Nonlinear Identification of an Unmanned Quadcopter Rotor Dynamics using RBF Neural Networks

Paulin Kantue
Research and Development Group
Uav4africa (Pty) Ltd
Johannesburg, South Africa
kantuep@uav4africa.co.za

Jimoh Olarewaju Pedro
School of Mechanical, Industrial and Aeronautical Engineering
University of the Witwatersrand
Johannesburg, South Africa
Jimoh.Pedro@wits.ac.za

***Abstract*—The unmodelled rotor dynamics in accelerated flight have a negative effect in the robustness and performance of an unmanned quadcopter, which could result in mission failure in adverse conditions or rotor faults. The nonlinear identification of an unmanned quadcopter rotor dynamics is investigated in this paper. The rotor dynamics are considered in terms of a first-order flapping dynamic model with the dynamics estimated using the radial basis function (RBF) neural networks. A RBF structure based on a continuous forward algorithm (CFA) is implemented for the estimation of a longitudinal rotor flapping dynamic coefficient. This was achieved through optimal input design by the maximization of the spectral density function and predicting the resonant frequency response from the RBF output. This was computed at various trim speeds and training data noise levels and compared with a linear model. The prediction accuracy and robustness to noise of the CFA algorithm proved that the proposed approach can result in better understanding of quadcopter flapping dynamic for high fidelity flight controller design.**

***Keywords*—Nonlinear identification, quadcopter, Unmanned Systems, rotor dynamics, radial basis functions**

I. Introduction

The direct approach of using system identification for the mathematical representation of unknown system dynamics is well-known. This is mainly due to the difficulty of obtaining accurate and practical models for control-design applications from first-principles approaches [1]. Moreover, nonlinear system identification has been utilized when the mapping from the observed data (input-output) to a cost function minimization regressor matrix is achieved only through a nonlinear model structure [2], [3]. The nonlinear modelling approaches used to characterize the dynamics of quadcopters are employed due to the complex interaction of aerodynamic forces and moments generated through rotating propellers acting on the frame encasing a stabilizing feedback controller, which in turn, provides input to the same propellers [4].

Although mechanically simple (compared to helicopters) and low maintenance, quadcopter still exhibits highly-coupled dynamics. However, the modelling and identification efforts of quadcopters have mainly been near steady-state conditions such as hover and constant forward flight [5], [6]. The simplistic assumption of ignoring rotor dynamics due to flapping angles and changes in the tip-path-plane (TPP) is only valid provided the destabilizing transients during accelerated flights, are within the control bandwidth of the flight controller [7], [8] such as PID control [9].

In transitional flight, it has been shown that compensation against blade flapping dynamics can only be achieved through accurate modelling instead of further flight controller tuning (in this case increasing the integral gain) [10]. This oversight on the aerodynamics could prevent accurate and robust tracking which is the primary requirement in autonomous high-speed manoeuvres or acrobatic flight. The problem of singularities during a quadcopter vertical loop maneuver was investigated by [11]. Underactuation during such a maneuver was resolved using energy-based control which would not have been possible otherwise.

A black-box approach has been used specifically to identify a quadcopter longitudinal dynamics using radial basis function (RBF) neural networks trained with minimal resource allocating network MRAN [12]. However, conventional two-stage approach of selecting RBF centers and widths, then computing the linear output weights through matrix pseudo-inversion, such as the Orthogonal Least square (OLS) algorithm [13], has often resulted in local minima issues and slow convergence [14]. To speed up convergence and optimize the computation of weights, center and widths, the continuous forward algorithm (CFA) [15], hybrid forward algorithm (HFA) [16] and more recently the extended Newton algorithm (ENA) [14] have been developed. We have decided to make use of modified CFA algorithm which uses a maximization function for its line search algorithm. As far as we know, this is the first time this has been applied for the modelling of quadcopter dynamics.

The estimation of quadcopter rotor flapping dynamic using RBF neural modelling is investigated in this paper. In Section II, the quadcopter rotor dynamics are described. Section III describes the identification methodology followed. The CFA algorithm for RBF modelling is introduced in Section IV. Section V discusses the results obtained followed by conclusions and future work in Section VI.

978-1-5386-4444-7/18/$31.00 ©2018 IEEE

II. Quadcopter Rotor Dynamics

The equations of motion of a quadcopter (shown in Figure 1) can be expressed with respect to the body-fixed reference frame [17]:

$$m\dot{\mathbf{v}} + m(\bar{\omega} \times \mathbf{v}) = \mathbf{F} \tag{1}$$

$$\mathbf{I}\dot{\bar{\omega}} + (\bar{\omega} \times \mathbf{I}\bar{\omega}) = \mathbf{M} \tag{2}$$

where $\mathbf{v} = [u\ v\ w]^T$ and $\bar{\omega} = [p\ q\ r]^T$ are the vehicle velocities and angular rates in the body-fixed frame respectively. $\mathbf{F} = [X\ Y\ Z]^T$ is the vector of external forces on the vehicle center of gravity and $\mathbf{M} = [L\ M\ N]^T$ is the vector of external moments. $\mathbf{I} = [I_{xx}\ I_{yy}\ I_{zz}]$ are the mass moment of inertia measured using a bifilar pendulum method, m is the measured mass of the vehicle. The angular orientation of the aircraft is described by the Euler angles (roll, ϕ, pitch, θ, and yaw, ψ, respectively). The generalized thrust force of an individual rotor is given as [18]:

$$T_i = C_{T_i}\rho(\Omega_i R_i)^2 \pi R_r^2 \tag{3}$$

where $i = [1-4]$ represents the number of the rotors. C_{T_i} is the ith rotor thrust coefficient which can be expressed as:

$$C_{T_i} = \frac{a_r \sigma_r}{2}\left(\theta_0\left(\frac{1}{3} + \frac{\mu_{ri}^2}{2}\right) + \frac{\mu_{z_{ri}} - \lambda_{0i}}{2}\right) \tag{4}$$

whereby the inflow velocities for the ith rotor can be expressed:

$$\lambda_{0i} = \frac{C_{T_i}}{2\eta_w\sqrt{\mu_{r_i}^2 + (\lambda_{0i} - \mu_{z_{ri}})^2}} \tag{5}$$

$$\mu_{ri} = \frac{\sqrt{(u_i - u_{wind})^2 + (v_i - v_{wind})^2}}{\Omega_i R_r} \tag{6}$$

$$\mu_{z_{ri}} = \frac{w_i - w_{wind}}{\Omega_i R_r} \tag{7}$$

$$\sigma_r = \frac{2c_r}{\pi R_r} \tag{8}$$

where a_r is the lift-curve slope of the propeller, c_r is the

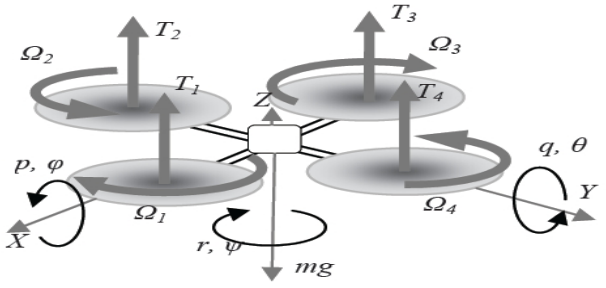

Fig. 1. Quadcopter Rotor forces and moments

root chord of the rotor blade (assuming constant chord), R_r is the rotor blade length, θ_0 is the pitch angle of the propeller (assuming no blade twist). The inflow velocity λ_{0i} is solved through an iterative approach. In the case of transitional flight, λ_{0i} will typically not have a steady-state value. At each ith rotor, the velocity components can be computed from body-fixed velocities u, v, w and angular rates p, q, r:

$$u_i = u + S_{u_i} L_{fr} r \sin\theta_{fr} \tag{9}$$

$$v_i = v + S_{v_i} L_{fr} r \cos\theta_{fr} \tag{10}$$

$$w_i = w + R_{x_i} q + R_{y_i} p \tag{11}$$

where L_{fr} is the length for airframe centre of gravity (COG) to rotor hub, $\bar{S}_u = [S_{u_1}, ..., S_{u_4}]$, similarly $\bar{R}_x$ and $\bar{R}_y$ are arrays of cross-product signs due to angular rates. θ_{fr} is the angular distance of each rotor to the airframe y axis. The rotor torque Q_r for the ith rotor can be computed:

$$Q_{r_i} = C_Q \rho(\Omega_i R_r)^2 \pi R_r^3 \tag{12}$$

where the torque coefficient is given:

$$C_Q = C_T(\lambda_{0i} - \mu_z) + \frac{C_{D_0}\sigma_r}{8}\left(1 + \frac{7}{3}\mu_{ri}^2\right) \tag{13}$$

Given that propeller aerodynamics are to solve numerically (given the span-wise twist/washout, taper from the root chord, vortex flow separation at the tips), wind tunnel data was used for a 10 by 4.5 inch propeller which is used on the H1 quadcopter [19]. This is shown in Figure 2. It is quite evident that propeller thrust decreases dramatically in forward flight (increased advance ratio due to negative pitch) above 15 m/s which coincides with more pronounced flapping dynamic which is not taken into consideration during controller design. Rotor-flapping dynamics was modelled by lumping each

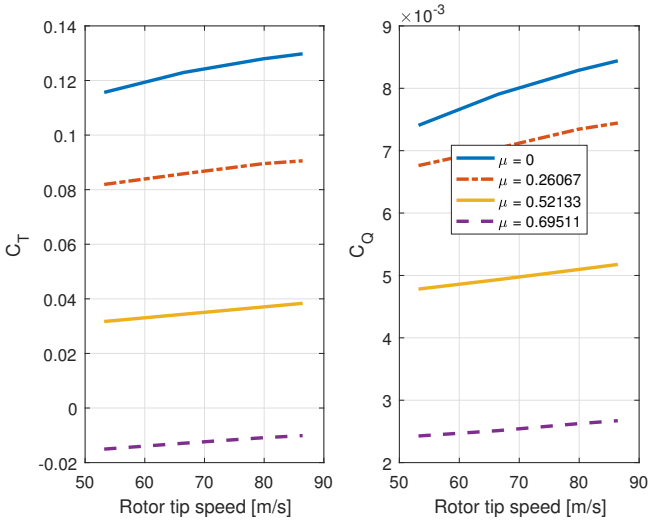

Fig. 2. Propeller Thrust/Torque coefficient - wind tunnel data

rotor and its flapping dynamic into tip-path plane dynamics described (excluding feathering hinge dynamics) [20], [21]:

$$\dot{b}_{1i} = -p - \frac{b_{1i}}{\tau_e} - \frac{1}{\tau_e}\frac{\delta b_{1i}}{\delta\mu_v}\frac{v_i - v_{wind}}{\Omega_i R_r} \tag{14}$$

$$\dot{a}_{1i} = -q - \frac{a_{1i}}{\tau_e} - \frac{1}{\tau_e}\left(\frac{\delta a_{1i}}{\delta\mu}\frac{u_i - u_{wind}}{\Omega_i R_r} + \frac{\delta a_{1i}}{\delta\mu_z}\frac{w_i - w_{wind}}{\Omega_i R_r}\right) \tag{15}$$

τ_e is the effective rotor time constant. The longitudinal dihedral derivative is given as:

$$\frac{\delta a_{1i}}{\delta \mu} = 2K_\mu \left(\frac{4\theta_0}{3} - \lambda_{0i} \right) \tag{16}$$

K_μ is the scaling coefficient to include the stabilizing effect. The longitudinal and lateral dihedral derivatives are equal in magnitude and both cause the rotor to flap away from the incoming air.

$$\frac{\delta b_{1i}}{\delta \mu_v} = -\frac{\delta a_{1i}}{\delta \mu} \tag{17}$$

The upward heave movement of the rotor causes a higher lift on the advancing blade which causes a moment on the rotor hub. The same stabilizer scaling coefficient is used:

$$\frac{\delta a_{1i}}{\delta \mu_z} = K_\mu \frac{16\mu_{ri}^2}{(1-\mu_{ri}^2/2)(8\left|\mu_{ri}\right| + a_r \sigma_r)} \tag{18}$$

Rotor flapping is the dominant effect on rotor moments.

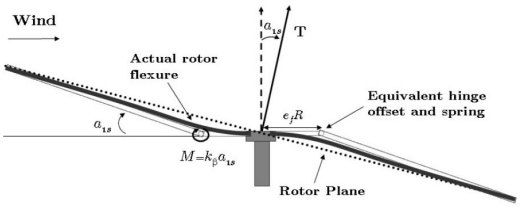

Fig. 3. Rotor flapping approximation [22]

The restraint is approximated using a linear torsional spring with constant stiffness coefficient K_β. This is illustrated in Figure 3. This results in a longitudinal (pitch) and lateral (roll) moments:

$$M_{k,lon} = K_\beta a_{1i} \tag{19}$$
$$L_{k,lat} = K_\beta b_{1i} \tag{20}$$

Once flapping occurs, the rotor thrust vector tilts and contributes to the body moments. Assuming the thrust vector tilts proportionally to the rotor flapping angles, the total rotor pitch and roll moments can be deduced as (N = 4 for a quadcopter):

$$L_r = \sum_{i=1}^{N} \left(K_\beta + Th_r\right) b_{1i} \tag{21}$$
$$M_r = \sum_{i=1}^{N} \left(K_\beta + Th_r\right) a_{1i} \tag{22}$$
$$N_r = \sum_{i=1}^{N} Q_i \tag{23}$$

where h_r is the distance between the rotor head and the center of gravity. The total rotor forces (not assuming small advance ratios) are given as:

$$X_r = -\sum_{i=1}^{N} T_i sin(a_{1i}) \tag{24}$$
$$Y_r = \sum_{i=1}^{N} T_i sin(b_{1i}) \tag{25}$$
$$Z_r = -\sum_{i=1}^{N} T_i cos(a_{1i}) cos(b_{1i}) \tag{26}$$

III. Rotor-flapping dynamic identification

A. Linear Model Structure

The linear model of the quadcopter was obtained through trimming the above model (including the rigid-body dynamics) at five trim conditions: hover, 5m/s, 8m/s, 12m/s, 17m/s and 20m/s. This resulted in the reduced linearised model given as:

$$\dot{u} = X_\theta \theta + X_q q + X_u u + \sum_{i=1}^{N} X_{a_{1i}} a_{1i} + X_{\delta_{lon}} \delta_{lon} \tag{27}$$
$$\dot{v} = Y_\theta \phi + Y_p p + \sum_{i=1}^{N} Y_{b_{1i}} b_{1i} + X_{\delta_{lat}} \delta_{lat} \tag{28}$$
$$\dot{w} = Z_\theta \theta + Z_q q + Z_w w + Z_{\delta_{mot}} \delta_{mot} \tag{29}$$
$$\dot{p} = L_p p + \sum_{i=1}^{N} L_{b_{1i}} b_{1i} + L_{\delta_{lat}} \delta_{lat} + L_{\delta_{rud}} \delta_{rud} \tag{30}$$
$$\dot{q} = M_q q + \sum_{i=1}^{N} M_{a_{1i}} a_{1i} + M_{\delta_{lon}} \delta_{lon} + M_{\delta_{mot}} \delta_{mot} \tag{31}$$
$$\dot{r} = N_r r + N_{\delta_{lat}} \delta_{lat} + N_{\delta_{rud}} \delta_{rud} \tag{32}$$
$$\dot{a_{1i}} = A_q^{1i} q + A_{a_{1i}}^{1i} a_{1i} + A_{\delta_{lon}}^{1i} \delta_{lon} \tag{33}$$
$$\dot{b_{1i}} = B_p^{1i} p + B_{b_{1i}}^{1i} b_{1i} + B_{\delta_{lat}}^{1i} \delta_{lat} \tag{34}$$

where N represents the number of rotors. The above equations form the basis of the neural network identification method described below and was arrived through zeroing coefficients below 1e-3. With specific focus on the rotor flapping dynamics, the pitch-rate response in the frequency domain can be described in terms of its natural frequency [23, pp. 302]:

$$\omega_n = \sqrt{-\sum_{i=1}^{N} M_{a_{1i}}} \tag{35}$$

The high values of flap stiffness (due to large hinge offset) which is the case for quadcopters, provides an increase in response frequency. Given maximization of the spectral density generated (defined in Section III-C), the associated frequency is assumed to be resonant frequency for a second order system defined as:

$$\omega_R = \omega_n \sqrt{1 - 2\zeta^2} \tag{36}$$

This then enables the computation of the rotor flapping dynamic (in this paper only longitudinal dynamics are observed and identified) based on the maximization of the signal spectral energy.

B. Model structure Determination

In order to avoid parameter estimation inaccuracies and unnecessary model complexity, the collinearity amongst dependent and independent variables must be quantified. The correlations between independent variables x_i and output variable y is computed as such [24, pp. 200]:

$$\rho_i = \frac{\sum_{k=1}^{N} (y(k) - \bar{y})(x_i(k) - \bar{x_i})}{\sqrt{\sum_{k=1}^{N} (y(k) - \bar{y})^2 \sum_{k=1}^{N} (x_i(k) - \bar{x_i})^2}} \tag{37}$$

Each signal is pre-conditioned prior through computing the signal mean and standard deviation. If both statistics are below 1e-3, the signal is removed for the data-set. Out of the remaining data-set, the four highest signals are stored for RBF training using the CFA algorithm.

C. Optimal Input Design

The main objective of flight testing for system identification is the formulation of the flight maneuvers that will minimize time and peak response while maximizing information content. This can only be achieved with a priori knowledge of the model structure and dynamic response [23, pp. 85]. Time domain maneuvers such as doublet, step and 3211 inputs can be analyzed using the power spectral function defined as [24, pp. 38]:

$$E(\omega) = 2\Delta t^2 \frac{1-cos\Omega}{\Omega^2} \times \left[\sum_{i=1}^{N} V_i^2 + 2\sum_{j=1}^{N} cosj\Omega \sum_{i=1}^{N} V_i V_{i+j}\right] \quad (38)$$

where N is the number of impulses with a time duration Δt and amplitude V and normalized frequency $\Omega = \omega\Delta t$. Figure 4 shows a typical input/output mapping for maneuver design. Figure 5 shows the normalized power spectral output as a function of step-size length. It can be noticed that the optimal step size is between 130-140 milliseconds.

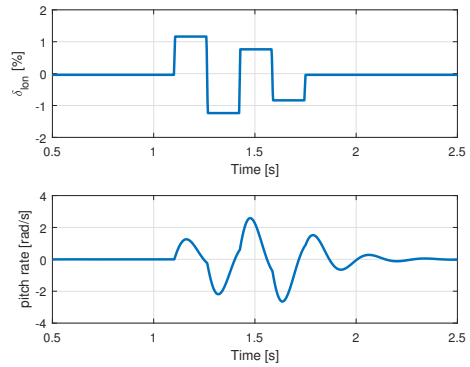

Fig. 4. Input/Output design for rotor dynamics identification

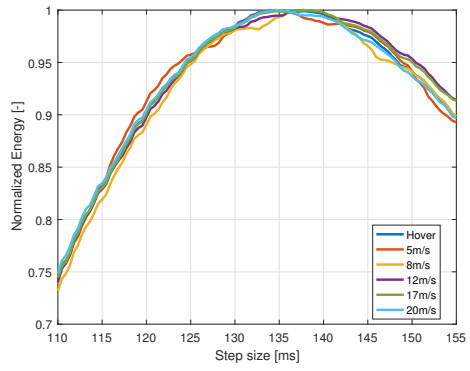

Fig. 5. Normalized power spectral function sensitivity with speed

IV. Neural Network Modelling

An analytical framework for both network construction and parameter optimization is at the centre of the continuous foward algorithm (CFA) developed by Peng et al [15]. Unlike the OLS and similar forward selection algorithm, the CFA algorithm optimizes the nonlinear parameters over the parameter space as the network architecture evolves. This results in an improved network modelling mechanism, as well as reduced memory storage and computational complexity. A multiple input-single-output (MISO) nonlinear RBF system with m hidden nodes is defined as:

$$\hat{\mathbf{y}} = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x}, \boldsymbol{\sigma}_i, \mathbf{c}_i) \quad (39)$$

where $\hat{y}$ is the network output, $\mathbf{x}$ is the input vector, $\phi_i(\mathbf{x}, \boldsymbol{\sigma}_i, \mathbf{c}_i)$ described as the nonlinear Gaussian function of the ith hidden node with the adjustable parameters of node center $\mathbf{c}_i$, node width $\boldsymbol{\sigma}_i$ and linear output weight w_i. The optimization of such parameters is achieved through the minimization of the sum squared error (SSE) defined as:

$$\mathbf{J}(\mathbf{w}, \boldsymbol{\sigma}, \mathbf{c}) = (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}) \quad (40)$$

where $\mathbf{y}$ is the output vector from the training set. Suppose the full regression matrix is represented by M candidates available for cost function minimization, there exists a subset k basis vectors such that the network outputs weight can be computed as:

$$\mathbf{w} = (\boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k)^{-1} \boldsymbol{\Phi}_k^T \mathbf{y} \quad (41)$$

where $\boldsymbol{\Phi}_k = [\phi_1, ..., \phi_k]$ represents regressor matrix subset required for cost function minimization. The cost function in (40) can then be expressed as:

$$\mathbf{J}(\boldsymbol{\Phi}_k) = \mathbf{y}^T \left[\mathbf{I} - \boldsymbol{\Phi}_k \left(\boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k\right)^{-1} \boldsymbol{\Phi}_k^T\right] \mathbf{y} \quad (42)$$

Further optimization will require the addition of a basis vector $\forall \phi \in (\phi_{k+1}, ..., \phi_M)$ into the subset regression matrix becoming $\boldsymbol{\Phi}_{k+1} = [\boldsymbol{\Phi}_k, \phi]$. Therefore the net reduction of the cost function due to the new regression matrix is given as:

$$\Delta \mathbf{J}_{k+1}(\phi) = \mathbf{J}(\boldsymbol{\Phi}_k) - \mathbf{J}([\boldsymbol{\Phi}_k, \phi]) \quad (43)$$

Based on [25], a residual matrix parameter can be defined:

$$\mathbf{R}_k = \begin{cases} \mathbf{I} - \boldsymbol{\Phi}_k \left(\boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k\right)^{-1} \boldsymbol{\Phi}_k^T, & 0 < k < M \\ \mathbf{I}, & k = 0 \end{cases} \quad (44)$$

such that a column vector ϕ and the output vector $\mathbf{y}$ are defined as:

$$\phi^{(k)} \triangleq \mathbf{R}_k \phi \quad \mathbf{y}^{(k)} \triangleq \mathbf{R}_k \mathbf{y} \quad (45)$$

where $\phi^{(0)} = \phi$ and $\mathbf{y}^{(0)} = \mathbf{y}$. The regressor and output parameters can then be recursively updated given as:

$$\phi^{(k)} = \phi^{(k-1)} - \frac{(\phi_k^{(k-1)})^T (\phi^{(k-1)})}{(\phi_k^{(k-1)})^T (\phi_k^{(k-1)})} \phi_k^{(k-1)} \quad (46)$$

whereby $\phi_k^{(k-1)}$ is the regressor output from the previous residual matrix computation. The properties of the residual matrix to enable the recursive calculation of hidden node regressors, is defined here [25]. Similarly the output of the kth basis vector can be computed in a similar fashion:

$$\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)} - \frac{(\phi_k^{(k-1)})^T (\mathbf{y}^{(k-1)})}{(\phi_k^{(k-1)})^T (\phi_k^{(k-1)})} \phi_k^{(k-1)} \quad (47)$$

The net cost function contribution can then be computed as:

$$\Delta \mathbf{J}_{k+1}(\phi) = \frac{[(\mathbf{y}^{(k)})^T \phi^{(k)}]^2}{(\phi^{(k)})^T \phi^{(k)}} \tag{48}$$

In order to simplify the implementation of (46) and (47), a $k \times M$ upper triangular matrix $\mathbf{A}$ can be introduced based on the k basis vectors used for cost function reduction:

$$\mathbf{A} \triangleq [a_{i,j}]_{k \times M} \tag{49}$$

$$a_{i,j} = \begin{cases} 0, & j < i \\ \phi_i^T \mathbf{R}_{i-1} \phi_i = (\phi_i^{(i-1)})^T (\phi_i^{(i-1)}), & j = i \\ \phi_i^T \mathbf{R}_{i-1} \phi_j = (\phi_i^{(i-1)})^T (\phi_j^{(i-1)}), & i < j < M \end{cases} \tag{50}$$

shown here in matrix format:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,M} \\ 0 & a_{2,2} & a_{2,3} & \cdots & a_{2,M} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & a_{k-1,M-k} & a_{k-1,M} \\ 0 & \cdots & 0 & 0 & a_{k,M} \end{bmatrix}$$

and $\mathbf{a}_y$ and $\mathbf{d}$ which are two $M \times 1$ vectors:

$$\mathbf{a}_y \triangleq [a_{i,y}]_{M \times 1}, \quad a_{i,y} = \mathbf{y}^T \mathbf{R}_k \phi_i = (\mathbf{y}^{(k)})^T \phi_i^{(k)} \tag{51}$$

$$\mathbf{d} \triangleq [d_i]_{M \times 1}, \quad d_i = \phi_i^T \mathbf{R}_k \phi_i = (\phi_i^{(k)})^T \phi_i^{(k)} \tag{52}$$

Furthermore, given that each basis vector is a function of width and centre in homogeneous space, the adjustable parameter of each hidden node can be grouped together such that:

$$\phi(\mathbf{x}(t), \boldsymbol{\sigma}, \mathbf{c}) = \Phi(\mathbf{x}(t), \omega) \tag{53}$$

where for $n \times 1$ input vector:

$$\omega = [\omega_0, \omega_1, ..., \omega_n] = [\boldsymbol{\sigma}, \mathbf{c}] \tag{54}$$

The net cost function contribution can then be computed as function of ω such that:

$$\Delta \mathbf{J}_{k+1}(\omega) = C^2(\omega)/D(\omega) \tag{55}$$

where for N training samples:

$$C(\omega) = (\mathbf{y}^{(k)})^T \phi^{(k)}(\omega) = \sum_{t=1}^{N} y^{(k)} \phi^{(k)}(\mathbf{x}(t), \omega) \tag{56}$$

$$D(\omega) = (\phi^{(k)}(\omega))^T \phi^{(k)}(\omega) = \sum_{t=1}^{N} (\phi^{(k)}(\mathbf{x}(t), \omega))^2 \tag{57}$$

The optimisation of the adjustable parameters results in the maximization of the net contribution to the cost function reduction for each $k+1$th hidden node added to the regression matrix. This is obtained through a conjugate gradient approach by obtaining the gradient of the cost function net contribution for each adjustable parameter ω. This is defined by differentiating (56) as given as:

$$\begin{aligned} &\frac{\partial(\Delta \mathbf{J}_{k+1}(\omega)}{\partial \omega_i} = \nabla \Delta \mathbf{J}_{k+1}(\omega) \\ =&\frac{2C(\omega)}{D(\omega)} \left(\mathbf{y}^{(k)} - \frac{C(\omega)}{D(\omega)} \phi^{(k)}(\omega) \right)^T \phi_{\omega_i}^{(k)}(\omega) \\ &i = 0, 1, ..., n \end{aligned} \tag{58}$$

where $\phi_{\omega_i}^{(k)}$ is the effect of the kth hidden node regressor due the adjustable parameter set ω_i along ith input vector element. This is defined as:

$$\begin{aligned} \phi_i^{(s)}(\omega) = \phi_i^{(s-1)}(\omega) - \frac{\partial a_{s,k+1}(\omega)}{\partial \omega_i} \frac{\phi_s^{(s-1)}}{a_{s,s}} \\ s = 1, ...k \quad i = 0, 1, ..., n \end{aligned} \tag{59}$$

where the gradient change of the elements in the $\mathbf{A}$ matrix can be defined as:

$$\frac{\partial a_{s,k+1}(\omega)}{\partial \omega_i} = (\phi_s^{(s-1)}(\omega))^T \phi_i^{(s-1)}(\omega) \tag{60}$$

The initialization of the algorithm at $k = 0$, the following can be established:

$$\begin{aligned} &\phi_{\omega_0}^{(0)}(\mathbf{x}(t), \omega) = -2\omega_0 \sum_{i=1}^{n} (x_i(t) - \omega_i)^2 \phi(\mathbf{x}(t), \omega) \\ &\phi_{\omega_i}^{(0)}(\mathbf{x}(t), \omega) = 2\omega_0^2 (x_i(t) - \omega_i) \phi(\mathbf{x}(t), \omega) \\ &i = 1, ..., n \end{aligned} \tag{61}$$

The initial values of the adjustable parameters are given as:

$$\omega_0^{(0)} = \left[\sum_{i=1}^{n} \sum_{t=1}^{N} (x_i(t) - \omega_i^{(0)})^2 / N \right]^{-1/2} \tag{62}$$

The conjugate gradient optimisation is then used to compute the $\omega_{k+1}^{(p)}$ adjustable parameter set that will be maximize $\Delta \mathbf{J}_{k+1}(\omega_{k+1}^{(p)})$. p is an iteration counter which is also used to adjust the gradient of the contribution $\nabla \Delta \mathbf{J}_{k+1}(\omega_{k+1}^{(p)})$ through a line search procedure. $\phi(\omega_{k+1}^{(0)})$ can then be computed along the elements of the triangular A matrix and then updated to $\phi(\omega_{k+1}^{(k)})$ for the $k+1$th hidden node according to (46). Similarly the contribution to the output vector of the $k+1$th hidden node can be computed according to (47). Finally, the reduction of the SSE can be computed:

$$SSE^{(k+1)} = SSE^{(k)} - \Delta \mathbf{J}_{k+1}(\omega_{k+1}^{(k)}) \tag{63}$$

The procedure is repeated until the net contribution falls below a certain threshold η (a value of 0.1 was used) or the SSE is below $1e-3$.

V. Results

The generation of identification data was obtained using a simulation model of the nonlinear quadcopter model, as described in Section II, which was developed using MATLAB S-functions and Simulink. Subsequently, the model structure determination for RBF nonlinear identification was performed using the correlation coefficients as described in Section III-B. The highest coefficients to the dependent variable were chosen for RBF training. Table I shows independent variables were chosen as a function of forward speed. It can be noticed that model structure becomes noise-independent above a certain forward speed. Given only one independent variable chosen at hover, it is very difficult to estimate flapping dynamic based on exciting pitch dynamics (this will be confirmed later).

TABLE I. RBF MODEL STRUCTURE SELECTION

Noise	Hover	5 m/s	12 m/s	20 m/s
Ideal	δ_{lon}	δ_{lon}, a_z	δ_{lon}, a_x, a_z	δ_{lon}, a_x, a_z
Low	δ_{lon}	δ_{lon}, a_z	δ_{lon}, a_x, a_z	δ_{lon}, a_x, a_z
High	δ_{lon}, w	δ_{lon}, a_z	δ_{lon}, a_x, a_z	δ_{lon}, a_x, a_z

The CFA algorithm as described in Section IV was used for training the RBF model structure. The number of neurons and resultant SSE once ΔJ_{k+1} had reached zero is also given (SSE is in brackets). This is shown in Table II. A maximum number of neurons of 4 was achieved while keeping the SSE low, even with reduced signal-to-noise ratio.

TABLE II. RBF TRAINING RESULTS - NEURONS(SSE)

Noise	Hover	5 m/s	8 m/s	12 m/s	17 m/s	20 m/s
Ideal	5 (7.43)	3 (0.59)	3 (0.47)	3 (0.56)	3 (0.36)	3 (0.55)
Low	3 (7.41)	3 (1.63)	4 (1.21)	3 (1.09)	3 (0.98)	3 (0.99)
High	2 (7.89)	3 (3.42)	3 (2.78)	3 (2.31)	3 (2.63)	4 (1.59)

Figure 6 shows the prediction accuracy of the RBF model as a function of forward speed and noise levels. It can be seen that a hover and in a high noise environment (bad sensors), and only one dependent variable for training, this produce really inaccurate results. However, at high forward speeds, the RBF prediction seems robust to noise levels while maintaining prediction accuracy.

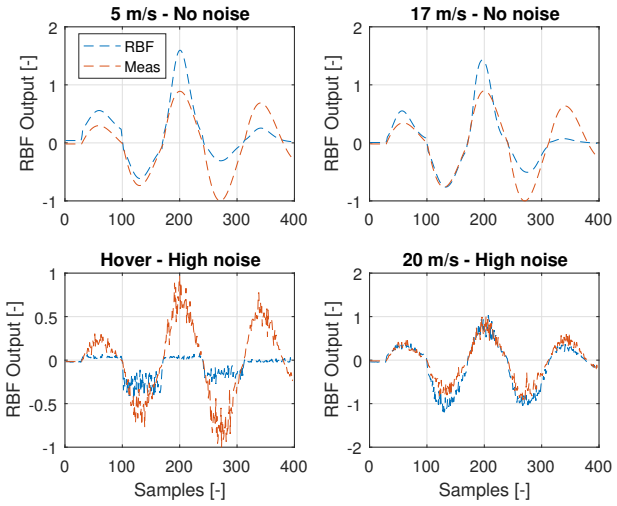

Fig. 6. RBF prediction accuracy with various noise levels/forward speed

The longitudinal rotor flapping dynamic coefficient M_{a_1} was computed as per Section III-A, which was based on the predicted outputs of the RBF model as a function of forward speed and noise levels. It is evident that the CFA algorithm combined with the optimization of the inputs design, achieved good results irrespective of noise levels. Although the damping ratio could be computed from the output signal, it was assumed at 0.261 (from the linear model).

VI. CONCLUSION

The nonlinear identification of unmanned quadcopter rotor dynamics was presented using CFA for RBF neural modelling. The combination of signal input design optimization using the spectral density parameter, the RBF model structure selection using the collinearity coefficient, and the prediction accuracy of the CFA algorithm, has proven an effective strategy for computing rotor flapping dynamic. The success of such a method depends on the forward speed of quadcopter, which is shown to be above 5 m/s. These results will aid in the future research of real-time identification of rotor dynamics using RBF neural networks and developing a possible framework for quadrotor neural fault detection and diagnosis.

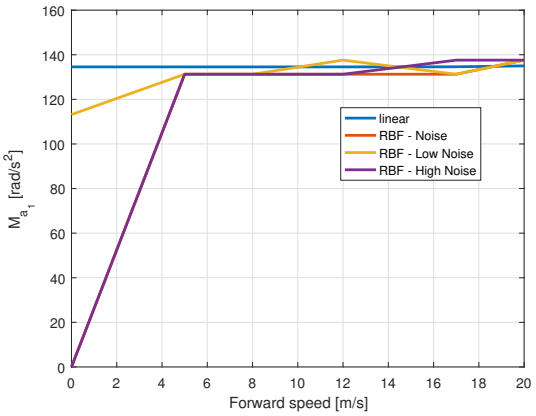

Fig. 7. Longitudinal rotor flapping dynamic coefficient with forward speed

REFERENCES

[1] B. Mettler, *Identification modeling and characteristics of miniature rotorcraft*, Springer, Ed. Springer Science & Business Media, 2013.

[2] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.-y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear Black-Box Modeling in System Identification : a Unified Overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[3] L. Ljung, *System Identification Theory for User*. Prentice Hall, 1999.

[4] A. Chovancová, T. Fico, E. Chovanec, and P. Hubinský, "Mathematical modelling and parameter identification of quadrotor (a survey)," in *Procedia Engineering*, vol. 96, 2014, pp. 172–181.

[5] I. M. Salameh, E. M. Ammar, and T. A. Tutunji, "Identification of quadcopter hovering using experimental data," *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2015*, pp. 3–8, 2015.

[6] X. Zhang, X. Li, K. Wang, and Y. Lu, "A survey of modelling and identification of quadrotor robot," *Abstract and Applied Analysis*, vol. 2014, 2014.

[7] V. Martinez, "Modelling of the flight dynamics of a quadrotor helicopter (Msc. Dissertation)," *Cranfield University*, pp. 1–289, 2007.

[8] H. Jin, X. Li, X. Zhang, and Y. Li, "Study on the Modeling Methods of the Quad-rotor Unmanned Helicopter," in *Control and Decision Conference*, no. 30, 2015, pp. 4855–4859.

[9] J. Li and Y. Li, "Dynamic analysis and PID control for a quadrotor," in *Mechatronics and Automation (ICMA), 2011 International Conference on*. IEEE, 2011, pp. 573–578.

[10] H. Huang, G. Hoffmann, S. Waslander, and C. Tomlin, "Aerodyamics and Control of Automous Quadrotor Helicopters in Aggressive Maneuvering," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 3277–3282.

[11] A. A. El-Badawy and M. A. Bakr, "Quadcopter Aggressive Maneuvers along Singular Configurations: An Energy-Quaternion Based Approach," *Journal of Control Science and Engineering*, vol. 4, 2016.

[12] M. F. Pairan and S. S. Shamsudin, "System identification of an unmanned quadcopter system using MRAN neural," *IOP Conference Series: Materials Science and Engineering*, vol. 270, pp. 1–10, 2017.

[13] S. Chen and S. a. Billings, "Recursive prediction error parameter estimator for non-linear models," *International Journal of Control*, vol. 49, no. 2, pp. 569–594, 1989.

[14] L. Zhang, K. Li, and E.-W. Bai, "A New Extension of Newton Algorithm for Nonlinear System Modelling Using RBF Neural Networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2929–2933, 2013.

[15] J.-x. Peng, K. Li, and G. W. Irwin, "A Novel Continuous Forward Algorithm for RBF Neural Modelling," *IEEE Transactions on Automatic Control*, vol. 52, no. 1, pp. 117–122, 2007.

[16] J. X. Peng, K. Li, and D. S. Huang, "A hybrid forward algorithm for RBF neural network construction," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1439–1451, 2006.

[17] B. Mettler and T. Kanade, "System Identification Modeling of a Model-Scale Helicopter," *Robotics*, vol. 17, no. 6, pp. 1–25, 2000.

[18] V. Gavrilets, B. Mettler, and E. Feron, "Human-inspired control logic for automated maneuvering of miniature helicopter," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, pp. 752–759, 2004.

[19] J. B. Brandt and M. S. Selig, "Propeller Performance at Low Reynolds Numbers," in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011, p. 1255.

[20] V. Gavrilets, "Autonomous Aerobatic Maneuvering of Miniature Helicopters," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[21] J. O. Pedro and P. Kantue, "Online aerodynamic parameter estimation of a miniature unmanned helicopter using radial basis function neural networks," in *ASCC 2011 - 8th Asian Control Conference*, 2011, pp. 1170–1175.

[22] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision Flight Control for A Multi-Vehicle Quadrotor Helicopter Testbed," *Control engineering practice*, vol. 19, no. 9, pp. 1023–1036, 2011.

[23] M. B. Tischler, "System identification methods for aircraft flight control development and validation," NASA, Tech. Rep. August, 1995.

[24] R. Jategaonkar, "Aerodynamic Modeling and System Identification From Flight Data Recent Applications At DLR," pp. 14–16, 2008.

[25] K. Li, J.-x. Peng, and G. W. Irwin, "A fast nonlinear model identification method," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1211–1216, 2005.