



ISSN NO. 2320-5407

Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/1242
DOI URL: <http://dx.doi.org/10.21474/IJAR01/1242>



INTERNATIONAL JOURNAL OF
ADVANCED RESEARCH (IJAR)
ISSN 2320-5407
Journal homepage: <http://www.journalijar.com>
Journal DOI:10.21474/IJAR01

RESEARCH ARTICLE

SYSTEM IDENTIFICATION OF UNMANNED HELICOPTER.

*Pragadeesh Suresh Babu, and SabareishPandian.

Department of Instrumentation and Control Engineering, National Institute of Technology, Trichy.

Manuscript Info

Manuscript History

Received: 12 June 2016
Final Accepted: 19 July 2016
Published: August 2016

Key words:-

Unmanned Helicopter, radial basis function, neural networks, particle swarm optimization.

Abstract

In this paper, we have developed a dynamic model to replicate the motions a helicopter. Unmanned helicopters have more advantage over manned as they are perfect for performing missions autonomously, which is vital to understand a dynamic model to enable the development of control and state estimation algorithms. This article presents a comprehensive method for identification of a nonlinear model of the unmanned helicopter. The accuracy of the developed model is verified by the comparison between predicted and actual responses from the model and the flight experiments, and between key parameters and the theoretical values. This report describes the process and results of the dynamic modeling of model-scale unmanned helicopter using system identification.

Copy Right, IJAR, 2016., All rights reserved.

Introduction:-

Helicopters are highly regarded because they are easily maneuverable and are able to hover, making them perfect for flying in closed environments while streaming data to the ground. Autonomous flight is necessary for these missions because of the fast and unstable dynamics of the aircraft, which are exacerbated by any transmission delays of sensor data and control inputs. The research and development of UAVs have gained much attention in academic and industrial fields. Researchers have explored the considerable potential of UAVs for various military and civilian applications. For successful application to these missions, control system for an autonomous flight should be designed and implemented. Compared to conventional fixed-wing aircraft, obtaining a state-space model for helicopter is more difficult. The complexity of helicopter flight dynamics makes modeling difficult, and without a good model of the flight-dynamics, the flight-control problem becomes inaccessible. A system model can be obtained in two ways, through first principles modeling and system identification. A pure first principle modeling is unsuitable for a small-scale helicopter because this approach requires extensive knowledge of helicopter flight mechanics. Moreover, unlike in a quad copter, the helicopter has coupling between its various parameters which has to be taken into account and thus makes the modeling much more difficult. Helicopter rotors add dynamics that couple with the rigid body fuselage motions and the surrounding flowfield, introducing complex aerodynamics which manifest in part as rotor/wake and rotor/fuselage interactions. We therefore go for system identification method to get the dynamic model.

Corresponding Author:-Pragadeesh Suresh Babu.

Address:- Department of Instrumentation and Control Engineering, National Institute of Technology,

Helicopter Dynamics:-

The dynamics of the helicopter are represented as a rigid body which can be coupled to additional dynamics such as the rotor or engine/drive-train dynamics. Including these subsystems improves the accuracy of the model making it more consistent for a range of frequencies.

Angular Dynamics:-

The frequency response of the rolling and pitching rates p and q to the lateral and longitudinal cyclic inputs δ_{lat} , δ_{lon} shows a pronounced underdamped second order behaviour. In this modelling approach, the lateral and longitudinal blade flapping dynamics are represented by two coupled first-order differential equations.

Heave Dynamics:-

With respect to heave dynamics, a first order system can successfully capture this part of the helicopter dynamics.

Yaw Dynamics:-

Here the yaw response has a nature of second order system. The differential equations used for state space model are as follows.

$$\begin{aligned}\dot{r} &= N_r r + N_{ped} (\delta_{ped} - r_{fb}) \\ \dot{r}_{fb} &= -K_{r_{fb}} r_{fb} + K_r r\end{aligned}$$

State Space Model:-

The aim of system identification is to construct a suitable model, such that the input output behavior of the model approximates the input-output behavior of the helicopter system, i.e. for a small positive constant.

$$\|y(k) - \hat{y}(k)\| \leq \varepsilon \forall k > 0$$

$$y_i(k+d) = F_i (\bar{u}(k), \dots, \bar{u}(k - n_1 + 1), \bar{y}(k - 1), \dots, \bar{y}(k - n_1 + 1)) \quad i=1, 2, \dots, p$$

Where n_1 is the order (or equivalent delay) of the system, d is the relative degree of the function.

A 6-degrees-of-freedom linear rigid body helicopter model with first-order approximation is given by a differential equation.

$$\dot{X} = AX + BU$$

The system identification analysis used a control vector consisting of the four pilot joystick inputs, consisting of the roll angle, pitch angle, translational velocities, rotational velocities, and rotor speed. These are written symbolically as-

$$u^T = [\delta_{thr} \quad \delta_{lon} \quad \delta_{lat} \quad \delta_{dir}]$$

$$z^T = [\phi \quad \theta \quad u \quad v \quad w \quad p \quad q \quad r \quad \Omega]$$

Where u, v, w : body-coordinate velocity, Ω, ϕ, θ : roll, pitch, yaw angle and p, q, r : roll, pitch, yaw rate respectively. The key dynamics of the helicopter are seen from reference to the system's Eigen values and eigen vectors. The first 4 roots are on the real axis, of which 2 are stable and 2 are unstable. The unstable modes involve the horizontal velocities with both altitudes and longitudes. The stable modes involve horizontal and vertical velocities. The eigen value 5 is associated with heavy response. The Eigen pair 6 and 7 is associated with the closed-loop yawing mode resulting from the active yaw damping system. The pitching mode (eigenvalues #8-9), which has a considerable roll coupling component, has a frequency that is nearly exactly the square root of the pitch flap spring. The coupled rolling mode with slight pitching component (Eigen values #10-11), has a frequency that corresponds to the square root of the roll flap spring.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\Omega} \\ \dot{a} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -g & X_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & Y_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Z_w & 0 & 0 & 0 & Z_{\Omega} & 0 & 0 \\ 0 & 0 & 0 & L_v & 0 & 0 & 0 & 0 & 0 & 0 & L_b \\ 0 & 0 & M_u & 0 & 0 & 0 & 0 & 0 & 0 & M_a & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & T_{\Omega} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -\frac{1}{\tau} & \frac{A_b}{\tau} \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & \frac{B_a}{\tau} & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ u \\ v \\ w \\ p \\ q \\ r \\ \Omega \\ a \\ b \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ N_{thr} & 0 & 0 & N_{dir} \\ T_{thr} & 0 & 0 & 0 \\ 0 & \frac{A_{lon}}{\tau} & \frac{A_{lat}}{\tau} & 0 \\ 0 & \frac{B_{lon}}{\tau} & \frac{B_{lat}}{\tau} & 0 \end{bmatrix} \begin{bmatrix} \delta_{thr} \\ \delta_{lon} \\ \delta_{lat} \\ \delta_{dir} \end{bmatrix}$$

Fig1:-State Space Model.

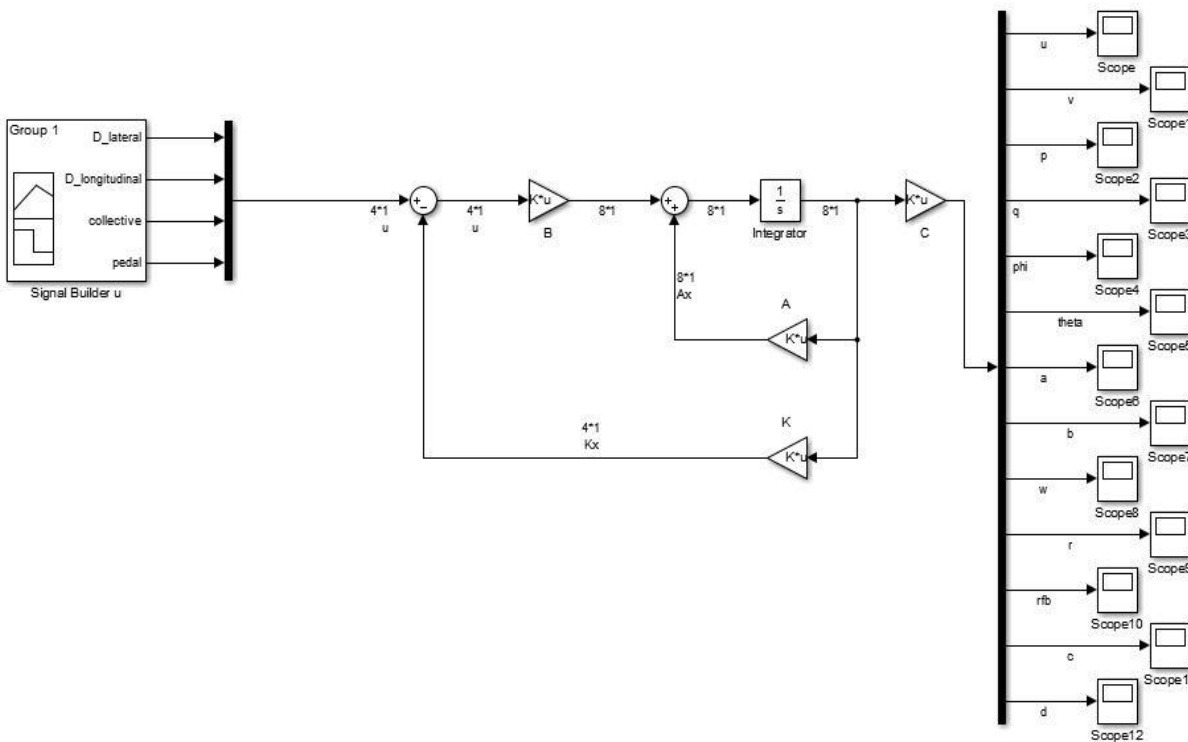


Fig 2:-MATLab Simulink Model.

Radial Basis Function:-

The idea of Radial Basis Function derives from the theory of function approximation. Their main features are: They are two-layer feed-forward networks. The hidden nodes implement a set of radial basis functions. The output nodes implement linear summation functions as in an MLP. The network training is divided into two stages: first the weights from the input to hidden layer are determined, and then the weights from the hidden to output layer. The training/learning is very fast. The networks are very good at interpolation.

The exact interpolation of a set of N data points requires every one of the D dimensional input vector $p = \{x : i = 1, \dots, D\}$ to be mapped onto the corresponding target output t^p . The goal is to find a function $f(x)$ such that

$$f(x^p) = t^p \forall p = 1, \dots, N$$

This approach introduces a set of N basis functions, one for each data point. The output of the mapping is then taken to be a summation of the basis functions, i.e.

$$f(x) = \sum_{p=1}^N w_p \phi(\|x - x^p\|)$$

The idea is to find the “weights” w_p such that the function goes through the data points. It is easy to determine equations for the weights:

$$f(x^q) = \sum_{p=1}^N w_p \phi(\|x^q - x^p\|) = t^q$$

The radial basis network is a two-layer network. First, in layer 1 of the RBF network, instead of performing an inner product operation between the weights and the input (matrix multiplication), we calculate the distance between the input vector and the rows of the weight matrix. The second layer is the linearization layer.

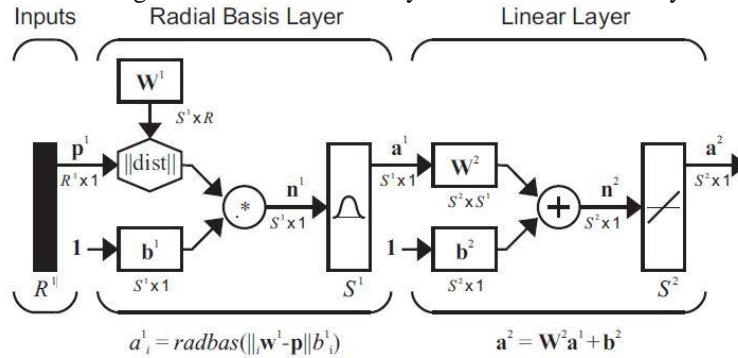


Fig3:-Radial Basis Network.

Model Verification:-

We first find out the state space model of the system from the data obtained from different flight experiments. The various values are substituted into the state space model (Fig1) and obtain the state space system as follows.

A =

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
x1	0.01793	-0.02816	-0.04905	-0.04989	-0.08618	-8.924	-31.87	-2.033	0	0	0
x2	-0.001013	-0.06121	0.1759	0.04767	8.552	-0.0003709	-0.1059	14.62	0	0	0
x3	-0.124	0.09871	-0.1504	-0.03949	0.5321	0.02909	0.4673	164.2	0	0	0
x4	-0.001578	-0.06033	-0.01114	-0.1123	-0.04229	0.3492	76.89	-1.131	0	0	0
x5	0	0	1	0	0	0	0	0	0	0	0
x6	0	0	0	1	0	0	0	0	0	0	0
x7	0.1267	-0.0211	-0.04932	-0.3623	-0.1562	-1.008	-8.204	-1.223	0	0	0
x8	0.005445	-0.1899	-0.4116	0.00926	-1.304	0.03748	0.3583	-12.64	0	0	0
x9	0.08342	-0.6959	2.356	0.2394	-1.151	0.8609	19.96	-7.583	-2.804	79.26	25.28
x10	0.3303	-0.09546	-2.542	-0.1585	-0.6653	2.354	26.29	-11.5	-39.87	-7.825	-180.4
x11	1458	-2737	-3372	1571	7335	1.352e+04	2.277e+05	1.464e+05	-3472	2.997e+05	-1.146e+06

Fig4:-“A” matrix of the state space system.

B =

	u1	u2	u3	u4
x1	0	0	0	0
x2	0	0	0	0
x3	0	0	0	0
x4	0	0	0	0
x5	0	0	0	0
x6	0	0	0	0
x7	0.03904	-0.1287	0	0
x8	0.2049	0.02395	0	0
x9	0	0	-77.52	0
x10	0	0	48.46	-40.98
x11	0	0	0	0

Fig5:- “B” matrix of the state space system.

We then run the code in MATLAB to find the fit between the estimated model and the helicopter model. There may not always be a perfect fit as there will always be some kind of discrepancy in the data, error in measurement readings and the inability of the network to keep up with the changes in the data etc.

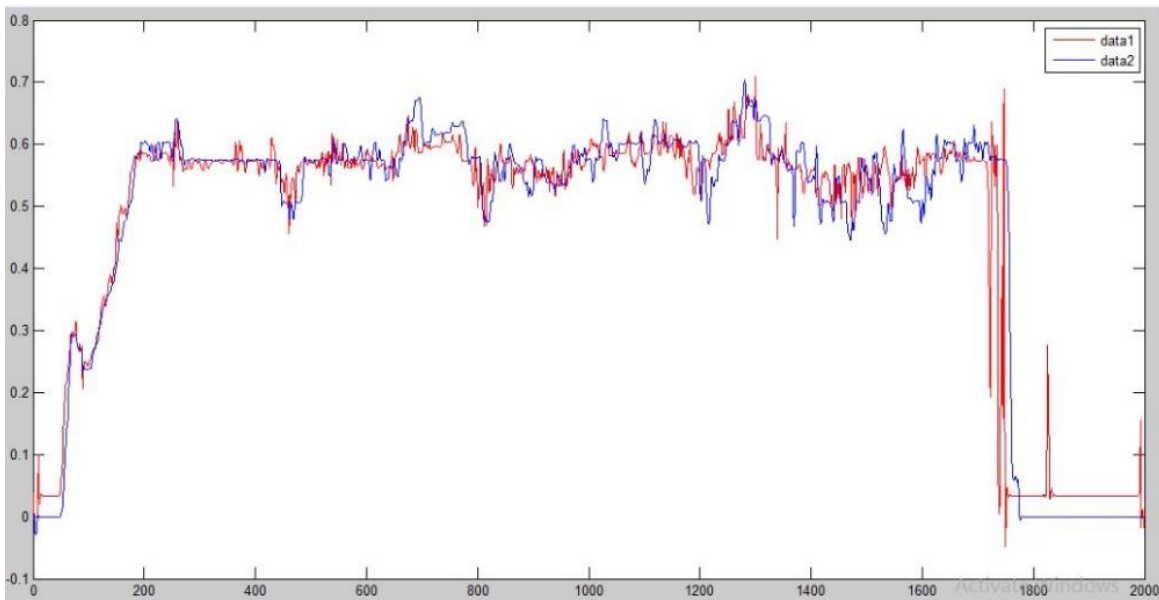


Fig6:- Yaw Rate (Blue=Actual o/p, Red=Network o/p)

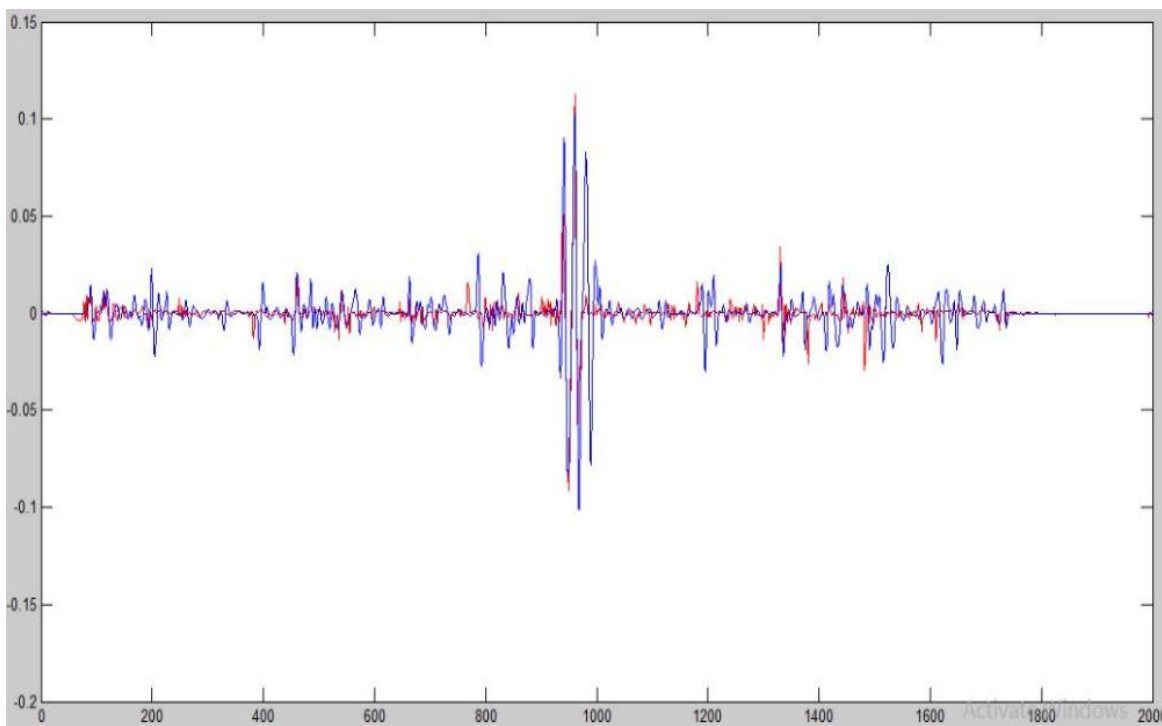


Fig 7:- Roll Angle (Blue=Actual o/p, Red=Network o/p)

Particle Swarm Optimization:-

Particle Swarm Optimization (PSO) is a swarm intelligence algorithm. It is based on the behavior of birds flocking together when they search for food. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the

food? The effective one is to follow the bird which is nearest to the food. In the proposed algorithm, an agent of the swarm, called a particle, learns from the best position that it has occupied and also the best position that any particle of the swarm might have encountered. These positions are saved in memory of each particle and constantly updated to direct the swarm to the global best position. The best position of a particle is called cognitive index - $pBest$, and the best of the swarm the social index - $gBest$. The equations that govern the change in positions of a particle are:

$$V_{i+1} = w * V_i + (C_p * r_1 * (p_{besti} - X_i)) + (C_g * r_2 * (g_{best} - X_i))$$

$$X_{i+1} = X_i + V_{i+1}$$

The flow of the PSO algorithm can be described by Algorithm 1, m is the number of particles. Where x is the position of the particles, i.e., the potential solutions, v is the velocity of the particles, f_{lbest} and x_{lbest} is the best fitness value and position for individual particles. f_{gbest} and x_{gbest} is the best fitness value and position for the group.

The algorithm is as follows-

1. Initialize the number of particles- m value.
2. **while** ($k < \text{max. iterations}$) **do**
3. **for** (each particle $j \in [1 \dots m]$) **do**
4. $f \leftarrow \text{PSO-FITNESS}(x[j])$
5. **if** $f < f_{lbest}[j]$ **then**
6. $f_{lbest}[j] \leftarrow f$
7. $x_{lbest}[j] \leftarrow x[j]$
8. **end if**
9. **if** $f < f_{gbest}$ **then**
10. $f_{gbest} \leftarrow f$
11. $x_{gbest} \leftarrow x[j]$
12. **end if**
13. $v[j, k+1] = wv[j, k] + c_1 \phi_1(x_{lbest}[j, k] - x[j, k]) + c_2 \phi_2(x_{gbest}[j, k] - x[j, k])$
14. $x[j, k+1] = x[j, k] + v[j, k+1]$
15. **end for.**
16. **end while.**

Here we take the number of particles to be 100 and the number of iterations to be 1000.

Multi Layer Perceptron (MLP):-

Artificial Neural Networks provides a method for learning arbitrary mapping between two data sets. A typical ANN contains a number of adjustable parameters called weights. Supervised learning involves finding a set of weights that minimizes the mapping error. Here mapping error is defined as the difference between observed output and NN's output. Fig.8 shows a typical MLP ANN [4, 6].

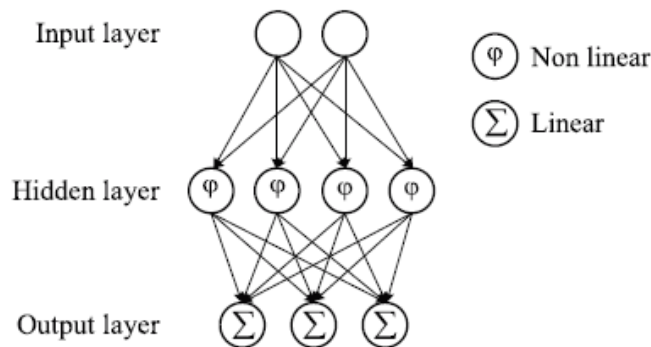


Fig 8:- Typical MLP network

In this case also, we find out the state space model of the system which is as follows.

a =

-9.7979	0	0	0	-0.9529	0	0	0	0	0	-9.8100	0
0	-9.7979	0	0.9529	0	0	0	0	0	0	9.8100	0
0	0	-0.9330	0	0	0	0	0	0	0	0	0
0	-18.5624	0	-19.2293	0	0	0	0	0	0	0	0
0.0352	0	0	0	-0.1000	0	0	0	0	0	0	0
0	0	0	0	0	-8.0085	0	0	0	0	0	0
1.0000	0	0	0	0	0	0	0	0	0	0	0
0	1.0000	0	0	0	0	0	0	0	0	0	0
0	0	1.0000	0	0	0	0	0	0	0	0	0
0	0	0	1.0000	0	0	0	0	0	0	0	0
0	0	0	0	1.0000	0	0	0	0	0	0	0
0	0	0	0	0	1.0000	0	0	0	0	0	0
0	0	0	0	0	0	1.0000	0	0	0	0	0

Fig 9:- "A" matrix of state space system.

b =

0	0	0	0
0	0	0	0
-0.2675	0	0	0
0	-18.7063	0	0
0	0	0.1000	0
0	0	0	0.4823
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Fig 10:- "B" matrix of state space system.

Model Verification:-

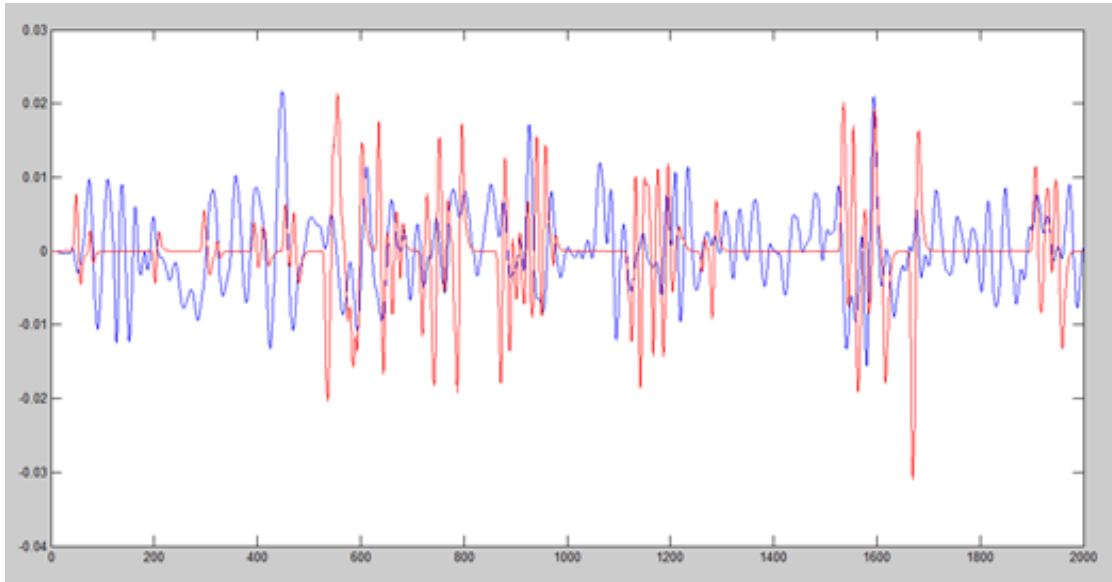


Fig11:-Pitch Rate (Blue=Actual o/p, Red=Network o/p)

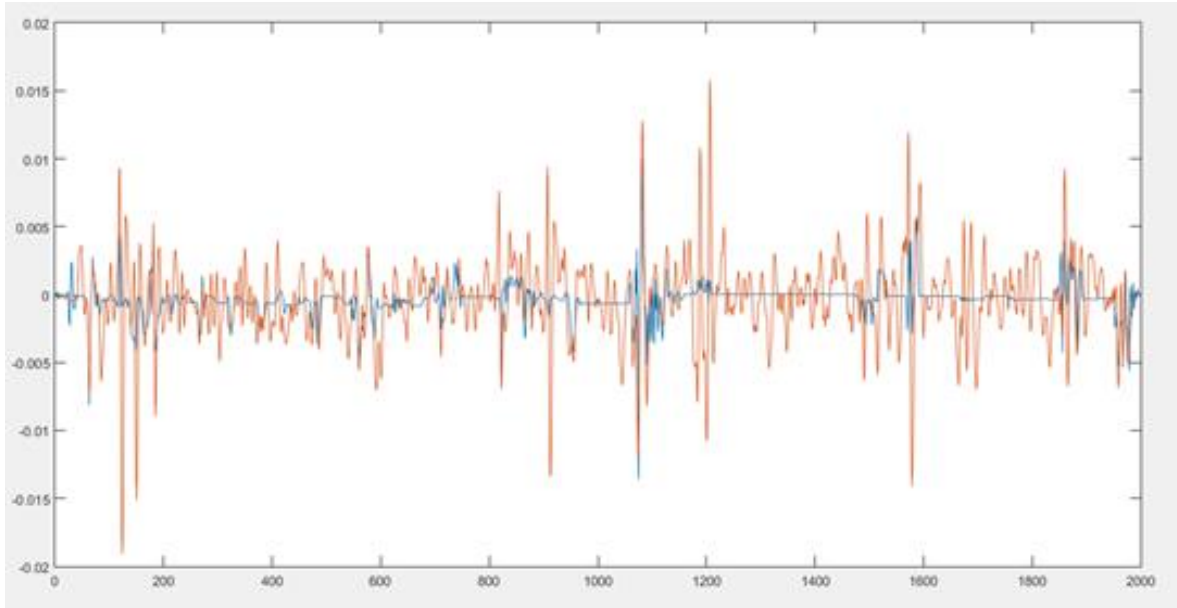


Fig12:-Climb Rate (Blue=Actual o/p, Red=Network o/p)

Conclusion:-

The overall consistency of these models provides good evidence that the identified models are accurate and that the methods used are appropriate for identifying miniature helicopters. We can see from the graphs that we have not obtained the best fit possible as there are some discrepancies in the data. PSO is a powerful optimization tool in the realm of control theory, particularly for the time-domain state space systems, where performance tuning has not been well-studied in the past. When analysing identification results, care must be taken by the user to consider what frequency range is of interest. Concerning the different methods and approaches investigated in the thesis, it can be stated that a general agreement across all methods has been seen in the estimated models. We can therefore conclude here the radial basis gives a better model to work with as the graphs show a better fit to the model than particle swarm optimization method. To improve upon PSO method, we must go into quantum optimization.

References:-

1. **Nonlinear system identification of a small-scale unmanned helicopter** by Shuai Tang, Zhiqiang Zheng, Shaoke Qian, Xinye Zhao.
2. **Using of Particle Swarm for Performance Optimization of Helicopter Rotor Blades** by Giorgio Guglieri
3. **Control System Design for Rotorcraft-based Unmanned Aerial Vehicles using Time-domain System Identification** by David Hyunchul Shim, Hyoun Jin Kim, and Shankar Sastry.
5. **System Identification of Small-Size Unmanned Helicopter Dynamics** by Bernard Mettler, Mark B. Tischler and Takeo Kanade
6. **Model development, system identification, and control of a quadrotor helicopter** by Matthew Rich
7. <http://www.swarmintelligence.org/tutorials.php>