

VEHICLE INTEGRATED STABILITY CONTROL USING HYBRID FUZZY C-MEAN CLUSTERING-ADAPTIVE BACK PROPAGATION SCHEME

M.Harly^(1,2), I. N. Sutantra⁽¹⁾ Mauridhi H P⁽¹⁾

¹⁾ PostGraduate School of Electrical Engineering
Insitut Teknologi Sepuluh Nopember
E-mail : hery@ee.its.ac.id

²⁾ Vocational Education Development Center
E-mail: mohamedharly@yahoo.com

ABSTRACT

Most vehicles accident were caused by instability vehicle motion. The instability just occurs cause four former integration controls (Feed-forward control, H^∞ control, Nonlinear Predictive control, Robust control) can not adapt to driving condition (vehicle, drivers character and environment), which always change their structure and parameter at high speed. This obstacle results the controlled variable of stability such as Yaw-Rate (YR) , Vehicle Side Slip (VSS) , Roll Angle (RA) cannot fulfill control targets, instability vehicle direction and then cause accident.

This paper propose a new integration control design exploits combined Multi Dimension Fuzzy C-Mean Clustering (MDFC) and Adaptive Back-propagation Control (ABC). ABC consist of NN-Plant and NN-Controller. Architecture NN-Plant results from genetically optimized hybrid fuzzy neural network (gHFNN) while NN-Controller from multi-layer neural network (MLN) with single hidden layer. Instead of three former vehicle dynamics model like decoupling of linear to nonlinear plant, two dimension to three dimension plant and ESP-4WS-AS plant, which are imprecise to build a driving condition model, will be proposed a “Three in one dynamics system (TODS)” plant, which can represents the best model of interaction among vehicle dynamics, driver characters and environment. The solving methodology are arrange like follow, First step vary 2.187.000 TODS real time data’s is realized by test drive a vehicle equipped with electronics stability program(ESP), four wheel steering (4 WS) and active suspension (AS), which covers 6 dimensions vector (YR from yaw-rate sensor, VSS from lateral acceleration sensor, RA from body level sensor, T_{DYC} from ESP actuator, δr from rear steering actuator, M_{AS} from suspension actuator). Data’s inputs to MDFC to cluster 810 centers. Second step is the training process to update the optimized architecture and parameters of NN-Plant uses all centers based on genetic algorithm (GA), LSE and BP. Third step is the training process to update optimized NN-Controller's architecture and parameters uses input reference and desired input of updated NN-Plant based on constructive back propagation (CBP). Fourth step is validating and testing of ABC use all data’s of TODS.

An experiment and simulation is completely setup to prove the performance of Hybrid MDFC-ABC integrated control, when is compared with four former integrated control method to control TODS. The simulation result in the form of rank shows that topmost sequence performance is Hybrid MDFC-ABC, then robust control, H-infinite control, NLPC, No-integration control and feed-forward control.

KEY WORDS

Three in one dynamics system, Multi Dimension Fuzzy Clustering & Adaptive Back propagation Control

1. Introduction

One of the important parameter in vehicle design is stability of vehicle direction, which is according to the driver’s desire (Output Target). From 1500 accident 877 accident(65%) because of driver factor 301 accident(22%) because of environmental factor and 182 accident(13%) because of vehicle factor.[2] Implicitly can be concluded, that the accident can be happened because of vehicle cannot adapt the character of driver and environment. Stability output vector which is also referred as Vehicle State generally consist of 6 variables[3]. Vehicle state, which related major to stability are Yaw-Rate (YR) , Vehicle Side Slip (VSS) or can be represented with Lateral Velocity (V_y) , Rolling Angle (RA). Vehicle state, Which minor relate to stability are Pitching Angle (PA) , Axial Velocity (V_z) and Longitudinal Velocity (V_x). Vehicle dynamics is very complex because it’s unknown structure and its uncertainty parameter, so that cause the result of integrated control will still has Sum-Error and Error-Square. Currently there are 3 important stability control subsystem that is Electronics Stability Program (ESP/DYC)[12] Active Steering like 4-WS[19]or ESAS[7]and Active Suspension (AS)[15]. Both of them are not integrated or integrated as controlling vehicle stability.

Four integrated method which have been used and also introduced by former researcher. Kozuya Kitajima has designed and simulated two integrated control first is Feed-

forward integration method [11], which coordinate activity of decoupling result of 3 stability control system in serial process depended priority scale. Its output vector is VSS , YR and RA and its control vector are Wheel torque (T_{VDC}), Rolling torque (M_{AS}) and Rear steering angle (δ_r). Performance of Feed-forward lower than which without integration (figure 22) because works alternately and a assumption that other sensor of control system act as disturbance input. Second integration method is H^∞ control [11] which its target to yield a integrated control by minimizing value of H^∞ gain. Optimizing will be done for the transfer function of DYC-4WS-AS decoupling result to disturbance Input. Control vector are T_{VDC} , M_{AS} , δ_r based on output measurement, which passing loops gain. Output vector are VSS , YR , RA . Performance H^∞ integrated control is better than both feed-forward and which without integration at a certain frequency area. Its weakness is that in determination of appropriate loop gain at low frequency area where trouble emerges to influence ability of robust performance. If we decide imprecise high frequency area, where indefinite model structure and uncertainty parameter are happened will influence its robust stabilization. Result of control vector output of vehicle is still big, if setting of balance area which controls him in external area of its robust design.

Third Integration method was introduced by ShiniChirnto through application of Nonlinear Predictive Control (NLPC) [9] for ESP and 4WS. State variable is $X=(X_1-X_2)$. Its state variable will follow goals trajectory of S_1 of partition of trajectory state $S = (S_1, S_2)$, with error tracking state ($e = X_1 - S_1$). Control vector are chosen to be four wheel torque (T_i) and rear steering gain (ξ_i). The approach of integration control this is to get a desired control vector by optimizing of tracking error function (Performance Index) to control vector. Weakness of NLPC is difficulty in modeling of vehicle dynamics plant, because it contains unmodeled structure at certain frequency. This mistake will result imprecise prediction so that its performance not be perfectly.

Fourth Integration method was developed by M. Lakehal Ayat is referred as robust control [8] and used to integrate ESP and AS. Its decoupling process is based on model 2-D (V_x, V_y, YR) to 3-D (PA, RA, V_z). Weakness of robust integration which applied in here are that the performance of tracking to only be good at low V_x until 100 km / hour, noise sensitivity is good to all frequency whereas disturbance rejection is only good at low frequency until middle. The uppermost weakness is that sensitivity to the change of nonlinearity plant at high frequency so that its control performance is not accurate. Robust performance is better compared to the three others.

In general weakness of the four former integration control systems can be explain like follow, first difficulty to identify dynamics of driving condition itself which always change its structure and parameter over time such as vehicle (unknown vehicle dynamics function), environment (unpredictable wheel-ground) and driver (drivers character), which in this research is referred as Tree in One Dynamics

System (TODS). Second, this TODS certainly cannot be reckoned by former integrated control system. (figure 1).

Therefore, this paper introduce an integration control based on Artificial Intelligent which combine fuzzy clustering and Adaptive Back-propagation Control (ABC) [4] ABC can be realized by simulation of Model Reference Neural Network (MRNNC). MRNNC consists of Neural Network Plant (NNP) and of Neural Network Controller (NNC) The following strategy scheme will be provided, first identify TODS by driving test to get 2.187.000 data sets as input of Advanced Neural Fuzzy such as Multi Dimension Fuzzy C-Means Clustering (MDFC) [5]. Clustering process is took place to get 810 convergence centers. Data's to MDFC result from real time test drive according to the variation of level driver ability and condition of environment (wheel-ground, maneuver etc) like seen at figure 4. Centers are used to train NNP. To train NNC use reference data input and desired output such as the input of updated NNP while its output is the same value with reference data. Architecture of NNP is build from an advance architecture of genetically optimized hybrid fuzzy neural network (gHFNN), which construct from combined FNN and PNN [6]. Optimized architecture of gHFNN designed by genetic algorithms (Gas) Learning to update parameter of FNN as a premise part of rule-base structure of gHFNN use standard back propagation (BP) and to update parameter of PNN as a consequent part of rule-base structure of gHFNN use standard least square method (LSE). NNC results from multi layer network with a single hidden layer (MLN) [13] Learning for MLN to get the optimum number of hidden layers node and parameter use constructive back propagation (CBP).

Performance of MDFC-ABC integration control results smaller yaw rate error ($\Delta YR < 3^\circ/\text{sec}$), Vehicle Side Slip error smaller ($\Delta VSS < 5^\circ$), Phase Plane (VSS To YR Phase $\Rightarrow 0$), Roll Angle error and its rate smaller ($\Delta RA < 1,5^\circ$, $d(RA) / dt \Rightarrow 0$), after passing simulation test then four former integration control method (Feed-forward, H^∞ , NLPC, Robust).

This paper will be arranged with the following sequence. First session is introduction to review about kind of handling problems for which vehicle were equipped by former integration control, our proposed solution and how to validate the result of experiment and simulation. In second chapter will be explained methodology such as the different approach between former modeling method and identification TODS by MDFC-NNP, how to designed genetically optimized architecture of ABC/MRNNC and its parameter learning process. Third session review the simulation and discussion. Finally fourth session is the conclusion.

2. Former Vehicle Dynamics Model

The former control integration method formulate the vehicle dynamics in determined system, it means to be assumed that all pre-viewable disturbance such as cornering stiffness, damper constant, spring stiffness etc.

don't change over time. There are 3 scheme to develop a stability vehicle model, first by decoupling of 2-dimension system and 3-dimension system and use a robust integration control, second by decoupling of linear and nonlinear control, then for integration control apply a nonlinear predictive control (NLPC), third by decoupling of ESP, 4WS and AS to get vehicle stability model and use for integration control feed-forward or H^∞ control system. (figure 1).

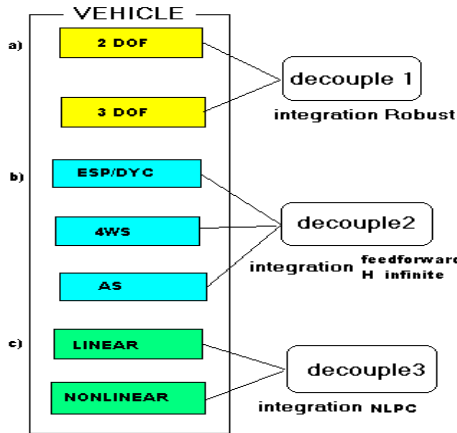


Figure 1 : Three Schemes of Former Vehicle Modeling

Decoupling produce a state space function with state variable (vehicle state), $X = YR, VSS, RA$ and control variable $U = T_{VDC}, \delta_r, M_{AS}$ are shown in figure 2. The form of state space is symbolized like below

$$A \cdot \dot{X} = B \cdot X + C \cdot U \quad (1)$$

$$Y = I \cdot X$$

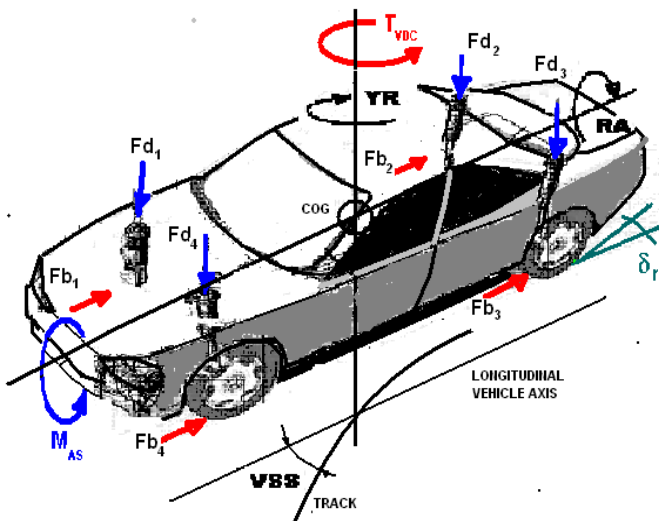


Figure 2 : Vehicle State and Control Variable of Subsystem (DYC, 4WS, AS)

or can be completely wrote in form of linear matrix [11]

$$\begin{bmatrix} 0 & M & M \cdot h & 0 \\ I_{xx} & 0 & I_{xz} & 0 \\ I_{xz} & M \cdot h & I_{zz} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{YR} - \dot{YR}_s \\ \dot{V} \\ \dot{RA} \\ \dot{RA} \end{bmatrix} = \begin{bmatrix} \frac{b \cdot C_{ar} - a \cdot C_{af} - M_s}{V_x} - M_s & -\frac{C_{af} + C_{ar}}{V_x} & 0 & 0 \\ -\frac{a^2 \cdot C_{af} + b^2 \cdot C_{ar}}{V_x} & \frac{b \cdot C_{ar} - a \cdot C_{af}}{V_x} & 0 & 0 \\ \frac{V_x}{M_s \cdot h} & 0 & -K_p & M_s \cdot g \cdot h - K_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta_r \\ T_{VDC} \\ M_{AS} \\ \delta_r \end{bmatrix} + \begin{bmatrix} C_{af} - \frac{V_x}{L \left(1 + \frac{K_{us}}{L} V_x^2\right)} \cdot \left(\frac{b \cdot C_{ar} - a \cdot C_{af} - M_s}{V_x} - M_s\right) \\ a \cdot C_{af} - \frac{V_x}{L \left(1 + \frac{K_{us}}{L} V_x^2\right)} \cdot \left(-\frac{a^2 \cdot C_{af} + b^2 \cdot C_{ar}}{V_x}\right) \\ -\frac{V_x}{L \left(1 + \frac{K_{us}}{L} V_x^2\right)} \cdot (-M_s \cdot h \cdot V_x) \\ 0 \end{bmatrix} \cdot \begin{bmatrix} \delta_r \\ T_{VDC} \\ M_{AS} \\ \delta_r \end{bmatrix} \quad (2)$$

Where , the constant parameters are :

- M = unsprung mass
- M_s = sprung mass
- C_{ar} = cornering stiffnes rear wheel
- C_{af} = cornering stiffnest front wheel
- L = wheel base
- K_{us} = understeer index
- a = distance center of weight to front axle
- b = distance center of weight to rear axle
- V_x = longitudinal velocity
- h = distance center of weight to ground
- I_{xx} = inertia moment of rolling
- I_{xz} = inertia moment of yaw-roll
- I_{zz} = inertia moment of yawing
- K_p = rolling damper constant
- K_θ = rolling stiffnes constant
- T_f = distance between left to right wheel
- R_w = radius of wheel

Previously three modeling scheme (decoupling 1, 2, 3) will be imprecise, especially in high frequency, when the vehicle structure (number of state space equation) and its parameter (matrix constant) start to change. Another side the direction of vehicle moving should not only be depended on vehicle but also driver character and environment.

To solve this weakness, this paper propose a new solution, in modeling approach by identification of TODS using combined MDFC-NNP.(figure 3).

3. Solving Methodology

MDFC-ABC integrated control design process to control TODS (figure 9) is realized by following algorithm, is illustrated by figure 3. The explanation of every step are also written clearly to make the experiment briefly.

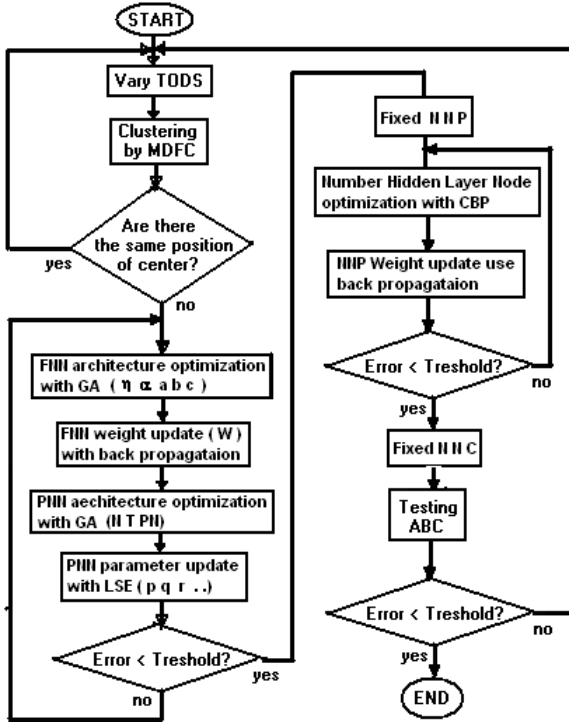


figure 3: Flow Chart Design of MDFC-ABC

4. Clustering TODS by MDFC

To realize the Identification process we have developed a prototype car, which was equipped by ESP, 4 WS and AS (fig 6). For communication of input-output each ESP-ECU, 4WS-ECU and AS-ECU[10] were applied a network transceiver under SAE J1939[20] with 0-8 bytes data field length, 1 Mbps max. bit rate, max. bus length > 40 m and max. number node >16 (fig 5). Scheme of plant identification is shown figure 4.

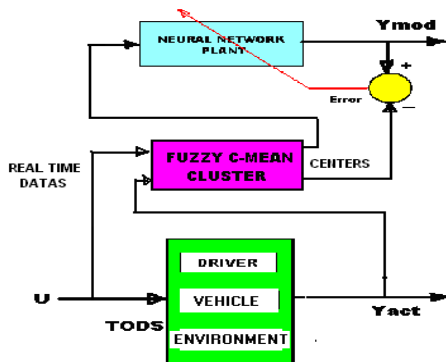


Figure 4: Combined MDFC-NNP as proposed model (TODS)



Figure 5: ESP,4WS and AS ECUs provided by network transceiver



Figure 6: TODS Prototype Vehicle equipped by ESP,4WS & AS

Entry data for MDFC will be explored by variation of driver level and environment condition. As representative of environment are the variation of maneuvers and wheel-ground friction. (Figure 7 & table 1). The popular maneuver are used, standard of J-Turning, Skyhook, S-Turning which are varied to kind level of wheel-ground friction. For classifying the drivers level require 10 housewife's for beginners, 10 taxi drivers for intermediates and 10 racers are chosen for masters. PCI card was used to transfer data from network for collecting input (sensors) and output (actuators) of each ECU's. Yaw rate sensor, lateral-longitudinal acceleration sensor, body level sensor are processed to get vehicle state such as YR, VSS, RA. Actuators signal like ESP hydraulic PWM, rear steering drive motor voltage, orifice stepper motor voltage for suspension damper are measured and processed as control variable T_{VDC} , δr and M_{AS} .

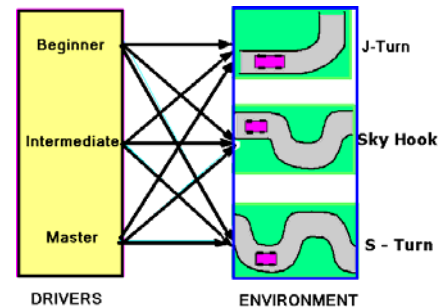


Figure 7: Variation to generate the data of TODS

Therefore to vary the data of TODS, hence the driver level and environment are used as guidance to list the condition of vehicle operation. Every driving condition (C_{i1} or C_{i2} or C_{i3}) is conducted by 10 different drivers who have the same level. Each driver should do maneuver 9 times every condition..

Table 1 : Setup for Variation of Driving Condition

ENVIRONMENT CONDITION		LEVEL OF DRIVER		
TURNING	WHEEL-GROUND FRICTION	BEGINNER	INTERMEDIATE	MASTER
			C_{11}	C_{12}
J-Manuver	Smooth asphalt	C_{21}	C_{22}	
	Ribbed asphalt	C_{31}	C_{32}	
Sky-hook	Gravel		Center Candidate	
	Smooth asphalt			
	Ribbed asphalt			
S-Manuver	Gravel			
	Smooth asphalt			
	Ribbed asphalt	C_{91}		C_{93}

Data set will be recorded every 0,1 m displacement of vehicle moving from starting until end the maneuver by wheel speed sensors. If distance of turning 90 m then a trip results 900 recorded data's. It means $27 \times 10 \times 9 \times 900 = 2.187.000$ sampling took place. Decide the number of candidate center are based on the critical operation of the driving occurs to each condition (C_{ij}) assumed 30 point. Finally we have $27 \times 30 = 810$ centers. Input data entry to MDFC are $X_j = \{YR_j, VSS_j, RA_j, T_{VDC(j)}, \delta r_j, M_{AS(j)}\}$ ($j = 1 : 2.187.000$) are shown in figure 8.

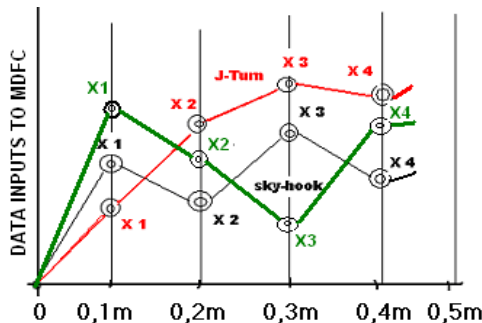


Figure 8: Variance of a driver does 3 turning in gravel, which are recorded every 0,1 m vehicle moving.

MDFC outputs the 810 significant centers after computational calculation based on the formula 3up to 5

Cost function for MDFC is described as follow:

$$J(U, C_1, \dots, C_c) = \sum_{i=1}^c J_i = \sum_{i=1}^c \sum_{j=1}^n U_{ij}^m \cdot d_{ij}^2 \quad (3)$$

To calculate the position of center is formulated [5]

$$C_i = \frac{\sum_{j=1}^n U_{ij}^m \cdot X_j}{\sum_{j=1}^n U_{ij}^m} \quad (i = 1: 810, j = 1: 2.187.000) \quad (4)$$

To fulfill element of matrix U we use a formula:

$$U_{ij} = \frac{1}{\sum_{k=1}^m \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \quad (5)$$

Algorithm to find a convergence centers, then it should flow like below:

Step 1

Fulfill initial condition value matrix *weight* U by a small random number [0:1].

Step 2:

Calculate all center positions C_i ($i = 1..810$) by using formula 4

Step 3:

Use formula 3 to calculate Cost Function J , stop calculation if the value of cost function lower than given threshold if higher than given threshold goes to step 4.

Step 4

Find new matrix U using formula 5 and then back to step 3.

If there are two or more center have the same position, repeat vary TODS, although we have never been met this obstacle.

After matrix U convergence we get 810 centers, which already to be inputted for training NNP.

5. Design Integrated Control

To design integrated control we exploits the adaptive back propagation control (ABC). ABC is developed from NNP to identify plant and NNC to control updated NNP based on reference value. Exactly NNC is the inverted of updated NNP. Because of complexity of TODS, we build NNP based on architecture of genetically optimized hybrid fuzzy neural network gHFNN. NNC as control part of ABC results from adaptive multi layer neural network (MLN), which always enable to adapt the change of TODS every time.

Fig 9 is the expression of ABC scheme with black arrows as the forward flow of control process and red arrow symbolize back propagation learning to update the new parameter (joint weight or synapse) and optimization architecture by GA_S .

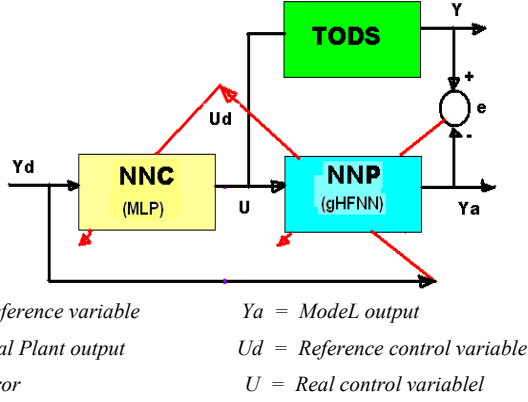


Figure 9: Scheme of Adaptive Back propagation Control (ABC)

5.1 Design Neural Network Plant (NNP)

NNP results from a optimized genetically hybrid fuzzy neural network (gHFNN). Then gHFNN consist of fuzzy relation neural network (FR-NN) as supporting of formation of the premise part of the rule-based structure of the gHFNN and polynomial neural network (PNN) as the consequence part of the rule-based structure of the gHFNN.

Step 1 FR-NN architecture optimization with GA (gFNN)

In order to enhance the of the FR-NN we use GA_s to adjust learning rate ($\hat{\eta}$), momentum coefficient ($\hat{\alpha}$) and parameter of gauss membership function (a, b, c). It means we have 5 items or 5 sub-chromosome, which should be structured in 5 bit groups

Fig.10 is shown how to arrange the bits and its normalization to optimized parameter values.

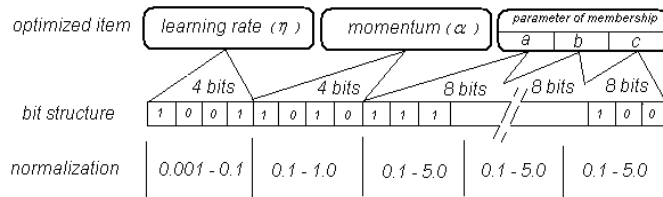


Figure 10: Mapping the gFNN optimized item on a chromosome

a) Learning rate ($\hat{\eta}$)

The 4 bits of the 1st sub chromosome are assigned to the binary bits for the selection of the learning rate. The 4 bits are decoded into a decimal format. The decimal value obtained is normalized into [0.001-0.1] and rounded off. The normalized integer value is given as the learning rate.

b) Momentum coefficient ($\hat{\alpha}$)

The 4 bits of the 2nd sub chromosome are assigned to the binary bits for the selection of the momentum coefficient. The 4 bits are decoded into a decimal format. The decimal value obtained is normalized into [0.1-1.0] and rounded off. The normalized integer value is given as the momentum coefficient.

c) Gauss membership function parameters (a, b, c)

Each 8 bits of the 3rd, 4th and 5th sub chromosome are assigned to the binary bits for the selection of the membership parameter a, b, c . Each 8 bits are decoded into a decimal format. The 3 decimal values obtained are normalized into [0.1-5.0] and rounded off. The 3 normalized integer values are given as a, b and c .

Step 2 Update the output joint weight of gFNN

If n is the number of selected rules, A is the updated gauss membership function, X is input antecedent vector and Y is output vector W is the output weight, we can express if then rule like below [6]

$$R_1 \quad \text{if } X_{1(1)} \text{ is } A_1 \dots \text{and } X_{k(1)} \text{ is } A_k \\ \text{then } C_{y1} = W_{01} + W_{11}X_1 + W_{k1}X_k$$

$$R_i \quad \text{if } X_{1(i)} \text{ is } A_{1i} \dots \text{and } X_{k(i)} \text{ is } A_{ki} \\ \text{then } C_{yi} = W_{0i} + W_{1i}X_i + W_{ki}X_k$$

$$R_n \quad \text{if } X_{1(n)} \text{ is } A_{1n} \dots \text{and } X_{k(n)} \text{ is } A_{kn} \\ \text{then } C_{yn} = W_{0n} + W_{1n}X_1 + W_{kn}X_k$$

Finally Y can be found by inference system

$$\bar{Y} = \sum_{i=1}^n f_i \\ = \frac{\sum_{i=1}^n \mu_i \cdot (W_{0i} + W_{1i} \cdot X_i + W_{ki} \cdot X_k)}{\sum_{i=1}^n \mu_i} \quad (6) \\ = \sum_{i=1}^n \bar{\mu}_i \cdot (W_{0i} + W_{1i} \cdot X_i + W_{ki} \cdot X_k) \quad (7)$$

Based this structure to update jointed weight can be formulated based on steepest gradient descend.

$$\Delta W_{ki} = 2 \cdot \eta \cdot (Y - \bar{Y}) \cdot \bar{\mu} \cdot X_k + \alpha \cdot (W_{ki}(t) - W_{ki}(t-1)) \quad (7)$$

Step 3 PNN architecture optimization with GA (gPNN)

Optimization of PNN architecture concerns the selection of the maximum number input variable (N), the polynomial order (T) and number of node in each layer (PN_s). There are 4 items which should be structured in 4 bit groups or sub-chromosomes. Fig.11 is shown the bit arrangement of the sub-chromosome and its normalization to optimized parameter values. We divide the chromosome into 4 sub-chromosomes as shown in the 1st sub-chromosome contains the max number of input variables goes to node, the 2nd sub-chromosome involves the order of the polynomial of the node, and the 3rd sub-chromosome (remaining bits) contains total input variable (n) coming to node at 2nd layer of gPNN represents the number of rule are developed gFNN device. and 4th sub-chromosome is the number of node (W) in 2nd layer of gPNN.

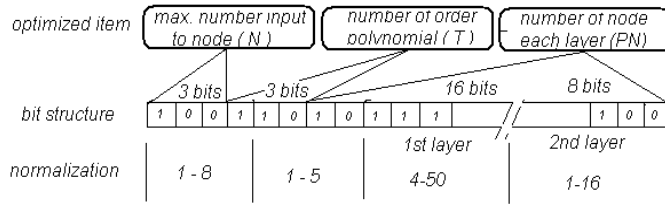


Figure 11: Mapping the gPNN optimized item on a chromosome

We create a population of chromosome for a PN, where each chromosome is a binary vector of bits. All bits for each chromosome are initialized randomly. The output of the layer 5 in the premise structure of the gFNN, is treated as the 1st layer of the consequence structure of gPNN, that is, $x_1=f_1$, $x_2=f_2, \dots, x_n=f_m$.

Each step of the genetic design of the 4 types of the parameters available within the PN is structured as follows:

a) Maximum number of input variables coming to each node in the corresponding layer

The first 3 bit of the given chromosome are assigned to the binary bits for the selection of the number of input variables. The selected 3 bits are decoded into decimal. The above decimal value is converted into [1-8] and rounded off. N denotes the maximal number of input variables entering the corresponding node (PN). The normalized integer value is then treated as the number of input variables (or input nodes) coming to the corresponding node.

b) Polynomial order of each node in each layer.

The 3bits of the 2nd sub chromosome are assigned to the binary bits for the selection of the order of polynomial. The 3bits are decoded into a decimal format. The decimal value obtained is normalized into [1 5] and rounded off. The normalized integer value is given as the polynomial order.

c). Total number n of node in the 1st layer (input layer of gPNN)

The remaining bits are assigned to the binary bits for the selection input variables. The remaining bits are divided by the value obtained such as number node in 1st layer. Each bit structure is decoded into decimal. The decimal value obtained is normalized into [4-50] and rounded off. n is the overall system's input in the 1st layer.

d) Total number W of nodes in 2nd layer or higher to be retained (selected) at the next generation of the gPNN.

The remaining bits are assigned to the binary bits for the selection input variables. The remaining bits are divided by the value obtained such as number node in 1st layer. Each bit structure is decoded into decimal. The decimal value obtained is normalized into [1- 16] and rounded off. W is the overall system's input in the 2nd layer.

The normalized integer value is then taken as the selected input variables while constructing each node of the corresponding layer. Here, if the selected input variables are multiple duplicated, the multiple-duplicated input variables are treated as single input variable.

Step 4

The vector of polynomial $\hat{a} = [p , q , r , \dots]$ coefficients is derived by minimizing the mean squared error between y_i and \hat{y}_i . To evaluate the approximation and generalization capability of a PN produced by each chromosome, we use the following fitness function (FF) like below:

$$FF = \frac{1}{1 - OF}$$

Where object function (OF) is

$$OF = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

The structure of consequence part of gPNN is linear, then to update the polynomial coefficient use least square error (LSE) based on forward learning.

5.2 Forward Learning (LSE)

Input-output data is state by linear function as follow:

$$\bar{X}_k = \bar{F}_{(U)_j} * \alpha_{jk} \quad (k=1..3, j=1..6) \quad (8)$$

Then to update general parameter in epoch z for p-entry data will be formulated as follow [5]

$$\Delta \alpha_z = \left[A_z^T (p^*j) * Q * A_z (p^*j) \right]^{-1} * A_z^T (p^*j) * Q * \Delta y_p \quad (9)$$

where Q is "forgetting factor" matrix λ (0 : 1)
 To generate new populations of the next generation, we carries out selection, crossover, and mutation operation using genetic information and the fitness values. Until the last generation, this step carries out by repeating previously steps such as fig 3
 The termination condition builds a sound compromise between the high accuracy of the resulting model and its complexity as well as generalization abilities.

Step 5

Optimized Architecture of NNP

Architecture of NNP is built gHFNN 1st until 5th layer as supporting of formation of the premise part of the rule-based structure of the gHFNN, 5th until 7th as the consequence part of the rule-based structure of the gHFNN. The input of NNP are $X_k = [TVDC, \delta_r, M_{AS}]$ and $Y_L = [YR_a, VSS_a, RA_a]$ as its output device . (figure 12)

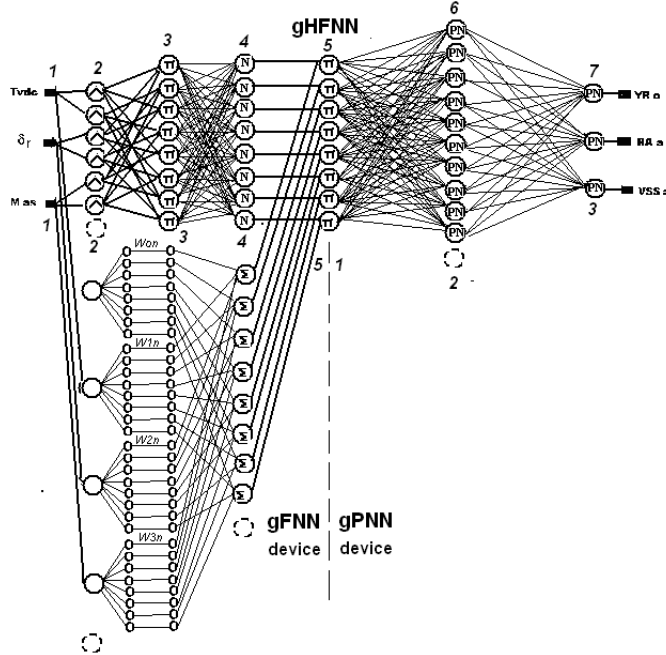


Figure 12: Architecture Adaptive Network of NNP (gHFNN)

Design Neural Network Controller (NNC)

Architecture of NNC

We use a three-layer network structure (MLN) with (L=3) input neuron, (h) hidden layer node and (k=3) output neuron for each layer as illustrated in fig 13. The network output or control input is expressed as formula 10 and 11 [13]

In output layer (3rd layer)

$$X_{k(3)} = \sum_{j=1}^h (\alpha_{kj} \cdot Y_{h(2)}) + b_k \tag{10}$$

And in hidden layer (2nd layer)

$$Y_{h(2)} = \alpha_{Lh} \left(\sum_{L=1}^3 (\alpha_{hL} \cdot Y_{L(1)}) + b_h \right) \tag{11}$$

Where output vector in output layer (3rd layer) is expressed

$$X_{k(3)} = [T_{VDC}, \delta_r, M_{AS}]$$

And input vector in 1st layer is

$$Y_{L(1)} = [YR_a, VSS_a, AS_a]$$

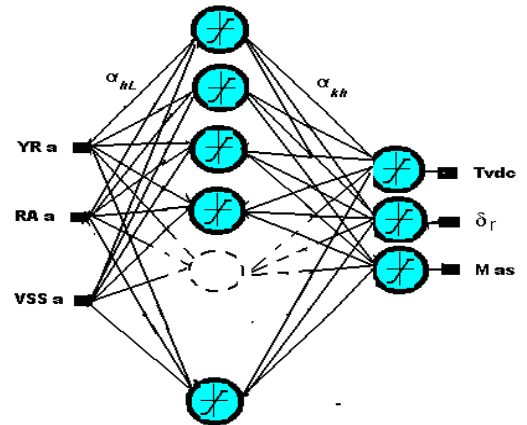


Figure 13 : Architecture Adaptive Network of NNC

Hidden Layer Node Optimization

In order to optimize the number of hidden layer node (h) and update the joint weight for every selected hidden layer node is developed the constructive back propagation algorithm [1]. is shown in fig 14.

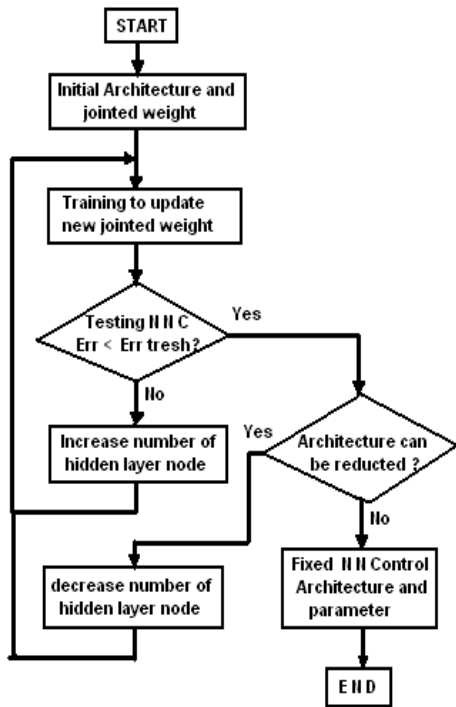


Figure 14: Flow Chart of Integration Control

Learning Process to update parameter NNC (MLN)

Learning process for NNC is selected use back-propagation learning, off-board for every epoch to update new parameters. Learning at backward process uses Gauss-Newton (GN).

$$\text{If } T_{VDC} = X_{1(3)}, \quad \delta_r = X_{2(3)}, \quad M_{AS} = X_{3(3)}$$

$$YR_a = Y_{1(1)}, \quad VSS_a = Y_{2(1)}, \quad AS_a = Y_{3(1)}$$

Non-linear function can be assumed as follow

$$X_{k(3)} = f(\bar{Y}, \bar{\alpha}) \quad \text{and} \quad \bar{X} = (X_{1(3)} \ X_{2(3)} \ X_{3(3)})$$

$$\bar{Y} = (Y_{1(1)} \ Y_{2(1)} \ Y_{3(1)}) \quad (12)$$

Data 810 (p) in epoch z produce overall measure-error

$$E_z = \sum_{p=1}^{810} \sum_{k=1}^3 \left(X_{pk} - \bar{X}_{pk} \right)^2 = \sum_{p=1}^{810} \sum_{k=1}^3 (e_{pk})^2 \quad (13)$$

Error vector per data, which its value are influenced by former parameter (α_{z-1})

$$e_p = \sum_{k=1}^3 [e_{pk}(\alpha_{z-1})] \quad (14)$$

so that its gradient descent in epoch z can be formulated [5]

$$g_z(\alpha_{z-1}) = \frac{\partial E_z}{\partial \alpha} = 2.J^T(\alpha_{z-1}) * e_p(\alpha_{z-1}) \quad (15)$$

Where J is Jacobean matrix of error vector, while the Hessian matrix of overall measure error [5] to be:

$$H_z(\alpha_{z-1}) = 2.[J^T(\alpha_{z-1}) * J(\alpha_{z-1}) + S] \quad (16)$$

$$\text{Value of } S = \sum \left[e_p \cdot \frac{\partial^2 e_z}{\partial \alpha, \partial \alpha^T} \right] \text{ then approach to } 0$$

Cause that the change of parameter based on Gauss-Newton (GN) generally should be described as follow.

$$\Delta \alpha = -[J_z^T(\alpha_{z-1}) * J_z(\alpha_{z-1})]^{-1} . J_z^T(\alpha_{z-1}) * e_p(\alpha_{z-1}) \quad (17)$$

or

$$\Delta \alpha = -H_z^{-1}(\alpha_{z-1}) * g_z(\alpha_{z-1}) \quad (18)$$

G M update new general parameter (epoch z) can be found from the change of general parameter and the effect of Learning Rate η as follow

$$\alpha_z = \alpha_{z-1} + \eta . \Delta \alpha \quad (19)$$

To find signal error each node in layer (fig 15) using Chain Rule Back propagation for every data entry (p) as follow:

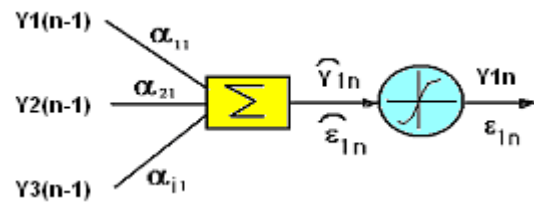


Figure 15: a Neuron of NNC

Signal Error in layer 3(output), which has node k (1...3)

$$\hat{\varepsilon}_{1(3)} = \frac{\partial E_p}{\partial \hat{Y}_{1(3)}} =$$

$$\frac{\partial E_p}{\partial Y_{1(3)}} * \frac{\partial Y_{1(3)}}{\partial \hat{Y}_{1(3)}} = \varepsilon_{1(3)} * \frac{\partial Y_{1(3)}}{\partial \hat{Y}_{1(3)}} =$$

$$-2(X_{1(3)} - Y_{1(3)})Y_{1(3)} \cdot (1 - Y_{1(3)}) \quad (20)$$

$$\begin{aligned} \hat{\varepsilon}_{2(3)} &= \frac{\partial E_p}{\partial \hat{Y}_{2(3)}} = \\ & \frac{\partial E_p}{\partial Y_{2(3)}} * \frac{\partial Y_{2(3)}}{\partial \hat{Y}_{2(3)}} = \varepsilon_{2(3)} * \frac{\partial Y_{2(3)}}{\partial \hat{Y}_{2(3)}} = \\ & -2(X_{2(3)} - Y_{2(3)})Y_{2(3)} \cdot (1 - Y_{2(3)}). \end{aligned} \quad (21)$$

$$\begin{aligned} \hat{\varepsilon}_{3(3)} &= \frac{\partial E_p}{\partial \hat{Y}_{3(3)}} = \\ & \frac{\partial E_p}{\partial Y_{3(3)}} * \frac{\partial Y_{3(3)}}{\partial \hat{Y}_{3(3)}} = \varepsilon_{3(3)} * \frac{\partial Y_{3(3)}}{\partial \hat{Y}_{3(3)}} = \\ & -2(X_{3(3)} - Y_{3(3)})Y_{3(3)} \cdot (1 - Y_{3(3)}). \end{aligned} \quad (22)$$

The change of weight, which joint node k in output layer to node h in hidden layer (2nd layer) is defined like bellow.

$$\Delta \alpha_{hk} = -\eta * H_{(ij)}^{-1} * \hat{\varepsilon}_{k(3)} * Y_{hk} \quad (23)$$

Signal Error in 2nd layer (hidden layer) has node h

$$\hat{\varepsilon}_{1(2)} = \frac{\partial Y_{1(2)}}{\partial \hat{Y}_{1(2)}} * \sum_{k=1}^3 \hat{\varepsilon}_{k(3)} * \frac{\partial \hat{Y}_{k(3)}}{\partial Y_{1(2)}} = \quad (24)$$

$$Y_{1(2)}(1 - Y_{1(2)}) * \sum_{k=1}^3 \hat{\varepsilon}_{k(3)} * \alpha_{1k}$$

$$\hat{\varepsilon}_{2(2)} = \frac{\partial Y_{2(2)}}{\partial \hat{Y}_{2(2)}} * \sum_{k=1}^3 \hat{\varepsilon}_{k(3)} * \frac{\partial \hat{Y}_{k(3)}}{\partial Y_{2(2)}} = \quad (25)$$

$$Y_{2(2)}(1 - Y_{2(2)}) * \sum_{k=1}^3 \hat{\varepsilon}_{k(3)} * \alpha_{2k}$$

$$\hat{\varepsilon}_{h(2)} = \frac{\partial Y_{h(2)}}{\partial \hat{Y}_{h(2)}} * \sum_{k=1}^3 \hat{\varepsilon}_{k(3)} * \frac{\partial \hat{Y}_{k(3)}}{\partial Y_{h(2)}} = \quad (26)$$

$$Y_{h(2)}(1 - Y_{h(2)}) * \sum_{k=1}^3 \hat{\varepsilon}_{k(3)} * \alpha_{hk}$$

The weighted factor, which joint node h in hidden layer to node L in input layer can be calculated by formula bellow.

$$\Delta \alpha_{Lh} = -\eta * H_{(ij)}^{-1} * \hat{\varepsilon}_{h(2)} * Y_{Lh} \quad (27)$$

6. Simulation

To realize the simulation of former integration control and ABC integration control which control TODS we were building two block scheme. First block consists of TODS as plant and former method (Feed-forward, H-infinite, NLPC, Robust) as controller (figure 16), and second block consists of The same character of TODS and ABC as controller.(figure17).

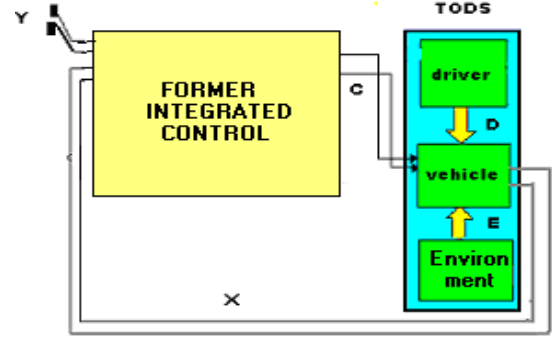


Figure 16: Block of Former Integration Control and TODS plant

Y = Reference (Y_R, VSS_s, RA_s) X = Vehicle State (Y_a, VSS_a, RA_a)

C = Control Vector ($T_{VDC}, \delta_r, M_{AS}$) D = Driver disturbance (δ_r, TR, BP)

E = Environment disturbance (F_{drag}, Ω, μ_i)

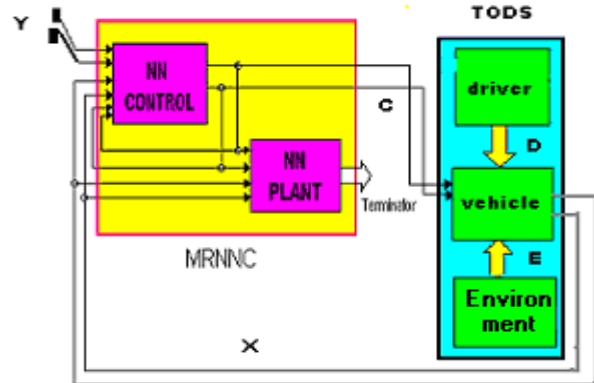


Figure 17: Block of MRNNC and TODS plant

Y = Reference (Y_R, VSS_s, RA_s) X = Vehicle State (Y_a, VSS_a, RA_a)

C = Control Vector ($T_{VDC}, \delta_r, M_{AS}$) D = Driver disturbance (δ_r, TR, BP)

E = Environment disturbance (F_{drag}, Ω, μ_i)

6.1 Control Target

Reference variable, which should be input to neural network control are defined how to get desired vehicle state

value, which affect stability of vehicle and convenience of passenger [3]

Error Yaw-Rate (YR)

$$\Delta YR_s = \frac{V_x \cdot \delta_f}{(l_f + l_r)} - YR_a \Rightarrow 0 \quad (28)$$

Error Vehicle Side Slip (VSS)

$$\Delta VSS_s = 0 - VSS_a \Rightarrow 0 \quad (29)$$

Error Rolling Angle (RA)

$$\Delta RA_s = 0 - RA_a \Rightarrow 0 \quad (30)$$

Data input and output for identification to be taken from 810 centers, which are clustered by MDPC is presented fig 18.

The learning, validation and testing process of NNP is shown in figure 19, how to get convergence of error. The number of data is 810, number of epoch 30, and number of segment 20. After running GA to optimized the architecture of gHFNN results 32 rules , learning rate 0.08 , momentum 0.6 , parameter of membership (a = 0.85, b = 2, c = 2.2) and for PNN structure are n = 32 , N = 4 , T = 2 , W = 16 and polynomial constants are [0.054 ; -0.023 ; 0.123 ; -1.2 ; 3.61 ; 0.167]. We design uniformly for selected PN node.

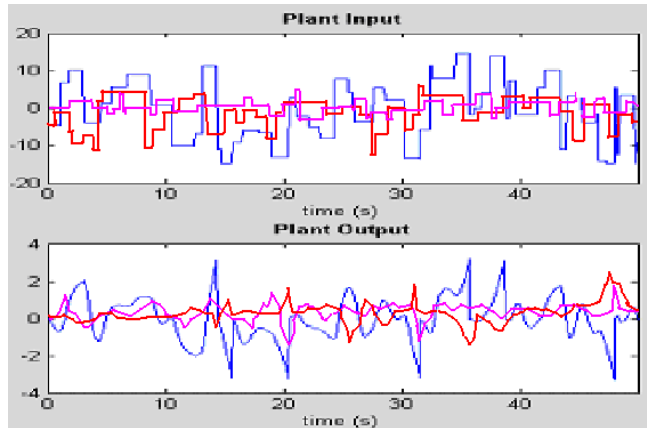


Figure 18: Preparing all Cluster to Training NNP

Center data for training NNP Input = T_{VDC} (blue), δr (violet), M_{AS} (red).

Output = YR (blue), RA (red), VSS (violet)

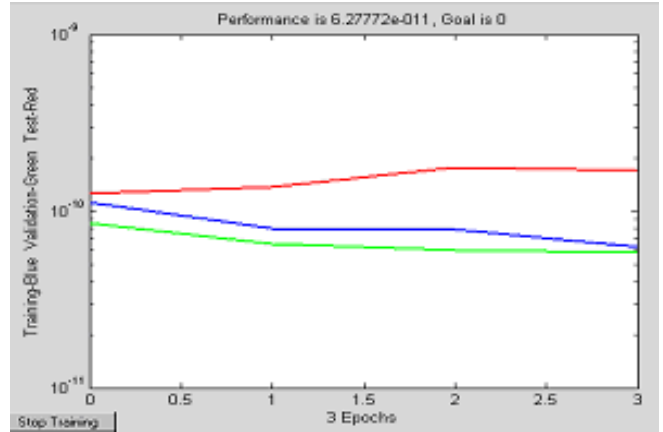


Figure 19: Training, Validation, Testing Process for NNP

Data for training NNC to be taken from the reference and desired input of update NNP with the same output value with the given reference. (Figure 20)

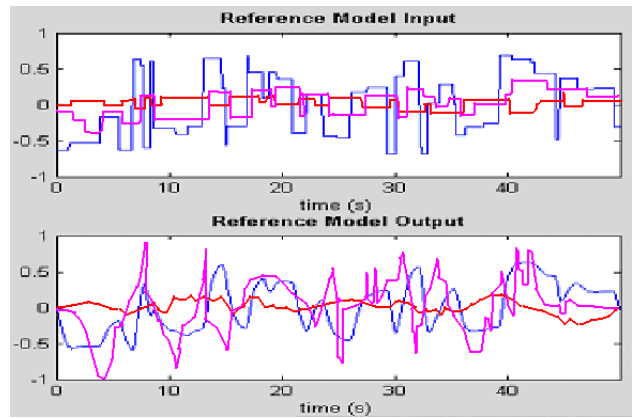


Figure 20: Preparing Data Reference for Training NNC

Reference data for training NNC Input = ΔYR (blue), ΔRA (red), ΔVSS (violet) Output = T_{VDC} (blue), δr (violet), M_{AS} (red).

Learning process of NNC will update the joint weight of optimized architecture specially the number of hidden layer node. (figure 21). The GA optimization results 48 hidden layer nodes, which the best generalization and don't meet locally optimum point during training, lowest error during testing.

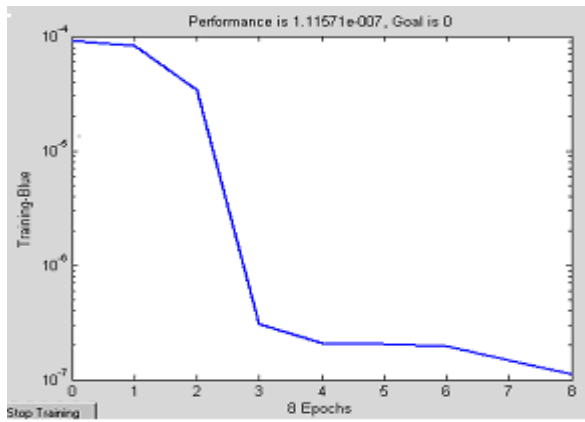


Figure 21: Training Process to Update NNC Parameter

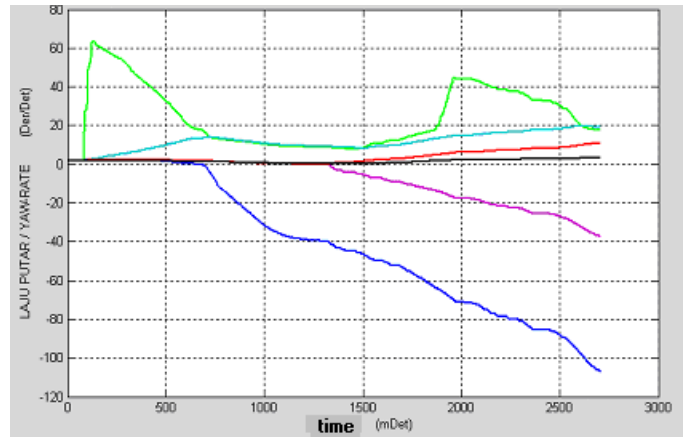


Figure 22: Error Yaw Rate Response (x 0,1)

Feedforward (dark blue)- H^∞ optimal (light blue)-Robust (violet)
NLPC (green)-AI (red)-Referensi (black)

7. Integrated Control Performance

Simulation is conducted comprehensively to 4 former integration system and MDFC-ABC integration. Vehicle states output, which are measured to be compared to reference to get (ΔYR , ΔVSS , ΔRA). To make moderate for comparing among one integration system with other integration system, hence has been done by assessment of response value in the form of rank. List of rank is shown in table 2. As reference for becoming rank list, hence the got error have been divided in to several group in according to its rank

Average of Error yaw rate (ΔYR)

$\Delta YR < 3^\circ/\text{det}$	Rank 1
$3^\circ/\text{det} < \Delta YR < 6^\circ/\text{det}$	Rank 2
$6^\circ/\text{det} < \Delta YR < 10^\circ/\text{det}$	Rank 3
$\Delta YR > 10^\circ/\text{det}$	Rank 4

Error Vehicle Side Slip (ΔVSS)

$\Delta VSS < 5^\circ$	Rank 1
$5^\circ < \Delta VSS < 10^\circ$	Rank 2
$10^\circ < \Delta VSS < 20^\circ$	Rank 3
$\Delta VSS > 20^\circ/\text{det}$	Rank 4

Average of Error Roll Angle (ΔRA)

$\Delta RA < 1,5^\circ$	Rank 1
$1,5^\circ < \Delta RA < 3,5^\circ$	Rank 2
$3,5^\circ < \Delta RA < 7,0^\circ$	Rank 3
$\Delta RA > 7,0^\circ$	Rank 4

Figure 19 is an example of error yaw rate response of TODS is controlled by all integration method.

Table 2: List of Simulation Result in the form of Rank

Manuver	Target	No Integrasi	FF Integrasi	H^∞ Integrasi	NLPC Integrasi	Robust Integrasi	AI
Small-J	VSS	2	3	2	4	1	1
	YR _r -YR _a	3	4	3	3	2	1
	RA	1	1	1	2	1	1
S	VSS	2	3	1	2	1	1
	YR _r -YR _a	3	3	2	3	1	1
	RA	2	2	2	4	1	1
J-Turn	VSS	3	4	3	2	2	1
	RA	2	3	3	1	1	1
	VSS	3	4	2	3	1	1
Fishhook	RA	2	4	1	3	1	1
	VSS	3	4	1	4	2	1
J with B	RA	2	3	1	3	1	1
J with B Low μ	VSS	3	4	2	2	2	1
	RA	1	3	3	2	2	1
Total		32	45	27	38	19	14

8. Simulation Result Discussion

According to total value in table 2 shows Robust gets 19, NLPC gets 38, H^∞ gets 27, Feed-forward gets 45 and No integrated gets 32. integration control by MDFC-ABC get 14. It means, that as low as total value as good as they yield their performance of integrated control is.

MDFC-ABC controls vehicle stability for all maneuver result the best performance (lowest rank), which other integrated control can not meet.

Sequence of Rank are then concluded as follow:
AI (MDFC-ABC) = excellent, Robust = very good, H^∞ = good, NLPC = adequate, Feed Forward = bad.

9. Conclusion

Based on simulation result, then it can be decided, that MDFC-ABC is the best choice for integrating of subsystem control of vehicle stability.

References

- [1] Mauridhi H.P., 2006, *Supervised Neural Network & Applications*, Graha Ilmu , ITS Indonesia.
- [2] I. N. Sutantra, 2002, *Automotive Technology & Application*, Guna Widya , Surabaya
- [3] Pacetka , 2005, *Tire & Vehicle Dynamics*, Delft University , Netherlands
- [4] Vojislav Kecman , 2005, *Learning & Sort Computing*, MIT press , USA
- [5] J.S.R. Yang, 1997, *Neuro-Fuzzy & Soft Computing*, Prentice Hill, USA
- [6] Sung-Kwun Oh, 2005, “Genetically Optimized Hybrid Fuzzy Neural Network Based on Linear Fuzzy Inference Rules”, *Intenational Journal of Control, Automation and System vol. 3 no 2 pp 183-194*, MOCIE, Korea.
- [7] Auto-Technology Fisita, 2005, “Global Chassis Control – The Networked Chassis”, *Fisita Magazine*, UK
- [8] M. Lakehal Ayat , 2000, “On Global Chassis Control Combined Braking, Cornering and Yaw Rate Control”, *Advance Vehicle Control proceeding*, Michigan /USA.
- [9] ShiniChiro, 1998, “Integrated Control of Four Wheel Steering and Wheel Torque Using Nonlinear Predictive Controller”, *FISITA journal*, Nihon University – Honda R&D , Japan
- [10] Sung-Su Kim, Seul Jung, 2006, “Hardware Implementation of Neural network Controller with an MCU and an FPGA for nonlinear System”, *International Journal of Control Automation and Systems vol 4 n0 5 pp 567-574*, KOSEF, Korea.
- [11] Kozuya Kitajima, 2000, “ H^∞ Control for Integrated Side Slip, Roll and Yaw Control for Ground Vehicle”, *Advance Vehicle Control Journal*, Michigan
- [12] Bosch R&D, “Electronics Stability Program”, 2005, Germany
- [13] Hyun Cheol Cho, M. Sami Fadali, 2005, “Neural Network Active Control of Structures with Earthquake Excitation”, *International Journal of Control, automation and Systems vol 3 no 2 pp 202-210*, MOST, Korea.
- [14] I.N Sutantra, 2004, “Design of Safety and comfort City Car with ESP Pump less”, *RUT research report* , ITS Indonesia
- [15] Agus Sigit P. 2005, “Design of Smart Chassis”, *Hibah Pasca Report Jurusan Teknik Mesin* , FTI-ITS.
- [16] M.Harly , Mauridhi HP , IN Sutantra , 2005, “Wheel Spin Control Through by Pass Trottle for SI Engine based on Fuzzy Logic” , *IES2005 proceeding* , EEPIS-ITS , Surabaya.
- [17] M.Harly , Mauridhi HP , IN Sutantra, 2006 , “Control of Continuous Transmission Ratio(CVT) for Vehicle Based on Model Reference Neural Network Control” , *IES2006 proceeding* , EEPS-ITS Surabaya.
- [18] IN Sutantra , M.Harly, 2005 ,” Implementation of Vehicle Vibration Control to improve stability and Comfort”, *ITS Vibration Conference* , TekFisika - ITS
- [19] IN Sutantra , 1999, “Desgn of four Wheel Steering Multi Function”, *Hibah Bersaing Report VII*, DikTi, ITS-Sby.
- [20] Don Leggiero, 2006, ‘M556 Electronics Subsystem Technical package OEM interfaces’, *Application Engineering Buletin*, Cummins Inc, Columbus Indiana.