International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Tuning Linearization Transformation using Back-Propagation Algorithm

S Janakiraman, Rajagopalan Devanathan

Abstract: The objective of linearization of a nonlinear system is to ensure smooth control of the linearized system through well-proven linear control methods. However, residual nonlinearities may still be present in a system after linearization either by design or due to mismatch between the system model and the actual plant. If the residual nonlinearities are not very significant, one can attempt to remove these by tuning the linearizing transformation by comparing the system to a linear canonical form. In this paper, we show how quadratic linearizing transformations of a three-phase horizontal gravity separator (TPS) model derived in an earlier paper by the authors can be tuned as in a neural network using error back-propagation by comparing it to a canonical linear model thus removing the nonlinearities within the tuning error limit.

Keywords: approximate linearization, back-propagation learning algorithm, control affine model, state space, three-phase horizontal gravity separator, neural network.

I. INTRODUCTION

Linearization attempts to eliminate the nonlinearities in a system and make the linearized system amenable to linear control. The approximate linearization due to Kang and Krener [1] attempts to eliminate the nonlinearities present in the system in a progressive way. The linearization of the nonlinear system is performed by coordinate change and state feedback. While the linearizing transformations eliminate the essential nonlinearities in the system, leaving only the higher order nonlinearities, it is possible to eliminate the residual nonlinearities by tuning the linearizing transformations while comparing the system to a canonical linear system of appropriate form as shown by Narendra and his co-workers [2-6]. The authors, in earlier papers [7, 8] have derived a control affine model for a three-phase horizontal gravity separator (TPS) considering the process to operate around an operating point. The significant quadratic nonlinearities present in the control affine model of TPS were eliminated through a coordinate change and state feedback leaving third and higher order nonlinearities in the system. In this paper, we show how even the remaining nonlinearities in the control affine model can be eliminated following the works of Narendra and his co-workers as cited above.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
S Janakiraman*, Centre for Automation and Robotics, Hindustan Institute of Technology and Science, Chennai, Tamil Nadu, India. Email: sjrjanakiraman@gmail.com
Rajagopalan Devanathan, Department of Electrical and Electronics, Hindustan Institute of Technology and Science, Chennai, Tamil Nadu, India. Email: devanathanr@hindustanuniv.ac.in

The main contributions of the paper are as follows.

- Quadratic linearizing transformations applied to a control affine model are tuned by comparing the system to a canonical system to remove third and higher order nonlinearities left in the system
- Multilayered neural network model for the tuning mechanism with back-propagation approach is adopted.
- Error minimization by gradient descent technique is used to derive tuning formulae for transformation coefficients considered as weights.
- Tuning of Quadratic linearization transformations to remove higher order nonlinearities as reported in the paper is new and not available in the literature.

The rest of the paper is organized as follows. Section II gives the background of the paper. Section III provides the necessary theory for the tuning of the transformation coefficients through a neural network architecture. Section IV gives an illustrative example and Section V gives the simulation results to verify the theory proposed. Section VI concludes the paper.

II. BACKGROUND

Narendra and Parthasarathy have applied a gradient neural network architecture to tune a nonlinear system by comparing it to a linear reference model [2]. Narendra and Parthasarathy have proposed a network architecture for the control of nonlinear system by tuning the transformation coefficients involved in linearization [3]. Levin and Narendra have presented their work on multilayered neural network for controlling a class of nonlinear systems through feedback linearization [4],[5]. Chen and Narendra presented their work on nonlinear adaptive control employing neural network with efficient switching between linear and nonlinear controllers [6]. Parvathy et al. [9] applied tuning of the linearization transformation coefficients to a permanent magnet motor model.

In earlier papers [7],[8], the authors have considered a TPS and derived a state space control affine model of the following form.

$$\dot{x} = Ax + Bu + f^{(2)}(x) + g^{(1)}(x)u + O^{(3)}(x,u) \qquad (1)$$

The matrices in (1) are of the form

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B=\begin{bmatrix} b_{11} & 0 & 0 \\ b_{21} & 0 & 0 \\ 0 & b_{32} & 0 \\ b_{41} & b_{42} & b_{43} \end{bmatrix}$$

$$f^{(2)}(x)=\begin{bmatrix} 0 \\ x_2^2 a_{122} \\ x_2^2 a_{131}+x_3x_2a_{132}+x_3^2a_{133} \\ 0 \end{bmatrix}$$

$$g^{(1)}(x)=\begin{bmatrix} x_1g_{11} & 0 & 0 \\ x_2g_{21} & 0 & 0 \\ 0 & x_3g_{32} & 0 \\ b_{141}^{(1)}(x) & b_{142}^{(1)}(x) & b_{143}^{(1)}(x) \end{bmatrix}$$

$$b_{141}^{(1)}(x)=x_2g_{41}+x_3g_{42}+x_4g_{43}$$

$$b_{142}^{(1)}(x)=x_2g_{44}+x_3g_{45}+x_4g_{46}$$

$$b_{143}^{(1)}(x)=x_2g_{47}+x_3g_{48}+x_4g_{49}$$

where the state variable x corresponds to deviation of water level, total of water and oil level in the separation chamber, oil level after the weir and gas pressure in the tank of TPS about their respective operating points, and u corresponds to deviation of the outflow rate of water, oil and gas of TPS about their respective operating points. The deviation in the inflow rate into TPS is considered as a disturbance and is not directly included in the linearization model. $O^{(3)}(x,u)$ corresponds to terms of order three or more involving x and u. $\{a_{ij}\}$, $\{a_{ijk}\}$, $\{b_{ij}\}$, $\{g_{jk}\}$are coefficients involving TPS process parameters and steady state operating values whose expressions are not relevant for discussion in this paper and hence omitted.

The quadratic terms in (1) are eliminated using the coordinate change and state feedback given by (2), (3) [10]-[11].

$$y=x+\phi^{(2)}(x) \tag{2}$$

$$u=\left(I+\beta^{(1)}(x)\right)\vartheta+\alpha^{(2)}(x) \tag{3}$$

for some functions $\phi^{(2)}(x)$, $\beta^{(1)}(x)$ and $\alpha^{(2)}(x)$. The superscript in brackets (j), j=1,2 correspond to the order of the homogeneous functions involved. Adopting the results of [6], [7], these functions can be expressed as follows.

$$\phi^{(2)}(x)=\begin{bmatrix} w_{11}^{(2)}x_1^2+w_{22}^{(2)}x_2^2 \\ w_{12}^{(2)}x_1x_2 \\ 0 \\ 0 \end{bmatrix}$$

$$\beta^{(1)}(x)=\begin{bmatrix} v_1x_1+v_2x_2 & 0 & 0 \\ 0 & v_3x_3 & 0 \\ \beta_{31}^{(1)}(x) & \beta_{32}^{(1)}(x) & \beta_{33}^{(1)}(x) \end{bmatrix}$$

where

$$\beta_{31}^{(1)}(x)=v_4x_1+v_5x_2+v_6x_3+v_7x_4$$

$$\beta_{32}^{(1)}(x)=v_8x_2+v_9x_3+v_{10}x_4$$

$$\beta_{33}^{(1)}(x)=v_{11}x_2+v_{12}x_3+v_{13}x_4$$

$$\alpha^{(2)}(x)=\begin{bmatrix} v_{14}x_1^2+v_{15}x_2^2 \\ v_{16}x_1^2+v_{17}x_2^2+v_{18}x_2x_3+v_{19}x_3^2 \\ \alpha_3^{(2)}(x) \end{bmatrix}$$

where

$$\alpha_3^{(2)}(x)=v_{20}x_1^2+v_{21}x_1x_2+v_{22}x_1x_3+v_{23}x_1x_4+v_{24}x_2^2$$
$$+v_{25}x_2x_3+v_{26}x_2x_4+v_{27}x_3^2+v_{28}x_3x_4$$
$$+v_{29}x_4^2$$

The expression for the coefficients, $\{w_{ij}\}$ $\{v_m\}$ involving process parameters and the steady state operating values are not relevant for our discussion and hence omitted.

III. TUNING THE TRANSFORMATION

Back-propagation algorithm is used for tuning the coefficients of the transformation. The canonical model represents a class of dynamic systems and is used as the reference model for tuning the transformation coefficients of the linearized TPS model. The controllable canonical model is represented by (4)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}+\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{4}$$

Back-propagation algorithm with a gradient descent approach is used to train the feedforward neural network representing the continuous functions of the transformation. Back-propagation algorithm enables calculation of the weight updates during the training of the neural network. The performance criterion in a layered network is the sum of the square of the difference (ε_i) between the reference model (o_i) and linearized system output (y_i) [3].

The error (ε_i) in a dynamic system is given by

$$\varepsilon_i(t)=o_i(t)-y_i(t)\text{, where } \{i\}\in\{1,2,3,4\} \tag{5}$$

The sum of squared error E is given by (6)

$$E=\frac{1}{2}\Sigma_{i=1}^{4}\varepsilon_i^2, \tag{6}$$

The gradient descent error minimization requires

$$\Delta w_{k,j}=-\eta\frac{\partial E}{\partial w_{kj}}\text{, where } k\,j\in\{11,22,12\} \tag{7}$$

where η is the learning constant.

The weight update for w_{kj} is given as follows.

$$w_{kj}(t+1)=w_{kj}(t)+\Delta w_{kj}(t), \tag{8}$$

where $k\,j\in\{11,22,12\}$ and (7) is given by

$$\Delta w_{kj}=-\eta\frac{\partial E(t)}{\partial w_{kj}(t)}=-\eta\frac{\partial E(t)}{\partial y_i(t)}*\frac{\partial y_i(t)}{\partial w_{kj}(t)} \tag{9}$$

where $k\,j\in\{11,22,12\}$ and η is the learning rate.

Similarly, we can write weight update rule for the input transformation coefficients as

$$v_m(t+1) = v_m(t) + \Delta v_m(t),\ m \in \{1,2,3,....29\} \quad (10)$$

where

$$\Delta v_m = -\beta \frac{\partial E(t)}{\partial v_m(t)} = -\beta \frac{\partial E(t)}{\partial y_i(t)} * \frac{\partial y_i(t)}{\partial x_i(t)} * \frac{\partial x_i(t)}{\partial u_i(t)} * \frac{\partial u_i(t)}{\partial v_m(t)} \quad (11)$$

where β is the learning rate.
From (5) and (6),

$$\frac{\partial E(t)}{\partial y_i(t)} = y_i(t) - o_i(t), i \in \{1,2,3,4\} \quad (12)$$

Also, from (2), one can write

$$\frac{\partial y_1(t)}{\partial w_{1,1}(t)} = x_1^2(t) \quad (13)$$

$$\frac{\partial y_1(t)}{\partial w_{2,2}(t)} = -x_2^2(t) \quad (14)$$

$$\frac{\partial y_2(t)}{\partial w_{1,2}(t)} = x_1(t)x_2(t) \quad (15)$$

The updated output transformation weights on substituting (12)-(15) in (9) and (8) we get (16)-(18)

$$w_{11}(t+1) = w_{11}(t) - \eta(y_1(t) - o_1(t))x_1^2(t) \quad (16)$$

$$w_{22}(t+1) = w_{22}(t) - \eta(y_1(t) - o_1(t))(-x_2^2(t)) \quad (17)$$

$$w_{12}(t+1) = w_{12}(t) - \eta(y_2(t) - o_2(t))x_1(t)x_2(t) \quad (18)$$

Similarly using (2), we can write

$$\frac{\partial y_1(t)}{\partial x_1(t)} = 1 + 2w_{11}x_1(t) \quad (19)$$

$$\frac{\partial y_2(t)}{\partial x_2(t)} = 1 + w_{12}x_1(t) \quad (20)$$

$$\frac{\partial y_3(t)}{\partial x_3(t)} = 1 \quad (21)$$

$$\frac{\partial y_4(t)}{\partial x_4(t)} = 1 \quad (22)$$

Writing (1) in discrete form we have

$$x_1(t) = x_1(t-1) - u_1(t)\left(b_{11} + g_{11}^{(1)}x_1(t)\right) \quad (23)$$

$$x_2(t) = x_2(t-1) - u_1(t)\left(b_{21} - g_{21}^{(1)}x_2(t)\right) - a_{22}x_2(t) + a_{122}x_2^2(t) \quad (24)$$

$$x_3(t) = x_3(t-1) - u_2(t)(b_{32} + g_{32}x_3(t)) + a_{32}x_2(t) + a_{33}x_3(t) + a_{131}x_2^2(t) + a_{132}x_2(t)x_3(t) + a_{133}x_3^2(t) \quad (25)$$

$$\begin{aligned} x_4(t) = {} & u_1(t)\left(-b_{41} + g_{41}^{(1)}x_2(t) + g_{42}^{(1)}x_3(t) - g_{43}^{(1)}x_4(t)\right) \\ & + u_2(t)\left(-b_{42} + g_{44}^{(1)}x_2(t) + g_{45}^{(1)}x_3(t) - g_{46}^{(1)}x_4(t)\right) \\ & + u_3(t)\left(-b_{43} + g_{47}^{(1)}x_2(t) + g_{48}^{(1)}x_3(t) - g_{49}^{(1)}x_4(t)\right) \\ & + x_4(t-1) \end{aligned} \quad (26)$$

Using (23) we can write,

$$\frac{\partial x_2(t)}{\partial u_1(t)} = \frac{-b_{21} + g_{21}^{(1)}x_2(t)}{1 - g_{21}^{(1)}u_1(t)} \quad (27)$$

Similarly, we can write for other state equations (24)-(26)

$$\frac{\partial x_2(t)}{\partial u_1(t)} = \frac{-b_{21} + g_{21}^{(1)}x_2(t)}{1 - g_{21}^{(1)}u_1(t)} \quad (28)$$

$$\frac{\partial x_3(t)}{\partial u_2(t)} = \frac{-b_{32} - g_{31}^{(1)}x_3(t)}{1 + g_{31}^{(1)}u_2(t)} \quad (29)$$

$$\frac{\partial x_4(t)}{\partial u_3(t)} = \frac{-b_{43} + g_{47}^{(1)}x_2(t) + g_{48}^{(1)}x_3(t) - g_{49}^{(1)}x_4(t)}{1 + g_{49}^{(1)}u_3(t)} \quad (30)$$

Similarly, using (3), one can write

$$\frac{\partial u_1(t)}{\partial v_1(t)} = -\vartheta_1(t)x_1(t) \quad (31)$$

$$\frac{\partial u_1(t)}{\partial v_2(t)} = \vartheta_1(t)x_2(t) \quad (32)$$

Substituting (12), (19), (27) and (31) into (11), we get

$$\Delta v_1 = -\beta\left(y_1(t) - o_1(t)\right)\left(1 + 2w_{1,1}x_1(t)\right) * \left(\frac{-b_{11} - g_{11}^{(1)}x_1(t)}{1 + g_{11}^{(1)}u_1(t)}\right)\left(-\vartheta_1(t)x_1(t)\right) \quad (33)$$

Substituting (12), (20), (28) and (32) into (11), we get

$$\Delta v_2 = -\beta\left(y_2(t) - o_2(t)\right)\left(1 + w_{1,2}x_1(t)\right) * \left(\frac{-b_{21} + g_{21}^{(1)}x_2(t)}{1 - g_{21}^{(1)}u_1(t)}\right)\left(\vartheta_1(t)x_2(t)\right) \quad (34)$$

Similarly, the change in weights Δv_m, $m=3,4,...29$ can be obtained by substituting (19)-(22) and (27)-(30) into (11).

The updated input transformation weights on substituting (33), (34) in (11) and (10) we get

$$v_1(t+1) = v_1(t) - \beta\left(y_1(t) - o_1(t)\right)\left(1 + 2w_{1,1}x_1(t)\right) * \left(\frac{-b_{11} - g_{11}^{(1)}x_1(t)}{1 + g_{11}^{(1)}u_1(t)}\right)\left(-\vartheta_1(t)x_1(t)\right)$$

$$v_2(t+1) = v_2(t) - \beta\left(y_2(t) - o_2(t)\right)\left(1 + w_{1,2}x_1(t)\right) * \left(\frac{-b_{21} + g_{21}^{(1)}x_2(t)}{1 - g_{21}^{(1)}u_1(t)}\right)\left(\vartheta_2(t)x_2(t)\right)$$

Similarly, the weight updates $v_m(t+1)$, $m=3,4,....29$ can be obtained.

IV. ILLUSTRATIV E EXAMPLE

Let

$$\dot{x} = Ax + Bu + f^{(2)}(x) + g^{(1)}(x)u + O^{(3)}(x,u) \quad (35)$$

where

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -12 & 0 & 0 \\ 0 & 1.4 & -0.13 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -11 & 0 & 0 \\ 0.29 & 0 & 0 \\ 0 & -0.04 & 0 \\ -0.33 & -0.83 & -4.3 \end{bmatrix}$$

$$f^{(2)}(x) = \begin{bmatrix} 0 \\ 0.64x_2^2 \\ x_2^2 - 0.25x_3x_2 - 0.72x_3^2 \\ 0 \end{bmatrix}$$

$$g^{(1)}(x) = \begin{bmatrix} 0.4x_1 & 0 & 0 \\ -0.2x_2 & 0 & 0 \\ 0 & -5.7e^{(-4)}x_3 & 0 \\ b_{141}^{(1)}(x) & b_{142}^{(1)}(x) & b_{143}^{(1)}(x) \end{bmatrix}$$

$$b_{141}^{(1)}(x) = -0.037x_2 - 0.016x_3 - 0.037x_4$$

$$b_{142}^{(1)}(x) = 1.7x_2 + 0.93x_3 + 1.4x_4$$

$$b_{143}^{(1)}(x) = -0.58x_2 - 0.28x_3 - 0.16x_4$$

The coordinate change (2) upon solution of homological equation is given by

$$y = x + \phi^{(2)}(x) \qquad (36)$$

where

$$\phi^{(2)}(x) = \begin{bmatrix} 1.06x_1^2 - 10.7x_2^2 \\ 0.63x_2^2 - 0.028x_1^2 \\ 0 \\ 0 \end{bmatrix}$$

The input transformation

$$u = (I + \beta^{(1)}(x))\vartheta + \alpha^{(2)}(x) \qquad (37)$$

where

$$\alpha^{(2)}(x) = \begin{bmatrix} 1.22x_1^2 + 23.73x_2^2 \\ 0.97x_1^2 + 3.1x_2^2 - 6.25x_2x_3 - 18x_3^2 \\ \alpha_3^{(2)}(x) \end{bmatrix}$$

where

$$\begin{aligned} \alpha_3^{(2)}(x) = {} & -0.26x_1^2 - 0.0037x_1x_2 - 0.0037x_1x_3 \\ & -0.0017x_1x_4 - 2.57x_2^2 + 1.24x_2x_3 \\ & +0.012x_2x_4 + 3.46x_3^2 + 0.0049x_3x_4 \\ & -0.0009x_4^2 \end{aligned}$$

$$\beta^{(1)}(x) = \begin{bmatrix} -2.09x_1 - 0.57x_1 & 0 & 0 \\ 0 & -0.14x_3 & 0 \\ \beta_{31}^{(1)}(x) & \beta_{32}^{(1)}(x) & \beta_{33}^{(1)}(x) \end{bmatrix}$$

where

$$\beta_{31}^{(1)}(x) = 0.1445x_1 + 0.035x_2 - 0.0037x_3 - 0.009x_4$$

$$\beta_{32}^{(1)}(x) = -0.0256x_2 - 0.053x_3$$

$$\beta_{33}^{(1)}(x) = -0.135x_2 - 0.065x_3 - 0.037x_4$$

The linearized state space model of the TPS (38) is obtained by applying (36) and (37) into (35).

$$\begin{aligned} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = {} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -12 & 0 & 0 \\ 0 & 1.4 & -0.13 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \\ & + \begin{bmatrix} -11 & 0 & 0 \\ 0.29 & 0 & 0 \\ 0 & -0.04 & 0 \\ -0.33 & -0.83 & -4.3 \end{bmatrix} \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \\ \vartheta_3 \end{bmatrix} + O^{(3)}(y, \vartheta) \end{aligned} \qquad (38)$$

$O^{(3)}(y,\vartheta)$ corresponds to terms of order three or more involving y and ϑ.

V. SIMULATION RESULTS

Fig.1. shows the nonlinear system together with the linearizing transformations implemented in Matlab simulink environment. The update law block continuously monitors the error and the weights are updated for the output and input transformation as shown in fig.1. Various training runs are made on the simulation using different step inputs. A typical error convergence is as shown in fig.2. In this case, the mean squared error for 300 iterations was found to be 0.0047 which is below the stopping value of 0.005 used. After tuning is completed the tuned system is separated and tested for linear response with different inputs. The system was tested with step inputs. Fig.3 & 4 show the response of y_1 and y_2 for step inputs ϑ_1=0.0001, 0.0002, 0.0003 & 0.0004 with ϑ_2=0 and ϑ_3=0. Fig.5. shows the response of y_3 for step inputs ϑ_2=0.001, 0.002, 0.003 & 0.004, ϑ_1=0.001 and ϑ_3=0. Fig.6. shows the response of y_4 for step inputs ϑ_3=0.001, 0.002, 0.003 & 0.004, ϑ_1=0.001 and ϑ_2=0.001. From fig.3-6, it is clear that the tuned system exhibits a uniform response to inputs which is typical of a linear system.

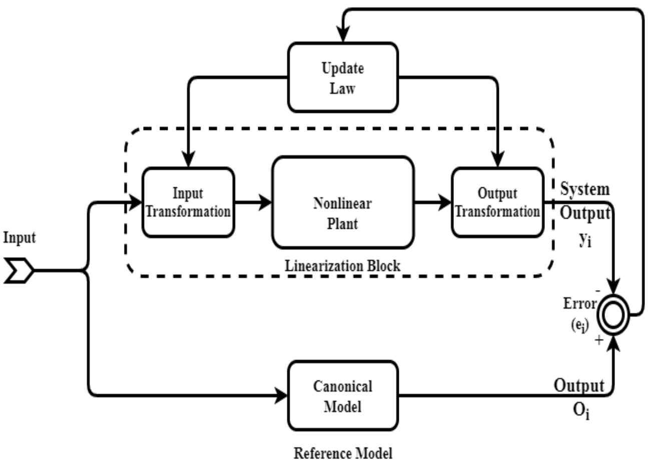

Fig. 1. Tuning transformation coefficients

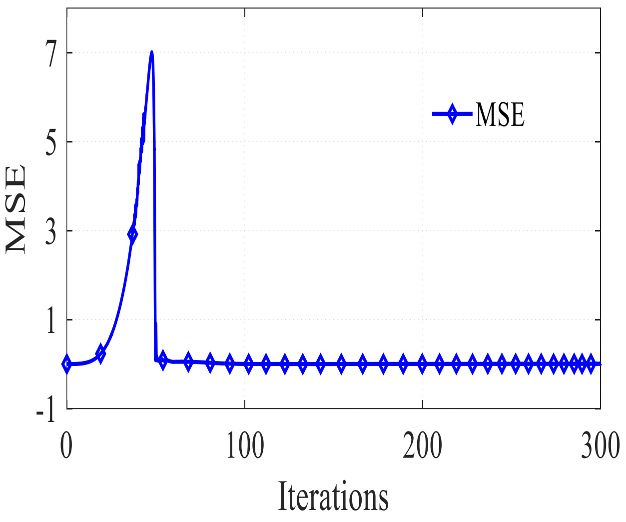

Fig. 2. Mean Squared Error of yi.

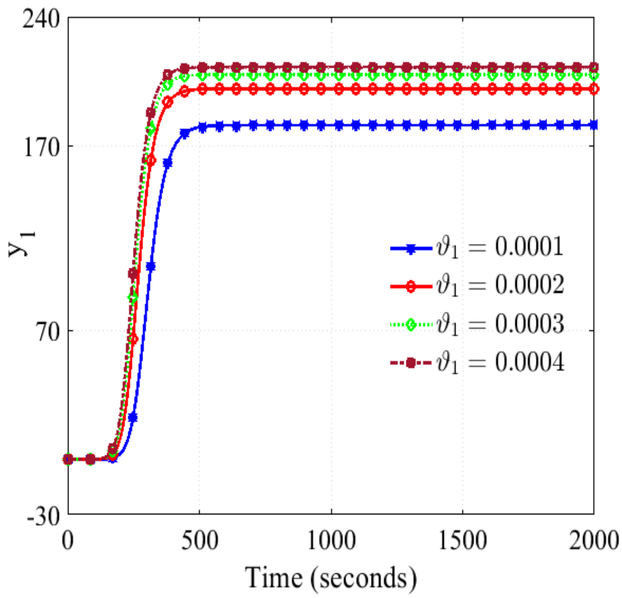

Fig. 3. Linearized response (y1) after tuning weights.

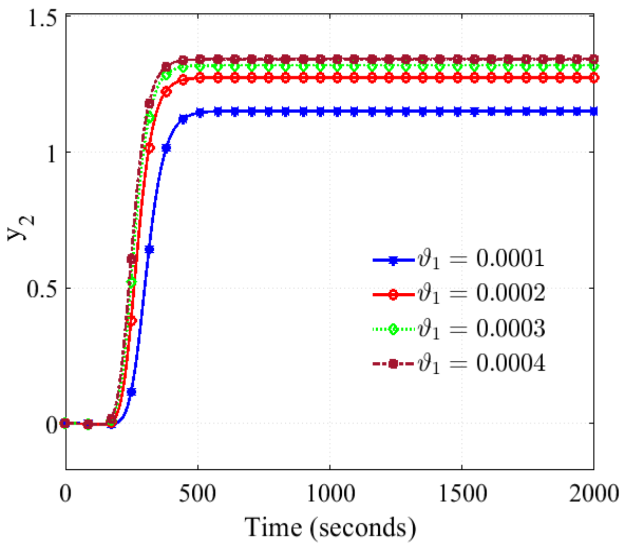

Fig. 4. Linearized response (y2) after tuning weights.

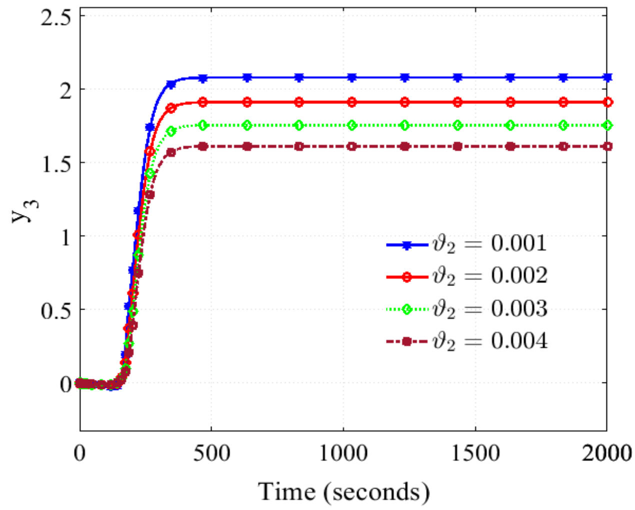

Fig. 5. Linearized response (y3) after tuning weights.

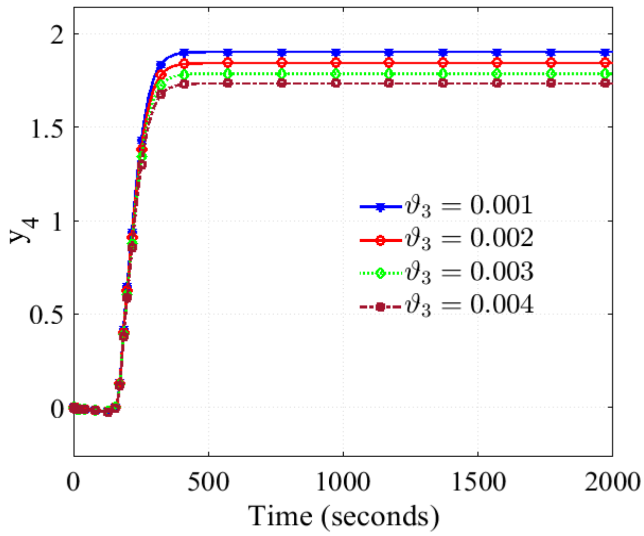

Fig. 6. Linearized response (y4) after tuning weights.

VI. CONCLUSION

In this paper, we show how following the work of Narendra and his co-workers, linearizing transformations can be tuned against a representative canonical model thus completely neutralizing any residual nonlinearities which may be present in the system upon the application of linearizing transformations. This method can also be applied to the case of a plant which differs from its nonlinear model. Linearizing transformation is first applied to the nonlinear model and then the transformations are tuned with the plant output but using the model equations for the tuning formula. This constitutes our future work.

REFERENCES

1. W. Kang and A. J. Krener, "Extended quadratic controller normal form and dynamic state feedback linearization of nonlinear systems," *SIAM J. Control Optim.*, vol. 30, no. 6, pp. 1319–1337, Nov. 1992.
2. K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

3. K. S. Narendra and K. Parthasarathy, "Neural networks and dynamical systems," *International Journal of Approximate Reasoning*, vol. 6, no. 2, pp. 109–131, Feb. 1992.
4. A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks: controllability and stabilization," *IEEE Trans. Neural Netw.*, vol. 4, no. 2, pp. 192–206, Mar. 1993.
5. A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks. II. Observability, identification, and control," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 30–42, Jan. 1996.
6. L. Chen and K. S. Narendra, "Nonlinear adaptive control using neural networks and multiple models," *Automatica*, vol. 37, no. 8, pp. 1245–1255, Aug. 2001.
7. S. Janakiraman and R. Devanathan, "Quadratic linearization of three phase horizontal gravity separator," in *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)*, Kollam, 2017, pp. 1–6.
8. S. Janakiraman and R. Devanathan, "An Analysis of Quadratic Linearization of Three Phase Horizontal Gravity Separator," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, Thessaloniki, 2018, pp. 1006–1011.
9. P. Ayalur Krishnamoorthy, K. Vijayarajan, and D. Rajagopalan, "Application of Quadratic Linearization for the Control of Permanent Magnet Synchronous Motor," *Advances in Power Electronics*, vol. 2011, pp. 1–11, 2011.
10. R. Devanathan, "Necessary and sufficient conditions for quadratic linearization of a linearly controllable system," *International Journal of Control*, vol. 77, no. 7, pp. 613–621, May 2004.
11. P. Ayalur Krishnamoorthy, K. Vijayarajan, and R. Devanathan, "Explicit Necessary and Sufficient Conditions for Quadratic Linearization: Explicit Necessary and Sufficient Conditions for Quadratic Linearization," *Asian J Control*, vol. 15, no. 1, pp. 225–236, Jan. 2013.

AUTHORS PROFILE

S. Janakiraman completed his undergraduate studies B.E.[Instrumentation and Control Engineering] from University of Madras and he completed post graduate M.E.[Applied Electronics] from Anna University Chennai. He is presently working as an Assistant Professor in Hindustan Institute of Technology and Science, Chennai. He is a member of IEEE, ISA and life member of ISTE. He has published several articles in international conferences and journals.

Rajagopalan Devanathan received his Ph.D. and M.Sc. (Engg) from Queen's University, Kingston, Ontario, Canada, and his B.E. and M.E. degrees from Indian Institute of Science, Bangalore, India. He worked in Electronics Commission, Govt. of India. Dr. Devanathan was faculty of Nanyang Technological University (NTU), Singapore, for over two decades. Currently he is serving as Professor-Emeritus in Hindustan Institute of Technology and Science, Chennai. Dr. Devanathan has published over 120 papers in international and national conference proceedings and journals. He is Life Senior Member of IEEE. He has received awards from IEEE Education Society and NTU. He has chaired and co-chaired international conferences organized under IEEE.