller, and then, attitude controller outputs \(U_2, U_3\) and \(U_4\) to quadrotor model. Finally, the model integrates the disturbance from wind field, outputs the state of next time step and feeds back to all controllers [25].

**PID Controller**

The transfer function of a PID controller has the following form:

\[ G_c(s) = K_p + \frac{K_i}{s} + K_D s \]  

(20)

Where \( K_p \), \( K_i \), and \( K_D \) are the proportional, integral, and derivative gains, respectively. Another useful equivalent form of the PID controller is:

\[ G_c(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{1}{T_d s}\right) \]  

(21)

where \( T_i = K_p/K_i \) and \( T_d = K_p/K_D \). \( T_i \) and \( T_d \) are known as the integral and derivative time constants, respectively. The discrete-time equivalent expression for PID control used in this paper is given as:

\[ u(k) = K_p e(k) + K_i T_s \sum_{i=1}^{n} e(i) + \frac{K_D}{T_s} \Delta e(k) \]  

(22)

Here, \( \Delta e(k) = e(k) - e(k-1) \) and \( e(k) \) is the error between the reference and the process output, \( u(k) \) is the control signal and \( T_s \) is the sampling period for the controller.

The parameters of the PID controller \( K_p \), \( K_i \), and \( K_D \) or \( K_p \), \( T_i \) and \( T_d \) can be manipulated to produce various response curves from a given process. [14, 16, 26].

**RBF Neural Network**

RBFNN is a three layer feed forward neural network. The mapping from input to output is nonlinear, but from hidden layer to output layer is linear. Learning rate is quickened greatly and the problem of local minimum is avoided. A typical RBFNN configuration is shown in Figure 3.
From general block diagram of radial basis function neural network, the inputs are denoted as \( X = [x_1, x_2, ..., x_n]^T \) in vector form. Here is a vector \( H = [h_1, h_2, ..., h_m]^T \) with \( m \) elements in the hidden layer. The \( h_m \) is named the radial basic function [27].

\[
    h_j = \exp\left(-\frac{\|x - c_j\|^2}{2b_j^2}\right), \quad j = 1, 2, ..., m
\]  

(23)

Where the operator \( \| \| \) represents a \( p \)-norm, also known as Euclidean 2-norm, and the vector \( C_j = [c_{j1}, c_{j2}, ..., c_{jm}]^T \) is the node center of basic function and \( b_j \) its radius accordingly. This is known as Gaussian radial basic function. The output layer is the manufactured output \( \tilde{y} \). Therefore, the network output \( \tilde{y} \) can be written as follows:

\[
    \tilde{y} = \sum_{j=1}^{m} w_j h_j
\]  

(24)

where real number parameters \( w_j \), \( \forall j=1,2,....,m \), are the weights. The error \( (y(k) - \tilde{y}(k)) \) between system output response \( y(k) \) and RBF output \( \tilde{y}(k) \) is used to regulate the networks parameters. Moreover, this neural network effectiveness is evaluated by a performance function defined as the squared estimation error:

\[
    J_I = \frac{1}{2}|y(k) - \tilde{y}(k)|^2
\]  

(25)

**Design of Adaptive PID Control Based on RBFNN**

Combining RBFNN with conventional PID controller can form a hybrid control strategy, and its structure is shown in Figure 4. RBFNN is used to tune parameters of the conventional PID Controller to keep system stable, and it is adjusted according to the error between the input \( r_i(k) \) and output \( y_i(k) \) in its loop. The aim of this hybrid strategy is to make the conventional PID controller more adaptive. If there are disturbances, the adaptive PID controller based on RBFNN will take effect more successfully. This approach not only ensures the stability and robustness of the system but also improves the tracking performance. So the proposed hybrid strategy is used to control multi input and multi-output (MIMO) coupling system [28-30].

The gradient descent method is then applied with the updating algorithms to get output weight, node center and radius parameter stated as follows [31]:

\[
    w_j(k) = w_j(k-1) + \eta[y(k) - \tilde{y}(k)]h_j + \gamma[w_j(k-1) - w_j(k-2)]
\]  

(26)
\[ \Delta \sigma_j = \frac{\|x-c_j\|^2}{\sigma_j^2} \]  
\[ \Delta \sigma_j(k) = \sigma_j(k-1) + \eta \Delta \sigma_j + \gamma [\sigma_j(k-1) - \sigma_j(k-2)] \]  
\[ \Delta \sigma_{ji} = \frac{[y(k) - \bar{y}(k)]^T x_j - c_{ji}}{\sigma_j^2} \]  
\[ c_{ji}(k) = c_{ji}(k-1) + \eta \Delta c_{ji} + \gamma [c_{ji}(k-1) - c_{ji}(k-2)] \]

\( \gamma \) is the momentum factor, and \( \eta \) is the learning rate of the neural network. This updating algorithm includes the learning capability. The PID controller parameters \( k_p, k_d, k_i \) are regulated by the RBF-PID controller is given as follows; the system error \( e(k) \) for the unit negative feedback control system can be written as:

\[ e(k) = r(k) - y(k) \]  

Where \( r(k) \) is the reference command. Furthermore, the efficiency of this adaptive controller is estimated by a performance function defined as the squared error:

\[ E(k) = \frac{1}{2} e^2(k) \]

The adaptive controller output \( u(k) \) can be represented in the updating algorithm as:

\[ u(k) = u(k-1) - \Delta u(k) = u(k-1) + k_p e_p(k) + k_i e_i(k) + k_d e_d(k) \]  

where \( k_p, k_d \) and \( k_i \) are proportional gain, derivative gain, and integral gain respectively. The \( e_p(k), e_i(k) \) and \( e_d(k) \) are defined as proportional error function, integral error function and derivative error function respectively as follows:

\[ e_p(k) = e(k) - e(k-1) \]  
\[ e_d(k) = e(k) - 2e(k-1) + e(k-2) \]  
\[ e_i(k) = e(k) \]

The gradient descent method is applied with the chain rule to infer the regulating rules for the \( k_p, k_d \) and \( k_i \) to minimize the performance index function \( E(k) \) as follows:

\[ \Delta k_p = -\mu \frac{\partial E}{\partial k_p} = -\mu \frac{\partial E}{\partial y} \frac{\partial u}{\partial k_p} = \mu e(k) \frac{\partial y}{\partial u} e_p(k) \]  
\[ \Delta k_i = -\mu \frac{\partial E}{\partial k_i} = -\mu \frac{\partial E}{\partial y} \frac{\partial u}{\partial k_i} = \mu e(k) \frac{\partial y}{\partial u} e_i(k) \]  
\[ \Delta k_d = -\mu \frac{\partial E}{\partial k_d} = -\mu \frac{\partial E}{\partial y} \frac{\partial u}{\partial k_d} = \mu e(k) \frac{\partial y}{\partial u} e_d(k) \]  

Where, \( \frac{\partial y}{\partial u} \) is Jacobean matrix denoting the sensitivity relating system output \( y(k) \) to controller output \( u(k) \) and \( \mu \) is the learning rate for the adaptive PID algorithm. The inputs, output response are three inputs in the RBF algorithm, output response \( x_1 = y(k) \), delayed output response \( x_2 = y(k-1) \), and controller output \( x_3 = u(k) \). The
manufactured output $\bar{y}(k)$ of RBF will approach the system output $y(k)$ after online learning. Therefore $\frac{\partial y(k)}{\partial u(k)}$, is very close to $\frac{\partial \bar{y}(k)}{\partial \bar{u}(k)}$ and one can conclude from (26) and (27),

$$\frac{\partial y(k)}{\partial u(k)} \approx \frac{\partial \bar{y}}{\partial \bar{u}} = \frac{\partial}{\partial x^3} \sum_{j=1}^{m} w_j e^{2\sigma_j} \left( - \frac{\|X - c_j\|^2}{2\sigma_j^2} \right) = \sum_{j=1}^{m} w_j h_j \frac{\partial}{\partial x^3} \left( - \frac{X^T X - X^T C_j - C_j^T X + C_j^T C_j}{2\sigma_j^2} \right)$$

Thus, the updating algorithm for the adaptive PID based on RBF can be derived as:

$$\Delta k_p = \mu e(k) e_p(k) \sum_{j=1}^{m} w_j h_j \frac{c_{j3} - u(k)}{\sigma_j^2}$$

$$\Delta k_i = \mu e(k) e_i(k) \sum_{j=1}^{m} w_j h_j \frac{c_{j3} - u(k)}{\sigma_j^2}$$

$$\Delta k_d = \mu e(k) e_d(k) \sum_{j=1}^{m} w_j h_j \frac{c_{j3} - u(k)}{\sigma_j^2}$$

The PID parameters $k_p$, $k_i$ and $k_d$ are automatically readjusted by online learning algorithm of RBF and (41-43) to keep the system error $e(k)$ zero.

**Simulation Experiment and Results**

The simulation adopts quadrotor system. Proposed algorithm and controller are adopted in the simulation. The RBFNN adopts a 3-5-1 structure, center parameter of hidden nodes is allocated $c_{ij} \in [-4, 4]$, $i = 1, 2, 3, j = 1, 2, 3, 4, 5$ the width parameter of Gauss Basis Function is $\sigma_j \in [0, 4]$, $j = 1, 2, 3, 4, 5$ and the connection weight of j th hidden node to output node is $w_j \in [0, 1]$, $j = 1, 2, 3, 4, 5$. Then, the connection weights, the center parameters and the width parameters of Gauss Basis Function are trained on-line. Then the MIMO adaptive PID decoupling control based on RBFNN is applied to quadrotor system, and simulations are made to verify its effect. Learning rate $\eta = 0.1$, inertial coefficient $\alpha = 0.15$. The simulation time is 20 seconds, and sampling time is 0.01 second. The PID parameters $k_p$, $k_i$ and $k_d$ are adjusted by self-learning RBFNN until error approach zero. The simulation results are shown in Figure 5 and Figure 6. Jacobian output for Z is shown in Figure 7 and w variation in Figure 8. The simulation results show that outputs of the controller can match the output of the closed-loop controlled plant excellently. We can see that the system output track the reference input satisfactorily. So the adaptive PID decoupling control based on RBFNN has better dynamic characteristic and the self-learning ability.
CONCLUSIONS

A novel adaptive PID decoupling control strategy based on Radial Basis Function (RBF) Neural Network (NN) is presented in this paper. The Proposed decoupling controller has advantages of both self-learning ability of neural network and simplicity of PID controller. First, the PID controller was designed to control the system, then NN control was used as supervisory controller over the PID controller; a NN controller supervises the operation of the PID controller, by tuning the PID controller gains, and optimizing them according to the operating conditions of the system. It can control the system under the circumstance of unknown state perfectly and modify parameters of
the controller on-line. Results show that the proposed controller has the adaptability, strong robustness and satisfactory control performance in the complex and strong coupling system. The adaptive control strategy provides valuable reference for actual application.

**REFERENCES**


C. Balas, "Modelling and Linear Control of a Quadrotor," Cranfield University, 2007.

M. S. Shaikh, "Quadrocopter Fuzzy Flight Controller," Department of Technology at Örebro University.


