

# Fuzzy-PID Control via Genetic Algorithm-Based Settings for the Intelligent DC-to-DC Step-Down Buck Regulation

Anas N. Al-Rabadi and Mahmoud A. Barghash

**Abstract** – This article presents further testing and verification results for a previously introduced new intelligent regulation method that controls the power-electronic Buck converter utilizing a small-signal model of the pulse width modulation (PWM) switch. The implemented intelligent control method uses a fuzzy-PID controller that is tuned using the global search method of genetic algorithm (GA). The presented results further verifies that the previously used intelligent hierarchical regulation method using the GA-tuned fuzzy-PID controller produces the desired Buck performance for wide spectrum of parameter values despite the occurrence of high amplitude noise.

**Index Terms** – Buck DC-to-DC converter, fuzzy control, genetic algorithms, intelligent regulation, switching-mode step-down converter.

## 1. INTRODUCTION

Small-signal modeling of the open-loop power converters has recently received an increasing attention, due to the fact that these models are the basis to extract accurate transfer functions which are essential in the feedback control design [7, 31]. They are used to design reliable high performance regulators, by enclosing the open loop DC-to-DC power converters in a feedback loop, to keep the function of the system as close as possible to the desired performance by counteracting the outside noise in the (a) source voltages, (b) pulse width modulator (PWM) duty ratio, and (c) load current.

Power converters generally operate in (a) Continuous Conduction Mode (CCM) or (b) Discontinuous Conduction Mode (DCM) [7, 31]. The CCM mode is desirable, as the output ripple of the DC-to-DC power converter is very small when compared to the DC steady state output. A linearized small-signal model is constructed to examine the dynamic behaviors of the converter, due to the fact that noise is of small signal variations.

This work was accomplished with the support from the Deanship of Academic Research (DAR) at The University of Jordan under financial grant number (733).

A. N. Al-Rabadi (Corresponding Author) is currently an Associate Professor with the Computer Engineering Department at The University of Jordan, Amman-11942-Jordan; phone: +962 79 6445364; e-mail: a.alrabadi@ju.edu.jo; <http://www.ju.edu.jo/sites/academic/a.alrabadi>.

M. A. Barghash is currently an Associate Professor with the Industrial Engineering Department at The University of Jordan; e-mail: mabargha@ju.edu.jo.

Using this small-signal model, the necessary open-loop transfer functions can be determined and plotted using Bode plots in order to use compensation to the PWM power converters, to meet the desired nominal operating conditions, through the application of various control methods. These control methods include: (a) frequency analysis in the classical control theory [32], (b) time analysis in the modern control theory [32], (c) both frequency analysis and time analysis domains in the post modern (digital and robust) control theory, and (d) soft computing (e.g., fuzzy logic, neural networks, and genetic algorithms) in the intelligent control methodology [2-4, 8, 10, 13-15]. These control methods can be applied to the models of power converters that usually work with only one specific control scheme which is PWM through either duty-ratio control or current programming control [31]. In this research, the duty-ratio control is used, in which the switch ON-time is controlled externally by comparing a saw tooth ramp with the controller voltage [31].

Several modeling approaches of the PWM power converters do exist. These approaches are separated into three main categories. The first modeling category aims towards modeling the whole PWM converters. Examples for this category are (a) volt-second and current-second (charge) balance approach and (b) state-space averaging approach [7, 31]. These approaches suffer from inaccurate results in the high-frequency range. The second modeling category aims more specifically towards modeling what is called the converter cell, that includes modeling the basic cell of the PWM converter, and ignoring the input DC voltage source and the output RC filter in the model, where the cell includes the PWM switch with the inductors and the capacitors associated with it. An example for this category is the averaged modeling approach [7, 31]. This approach also suffers from inaccurate results in the high-frequency range. The third modeling category aims more specifically to model the PWM switch.

In general, the previously mentioned modeling approaches utilize four techniques. The first technique is the sampled-data representation technique. The second technique is the averaged technique. The third technique is the exact small-signal analysis technique [7, 31], and the fourth technique combines the averaged technique and the sampled-data technique [7, 31]. The averaged technique represents the easiest and the most widely used technique. It can be used to determine the various impedances and transfer functions for the converters. The basic characteristics of this technique are (1) it uses the averaging technique of voltages and currents

and (2) it gives accurate low-frequency results but inaccurate high-frequency results.

The production of averaged models can be accomplished for the nonlinear PWM switch as well for the converter system as a whole. This switch is usually a single-pole double-throw (SPDT) switch. It is this switch which is responsible for switching the converter from one configuration to another during each switching period. These models, derived for the PWM switch, are usually easier than the derivation of converter models. Yet, it has the limitation of the fact that not all of the converter topologies have the same PWM switch arrangement [31].

Exact small-signal technique [7, 31] is very accurate to a wide range of frequencies. This technique can be applied to any converter system that is (a) periodic, (b) time-varying, and (c) piecewise linear. The trade off for the high accuracy occurs in the complexity of the matrix manipulations and the time consumed to produce the exact results. Yet, it has a great advantage of being automated through the use of computer-aided design (CAD) software packages.

Sampled data technique is based on the generation of a difference equation that describes the propagation of a point on a converter waveform from one cycle to another. It is usually used to derive an accurate response for the PWM current mode control. Yet, the price is paid again through the limitation of the upper-frequency range, to be limited to half of the switching frequency. The fourth modeling technique combines the averaged technique and the sampled-data technique, in an effort to gain the main benefits of each technique. However, this technique, while improved, is also inaccurate [7, 31].

Thus, it can be observed that there is a need to develop a model applicable to various regulating schemes, including the most used scheme which is the PWM duty ratio and current mode control scheme. Therefore, a small-signal modeling approach which is applicable to any power converter system represented as a two-port network has been introduced [7]. This was done through the modeling of the nonlinear part in the power converter system, which is the PWM switch.

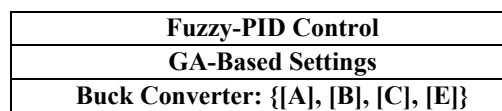
To deal with reasoning, that is approximate rather than exact, fuzzy logic is used which is a form of many-valued logic and is derived from fuzzy set theory. In contrast with "crisp logic", where binary sets have two-valued logic, fuzzy logic variables may have a truth value that ranges in degree between "0" and "1". In another formulation, one can point out that fuzzy logic is a superset of the conventional (Boolean) logic that has been extended to handle the concept of partial truth which is the truth values between completely true and completely false. In addition, when linguistic variables are used, these degrees can be managed by specific functions.

Fuzzy logic has been applied successfully into several fields in social and technical sciences such as in social psychology, expert systems, artificial intelligence, control theory and engineering [1, 6, 14, 17, 19-20, 25, 28, 34-35, 38-39, 43, 45, 47] that lead to the design of many variants of fuzzy controllers that effectively control noisy systems.

The global search heuristic called genetic algorithm (GA) mimics the process of natural evolution. This heuristic algorithm is frequently used to generate useful solutions to several optimizations and search problems that are widely used in many applications such as in bioinformatics, computational sciences, economics, mathematics, physics, and engineering [13-14, 18, 21-22, 24, 26-27, 29, 33, 36, 41-42, 44, 48, 50]. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using naturally-inspired operations such as inheritance, mutation, selection, and crossover.

A typical GA requires (a) a genetic representation of the solution domain and (b) a fitness function to evaluate the solution domain. In GA, a population of strings called chromosomes or genotype of the genome, which encode candidate solutions to an optimization problem (called individuals, creatures, or phenotypes), evolves toward better solutions. Usually, solutions are represented as strings of "0"s and "1"s, but other encoding schemes are also used. The evolution usually starts from a population of randomly generated individuals and occurs in generations, where, in each generation, the fitness of each individual in the population is evaluated, multiple individuals are stochastically selected (based on their fitness), and then modified using the corresponding GA operations to form a new population. The new population is then used in the next iteration of the GA, where usually the GA terminates when either a maximum number of generations have been produced or a satisfactory fitness level for the population has been reached.

Fig. 1 illustrates the layout of the Buck-based control method that is used in this article. In Fig. 1, the first layer presents the state-space representation of the Buck converter, the second layer presents the GA-based settings to achieve the needed dynamic performance, and the third layer presents the implemented fuzzy-based PID controller.



**Fig. 1.** Buck power-electronic converter hierarchical control method which is used in this article.

Although several previous approaches have been presented for the purpose of controlling switching-mode converters [5, 9, 11-12, 16, 30-31], the intelligent control method which is presented in this research using GA-settings of the fuzzy-PID controller presents further additional testing and verification results to the previously obtained ones [13, 14] using intelligent Buck regulation for wide spectrum of parameter values despite the existence of high-amplitude noise.

The remainder of this article is organized as follows: Section 2 presents basic background on the Buck power converter, fuzzy logic, and genetic algorithms. Section 3 presents the illustration of the used method of the genetic algorithm-based settings of the fuzzy-PID controller for controlling the utilized Buck converter. Section 4 presents the simulation results for the application of GA-tuned fuzzy-PID controller on the state-space model of the Buck converter for both of the input-to-output and control-to-output transfer functions in the existence of high-amplitude noise. Conclusions and future work are presented in Section 5.

## 2. BACKGROUND

Important background on the DC-to-DC step-down Buck converter, fuzzy logic and genetic algorithms, that will be utilized in later sections, is presented in this section.

### 2.1. Switching Mode Power Supply: The Application of the Averaged Modeling Approach and the New Small-Signal Model for the PWM Converters

Various averaged modeling techniques used to model the PWM converters do occur. These techniques include (a) volt-second and current-second balance approach, and (b) state-space averaging approach [7, 31]. These techniques are used to model the converter systems as a whole, as well as to model the pulse width modulation (PWM) switch by itself. Yet, these techniques are valid for the low-frequency range, and they give inaccurate results for the dynamic behaviors of the power converters in the high-frequency ranges [7, 31]. Another modeling approach that focuses on modeling the converter-cell, instead of the converter as a whole, is used to get averaged models for the PWM converters. This approach is also useful for the low-frequency ranges, but not useful for the high-frequency ranges. One major advantage of these techniques is the fact that they are easy to implement, and the results obtained are not in complicated forms.

#### 2.1.1. The Averaged Modeling Approach and its Application upon the Buck DC-to-DC Power Converter

Averaged modeling approach aims to produce an averaged model for a specific cell of the PWM converters. This cell is shown in Fig. 2, where this basic cell is used to explore the DC behaviors and the AC small-signal dynamic behaviors of the PWM Buck converter.

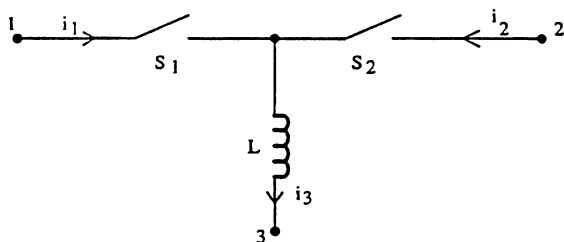


Fig. 2. The fundamental cell of the PWM converter.

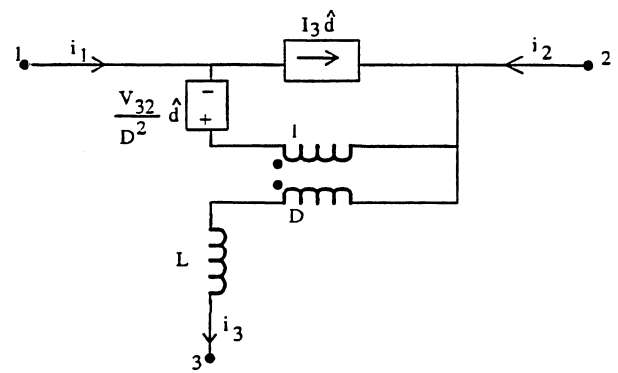


Fig. 3. The DC and AC small-signal averaged model of the converter-cell that was shown in Fig. 2.

The DC and AC small-signal averaged model of the converter-cell which is shown in Fig. 2, can be produced as shown in Fig. 3, where  $D$  is the DC value of the duty ratio,  $\hat{d}$  is the small-signal perturbation of the duty ratio, and  $V_{32}$  is the DC voltage between terminals 3 and 2.

The AC small-signal and DC averaged model, shown in Fig. 3, will be used to derive the corresponding control-to-output, input-to-output, input impedance, and the control-to-input current transfer functions for the Buck converter. Also, the averaged model will be used to derive the input-to-output, control-to-output, input impedance, and control-to-input current average transfer functions for the PWM Buck converter. Fig. 4 shows a typical Buck converter.

One assumes a small-signal perturbation  $\hat{v}_g$ , in the DC voltage source  $V_g$ , and that  $|V_g| \gg |\hat{v}_g|$ . After the implementation of the averaged model that was shown previously, we get the following small-signal model for the PWM Buck converter as shown in Fig. 5.

By nulling the input  $\hat{v}_g$ , we get the following control-to-output transfer function [7]:

$$\frac{\hat{v}_o}{\hat{d}} = V_g \frac{1}{1 + (L/R)s + LCs^2} \quad (1)$$

Also, by nulling the input  $\hat{d}$ , we get the following input-to-output transfer function [7]:

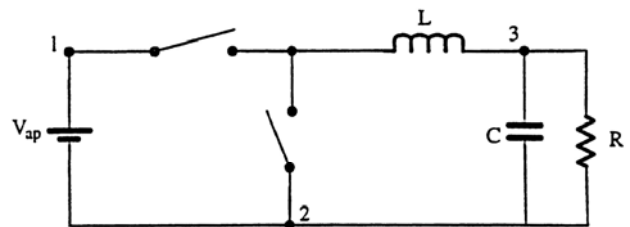
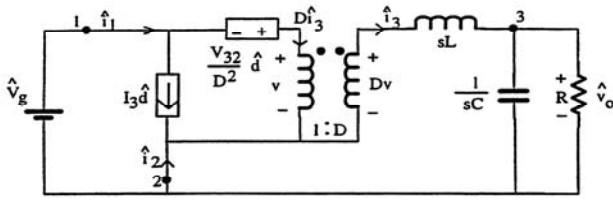


Fig. 4. The DC-to-DC power-electronic Buck converter.



**Fig. 5.** The AC small-signal model of the Buck converter operating in the continuous conduction mode.

$$\frac{\hat{v}_o}{\hat{v}_g} = D \frac{1}{1 + (L/R)s + LCs^2} \quad (2)$$

There are other transfer functions of interest for the Buck converter such as input impedance and control-to-input current transfer functions. To get the input impedance ( $\hat{v}_g / \hat{i}_1$ ), we null the input  $\hat{d}$ , so we get the following equation [7]:

$$\frac{\hat{v}_g}{\hat{i}_1} = \frac{R}{D^2} \frac{1 + (L/R)s + LCs^2}{1 + RCs} \quad (3)$$

To obtain the control-to-input current transfer function ( $\hat{i}_1 / \hat{d}$ ), we null the input  $\hat{v}_g$ , so we get the transfer function [7]:

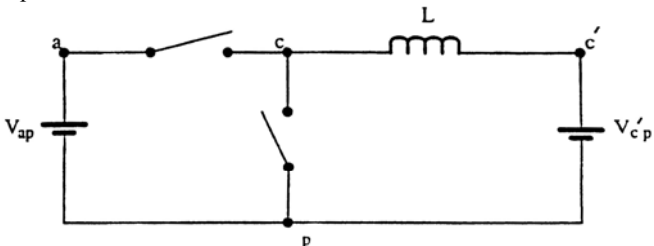
$$\frac{\hat{i}_1}{\hat{d}} = \frac{V_g D + RI_3}{R} \frac{1 + \left(\frac{V_g DRC + LI_3}{V_g D + RI_3}\right)s + \left(\frac{LI_3 RC}{V_g D + RI_3}\right)s^2}{1 + (L/R)s + LCs^2} \quad (4)$$

### 2.1.2. New Method for Obtaining an Exact Model of the PWM Switch in the Duty Ratio Programming Mode

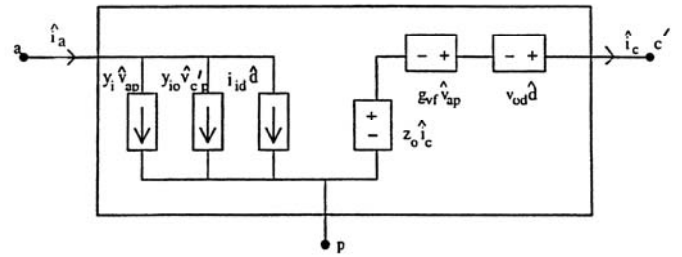
A new approach is developed to formulate a new model for the PWM nonlinear switch [7] within this subsection. The Buck converter will be used now as the basic model to extract the corresponding two-port network parameters. The main reason that the Buck is used over the other PWM converters is the fact that the Buck converter is a second order system with a simple structure. This will be reflected upon the simplicity of the results that will be obtained.

Due to the fact that the ripple voltage is comparatively much smaller than the DC voltage across the output capacitor (as the Buck converter is operating in the CCM), the capacitor will be replaced with a constant DC voltage source  $V_{c'p}$ . This is illustrated in Fig. 6.

As can be observed from Fig. 6, the two-port augmented equations can be written as follows:



**Fig. 6.** An alternative Buck configuration.



**Fig. 7.** Circuit model for Equations (5) and (6).

$$\hat{i}_a = y_i \hat{v}_{ap} + y_{io} \hat{v}_{c'p} + i_{id} \hat{d} \quad (5)$$

$$\hat{v}_{c'p} = z_o \hat{i}_c + g_{vf} \hat{v}_{ap} + v_{od} \hat{d} \quad (6)$$

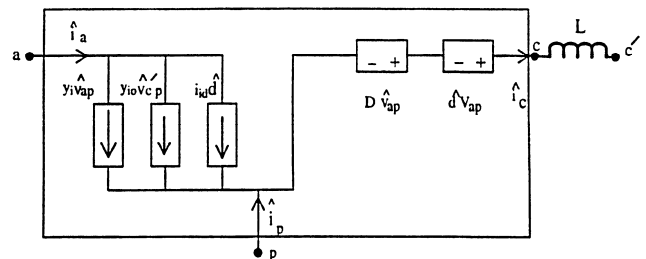
As observed from Fig. 7, a circuit model for the two-port augmented equations, which are represented by Equations (5) - (6), can be constructed.

Developing a new model for the PWM switch, which is the nonlinear part of the PWM converter, is the main objective. This model can be constructed directly by replacing the values of the parameters  $\{y_i, y_{io}, i_{id}, z_o, g_{vf}, v_{od}\}$  in their simplest form [7] in Equations (5) - (6). Thus, the mathematical model will be as follows [7]:

$$\begin{aligned} \hat{i}_a &= y_i \hat{v}_{ap} + y_{io} \hat{v}_{c'p} + i_{id} \hat{d} \\ &= \frac{\beta_1 \beta_2 \gamma (1 + j\omega DT_3) + 1 - j\omega DT_3 - \beta_1 - \beta_2}{T_s \omega^2 L (1 - \beta_1 \beta_2)} \hat{v}_{ap} + \\ &\quad + \frac{jD}{\omega L} \hat{v}_{c'p} + I_x - \frac{j\beta_2 (1 - \beta_1) V_{ap}}{\omega L (1 - \beta_1 \beta_2)} \hat{d} \end{aligned} \quad (7)$$

$$\begin{aligned} \hat{v}_{c'p} &= z_o \hat{i}_c + g_{vf} \hat{v}_{ap} + v_{od} \hat{d} \\ &= -j\omega L \hat{i}_c + D \hat{v}_{ap} + V_{ap} \hat{d} \end{aligned} \quad (8)$$

One can note that the parameter  $z_o$  represents an inductor. Thus, we can "pull" the  $z_o$  parameter outside the circuit model, which is equivalent to the mathematical model which is represented by Equations (7) - (8), as the  $z_o$  parameter is merely an inductor impedance, which is then multiplied by the corresponding path current  $\hat{i}_c$ , to form a voltage source ( $z_o \hat{i}_c$ ) in series with the voltage sources ( $g_{vf} \hat{v}_{ap}$ ) and ( $v_{od} \hat{d}$ ). The result of this process is shown in Fig. 8.



**Fig. 8.** A derived new circuit model.

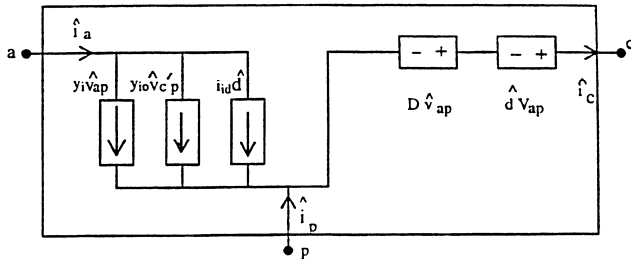


Fig. 9. Circuit model for the PWM switch.

As observed from Fig. 8, we can recognize that the circuit model between terminals  $\{a, p, c\}$  is merely the switch between these terminals in the original Buck converter circuit. So, the equivalent switch model in terms of the perturbations  $\{\hat{v}_{ap}, \hat{v}_{cp}, \hat{d}\}$  is shown in Fig. 9. Now we need to obtain the switch model in terms of the perturbations  $\{\hat{v}_{ap}, \hat{v}_{cp}, \hat{d}\}$  instead of the perturbations  $\{\hat{v}_{ap}, \hat{v}_{cp}, \hat{d}\}$ . To do so, we note from Fig. 8 that:

$$\hat{v}_{cp} = \hat{v}_{cp} - j\omega L \hat{i}_c \quad (9)$$

As can be seen from Fig. 8, we note that the common node ( $c'$ ) corresponds to the node ( $c$ ) in the Buck converter in Fig. 6. After multiplying both sides of Equation (9) by the parameter  $y_{io}$ , we get:

$$y_{io} \hat{v}_{cp} = y_{io} \hat{v}_{cp} + D \hat{i}_c \quad (10)$$

Therefore, we can replace the term  $\{y_{io} \hat{v}_{cp} + D \hat{i}_c\}$  instead of the term  $\{y_{io} \hat{v}_{cp}\}$  in the previously derived switch model shown in Fig. 9, in order to make the new model contains the perturbations  $\{\hat{v}_{ap}, \hat{v}_{cp}, \hat{d}\}$  instead of the perturbations  $\{\hat{v}_{ap}, \hat{v}_{cp}, \hat{d}\}$ . The new switch model will be constructed as shown in Fig. 10.

To reduce the number of the dependent current sources that appear in the new switch model, which are four dependent current sources, we will try to reduce the number of terms in the previous mathematical switch model. One has:

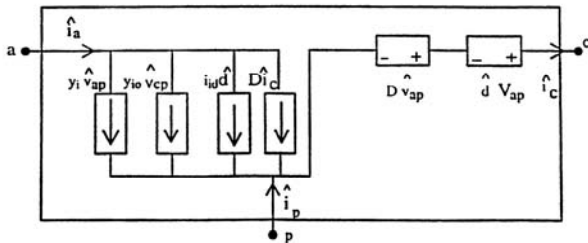


Fig. 10. Alternative circuit model for the PWM switch.

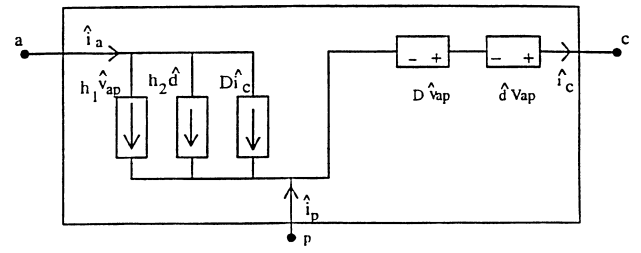


Fig. 11. The new small-signal model of the PWM switch.

$$\hat{v}_{cp} = V_{ap} \hat{d} + D \hat{v}_{ap} \quad (11)$$

and as ( $\hat{v}_{cp} = z_o \hat{i}_c + g_{vf} \hat{v}_{ap} + v_{od} \hat{d}$ ), we obtain:

$$\therefore \hat{v}_{cp} = V_{ap} \hat{d} + D \hat{v}_{ap} - j\omega L \hat{i}_c \quad (12)$$

To develop the first reduced mathematical switch model, we note that  $\hat{i}_a = y_i \hat{v}_{ap} + y_{io} \hat{v}_{cp} + i_{id} \hat{d}$ . Substituting Equation (12) in Equation (5), and after the collection of the similar terms, we get the following reduced-form equation:

$$\hat{i}_a = (y_i + y_{io} D) \hat{v}_{ap} + (y_{io} V_{ap} + i_{id}) \hat{d} - j y_{io} \omega L \hat{i}_c \quad (13)$$

One obtains the following equation [7] by substituting the values of  $\{y_i, y_{io}, i_{id}\}$  in Equation (13):

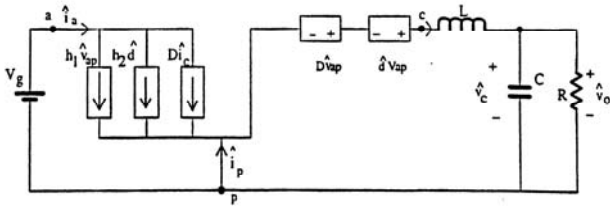
$$\hat{i} \left( \frac{e^{-j\omega T_s} (1 + j\omega DT_s) + 1 - j\omega DT_s - e^{-j\omega DT_s} - e^{-j\omega DT_s}}{T_s \omega^2 L (1 - e^{-j\omega T_s})} + \frac{jD^2}{\omega L} \right) \hat{v}_{ap} + \left( I_x + \frac{jDV_{ap}}{\omega L} - \frac{j e^{-j\omega DT_s} (1 - e^{-j\omega DT_s}) V_{ap}}{\omega L (1 - e^{-j\omega T_s})} \right) \hat{d} + D \hat{i}_c \quad (14)$$

Equation (11) is the other equation of the model. So, Equations (11) and (14) represent the final reduced mathematical model of the PWM switch, replacing the model represented in Fig. 9. The final equivalent circuit model of the switch mathematical model (which is represented by Equations (11) and (14)) is as shown in Fig. 11 [7], where:

$$h_1 = \frac{e^{-j\omega T_s} (1 + j\omega DT_s) + 1 - j\omega DT_s - e^{-j\omega DT_s} - e^{-j\omega DT_s}}{T_s \omega^2 L (1 - e^{-j\omega T_s})} + \frac{jD^2}{\omega L} \quad (15)$$

$$h_2 = I_x + \frac{jDV_{ap}}{\omega L} - \frac{j e^{-j\omega DT_s} (1 - e^{-j\omega DT_s}) V_{ap}}{\omega L (1 - e^{-j\omega T_s})} \quad (16)$$

Figure 11 shows the new switch model which is expected to be an exact small-signal model since the mathematical equations, upon which the whole derivation process was built, are exact. Also, one notes that two of the dependent current sources are frequency-dependent, which is uncommon for current or voltage dependent sources.



**Fig. 12.** Equivalent circuit model of the PWM Buck converter, which is obtained through the application of the new small-signal model of the PWM switch.

### 2.1.3. Examining the New Small-Signal Model: The Implementation of the New Small-Signal Model of the PWM Switch on the Buck DC-to-DC Power Converter

The new small-signal model of the PWM switch, that was developed in the previous sub-section, will be examined in this subsection on the PWM Buck converter. The control-to-output, input-to-output, input impedance, and control-to-input current transfer functions will be derived for the Buck converter, using the new small-signal model of the PWM switch. These transfer functions will be compared to the corresponding transfer functions for the averaged modeling approach and the exact transfer functions for the Buck converter.

Applying the PWM switch model that was developed previously for the Buck power converter, one obtains the equivalent circuit model as shown in Fig. 12.

One assumes that the input DC voltage source,  $V_g$ , has small-signal perturbation,  $\hat{v}_g$ , and that  $|V_g| \gg |\hat{v}_g|$ . To determine the system quadruple  $\{[A], [B], [C], [E]\}$  for the Buck model shown in Fig. 12, for the inputs  $\hat{d}$  and  $\hat{v}_g$ , we null the DC voltage source  $V_g$ . Then, the following equations can be developed for the Buck model shown in Fig. 12, where  $\hat{i}_l$  is the inductor current,  $\hat{i}_c$  is the capacitor current, and  $\hat{i}_o$  is the output current that flows in the output resistor. Hence, we have the following equations:

$$\hat{v}_o = \hat{v}_c, \quad \hat{i}_l = \hat{i}_c + \hat{i}_o = C\dot{\hat{v}}_c + \frac{1}{R}\hat{v}_o \quad (17)$$

$$\therefore \dot{\hat{v}}_c = \frac{1}{C}\hat{i}_l - \frac{1}{RC}\hat{v}_c \quad (18)$$

$$\hat{v}_{ap} = \hat{v}_g \quad (19)$$

$$-D\hat{v}_g - \hat{d}V_g + L\dot{\hat{i}}_l + \hat{v}_c = 0$$

$$\therefore \dot{\hat{i}}_l = \frac{D}{L}\hat{v}_g + \frac{V_g}{L}\hat{d} - \frac{1}{L}\hat{v}_c \quad (20)$$

The corresponding output equation is ( $y = \hat{v}_o = \hat{v}_c$ ). For the

converter states  $x$  and inputs  $u$ , where  $x = \begin{bmatrix} \hat{i}_l \\ \hat{v}_c \end{bmatrix}$  and  $u = \begin{bmatrix} \hat{v}_g \\ \hat{d} \end{bmatrix}$ ,

the system quadruple  $\{[A], [B], [C], [E]\}$  will be [7]:

$$A = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix}, \quad B = \begin{bmatrix} D/L & V_g/L \\ 0 & 0 \end{bmatrix}, \quad C = [0 \quad 1], \\ E = [0 \quad 0].$$

We null the input  $\hat{v}_g$  in order to obtain the control-to-output transfer function. The new system quadruple  $\{[A], [B], [C], [E]\}$  will be as follows [7]:

$$A = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix} \quad (21)$$

$$B = \begin{bmatrix} V_g/L \\ 0 \end{bmatrix} \quad (22)$$

$$C = [0 \quad 1] \quad (23)$$

$$E = [0] \quad (24)$$

In order to find the control-to-output transfer function from the system quadruple represented by Equations (21) - (24), we apply the Laplace transformation to both sides of the state and output equations represented by system state-space equations  $\dot{x}(t) = Ax(t) + Bu(t)$  and  $y(t) = Cx(t) + Eu(t)$ . After rearranging the resulting terms, we get the following general input-to-output transfer function:

$$\frac{y}{u} = C(sI - A)^{-1}B + E \quad (25)$$

Applying Equations (21) - (24) in Equation (25), for the circuit values of  $\{V_g = 15 \text{ V}, R = 18.6 \Omega, D = 0.4, f_s = 40.3 \text{ kHz}, \hat{D} = 1 - D = 0.6, L = 58 \mu\text{H}, C = 5.5 \mu\text{F}\}$ , and to investigate the accuracy of the new PWM switch model, we compare the control-to-output frequency response plots of the PWM Buck converter, which is obtained through the application of the new PWM switch small-signal model, with both of the exact and the averaged control-to-output frequency responses, as shown in Fig. 13.

We null the input  $\hat{d}$  in order to obtain the input-to-output transfer function. The system quadruple  $\{[A], [B], [C], [E]\}$  will be as follows [7]:

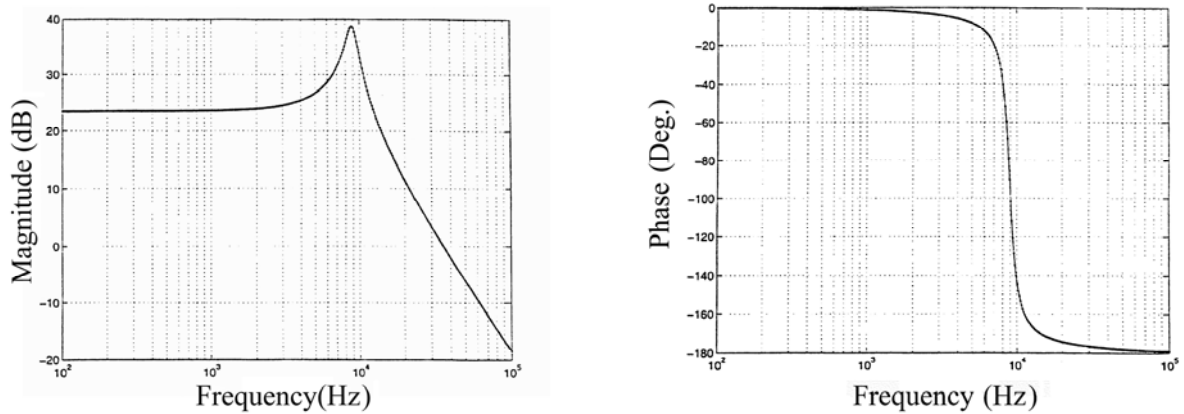
$$A = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix} \quad (26)$$

$$B = \begin{bmatrix} D/L \\ 0 \end{bmatrix} \quad (27)$$

$$C = [0 \quad 1] \quad (28)$$

$$E = [0] \quad (29)$$

By the application of Equations (26) - (29) in Equation (25), for the circuit values of  $\{R = 18.6 \Omega, D = 0.4, f_s = 40.3 \text{ kHz}, \hat{D} = 1 - D = 0.6, L = 58 \mu\text{H}, C = 5.5 \mu\text{F}\}$ , and to investigate the accuracy of the new PWM switch model, we compare the input-to-output frequency response plots of the PWM Buck converter, which is obtained through the application of the new PWM switch small-signal model, with both of the exact and the averaged input-to-output frequency responses, as shown in Fig. 14.



**Fig. 13.** The control-to-output magnitude and phase frequency response of the Buck converter operating in CCM: exact (solid line), averaged (dotted line), and the new model (dashed line), where inner parallel lines correspond to the values in plots.

One can note, from the previous frequency response plots for both of the control-to-output and the input-to-output transfer functions of the PWM Buck converter which is operating in the CCM, that an excellent match occurs between the exact and the new model results, as well as between the averaged and the new model results.

The observed results indicate, for the time being, that the newly utilized small-signal model of the PWM switch is, in fact, an accurate model [7].

The effect of the new source coefficients  $h_1$  and  $h_2$  that exist in the new model of the PWM switch, does not appear in the case of the control-to-output and input-to-output transfer functions. So, we need the input impedance and the control-to-input current transfer functions to investigate the effect of the new source coefficients  $h_1$  and  $h_2$ , respectively.

Referring to Fig. 12, and considering the input current  $\hat{i}_a$  to be the output, we get the following output equation  $y = \hat{i}_a$ .

For the converter states  $x$  and the inputs  $u$ , where  $x = \begin{bmatrix} \hat{i}_l \\ \hat{v}_o \end{bmatrix}$

and  $u = \begin{bmatrix} \hat{v}_g \\ \hat{d} \end{bmatrix}$ , the system quadruple  $\{[A], [B], [C], [E]\}$  will be as follows [7]:

$$A = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix}, \quad B = \begin{bmatrix} D/L & V_g/L \\ 0 & 0 \end{bmatrix}, \quad C = [D \quad 0], \\ E = [h_1 \quad h_2].$$

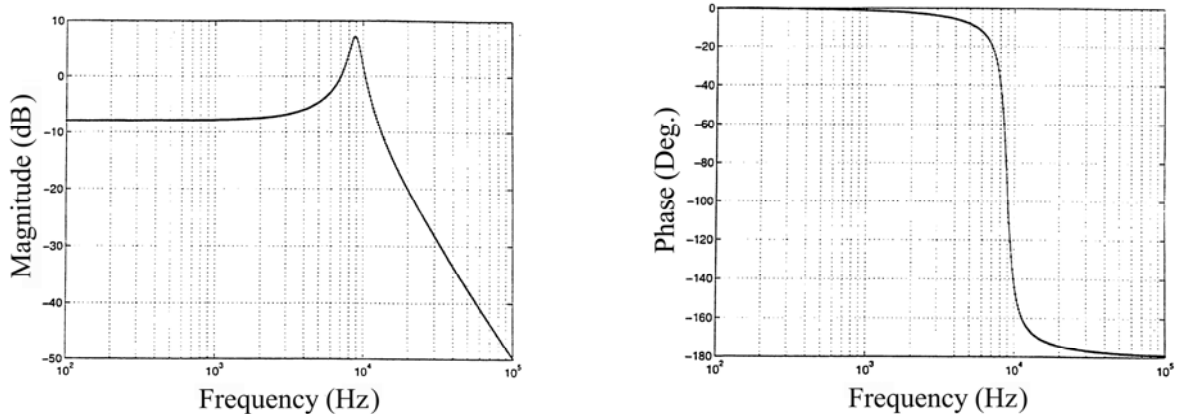
One nulls the input  $\hat{v}_g$  in order to find the control-to-input current transfer function. Thus, the new system quadruple  $\{[A], [B], [C], [E]\}$  will be as follows [7]:

$$A = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix} \quad (30)$$

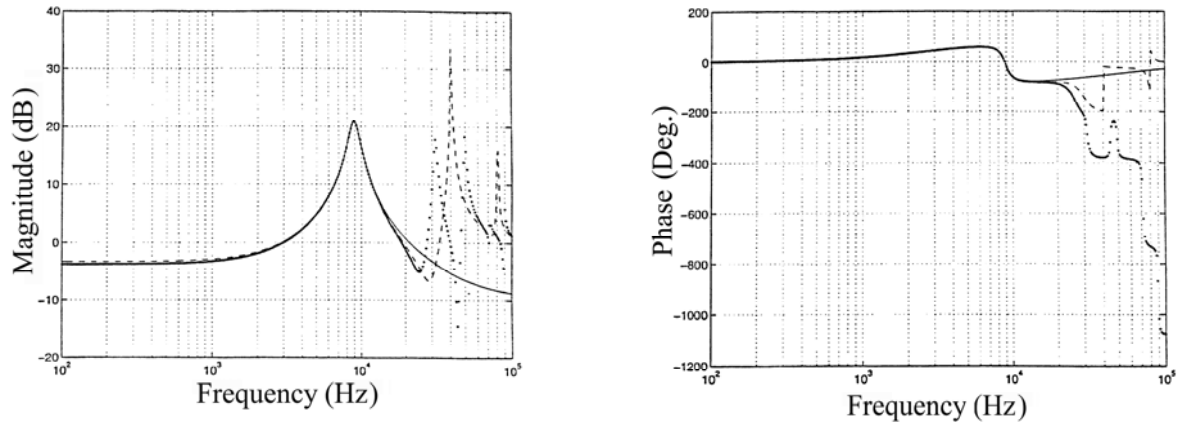
$$B = \begin{bmatrix} V_g/L \\ 0 \end{bmatrix} \quad (31)$$

$$C = [D \quad 0] \quad (32)$$

$$E = [h_2] \quad (33)$$



**Fig. 14.** The input-to-output magnitude and phase frequency response of the Buck converter operating in CCM: exact (solid line), averaged (dotted line), and the new model (dashed line), where inner parallel lines correspond to the values in plots.



**Fig. 15.** The control-to-input current magnitude and phase frequency response of the Buck converter operating in CCM: exact (solid line), averaged (dotted line), and the new model (dashed line), where inner parallel lines correspond to the values in plots.

One uses the general input-to-output transfer function which is represented by Equation (25) in order to find the control-to-input current transfer function from the system quadruple represented by Equations (30) - (33).

Applying Equations (30) - (33) in Equation (25), for the circuit values of  $\{V_g = 15 \text{ V}, R = 18.6 \Omega, D = 0.4, f_s = 40.3 \text{ kHz}, D' = 1 - D = 0.6, L = 58 \mu\text{H}, C = 5.5 \mu\text{F}\}$ , and to investigate the accuracy of the new PWM switch model, we compare the control-to-input current frequency response plots of the PWM Buck, which is obtained through the application of the new PWM switch small-signal model, with both of the exact and the averaged control-to-input current frequency responses, as shown in Fig. 15.

In order to obtain the input impedance transfer function, we null the input  $\hat{d}$ . Thus, the system quadruple  $\{[A], [B], [C], [E]\}$  will be as follows [7]:

$$A = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix} \quad (34)$$

$$B = \begin{bmatrix} D/L \\ 0 \end{bmatrix} \quad (35)$$

$$C = [D \quad 0] \quad (36)$$

$$E = [h_1] \quad (37)$$

Applying Equations (34) - (37) in Equation (25), for the circuit values of  $\{V_g = 15 \text{ V}, R = 18.6 \Omega, D = 0.4, f_s = 40.3 \text{ kHz}, D' = 1 - D = 0.6, L = 58 \mu\text{H}, C = 5.5 \mu\text{F}\}$ , and to investigate the accuracy of the new PWM switch model, we compare the input impedance frequency response plots of the PWM Buck, which is obtained through the application of the new PWM switch small-signal model, with both of the exact and the averaged input impedance frequency responses, as shown in Fig. 16.

We see, from the previous frequency response plots of the transfer functions of the PWM Buck converter operating in the CCM, that a good match occurs between the exact and the

new model results, as well as between the averaged and the new model results, for the frequency range up to half of the switching frequency [7], although a mismatch occurs between the exact and the new model results, as well as between the averaged and the new model results, for the frequency range higher than half of the switching frequency [7].

In an overall performance evaluation, the new small signal model behaves in a much accurate response than the older averaged modeling approach.

## 2.2. Fuzzy Logic

Computational fuzzy logic is considered as an efficient tool for embedding structured human knowledge into useful algorithms that has a large number of existing applications in human sciences, natural sciences and engineering applications [1, 6, 14-15, 17, 19-20, 23-25, 28, 34-35, 37-40, 43-47, 49]. It is a precise engineering tool developed to do a good job of trading off precision and significance. As in human reasoning and inference, the truth of any statement, measurement, or observation is a matter of degree. This degree is expressed through membership functions that quantify (i.e., measure) the degree of belonging of some (crisp) input to given fuzzy subsets. Fig. 17 shows the difference between crisp set used in crisp logic and fuzzy set used in fuzzy logic.

Membership or belonging is considered as a fundamental concept within the field of set theory. In the classical crisp convention, there are two possibilities that  $x$  belongs to  $A$  or it does not. This can be compared to fuzzy notion where a membership function describes the degree of belonging. Thus, crisp sets all have precise boundaries, where fuzzy sets have imprecise boundaries. The membership  $\mu$  is "0" or "1" for the crisp sets and  $(0 \leq \mu \leq 1)$  for the fuzzy sets. For Fig. 17(a), the set is crisp in that:

$$\mu = \begin{cases} 1, & x_1 < x < x_4 \\ 0, & \text{otherwise} \end{cases} \quad (38)$$

and for Fig. 17(b), the set is fuzzy in that:



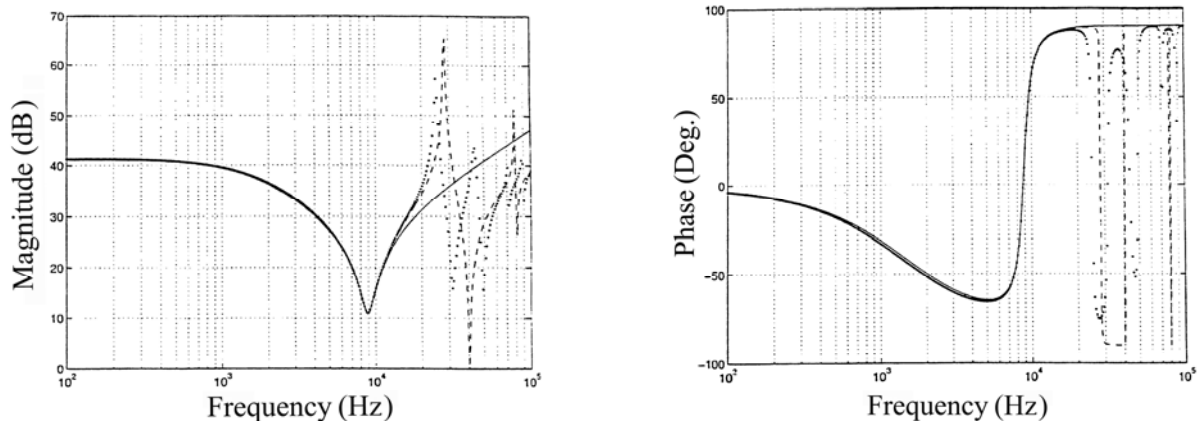


Fig. 16. The input impedance magnitude and phase frequency response of the Buck converter operating in CCM: exact (solid line), averaged (dotted line), and the new model (dashed line), where inner parallel lines correspond to the values in plots.

$$\mu = \begin{cases} [0,1], & x_1 < x < x_2 \text{ and } x_3 < x < x_4 \\ 1, & x_2 < x < x_3 \\ 0, & \text{otherwise} \end{cases} \quad (39)$$

As in crisp sets, fuzzy sets are subject to set operations such as union, intersection and complement. There are many functions that describe the union and intersection operations, where the mostly used ones in fuzzy logic are the *max* and *min* functions as follows:

Union:  $\mu_C(x) = \mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$  (40)

Intersection:  $\mu_C(x) = \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$  (41)

Fuzzy logic, as previously stated, is based on representing human reasoning as a classical binary relation. The concept of relation is general; it is based on the concept of ordered pairs (a, b), where a relation from A to B (or between A and B) is any subset  $\mathfrak{R}$  of the Cartesian product A x B. We say that  $\{a \in A, b \in B\}$  are related by  $\mathfrak{R}$ .

Usually, fuzzy logic is expressed in terms of (if ... and ... then) form. Fig. 18 shows an example of this (if ... and ... then) rule where the actual meaning of the (if ... and ... then) rules is (if x is  $A_i$  and y is  $B_j$  then z is  $C_k$ ).

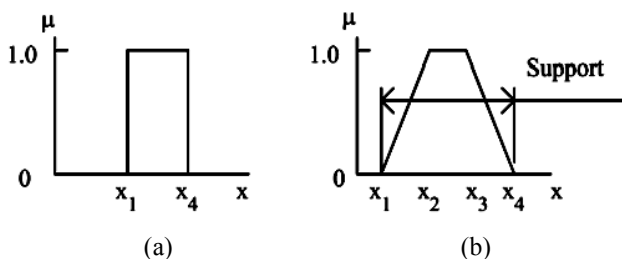


Fig. 17. An example of two types of sets: (a) crisp set and (b) fuzzy set.

The mathematical fuzzy interpretation of AND is the intersection. For example, the intersection of  $A_i$  and  $B_j$  is treated using the *min* function as follows:

$$\mu_{A_i \cap B_j} = \min(\mu_{A_i}, \mu_{B_j}) = \alpha \quad (42)$$

The membership value  $\alpha$  is called the power of the rule or the firing power. This part is then intersected with  $C_{ij}$  to simulate the “then” part of the (if...and...then) rule. This intersection is expressed as a clipped fuzzy rule as shown in Fig. 19.

A crisp input can cause the firing of several rules. This is interpreted as the aggregation or union of these rules, where the final part of the aggregation of the rules is usually interpreted as the *max* operation. An example of rule aggregation is shown in Fig. 20, and Fig. 21 shows an example of firing rules within fuzzy logic.

Defuzzification is the final part that is usually utilized within fuzzy implementation. The defuzzification process has several techniques including the center of area method where we divide the area into equally spaced rectangles and for each we find the membership function. This is shown for the rule aggregation in Fig. 20 as follows:

	$B_1$	...	$B_j$	$B_{j+1}$	...	$B_m$
$A_1$	$C_{11}$	...	$C_{1j}$	$C_{1,j+1}$	...	$C_{1,m}$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
$A_i$	$C_{i1}$	...	$C_{ij}$	$C_{i,j+1}$	...	...
$A_{i+1}$	$C_{i+1,1}$	...	$C_{i+1,j}$	$C_{i+1,j+1}$	...	$C_{i+1,m}$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
$A_n$	$C_{n,1}$	...	$C_{n,j}$	$C_{n,j+1}$	...	$C_{n,m}$

Fig. 18. An example of a decision table which is represented by (if ... and ... then) rules.

$z_k$	10	20	30	40	50	60	70
$\mu_{agg}$	2/3	2/3	2/3	1/3	1/6	1/6	1/6

When implementing defuzzification using the center of area method, one obtains, for the above example, the following defuzzified value:

$$z_c = \frac{\sum_{k=1}^{q-1} z_k \mu_{agg}(z_k)}{\sum_{k=1}^{q-1} \mu_{agg}(z_k)}$$

$$= \frac{10\left(\frac{2}{3}\right) + 20\left(\frac{2}{3}\right) + 30\left(\frac{2}{3}\right) + 40\left(\frac{1}{3}\right) + 50\left(\frac{1}{6}\right) + 60\left(\frac{1}{6}\right) + 70\left(\frac{1}{6}\right)}{\frac{2}{3} + \frac{2}{3} + \frac{2}{3} + \frac{1}{3} + \frac{1}{6} + \frac{1}{6} + \frac{1}{6}}$$

$$= 29.41$$

### 2.3. Genetic Algorithms

Basic background for the evolutionary-based algorithms is presented in this subsection. In general, evolutionary computing (EC) is one type of “black box” global optimization methods that has been successfully implemented to solve for many difficult nonlinear problems. An EC implements the idea which was proposed by Darwin as an explanation of the biological world surrounding us which is the "Evolution by Natural Selection". By evolution, we mean the change of the genes that produce a structure. The result of this evolution is the survival of the fittest and the elimination of the unfit. Darwin's theory of evolutionary selection states that variation within species occurs randomly and that the survival or extinction of each organism is determined by that organism's ability to adapt to its environment.

This powerful and simple EC idea has been implemented in algorithms such as in genetic algorithms (GA) and genetic programming (GP), and found wide spectrum of applications in several natural and applied fields [13-14, 18, 21-22, 24, 26-27, 29, 33, 36, 41-42, 44, 48, 50].

The difference between GA and GP is the representation of the problem and consequently the set of genetic operators used to obtain the solution; GA uses string representation and the

consequent genetic operators, while GP uses tree representation and the consequent genetic operators. Fig. 22(a) represents the general optimization using the EC method, where iterations on this flow diagram are made until the actual output matches exactly the desired output (i.e., without error) or the actual output mismatches the desired output within an acceptable range of error.

The idea of genetic algorithms is based on the simulation of life, where the first step is usually to represent the problem variables as chromosomes also called genomes. The common operations within a GA are as follows (cf. Fig. 22(b)):

(a) Initialization: within this step, the chromosomes are generated randomly to cover the search space and in some special cases the population is seeded with special solution or optimal solutions.

(b) Selection: there are several types used for selection such as: (1) fitness proportionate selection or roulette-wheel selection (a single random number is used), (2) stochastic universal sampling (multiple random numbers are generated for selection), (3) tournament selection (best individuals are always selected), (4) truncation selection (a portion of the population is selected), and (5) elitism or elitist selection (where the best individual(s) are always selected).

(c) Crossover: this operation involves the combination of genes from two parents to produce offsprings. There are several variants of crossover: (1) single point crossover where a fixed position is selected in both parents and then the contents beyond that crossover point are swapped, (2) multiple crossover points, (3) cut and slice crossover (change in length between the parents and the children), and (4) uniform crossover where a random number is generated and, if it is greater than a threshold value, then swapping is performed.

(d) Mutation: this process involves the reproduction of an erroneous copy of the individual, in which a random number is generated where if it is greater than a threshold value then the zero binary value is changed to one. This part is added to increase the diversity.

(e) Copying: this process involves the reproduction of an exact copy of the individual.

(f) Termination: where a certain number of generations is reached, or an acceptable solution is reached, or no change in the optimal solution is reached.

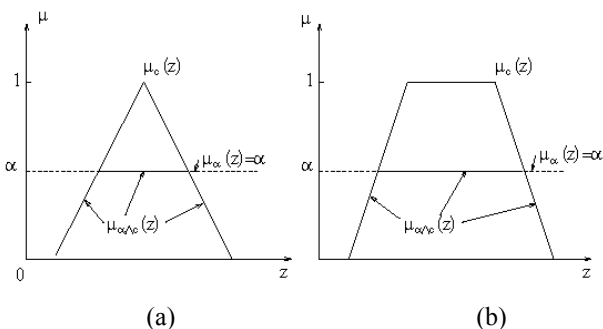


Fig. 19. Clipped triangular and trapezoidal membership functions.

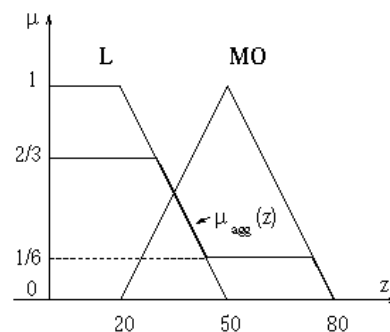


Fig. 20. An example of fuzzy rule aggregation with firing powers  $\alpha_1 = 2/3$  and  $\alpha_2 = 1/6$ .

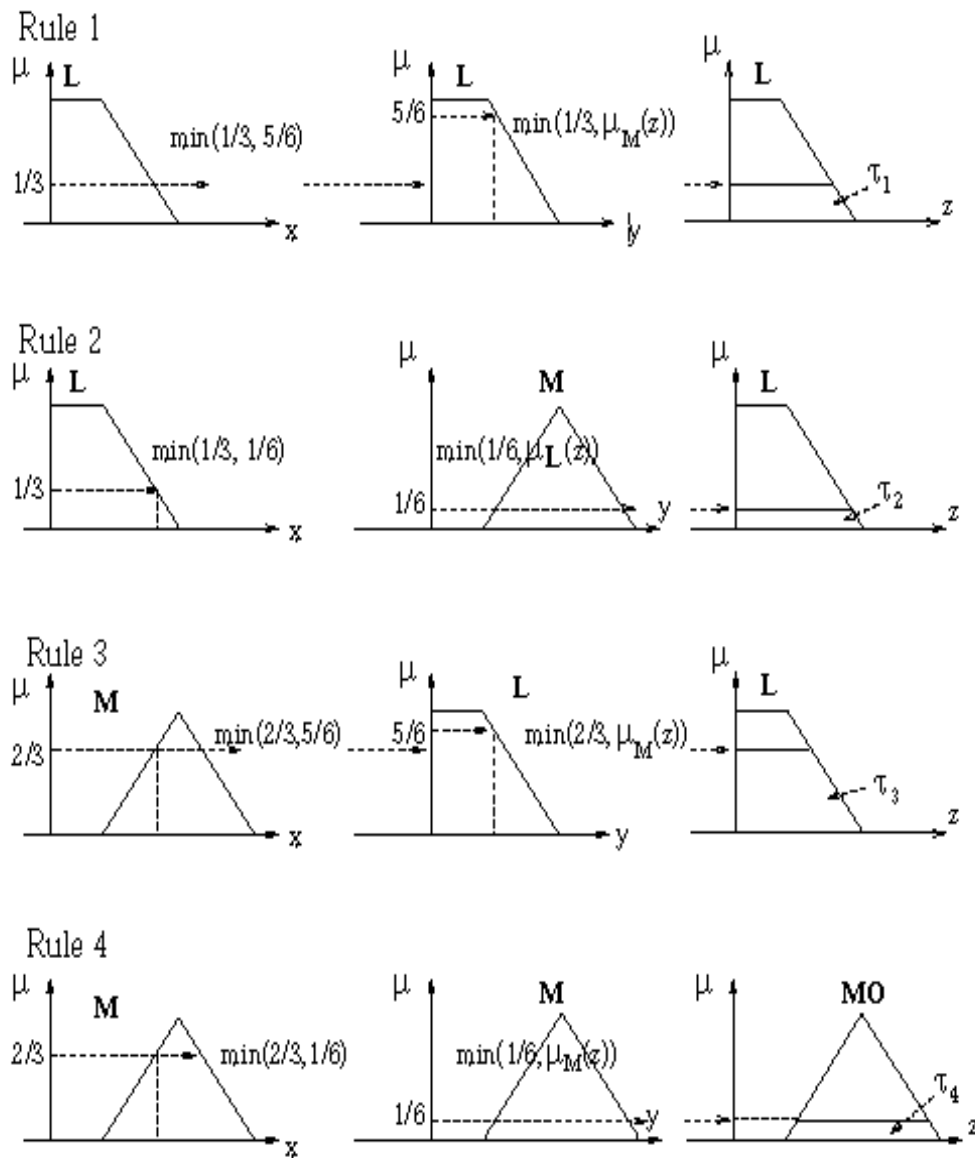


Fig. 21. Several rules applied in fuzzy logic with various firing powers  $\tau_i$ .

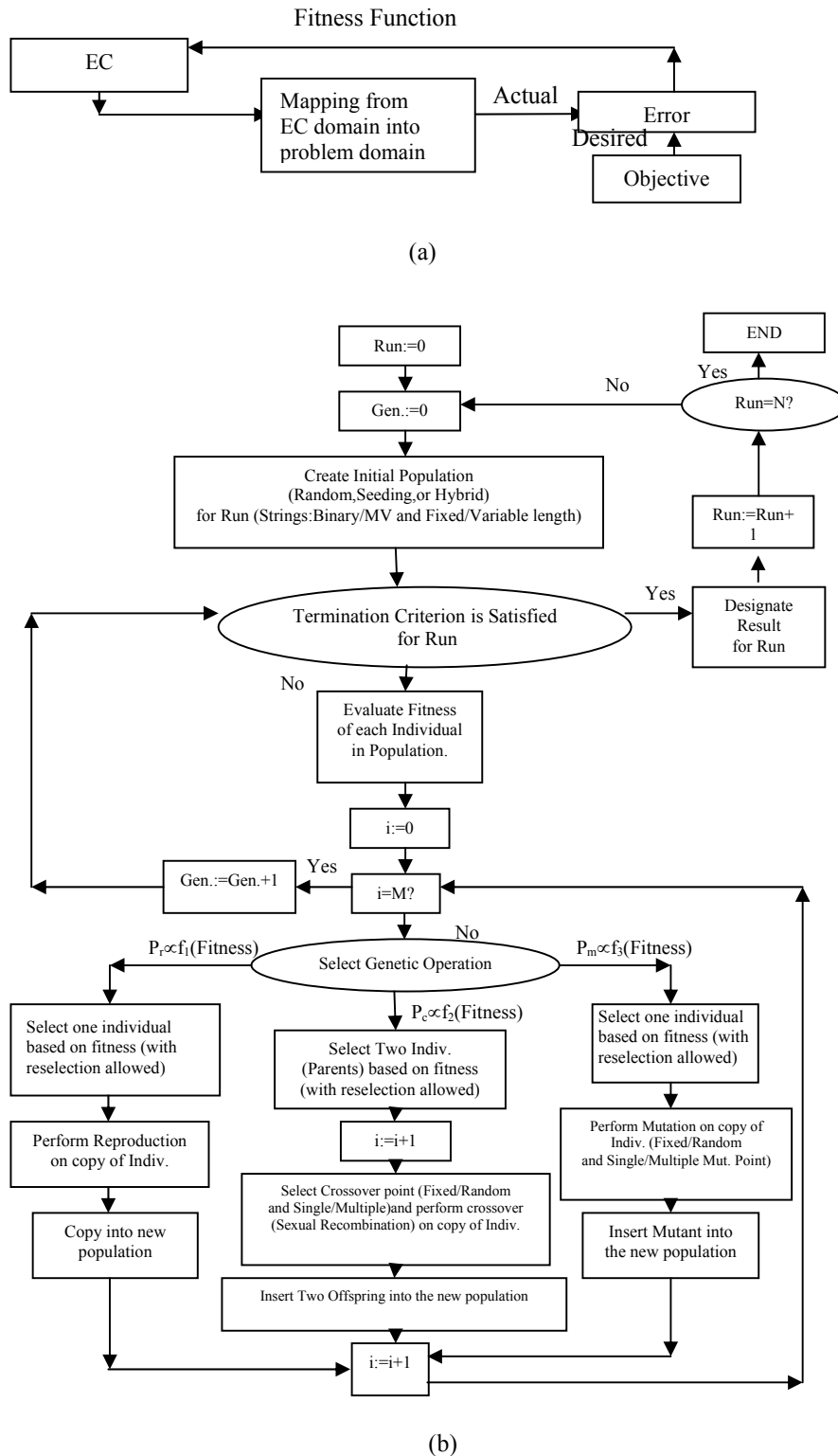


Fig. 22. Evolutionary computation (EC) and GA: (a) block diagram showing the mechanism of problem solving utilizing EC, and (b) flow graph of a generally-used GA.

A general flow diagram of a GA is illustrated in Fig. 22(b), where  $Run$  is the current run number,  $N$  is the maximum number of runs,  $Gen.$  is the current generation number,  $M$  is the population size,  $i$  is the current individual in the population,  $P_r$  is the probability of reproduction,  $P_c$  is the probability of crossover,  $P_m$  is the probability of mutation where  $(P_r + P_c + P_m = 1.0)$ . In Fig. 22(b), the result of looping over  $Gen.$  is the best-of-run individual, the result of looping over  $Run$  is the best-of-all individual, and the result of looping over  $i$  is the best-of-generation individual. Iterations in Fig. 22(b) continue until an optimal solution is obtained. Since the EC algorithms are try-and-check (i.e., try-and-error) probabilistic search algorithms (i.e., depends on the reduction of error in the search process to produce a solution), the EC program may have to perform so many iterations (as in Fig. 22(b)) to produce the desired solution to a problem. Thus, and although EC methods produce in many occasions new solutions that humans never made before, it is in general highly advisable to consider EC as one final option for problem solving (i.e., when other methods fail to solve the problem), since EC acts like a “black box” that produces solutions without showing a detailed step-by-step method.

From Fig. 22(b), the evolutionary algorithm has many variants. Yet, a canonical form for all of these variants exist. Fig. 23 illustrates one possible canonical diagram for evolutionary computing, where selecting survivors means (1) selection of parents and (2) generation of offspring.

The EC canonical diagram, shown in Fig. 23, characterizes the canonical implementation of various types of EC such as GA, and as stated previously, the only difference between GA and other EC (such as the GP) will be in (1) the internal representation of chromosomes operated upon and (2) the types of internal operations used accordingly. Fig. 24 shows an example of several important GA operations.

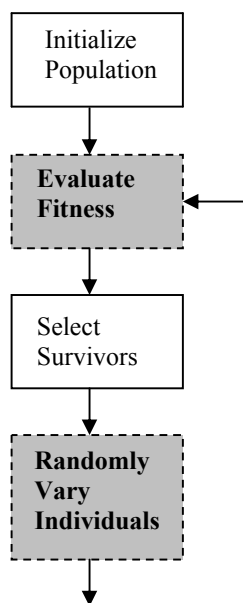


Fig. 23. A canonical flow graph for evolutionary methods.

### 3. GENETIC-BASED SETTINGS FOR THE BUCK-BASED FUZZY CONTROL

Basic Simulink and MATLAB setups, and the GA-based settings of the fuzzy controller, that will be used in Section 4 to obtain the fuzzy-PID control results, are presented in this section.

#### 3.1. Simulink and MATLAB Setups

Within MATLAB, solvers are divided into two main types of (a) fixed-step solvers and (b) variable-step solvers. Both types of solvers compute the next simulation time as the sum of the current simulation time and a quantity known as the step size. With a fixed-step solver, the step size remains constant throughout the simulation. On the contrast, with a variable-step solver, the step size can vary from step to step, depending on the model's dynamics. In particular, a variable-step solver reduces the step size when the model's states are changing rapidly in order to maintain accuracy and increases the step size when the system's states are changing slowly in order to avoid taking unnecessary steps.

Control type within the Simulink solver configuration allows selecting either of these two types of solvers. Fixed-step solvers have lower chances of missing an event in the model as compared to a variable-step solver that may cause the simulation to miss error conditions that can occur on a real-time computer system. Thus, for this work, fixed-step solvers are used for a step size of 0.001s to ensure capturing all of the dynamics occurring in the Buck system. If the step size is chosen less than this value, it will be highly time consuming for the GA code to run as the model will take large amount of time to run. Other step-size values such as 0.01s were tested but the model results were not as accurate as that of 0.001s.

Other configurations for the MATLAB solvers are (a) continuous time and (b) discrete time, where continuous time solvers can handle both of the discrete and continuous blocks which is the case for the analyzed system.

Therefore, we have chosen the continuous solvers. Within the prospect of continuous systems, we can use (a) implicit solvers or (b) explicit solvers by using implicit or explicit functions. The implicit solvers are more time consuming than explicit solvers, and thus explicit solvers were used with the Runge-Kutta (RK4) model because of the optimization part.

#### 3.2. Genetic-Based Settings for the Centers of Fuzzy Membership Functions

In this case, the centers within Fig. 25 for the inputs and outputs can be set by the GA algorithm and not the gains  $\{k_e, k_{de}, \text{Alpha}, \text{Beta}\}$ . The Simulink is then executed with the generated fuzzy logic variable. The sum-of-square error (SSE) is calculated as the fitness value.

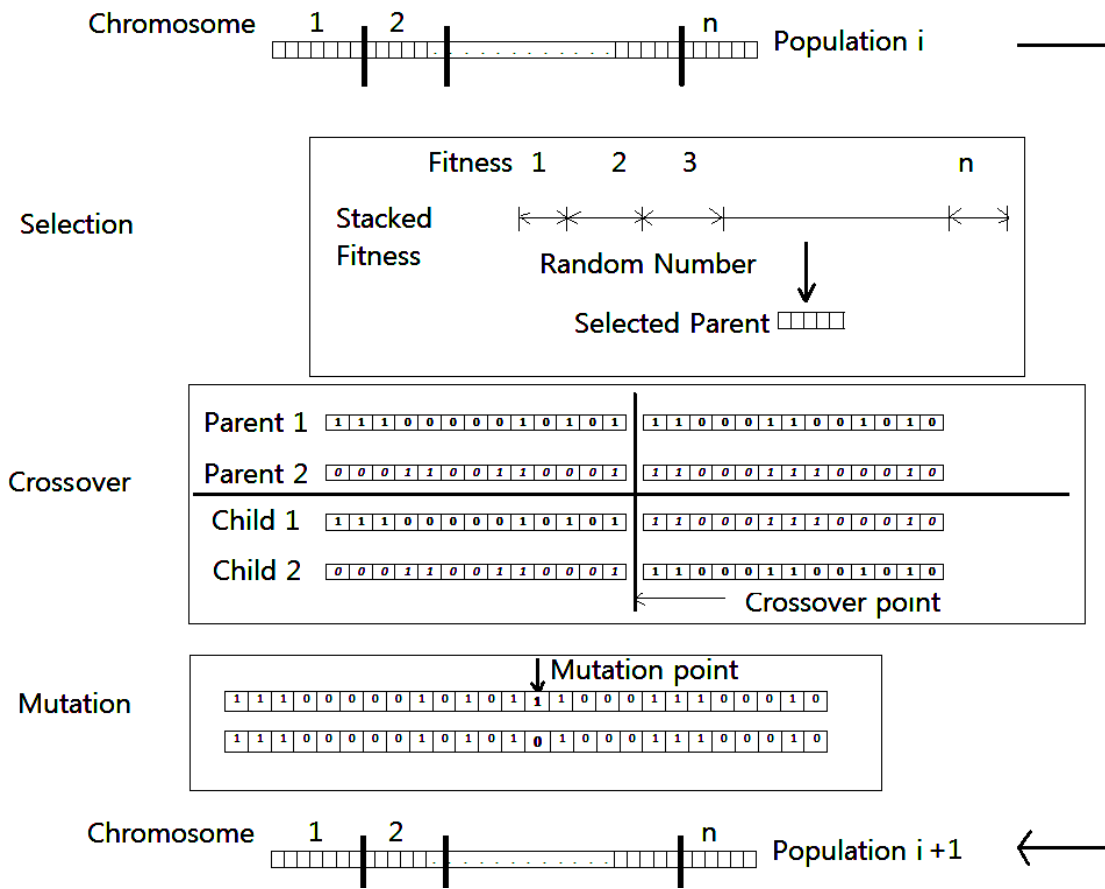


Fig. 24. An illustrative example of important GA operations.

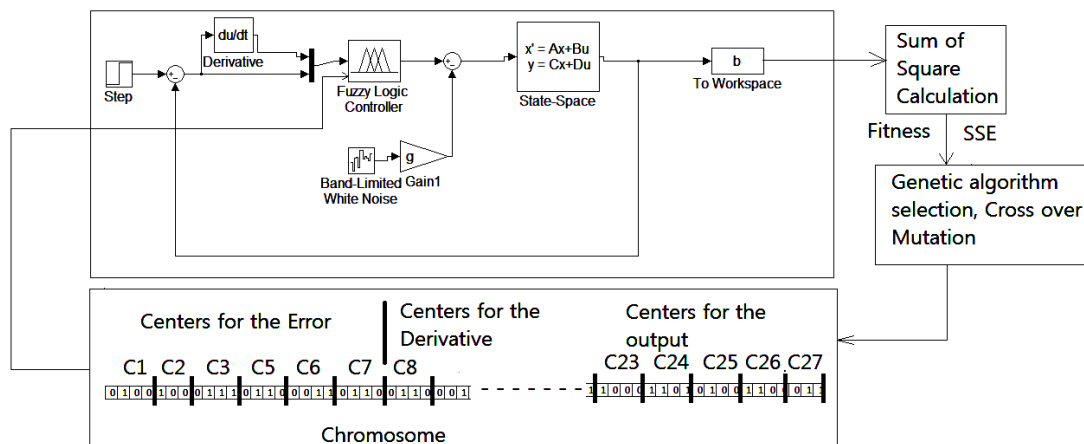


Fig. 25. The GA settings for the centers of the inputs and outputs within the Buck system.

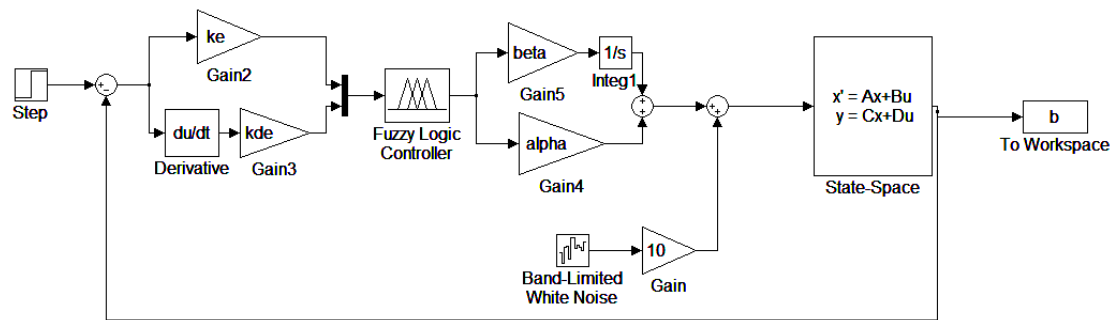


Fig. 26. The utilized Simulink block diagram for the GA-based settings of the fuzzy variables within the Buck system.

The GA, which was explained earlier, then regenerates a new population using the selection, crossover and mutation genetic operators. However, this can lead to lengthy GA runs and thus the convergence time for the GA could be very high. Therefore, this setup was not used in this research.

**3.3. Fuzzy-PID Case: Genetic-Based Settings for the Fuzzy Variables**

Alternative to the method shown in sub-section 3.2, Fig. 26 shows the Simulink block diagram that is used in this work for the GA settings of the fuzzy-PID variables, where GA changes the chromosomes for the settings of the fuzzy variables (cf. Fig. 28). The fuzzy-PID method shown in Fig. 26 is a standard commonly used PID control form which is utilized in several other applications [15, 24, 44].

**4. GENETIC-BASED SETTINGS FOR THE FUZZY VARIABLES OF THE BUCK CONVERTER**

The simulation results for the GA settings of the fuzzy variables for both of the input-to-output and control-to-output Buck transfer functions are presented in this section.

**4.1. Genetic-Based Settings for the Fuzzy Variables for the Input-to-Output Buck Transfer Function**

The error is calculated, in Fig. 26, first using the summing function as the difference between the input and the output. Followed to that, the proportional part and the derivative part are calculated and multiplied by the counterpart gain { $k_e$ ,  $k_{de}$ }.

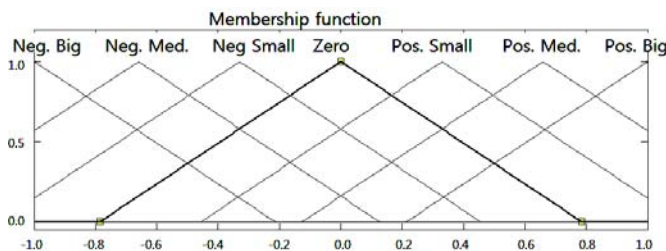


Fig. 27. Fuzzy sets for the error, derivative and the output.

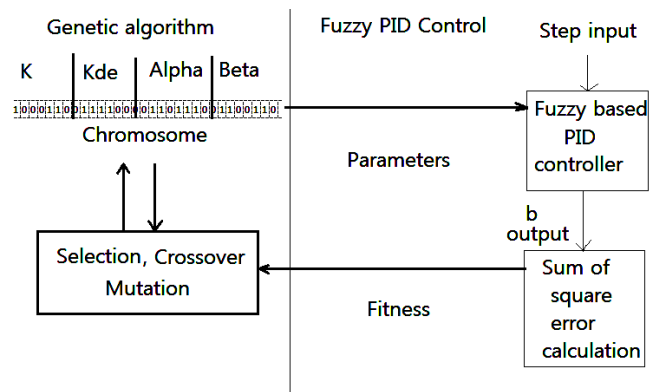


Fig. 28. Block diagram that presents the utilized interaction between the GA-based settings and fuzzy-PID controller.

After that, this is used as an input to a multiplexer, and then these two inputs (i.e., proportional and derivative) are used as an input to the fuzzy logic part. These are then fuzzified using the fuzzy sets and membership functions shown in Fig. 27.

Fuzzified variables are then processed using the Mamdani-type fuzzy system using the rules in Table (1). The centroid type defuzzification system is then estimated. The output is then multiplied by the corresponding gains {Alpha, Beta} and then integration is used.

As illustrated, Fig. 28 shows the block diagram of the interaction between the genetic algorithm part and the fuzzy-PID controller part that is used in this work, and Fig. 29 illustrates a sample run for the utilized fuzzy control. The GA is based on representing the different parameters { $k_e$ ,  $k_{de}$ , Alpha, Beta} as a chromosome. The fuzzy-PID controller runs the model with the selected values for these parameters and passes the output to an M-file which estimates the sum of the square error (SSE). This in turn is treated as the fitness function. The GA then performs the genetic operations of selection, crossover and mutation on the chromosomes and produces a new population which in turn uses the fuzzy-PID controller model to estimate the fitness. This cycle continues until a suitable minimum value is reached for termination.

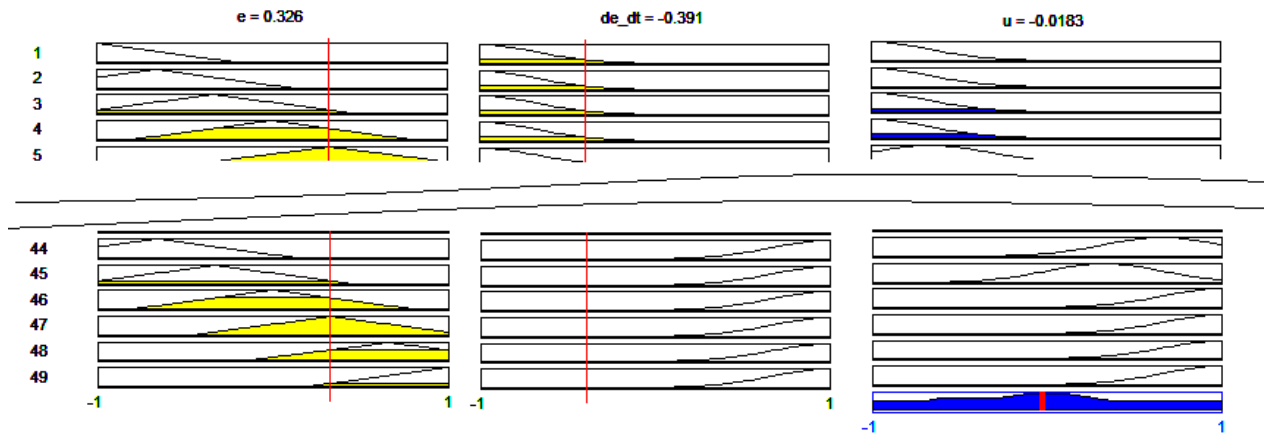


Fig. 29. A sample run for the used fuzzy control method.

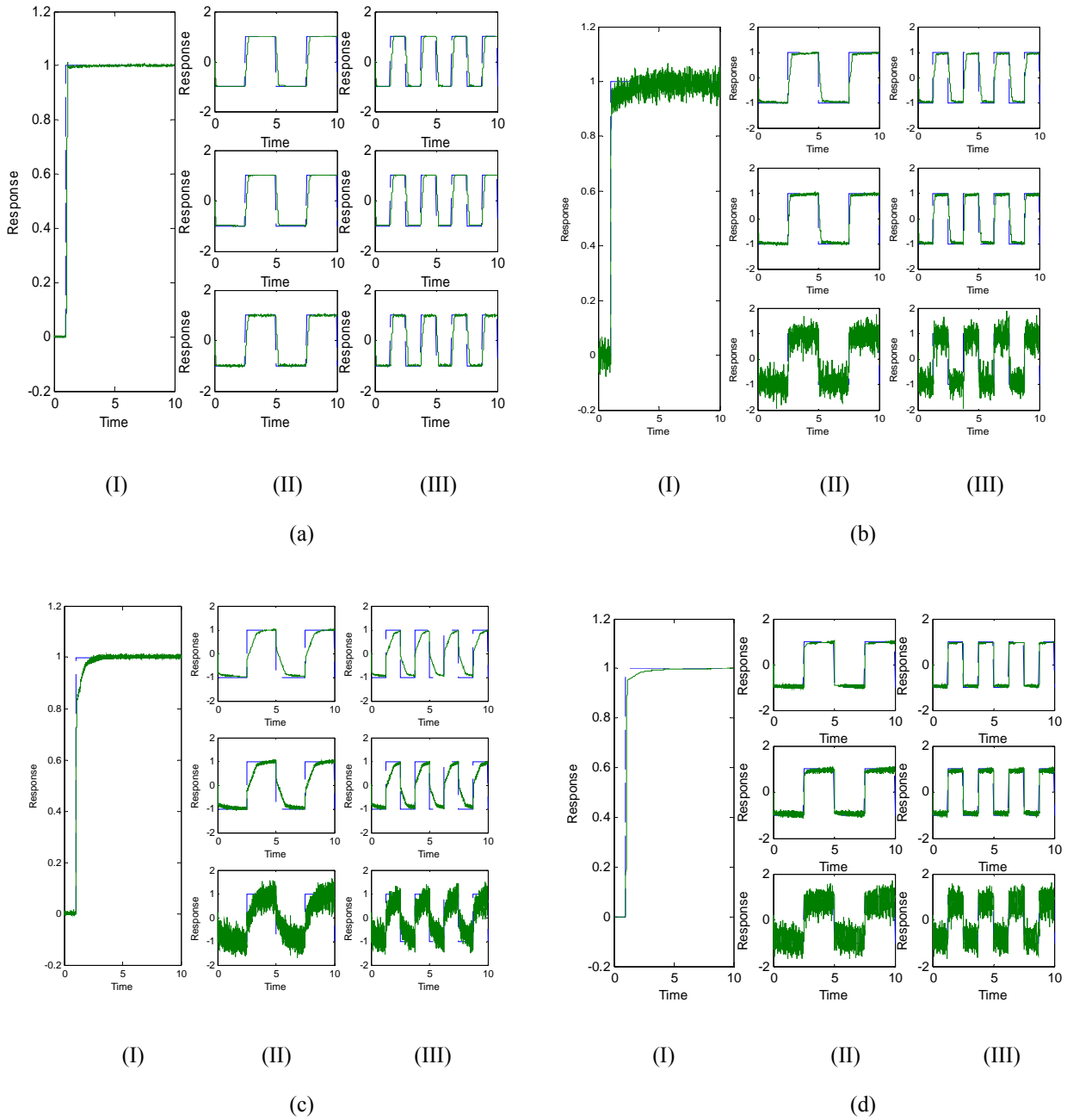
Fig. 30 presents the simulation results for the input-to-output Buck transfer function using the state-space matrices  $\{A = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix}, B = \begin{bmatrix} D/L \\ 0 \end{bmatrix}, C = [0 \quad 1], E = [0]\}$ , with noisy inputs for one step function and three square

wave functions, where the noise in the first square wave is 0.1:1 of the signal, the noise in the second square wave is 1:1 of the signal, and the noise in the third square wave is 10:1 of the signal.

Table 1. Rules for the utilized Mamdani-type fuzzy system.

$\frac{de}{dt}$ \ e	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB





**Fig. 30.** Simulation results for the Buck input-to-output: (a) using system values  $\{D = 0.4, R = 18.6 \Omega, L = 5.8 \text{ H}, C = 0.55 \text{ mF}\}$ , (b) using system values  $\{D = 0.4, R = 18.6 \Omega, L = 580 \text{ mH}, C = 55 \mu\text{F}\}$ , (c) using system values  $\{D = 0.4, R = 18.6 \Omega, L = 30 \text{ mH}, C = 55 \mu\text{F}\}$ , and (d) using system values  $\{D = 0.4, R = 18.6 \Omega, L = 58 \text{ mH}, C = 55 \mu\text{F}\}$ .

To ensure that the system is tested well, different type of inputs were used including (a) step input in Figs. 30(I), (b) 0.2 Hz pulse input with noise levels {0.1:1, 1:1, 10:1} in Figs. 30(II), and (c) 0.4 Hz pulse input with noise levels {0.1:1, 1:1, 10:1} in Figs. 30(III). The different noise levels are used to test the ability of this system to reject the existing noise.

Fig. 30 shows that the system has good robustness against noise and possesses good accuracy for the steady-state response. However, the settling time is somewhat high, where these values are obtained as a result of the optimization process and shows the best obtained performance that the system can perform under the aforementioned conditions.

The system shows a low steady-state error, but comparatively bad noise rejection especially for high noise level of 10:1 when compared to the signal. The system gets worse for higher frequency values as the final steady-state value might not be reached for the used step time.

#### 4.2. Genetic-Based Settings for the Fuzzy Variables for the Control-to-Output Buck Transfer Function

The Buck dynamic system is then tested for the important control-to-output transfer function as shown in Fig. 31, where Fig. 31 presents the simulation results for the control-to-output Buck transfer function using the state space matrices  $\{A = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix}, B = \begin{bmatrix} V_g \\ 0 \end{bmatrix}, C = [0 \quad 1], E = [0]\}$ . The

Buck system is simulated for different types of inputs including (a) step input in Figs. 31(I), (b) 0.2 Hz pulse input with noise levels {0.1:1, 1:1, 10:1} in Figs. 31(II), and (c) 0.4 Hz pulse input with noise levels {0.1:1, 1:1, 10:1} in Figs. 31(III). The different noise levels are used to test the ability of this system to reject the corresponding noise.

The results in Fig. 31 show an acceptable Buck performance in terms of the steady-state value. Fig. 31 also shows a rapid response, but this system has the drawback of slight overshoots for small periods of time, especially at the beginning rising edges of the step and square wave signals, where these small overshoots can be ignored as they have usually a comparatively negligible effect on the overall performance of the Buck dynamic system.

The results in Fig. 31 show an acceptable steady-state Buck performance despite small overshoots that have usually a comparatively negligible effect on the overall performance.

### 5. CONCLUSIONS AND FUTURE WORK

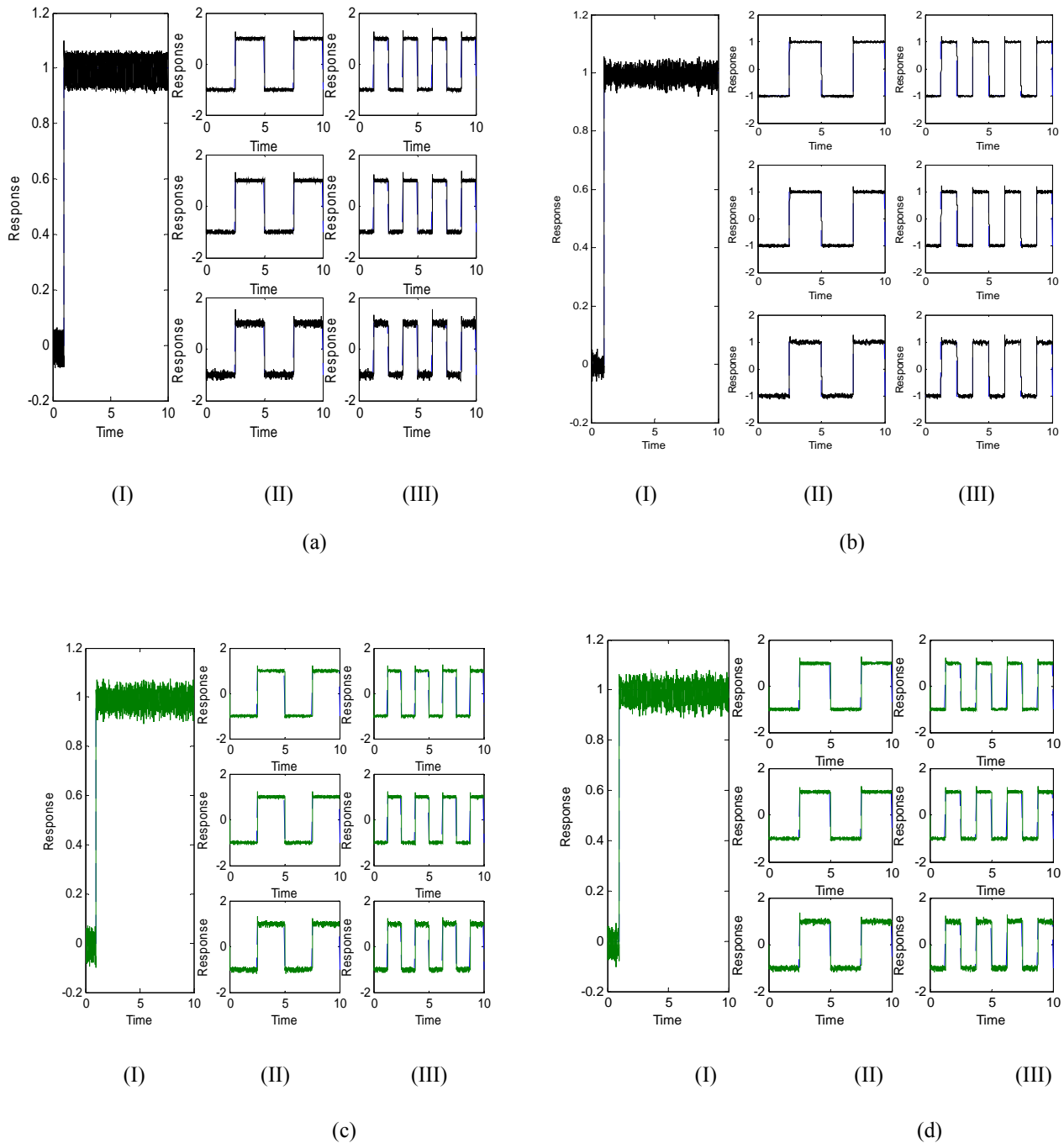
Further testing and verification for the implementation of a previously introduced new hierarchical intelligent regulation upon the electronic Buck power converter, using a newly developed small-signal model of the pulse width modulation switching, is introduced in this article. The hierarchical intelligent control uses the GA-based settings of the fuzzy-PID controller to counteract the existence and effect of high-amplitude noise. The additional experimental simulation results show that the presented control method, which is utilized using the new PWM small-signal model, succeeds in

minimizing the effect of noise even when noise is of several levels higher than the Buck-generated output signal.

The investigation of the application of the presented intelligent control method upon other important DC-to-DC power-electronic converters, e.g., Boost, Buck-Boost and C'uk converters, will be conducted in future work.

### REFERENCES

- [1] S. C. Abou, M. Kulkarni and M. Stachowicz, "Actuated Hydraulic System Fault Detection: A Fuzzy Logic Approach," *Engineering Letters*, 18:1, February 2010.
- [2] K.K. Ahn and N.B. Kha, "Modeling and Control of Shape Memory Alloy Actuators Using Preisach Model, Genetic Algorithm and Fuzzy Logic," *Mechatronics*, Vol. 18, pp. 141–152, 2008.
- [3] E. Akin, M. Kaya and M. Karakose, "A Robust Integrator Algorithm With Genetic Based Fuzzy Controller Feedback for Direct Vector Control," *Computers and Electrical Engineering*, Vol. 29, pp. 379–394, 2003.
- [4] M.S. Alam and M.O. Tokhi, "Hybrid Fuzzy Logic Control With Genetic Optimization for a Single-Link Flexible Manipulator," *Engineering Applications of Artificial Intelligence*, Vol. 21, pp. 858–873, 2008.
- [5] D. Alejo, P. Maussion and J. Faucher, "Multiple Model Control of a Buck Dc/Dc Converter," *Mathematics and Computers in Simulation*, Vol. 63, pp. 249–260, 2003.
- [6] H. Aljoumaa and D. Soffker, "Multi-Class Classification Approach based on Fuzzy-Filtering for Condition Monitoring," *IAENG International Journal of Computer Science*, 38:1, pp. 66–73, 2011.
- [7] A.N. Al-Rabadi, *An Approach to Exact Modeling of the PWM Switch*, M.Sc. Thesis, Electrical and Computer Engineering Department, Portland State University, 1998.
- [8] A.N. Al-Rabadi, "Intelligent Control of Singularly-Perturbed Reduced Order Eigenvalue-Preserved Quantum Computing Systems via Artificial Neural Identification and Linear Matrix Inequality Transformation," *IAENG International Journal of Computer Science*, Vol. 37, No. 3, pp. 210 – 223, 2010.
- [9] A.N. Al-Rabadi, "Artificial Neural Identification and LMI Transformation for Model Reduction-Based Control of the Buck Switch-Mode Regulator," *American Institute of Physics (AIP)*, In: IAENG Transactions on Engineering Technologies, Special Edition of the International MultiConference of Engineers and Computer Scientists 2009, AIP Conference Proceedings 1174, Vol. 3, pp. 202 – 216, 2009.
- [10] A.N. Al-Rabadi, "Intelligent Control of Reduced-Order Closed Quantum Computation Systems Using Neural Estimation and LMI Transformation," Springer-Verlag, In: Intelligent Control and Computer Engineering, Special Edition of the International MultiConference of Engineers and Computer Scientists 2010, Editors: Xu Huang, Oscar Castillo, and Sio-Iong Ao, 2010.
- [11] A.N. Al-Rabadi and O. M.K. Alsmadi, "Supervised Neural Computing and LMI Optimization for Order Model Reduction-Based Control of the Buck Switching-Mode Power Supply," *Int. Journal of Systems Science (IJSS)*, Taylor & Francis, Vol. 42, Issue 1, pp. 91–106, 2011.
- [12] A.N. Al-Rabadi and O. M.K. Alsmadi, "Soft Computing Using Neural Estimation with LMI-Based Model Transformation for OMR-Based Control of the Buck Converter," *IAENG Engineering Letters*, Vol. 17, No. 2, pp. 101 – 120, 2009.
- [13] A.N. Al-Rabadi, M.A. Barghash, and O.M. Abuzeid, "Intelligent Regulation Using Genetic Algorithm – Based Tuning for the Fuzzy Control of the Power Electronic Switching-Mode Buck Converter," *IAENG International Journal of Computer Science*, 38:4, 2011.
- [14] A.N. Al-Rabadi, M.A. Barghash, and O.M. Abuzeid, "Fuzzy Regulation for the Intelligent Control of Switching-Mode Buck Power-Electronic Converter Using Genetic Algorithm – Based Tuning," *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2012, IMECS 2012*, Hong Kong, pp. 839–859, 14–16 March 2012.



**Fig. 31.** Simulation results for the Buck control-to-output: (a) using system values  $\{V_g = 15 \text{ V}, C = 1 \text{ mF}, R = 1 \text{ k}\Omega, L = 5.8 \text{ H}\}$ , (b) using system values  $\{V_g = 20 \text{ V}, C = 2 \text{ mF}, R = 3 \text{ k}\Omega, L = 11 \text{ H}\}$ , (c) using system values  $\{V_g = 15 \text{ V}, C = 2 \text{ mF}, R = 1.5 \text{ k}\Omega, L = 5 \text{ H}\}$ , and (d) using system values  $\{V_g = 20 \text{ V}, C = 2 \text{ mF}, R = 1.5 \text{ k}\Omega, L = 7 \text{ H}\}$ .

- [15] R. Bandyopadhyay, U.K. Chakraborty and D. Patranabis, "Autotuning a PID Controller: A Fuzzy-Genetic Approach," *Journal of System Architecture*, Vol. 47, pp. 663-673, 2001.
- [16] A.J. Calderon, B.M. Vinagre and V. Feliu, "Fractional Order Control Strategies for Power Electronic Buck Converters," *Signal Processing*, Vol. 86, pp. 2803-2819, 2006.
- [17] J. Causaa, G. Karer, A. Núñez, D. Sáez, I. Škrjanc and B. Zupancic, "Hybrid Fuzzy Predictive Control Based on Genetic Algorithms For The Temperature Control of a Batch Reactor," *Computers and Chemical Engineering*, Vol. 32, pp. 3254-3263, 2008.
- [18] N.R. Cazarez-Castro, L.T. Aguilar and O. Castillo, "Fuzzy Logic Control With Genetic Membership Function Parameters Optimization for The Output Regulation of a Servomechanism With Nonlinear Backlash," *Expert Systems with Applications*, Vol. 37, pp. 4368-4378, 2010.
- [19] W.-J. Chang, W.-H. Huang and W. Chang, "Synthesis of Fuzzy Control for Inverter Pendulum Robot with H Infinity Performance Constraint," *IAENG International Journal of Computer Science*, 36:3, 2009.
- [20] F. Chetouane, S. Darenfed and P.K. Singh, "Fuzzy Control of a Gyroscopic Inverted Pendulum," *Engineering Letters*, 18:1, 2010.
- [21] C.-L. Chiang and C.-T. Su, "Tracking Control of Induction Motor Using Fuzzy Phase Plane Controller With Improved Genetic Algorithm," *Electric Power Systems Research*, Vol. 73, pp. 239-247, 2005.
- [22] T.C. Chin and X.M. Qi, "Genetic Algorithms for Learning the Rule Base of Fuzzy Logic Controller," *Fuzzy Sets and Systems*, Vol. 97, pp. 1-7, 1998.
- [23] R. Chinram and S. Malee, "L-Fuzzy Ternary Subsemirings and L-Fuzzy Ideals in Ternary Semirings," *IAENG International Journal of Applied Mathematics*, 40:3, 2010.
- [24] H.-J. Cho, K.-B. Cho and B.-H. Wang, "Fuzzy-PID Hybrid Control: Automatic Rule Generation Using Genetic Algorithms," *Fuzzy Sets and Systems*, Vol. 92, pp. 305-316, 1997.
- [25] C.-H. Chou, "Model Reference Adaptive Fuzzy Control: A Linguistic Space Approach," *Fuzzy Sets and Systems*, Vol. 96, pp. 1-20, 1998.
- [26] V. Congradac and F. Kulic, "HVAC System Optimization With CO<sub>2</sub> Concentration Control Using Genetic Algorithms," *Energy and Buildings*, Vol. 41, pp. 571-577, 2009.
- [27] R.P. Domingos, G.H.F. Caldas, C.M.N.A. Pereira and R. Schirru, "PWR's Xenon Oscillation Control Through A Fuzzy Expert System Automatically Designed By Means Of Genetic Programming," *Applied Soft Computing*, Vol. 3, pp. 317-323, 2003.
- [28] J. Dong, Y. Wang and G.-H. Yang, " $H_\infty$  and Mixed  $H_2/H_\infty$  Control of Discrete-Time T-S Fuzzy Systems With Local Nonlinear Models," *Fuzzy Sets and Systems*, Vol. 164, pp. 1-24, 2011.
- [29] H. Du and N. Zhang, " $H_\infty$  Control for Buildings With Time Delay in Control via Linear Matrix Inequalities and Genetic Algorithms," *Engineering Structures*, Vol. 30, pp. 81-92, 2008.
- [30] H. El-Fadil and F. Giri, "Robust Nonlinear Adaptive Control of Multiphase Synchronous Buck Power Converters," *Control Engineering Practice*, Vol. 17, pp. 1245-1254, 2009.
- [31] R.W. Erickson, *Fundamentals of Power Electronics*, 1<sup>st</sup> Edition, Chapman and Hall, 1997.
- [32] G.F. Franklin, J.D. Powell and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 3<sup>rd</sup> Edition, Addison-Wesley, 1994.
- [33] B. Grosman and D.R. Lewin, "Automated Nonlinear Model Predictive Control Using Genetic Programming," *Computers and Chemical Engineering*, Vol. 26, pp. 631-640, 2002.
- [34] H. Hagras, V. Callaghan and M. Colley, "Learning and Adaptation of an Intelligent Mobile Robot Navigator Operating in Unstructured Environment Based on a Novel Online Fuzzy-Genetic System," *Fuzzy Sets and Systems*, Vol. 141, pp. 107-160, 2004.
- [35] H. Han and C.-Y. Su, "Robust Fuzzy Control of Nonlinear Systems Using Shape-Adaptive Radial Basis Functions," *Fuzzy Sets and Systems*, Vol. 125, pp. 23-38, 2002.
- [36] J.V. Hansen, "Genetic Search Methods in Air Traffic Control," *Computers & Operations Research*, Vol. 31, pp. 445-459, 2004.
- [37] T. Hasuike and H. Katagiri, "Sensitivity Analysis for Random Fuzzy Portfolio Selection Model with Investor's Subjectivity," *IAENG International Journal of Applied Mathematics*, 40:3, 2010.
- [38] T. Hasuike, H. Katagiri and H. Ishii, "Efficient Strict Solution Methods for 0-1 Random Fuzzy Programming Problems Based on the Necessity Measure," *IAENG Int. Journal of Applied Mathematics*, 39:2, 2009.
- [39] H. Hedayati, "Connections Between Generalized Fuzzy Ideals and Sub-implicative Ideals of BCI-Algebras," *IAENG International Journal of Applied Mathematics*, 41:1, pp. 17-22, 2011.
- [40] T. Helmy and Z. Rasheed, "Independent Job Scheduling by Fuzzy C-Mean Clustering and an Ant Optimization Algorithm in a Computation Grid," *IAENG International Journal of Computer Science*, 37:2, 2010.
- [41] F. Herrera, M. Lozano and J.L. Verdegay, "A Learning Process for Fuzzy Control Rules Using Genetic Algorithms," *Fuzzy Sets and Systems*, Vol. 100, pp. 143-158, 1998.
- [42] F. Herrera, M. Lozano and J.L. Verdegay, "Tuning Fuzzy Logic Controllers by Genetic Algorithms," *International Journal of Approximate Reasoning*, Vol. 12, pp. 299-315, 1995.
- [43] C. Hua, X. Guan and G. Duan, "Variable Structure Adaptive Fuzzy Control for a Class of Nonlinear Time-Delay Systems," *Fuzzy Sets and Systems*, Vol. 148, pp. 453-468, 2004.
- [44] G. Jahedi and M.M. Ardehali, "Genetic Algorithm-Based Fuzzy-PID Control Methodologies for Enhancement of Energy Efficiency of a Dynamic Energy System," *Energy Conversion and Management*, Vol. 52, pp. 725-732, 2011.
- [45] S. Jassar, Z. Liao and L. Zhao, "Data Quality in Hybrid Neuro-Fuzzy based Soft-Sensor Models: An Experimental Study," *IAENG International Journal of Computer Science*, 37:1, 2010.
- [46] A. Kablan and W.L. Ng, "Enhancing High-Frequency Order Placement Strategies with Fuzzy Logic and Fuzzy Inference," *IAENG International Journal of Computer Science*, 37:4, 2010.
- [47] B. Kadmiry and D. Driankov, "A Fuzzy Flight Controller Combining Linguistic and Model-Based Fuzzy Control," *Fuzzy Sets and Systems*, Vol. 146, pp. 313-347, 2004.
- [48] C.L. Karr and L.M. Freeman, "Genetic-Algorithm-Based Fuzzy Control of Spacecraft Autonomous Rendezvous," *Engineering Applications of Artificial Intelligence*, Vol. 10, No. 3, pp. 293-300, 1997.
- [49] J. Kavikumar and A.B. Khamis, "Fuzzy Ideals and Fuzzy Quasi-ideals in Ternary Semirings," *IAENG International Journal of Applied Mathematics*, 37:2, 2007.
- [50] S.J. Kim, S. Oh, Y.G. Lee, M.G. Jeon, I.S. Kim and J.H. Kim, "A Control Methodology for the Feed Water Temperature to Optimize SWRO Desalination Process Using Genetic Programming," *Desalination*, Vol. 247, pp. 190-199, 2009.