Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-4 (2018) 358–363

PSO Based PID Controller for Quadrotor with Virtual Sensor

Yul Y. Nazaruddin, Angela Dian Andrini and Boby Anditio

Instrumentation and Control Research Group, Department of Engineering Physics,
Institut Teknologi Bandung, Bandung 40132, Indonesia
(Tel. +62-22-2504424; e-mail: yul@tf.itb.ac.id)

Abstract: Recent development and implementation on intelligent control system has lead to the development of virtual sensing system technology. Virtual sensing system allows immeasureable state variables to be accurately predicted, which is very beneficial to reduce the amount of sensors required to monitor and control a system, especially for the case of controlling a quadrator. This paper proposes a novel technique to design a PID control using virtual sensing system, consisting of Diagonal Recurrent Neural Network (DRNN) and Extended Kalman Filter (EKF), which predicts the immeasureable states of the quadrator system based on the current states and control inputs. A bio-inspired optimization technique, Particle Swarm Optimization (PSO), is proposed to be applied in DRNN to avoid any possibilities from local extreme condition. Further, a PSO based PID position controller is also developed to be integrated with the designed virtual sensing system to control a quadrator.

Keywords: Virtual Sensor, Diagonal Recurrent Neural Network, Extended Kalman Filter, Particle Swarm Optimization, PID Controller, Quadrotor

1. INTRODUCTION

Proportional Integral Derivative (PID) controllers have been implemented for a long time especially in various industries, and mostly applied for Single Input Single Output (SISO) systems. Basically, tuning methods of PID controller parameters are performed by adjusting the proportional, integral and derivative gains to make the output of a controlled system track a target set-point properly (Taeib et al., 2013). There are several techniques available to find the value of the gains such as Ziegler-Nichols, gain-phase margin, root locus, minimum variance, gain scheduling, etc., (see e.g. first literatures on the issues by Koivo and Tanntu, 1991, Aström and Hagglund, 1995), however, these methods were considered less than optimal for systems that are non-linear and of higher order. Although in general, until now PID type of controllers are still succesfully applied in many industrial processes as well as in many other applications, nevertheless it has still many limitations.

Further, quite often the implementation of control systems require more sophisticated and complex control objectives which need immeasureable state variables, instead of measureable ones. Those state variables may not usually be available to be measured because lack of any reliable sensors or the available sensors are mostly expensive. In addition, there are another problems that may occur while using the conventional sensors, for example the degradation of equipment performance, the sensor's performance decline and sensor's inability to detect the disturbance during measurement. Therefore, in such cases, a virtual sensing system or virtual sensor is usually required. Virtual sensor is used to predict values of complex systems variables by calculating information priori known in the controlled system using software (Liu et al. 2009, Nazaruddin and Aria, 2005, Nazaruddin and Astuti, 2009).

The designed virtual sensor consists of a Diagonal Recurrent Neural Network (DRNN) scheme for system modeling and an Extended Kalman Filter (EKF) as the estimator with inputs from DRNN. In many cases, DRNN applies several algorithms, including the well-known back-propagation (BP) algorithm during its training procces (Garro and Vazquez, 2015), however many algorithms can stay trapped in nondesirable solutions, which is non-optimum. In this investigation, the weights of DRNN will be optimized using Particle Swarm Optimization (PSO) method to avoid any local minima/maxima in its searching procedure (Garro and Vazquez, 2015). PSO is a population-based approach, which uses the swarm intelligence generated by the cooperation and competition between the particles in a swarm. It has been emerged successfully to a wide variety of search and optimization problems.

Furthermore, the virtual sensor scheme will be integrated with an intelligent PID control method, where an optimal gain values of the PID controller will be searched. For this purpose, once again the Particle Swarm Optimization (PSO) algorithm will be used to find the optimal value of the gains by spreading a number of particle in a certain range. In this case, an Integral Absolute Error (IAE) will be set to minimum.

The overall control scheme with virtual sensor will be tested to a quadrator model. The objective is to control the quadrotor by using the designed PSO based PID controller, where the position state variables of the quadrotor will be estimated using the proposed virtual sensor scheme. In this case, the application of virtual sensor will be simulated to replace the most commonly used GPS based measurement for

2405-8963 © 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.06.091

localization of a quadrator. Since the use of GPS based measurement increases the plant load, and it may cause a higher degree of uncertainty in several cases, as well as more expensive, then it is expected these shortcoming could be avoided by using virtual sensor. The results of simulation studies will also demonstrate the difference in performance between the states output of quadrotor with a manually-tuned PID controller and a PSO-tuned PID controller on the ideal quadrotor system, as well as the implementation of virtual sensor scheme with the PSO-tuned PID controller.

2. ESTIMATING VARIABLES USING VIRTUAL SENSOR SCHEME

Virtual sensor scheme is used to estimate immeasureable state variables of a system. As a requirement, modeling of the plant should be conducted. For this purpose, DRNN will be employed to model the plant. By deriving the inverse model of the plant, DRNN will be applied to model the immeasureable variable (or the primary variable), from the more easy to measure variable or measureable variable (or secondary variable). A number of input and output data is trained by DRNN with PSO method in the learning phase. The output of DRNN will be compared with the actual output of the system which is then used by Extended Kalman Filter (EKF) to estimate the immeasureable state variables of the systems. Fig. 1 illustrates the developed virtual sensing scheme with PSO as optimizer and EKF as an estimator.

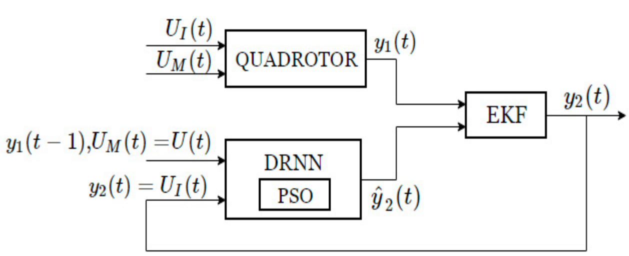

Fig. 1. Virtual Sensor Scheme

In the modeling phase of the quadrotor using DRNN, during the learning phase of DRNN, as input variables are the measureable input U_M (secondary variable) and immeasureable input U_I (primary variable), and as the output variable of the quadrotor is y_2.

2.1 Diagonal Recurrent Neural Network

Diagonal Recurrent Neural Network (DRNN) has self-feedback connection among the neurons in the hidden layer (Deng et al., 2005). The first layer of DRNN consists of inputs while the last layer is the network outputs. DRRN reduces the number of inter-connectivity among the neurons of the Fully Connected Recurrent Neural Network (FRNN), without reducing the performance considerably. In the development of virtual sensor, especially in the case of quadrotor, DRNN has the advantages of taking the control inputs $U(n)$, and both the last measureable states $y_1(n-1)$ and immeasureable states $y_2(n-1)$, to determine the next predicted immeasureable states $y_2(n)$. The variables $U(n)$, $y_1(n-1)$ and $y_2(n-1)$ are put together into one input vector to the neural network, denoted as $V(n)$. The mathematical model of the network is represented by the following equations

- Input Layer

$$S_j^1(n) = \sum_{k=1}^{N} W_{jk}^{1I} V(n) + W_j^{11} X_j^1(n-1) + W_j^{1b} \tag{1}$$

$$X_j^1(n) = f(S_j^1(n)) \tag{2}$$

- Hidden Layer

$$S_j^l(n) = \sum_{k=1}^{N} W_{jk}^{ll-1} X_j^l(n) + W_j^{ll} X_j^l(n-1) + W_j^{lb} \tag{3}$$

$$X_j^l(n) = f(S_j^l(n)) \tag{4}$$

- Output Layer

$$\hat{y}_2(n) = \sum_{j=0}^{N} W_j^{LO} X_j^L(n) \tag{5}$$

where $l = 2,3, \ldots, L$ and k is the number of neurons in the initial layer, and j is the number of neurons in the next layer. In every iteration, $V(n)$ is a k^{th} dimensional vector of the input layer. $S^l_j(n)$ is the sum input to the j^{th} neuron on the next layer, $X^l_j(n)$ is the output neuron to the j^{th} neuron in the next layer, $y_2(n)$ is the network output and $f(.)$ is the activation function of the neurons. N is the number of neurons in each layer, while L is the total number of hidden layers in the network and n is the iteration number at time t. The architecture of DRNN is shown in Fig. 2.

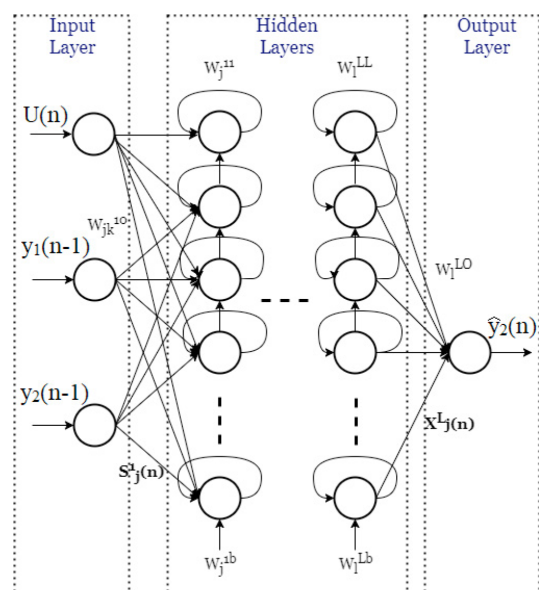

Fig. 2. Architecture of Diagonal Recurrent Neural Network (DRNN)

2.2 Extended Kalman Filter (EKF)

In general, EKF is one of the most efficient filter in predicting the true value of a measurement. In this case, it will serve as an estimator of the state variables and the immeasureable input. EKF will perform in two stages which are the prediction phase and the updating phase. The prediction phase is to make an estimation of the next state given the current state and the control input, while in the updating phase, the filter try to minimizing the error between its prediction and the actual measurement. In EKF, the state transition equation may be a linear state space model or many non-linear functions. The algorithm of EKF (Li et al., 2015) is shown in (6) to (12).

$$A = \left.\frac{df}{dx}\right|_{x} = \hat{x}_{t-1} \tag{6}$$

- Prediction phase

$$x_t^- = f(\hat{x}_{t-1}, \hat{u}_{t-1}) + W_t \tag{7}$$

$$P_t^- = AP_tA^T + Q \tag{8}$$

- Updating phase

$$z_t = y_t + Z_t \tag{9}$$

$$K_t = P_k^- C^T (CP_k^- C^T + R)^{-1} \tag{10}$$

$$\hat{x}_t = x_t^- + K_t(z_t - x_t^-) \tag{11}$$

$$P_t = (I - K_t C)P_t^- \tag{12}$$

$$y_2(t) = C\hat{x}_t \tag{13}$$

where x_t^- is defined as the prior status estimation derived from status transition equation at the time t-1, $f(x_{t-1}, u_{t-1})$ is the non-linear state transition functions of the system, W_t is the prediction error, x_t is the state variable, y_t is the observable output, Z_t is the observation error, P_t is the process covariance matrix, K_t is the Kalman gain and $y_2(t)$ is the immeasureable variables which is the output of EKF.

At first, the filter makes prediction based on its knowledge of the system including its non-linear functions, then the Kalman gain is calculated to observe how reliable the prediction and the actual measurement is, the smaller the error between measurement value and prediction value, means the filter needed only small amount of correction between its prediction, resulting in small Kalman gain. In general, this algorithm produces quite well prediction based on approximating the known variables even for the case of non-linear functions

3. PID OPTIMIZATION USING PSO

PID controller is widely used in various industrial application but it is usually difficult to find the optimal gains of the controller to reach the target outputs of the plant. Standard PID equation can be described by (14)

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{14}$$

where K_p , K_i , K_d is the proportional, integral and derivative gain respectively, $u(t)$ is control signal and $e(t)$ is the error signal where $e(t) = SP_{y2}(t) - y_2(t)$, with $SP_{y2}(t)$ is the target output or set-point and $y_2(t)$ is the system output.

An alternative technique to determine the gains of PID controller is by implementing Particle Swarm Optimization (PSO) algorithm, which is also to ensure that the gains are optimal. PSO is an evolutionary computing method which inspired by the movement of many animals such as, bird flocks and fish schools. This algorithm works like animal behaviour on finding foods and avoiding danger, where they will coordinate with each other to find the best position to settle. Likewise, PSO is directed by the movement of the best individual from the population, known as the social compound, and their own experience, known as the cognitive compound. The algorithm moves the set of solutions to find the best solution among them.

Given a set of solution as particle p, each particle has their own position $x_p \in \Re^N, p = 1,2,\ldots,M$ where N is the dimension of solution set and M is the number of particle. The particles movement is determined by the global best and personal best solutions which is updated every iteration, which determine by $G_{best} \in \Re^N$ and $P_{best} \in \Re^N$ where personal best is updated every time each particle found a new best experience and global best is updated every time a newly found best solution over each iteration and each particle. Every iteration, the particle position is updated by its velocity function below

$$v_p(i+1) = wv_p(i) + c_1 r_1 (P_{best,p}(i) - x_p(i)) + c_2 r_2 (G_{best}(i) - x_p(i)) \tag{15}$$

where w is the inertia coefficient of the particle movement, c_1 and c_2 is personal and social acceleration coefficient respectively, while r_1 and r_2 is a random number between (0,1). The coefficients w, c_1 and c_2 is determined by the following equation to ensure stability and convergence in the search (Clerc and Kennedy, 2002)

$$x = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}}; \phi = \phi_1 + \phi_2 > 4 \tag{16}$$

Clerc's constriction method commonly use $\phi = 4.1$ and $\phi_1 = \phi_2$ resulting $x \approx 0.7298$. So the inertia coefficient w, the acceleration coefficients c_1 and c_2 are determined by the following relations

$$w = x \tag{17}$$

$$c_1 = \phi_1 x \tag{18}$$

$$c_2 = \phi_2 x \tag{19}$$

Finally, after calculating the velocity, the particle position can be updated using the following equation

$$x_p(i+1) = x_p(i) + v_p(i+1) \tag{20}$$

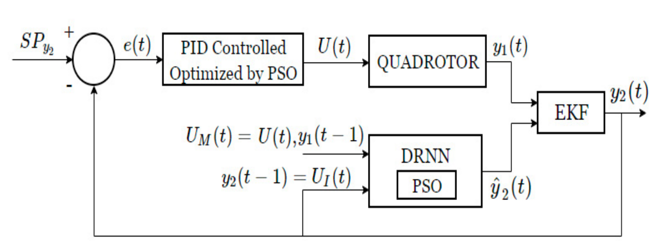

Fig. 3. The overall PID control scheme integrated with virtual sensor

The performance of PSO is observed by calculating the cost functions, namely Integral Absolute Error (IAE), or

$$IAE = \int_0^t \left| O(t) - Y(t) \right| dt \tag{21}$$

The overall control scheme integrated with virtual sensor scheme and PSO based PID controller design is illustrated in Fig. 3. Here, the PID controller gains are obtained by using PSO algorithm with Integral Absolute Error (IAE) as the cost function.

4. QUADROTOR MODEL

Quadrotor is considered to be one of the most challenging problem in control issue, because of its complexity, non-linear behavior, and underactuated characteristics. It means that quadrotor has more degree of freedom, which is three translational movements (x,y,z) and three angular movements (ϕ,θ,φ) than its controller inputs, which is only four $(f_T, \tau_x, \tau_y, \tau_z)$. The linearized full state quadrotor model according to Sabatino (2015), in a state space representation, applies the following states and the control inputs

$$x = [\phi\ \theta\ \varphi\ p\ q\ r\ u\ v\ w\ x\ y\ z]^T \in \Re^{12} \tag{22}$$

$$u = [f_T\ \tau_x\ \tau_y\ \tau_z]^T \in \Re^4 \tag{23}$$

where (ϕ,θ,φ) are the angular positions of quadrotor defined in angle along the x, y and z axis known as roll, pitch and yaw respectively, (p,q,r) are the angular velocities of the rotational movement, (x,y,z) are the translational positions along the x, y and z axis respectively, (u,v,w) are the translational velocities along the x, y and z direction. The control input $(f_T, \tau_x, \tau_y, \tau_z)$ are the thrust force, rotational torques in the x, y and z axis respectively. The state space model of the quadrator is given as follows

$$A = \left.\frac{\partial f(x,u)}{\partial x}\right|_{\substack{x=\bar{x}\\ u=\bar{u}}} = \begin{bmatrix} 0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&-g&0&0&0&0&0&0&0&0&0&0\\ g&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0 \end{bmatrix} \tag{24}$$

$$B = \left.\frac{\partial f(x,u)}{\partial u}\right|_{\substack{x=\hat{x}\\ u=\bar{u}}} = \begin{bmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&\frac{1}{I_x}&0&0\\ 0&0&\frac{1}{I_y}&0\\ 0&0&0&\frac{1}{I_z}\\ 0&0&0&0\\ 0&0&0&0\\ \frac{1}{m}&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix} \tag{25}$$

where g is the gravitation constant, I_x, I_y and I_z is the inertial term along the x, y and z axis respectively. In this investigation, it is assumed that not all of the states are made measureable, there are nine observable states and three unknown states. Since the objective is to replace GPS as a position sensor, the state positions (x,y,z) become unobservable while the other states are fully observable.

5. IMPLEMENTATION AND EVALUATION

A virtual sensor is used to estimate three positions (x,y,z) variables of quadrotor. At first, the network inputs, which is the states and control inputs $(\phi,\theta,\varphi,p,q,r,u,v,w,x,y,z,f_T,\tau_x,\tau_y,\tau_z)_t$ of the quadrotor are generated, and outputs, which is the next immeasureable state $(x,y,z)_{t+1}$ of the quadrotor by using the linearized state-space model derived by Sabatino (2015). The neural network will be trained using the input and output data, optimized by PSO, with following designed parameters

DRNN parameters :

Neural Network Layer	: [16 20 20 3]
Activation Function	: tan-sigmoid

PSO parameters :

N (number of set solution)	: 860
M (number of particle)	: 30
K (number of data set)	: 2000
Boundary of solution set	: [-1, 1]
Fitness Function	: IAE

During investigation, several trials of neural network structure were done and the best trade-off between prediction accuracy and computation time was found with [16 20 20 3] structure. The tan-sigmoid activation function, will limit the boundary of the search space into [-1, 1], and also for searching in a high dimensional space, its better to limit the velocity update to a minimum value so it is guaranteed to sweep the entire solution space. After thousand of iterations, the fitness of the best particle reached 1.622 on the normalized scale. After that, the EKF algorithm was implemented to approximate the unknown state variables and reduced the errors made by the DRNN. After the DRNN has been trained and the EKF model was validated, both algorithms were integrated to create a virtual sensor output.

For PID controller design, the quadrotor's position x and y have an interconnected relationship between the position and the angular position, especially pitch and roll. Since it is difficult for a standard PID structure to control the quadrotor, so in this case an improvement for controlling the quadrotors position x and y is made according to Ajmera and Sankaranarayanan (2015), but in this case PID type controller is implemented rather than PD algorithm as mentioned. For the z position, it is also controlled with the standard PID algorithm.

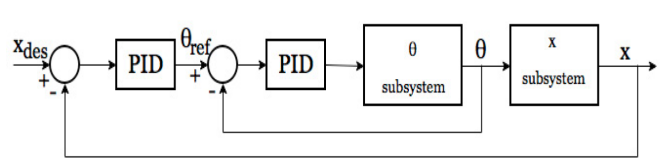

Fig. 4. Two degree of freedom PID Controller

The positions quadrotor x is controlled by using 2 PID controller, where the first controller will control the pitch angular movement which affect the other controller to produce control input for controlling the x position (see Fig. 4). According to Ajmera and Sankaranarayanan (2015), the mathematical model of PD controller can be written as

$$\tau_\theta = K_{p2}[K_{p1}(x_{des} - x) + K_{d1}(\dot{x}_{des} - \dot{x}) - \theta] + K_{d2}[K_{p1}(\dot{x}_{des} - \dot{x}) + K_{d1}(\ddot{x}_{des} - \ddot{x}) - \dot{\theta}] \quad (26)$$

From (26), a modification is made by adding an integral gain into the equation, so the equation for PID controller becomes

$$\tau_y = K_{p2}[K_{p1}(x_{des} - x) + K_{i1}\int(x_{des} - x)dt + K_{d1}(\dot{x}_{des} - \dot{x}) - \theta] + K_{i2}\int[K_{p1}(x_{des} - x) + K_{i1}\int(x_{des} - x)dt + K_{d1}(\dot{x}_{des} - \dot{x}) - \theta]dt + K_{d2}[K_{p1}(\dot{x}_{des} - \dot{x}) + K_{i1}\int(\dot{x}_{des} - \dot{x})dt + K_{d1}(\ddot{x}_{des} - \ddot{x}) - \dot{\theta}] \quad (27)$$

In this case, the total number of PID gains that must be determined to control the quadrotor positions (x,y,z) are 15 gains. All of them is going to be the particle position in PSO algorithm with IAE cost function. This is an excellent example to use PSO method for optimizing the PID coefficients since 2 degree of freedom PID controller is hard to determine because each variables will affect one another. It was observed that after 500 iterations, the optimal parameters of PID controller was found. Using quadrotor parameters shown in Table 1., the positions *x, y* and *z* of the quadrotor as results of PID control action are demonstrated in Figs. 5, 6 and 7. The quadrator system was subjected to different step changes during the control running.

In the implementation, three different schemes of PID control was used to demonstrate the difference in performance of each scheme, namely :

1. Control using manually-tuned PID controller on the ideal quadrotor system
2. Control using PSO-tuned PID controller on the ideal quadrotor system,
3. Control using PSO-tuned PID controller with immeasureable state variables estimated by virtual sensor

Table 1. Parameters of quadrotor model

I_x	0.01 kg m^2
I_y	0.01 kg m^2
I_z	0.2 kg m^2
m	0.5 kg.

Comparing the results from three different schemes, it can be concluded that

1. The response of x, y and z position of the quadrotor for the case of 2 (without virtual sensor scheme) and 3 (with virtual sensor scheme) are almost identical which reveals that the virtual sensor scheme is able to predict the positions state of the quadrotor accurately, so that the PID control action tracks the different changes of set-point quite satisfactorily.
2. The response of x, y and z position of the quadrotor for the case of 1 (PID manually tuned) shows worse performance compared to the case 2 and 3 (PSO tuned PID controller), especially for the z position where more damped response is demonstrated
3. The IAE values for manually-tuned PID controller versus PSO tuned PID controller are 7.0709 and 3.0995 respectively, which reveals the superiority of PID controller tuned with PSO.

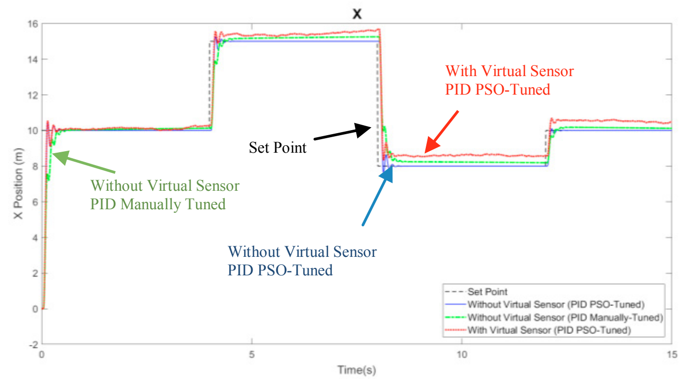

Fig. 5. x-position of the quadrotor subject to several step changes, for 3 different control schemes

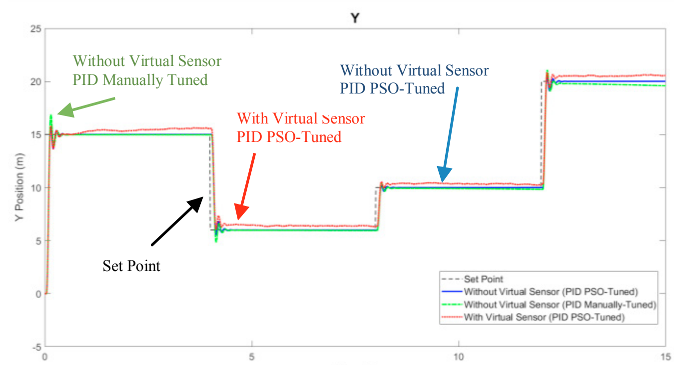

Fig. 6. y-position of the quadrotor subject to several step changes, for 3 different control schemes.

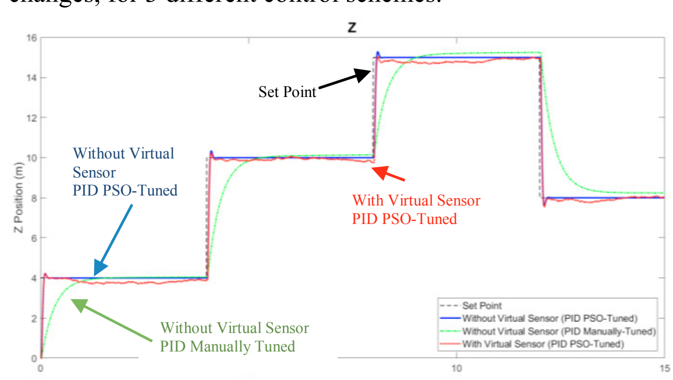

Fig. 7. z-position of the quadrotor subject to several step changes, for 3 different control schemes

As observed from the results of the simulation, estimation of positions show good performance during control course, however there might be a short-coming effect called 'dead reckoning' which is a lump of bias in the measurement over a long period of time. This problem usually occurs in the estimation of variables using numerical approach (Neto et al., 2013).

6. CONCLUSIONS

An alternative method in the development of PSO based PID controller for a quadrator model has been proposed. For all 12 state variables describing the quadrator model, 3 state variables, which are immeasureable variables, have been estimated using virtual sensing system. The virtual sensing system is derived using the DRNN method integrated with EKF, allowing the immeasureable states to be determined from the secondary variables. An optimization algorithm, PSO, is applied both in DRNN and in determining the optimal parameters of PID controller. The results show that the virtual sensing scheme predicts the immeasureable variable of the quadrator's state variables satisfactorily based on the information from measureable or secondary variables. Further, it is also shown that the proposed PSO based PID control scheme with virtual sensor for the quadrator, control the x, y and z position quite well to set-point changes. Comparing the results with manualy-tuned PID controller, a superior performance is demonstrated for the case of PSO based PID controller.

REFERENCES

Astrom, K.J. and Hagglund, T. (1995). *PID Controllers: Theory, Design, and Tuning*. Instruments Society of America.

Ajmera, J. and Sankaranarayanan, V. (2015), Trajectory Tracking Control of a Quadrotor, *Proceed. of International Conference on Control, Communication & Computing*

Clerc, M. and Kennedy, J. (2002), The Particle Swarm-Explosion, Stability, and Convergence in Multidimensional Complex Space, *IEEE Trans. Evolutionary Computation*, vol. 6, no. 1, pp. 58-73

Deng X., Xie, J., Guo, W., and Liu. J. (2005), A New Learning Algorithm for Diagonal Recurrent Neural Network, *International Conference on Natural Computation, Advances in Natural Computation*, pp. 44-50

Garro, B.A. and Vazquez, R.A. (2015) Designing Neural Networks Using Particle Swarm Optimization, Research Article, *Computational Intelligence in Neuroscience*, Vol. 2015

Koivo, H. and Tanttu, J. (1991). Tuning of PID controllers : Survey of SISO and MIMO techniques. *Proceed. IFAC Intell. Tuning Adapt. Control Symp.*, pp. 75–80

Ku, C.-C. and Li, K.Y. (1995), Diagonal Recurrent Neural Networks for Dynamic Systems Control, *IEEE Trans. Neural Networks*, vol. 6, no.1, pp. 144-156

Li, Q., Li, R., Ji, K. Ji, Dai, W. (2015), Kalman Filter and Its Application, *Proceed. of the 8th Intern. Conf. on Intelligent Networks and Intelligent Systems (ICINIS)*, pp.74-77

Liu, L., Kuo, S.M., and Zhou, M. (2009), Virtual Sensing Techniques and Their Applications, *Proceed. of the IEEE International Conference on Networking, Sensing and Control*, Okayama, Japan, March 26-29

Nazaruddin, Y.Y., and Aria, M. (2005), Adaptive-Predictive Control With Intelligent Virtual Sensor, *Proceed. of the 16th Triennial World Congress*, Prague, Czech Republic, pp. 263-268.

Nazaruddin, Y.Y. and Astuti, P. (2009), Development of Intelligent Controller with Virtual Sensing, *ITB J. Eng. Sci.* Vol. 41, No. 1, pp.17-36

Neto, P., Pires, J.N., and Moreira, A. P. (2013), 3-D Position Estimation from Inertial Sensing: Minimizing the Error from the Process of Double Integration of Accelerations, *39th Annual Conference of the IEEE Industrial Electronics Society (IECON 2013),* pp. 4026-4031

Sabatino, F. (2015), Quadrotor Control: Modelling, Nonlinear Control Design, and Simulation, *Master Degree Project*, KTH Electrical Engineering

Taeib, A., Lltaeif, A. and Chaari, A. (2013), A PSO Approach for Optimum Design of Multivariable PID Controller for Nonlinear Systems, International Conference on Control, Engineering & Information Technology (CEIT'13), *Proceed. Engineering & Technology*, Vol.2, pp. 206-210