APPLICATION OF A NONLINEAR GAUSSIAN ADAPTIVE PID WITH PSO OPTIMIZATION TO A BUCK CONVERTER

Erickson P. Puchta*, Ricardo Lucas*, Hugo V. Siqueira*, Mauricio S. Kaster*

*Departamento de Eletrônica
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Tecnológica Federal do Paraná
Ponta Grossa, PR, Brazil.

Emails: ericksonpuchta@gmail.com, ricardolucas@utfpr.edu.br, hugosiqueira@utfpr.edu.br, mkaster@utfpr.edu.br

Abstract— This work proposes the application of a Gaussian Adaptive PID (GAPID) to a power electronic converter in order to enhance its performance. The Gaussian function used to define the adaptive gains have some desired characteristics: it is lower and upper bounded and is smooth with smooth derivatives. A smooth function helps avoiding problems related to abrupt changes in the gains. However, there aremno algebraic methodology to design such controller. In this sense, the PSO algorithm is employed for optimization of the GAPID parameters. These parameters are linked to the linear PID parameters in order to preserve the same design constraints and, in addition, to present better robustness to load variations. Simulations and experimental results show that significant performance improvements can be achieved.

Keywords— Gaussian Adaptive PID Control, Adaptive Control, PSO Algorithm.

1 Introduction

For several years, power electronics have achieved its enhancements (lower losses, higher efficiency and so on) by means of new topologies ((Romaneli and Barbi, 2000) (Ceglia et al., 2004) (Costa et al., 2013)). It is still true that new topologies will continue to be explored, but actually several improvements have been reached by means of better control techniques applied to already known topologies ((Hwu and Yau, 2010) (Chen et al., 2013) (León-Aldaco et al., 2015)).

In this sense, the present work proposes the application of a Gaussian Adaptive PID to a power electronic converter in order to enhance its performance (Kaster et al., 2015). The Gaussian function used to define the adaptive gains has some desired characteristics: it is smooth with smooth derivatives and lower and upper bounded. A smooth function helps avoiding problems related to abrupt changes in the gains. It can be shown that for nonsmooth adaptive functions and depending on the specific design of the controller, the "equilibrium" state of the output can lead to chattering around the treshold point of the nonsmooth function. This can affect the circuit operation and reduce components lifetime (Kaster et al., 2011). A partially similar approach was used by (Isayed, 2007) in position control of a hard disk drive servosystem, where a hyperbolic secant function was used to adapt the proportional gain and a double gaussian-like function for the adaptive integral gain.

There are still no algebraic methodology to design such controller. To overcome this problem, metaheuristics may be employed in order to optimize the Gaussian function parameters. In this work, was used the particle swarm optimization method (PSO), a bio-inspired methodology based on the collective movement of a group of particles. The PSO algorithm is one of the most frequently candidates to solve optimization and search problems, having many examples of its application in the literature (Castro, 2009; Kennedy and Eberhart, 2001). It is important to remark that in previous experiments the genetic algorithm (Castro, 2009) was evaluated achieved good results, but PSO solve the task demanding considerably less computational effort.

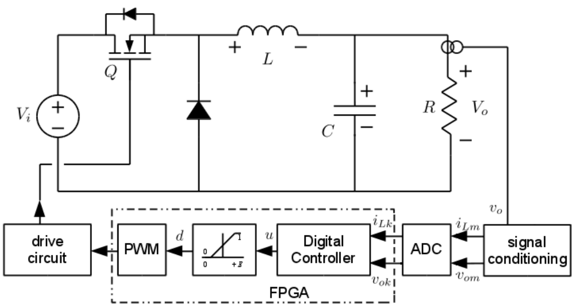

Figure 1: Schematics of the converter and the control system.

2 Application circuit

The step-down converter with its control circuit is presented in Figure 1.

The converter can be described by the state-space equations

$$\begin{aligned} L\frac{di_L}{dt} &= -v_c + uV_i \\ C\frac{dv_c}{dt} &= i_L - \frac{1}{R}v_c \end{aligned} \tag{1}$$

where L is the converter inductor, C is the capac-

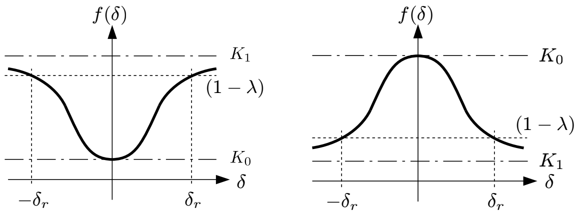

Figure 2: Gaussian function: upward (left) and downward (right).

itor, R is the load, V_i is the supply voltage, i_L and v_c are the state variables for inductor current and capacitor (output) voltage, and u is the control input. This system presents the following control input-to-voltage output transfer function

$$\frac{v_c(s)}{u(s)} = G_v(s) = \frac{V_i}{LCs^2 + \frac{L}{R}s + 1} \tag{2}$$

which is discretized using ZOH method as

$$G_v(z) = \frac{K}{ab}\left[1 + \frac{z-1}{a-b}\left(\frac{b}{z-e^{-aT}} - \frac{a}{z-e^{-bT}}\right)\right] \tag{3}$$

where a and b are the roots of the characteristic polynomial of eq. (2) and $K = V_i/LC$.

3 Gaussian Adaptive PID

In this work, a nonlinear adaptive PID controller is used, where the adaptive rule is based on a Gaussian function defined by

$$f(\delta) = k_1 - (k_1 - k_0)\mathrm{e}^{-q\delta^2} \tag{4}$$

where k_0 and k_1 are the upper and lower bounds of the function and q defines the degree of concavity.

Depending on which k_0 or k_1 is greater defines the shape of the function, as seen in figure 2.

The definition of parameter q is not straightforward. Indeed, the very notion about scaling and the range of reasonable values and its implications are difficult to stablish. A more understandable form comes by defining its value as a function of the percentage λ of the range from k_1 to k_0 and the expected reference error δ_r in that point, given by

$$q = -\frac{\ln\left(\frac{f(\delta)-k_1}{k_0-k_1}\right)}{\delta_r^2} = -\frac{\ln(1-\lambda)}{\delta_r^2}. \tag{5}$$

For example, one can choose that for a reference of 20 V, a reference error $\delta_r = 18\,\mathrm{V}$ corresponds in the gaussian function to 10% of the range from k_1 to k_0 ($\lambda = 0.1$), yielding $(1-\lambda) = 0.9$.

As mentioned before, the Gaussian function represents a smooth function with smooth derivatives and is upper and lower bounded. Being smooth, it avoids abrupt changes in the gains that can lead to problems in some systems, like the occurrence of chattering due to fast repetitive changes in the gains. Being bounded, the designer can establish clear limits for the gains in the whole error range.

An important assumption is to define the value of k_{0d} as zero. This constant represents the derivative gain when the set-point is achieved, i.e., when the error is null. In this situation, the controller presents only Proportional-Integral actions helping avoiding noise amplification issues.

The Gaussian function defined by eq (4) is used in all the three components of GAPID. As the function has three parameters, k_0, k_1 and q, and considering the assumption that $k_{0d} = 0$, the overall design problem consists of eight parameters to be defined. There is no methodology to solve the problem algebraically, so an optimization algorithm based on metaheuristics is employed in order to find an optimized solution to the problem, which allows to achieve the faster response with low overshoot.

There are two paths for the solution: the first considers all eight parameters free; the second considers the parameters linked to linear PID parameters following the rules defined in (6).

$$\begin{aligned} k_{0p} &= x \cdot k_p \quad , \quad k_{1p} = {}^1\!/\!_x \cdot k_p \\ k_{0i} &= y \cdot k_i \quad , \quad k_{1i} = {}^1\!/\!_y \cdot k_i \\ k_{1d} &= z \cdot k_d \end{aligned} \tag{6}$$

When using free parameters for optimization, it is possible extract the maximum performance of the control system. However, experimental results have shown that such system perform very well only for the designed circuit parameters, lacking necessary robustness to perform reasonably with different loads or input voltages.

The linked parameters case presents some advantages, like better robustness and being referenced to a PID design (user can easily design by employing traditional design procedures).

In this sense, the linked parameters case was choosen in this work. Therefore, the parameter set used in the optimization process comprises six elements: (x, y, z, δ_p, δ_i and δ_d), where δ_p, δ_i and δ_d refers to the previously defined reference error constant δ_r of each of the three gaussian functions. These parameters will be obtained by employing the PSO algorithm, which will be described in the next section.

4 PSO Optimization strategy

Particle swarm optimization (PSO) is a bio-inspired optimization methodology based on the collective movement of a group of particles, which forms a swarm. The PSO has exploration and exploitation capabilities which means it presents the conditions to explore the search space and, at the same time, explore the neighborhood of the solutions in order to refine them. The main important behavior present in the algorithm is that each particle travels around the search space according to two forces: the first attracts a single particle with random magnitude to the best location found by itself (*pbest*) and the second force attracts to the best location found in its neighborhood (*gbest*) (Kennedy and Eberhart, 2001). The speed and dynamic position of the particles are updated at each iteration until the swarm converge, according to eqs. (7) and (8) respectively (Castro, 2009):

$$v_p^{(it+1)} = wv_p^{(it)} + c_1 \cdot rand_1^{(it)} \left[pbest_p - x_p^{(it)}\right] + + c_2 \cdot rand_2^{(it)} \left[gbest_p - x_p^{(it)}\right] \quad (7)$$

$$x_p^{(it+1)} = x_p^{(it)} + v_p^{(it+1)}. \quad (8)$$

where:

w is the inertia weight factor;
c_1,c_2 are acceleration constants;
$pbest_p$ is the personal best;
$gbest_p$ is the global best;
$rand_n$ is a random number between 0 and 1;
$x_p^{(it+1)}$ is the current position of particle.

As can be seen, a single particle moves in the search-space, being its speed and its position dynamically updated according to its self-experience and collective experience. When better positions are discovered, they guide the movements of the swarm (do Nascimento et al., 2013). Thus, the movement evolution will flow until achieve the target, or, the convergence (Castro, 2009).

In optimization tasks it is necessary to define a cost function in order to promote a simulation procedure, in which the free parameters are chosen. In this work some methods were tested: setling-time by enveloping functions, Integral Absolute Error (IAE), Integral Square Error (ISE) and Integral Time Square Error (ITSE). The IAE function, presented in eq. (9), achieved the better results — small settling time and overshoot — and was selected. Its mathematical expression is given by:

$$IAE = \int_0^\infty |\delta(t)| dt \quad (9)$$

where $\delta(t)$ is the function of the error.

However, the computational intelligence field usually defines other function called fitness which has to be always increased. The biological inspiration in this area is that particles with high fitness find better positions in the search space during the collective movement. So, the fitness function is given by

$$fit = 1/cost. \quad (10)$$

Table 1: PSO parameters

Description	Value
Population size	20
Interactions	30
Initial inertia weight factor	0.9
End inertia weight factor	0.4
Acceleration constant ($c1$ and $c2$)	2.0

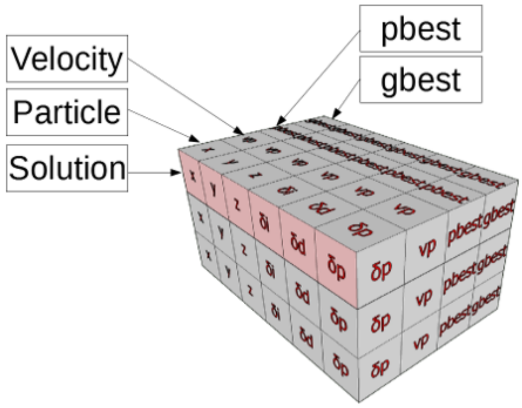

Figure 3: Organization of the swarm of particles

Therefore, the cost function defined in eq. (9) must be minimized, it means that we have to maximize the fitness defined in eq. (10).

As discussed, the choice of parameters for the PSO has a great impact on the optimization performance. Based on previous tests and using the above definitions, the selected parameters are shown in table 1.

The simplest structure for this application is a three-dimensional array, as shown in Figure 3, where every solution have six search spaces (x, y, z, δ_i, δ_d, δ_p), each search space has a set of particles and each particle has its own characteristics (velocity, *pbest* and *gbest*).

The basic PSO easily converges to a local minimum but this premature convergence is not interesting and may result in a vicious swarm. To circumvent this difficulty, random particles were spread along the search space in such a way to garantee some distance between them (they must not be concentrated).

In the PSO discussed in this work, the self-experience of each of the n-th solution (x_n, y_n, z_n, $\delta_{i,n}$, $\delta_{d,n}$, $\delta_{p,n}$) was used to achieve the collective solution for the parameters of the GAPID.

The evolution of the particles in each search space is presented in figure 4. It is easy to perceive the convergence of the swarm towards a common value after some iterations in all search spaces (more than 80% of the particles converge in up to 15 iterations).

The temporal evolution of the best, worst and average fitness of the swarm until reaching the optimal is shown in figure 5.

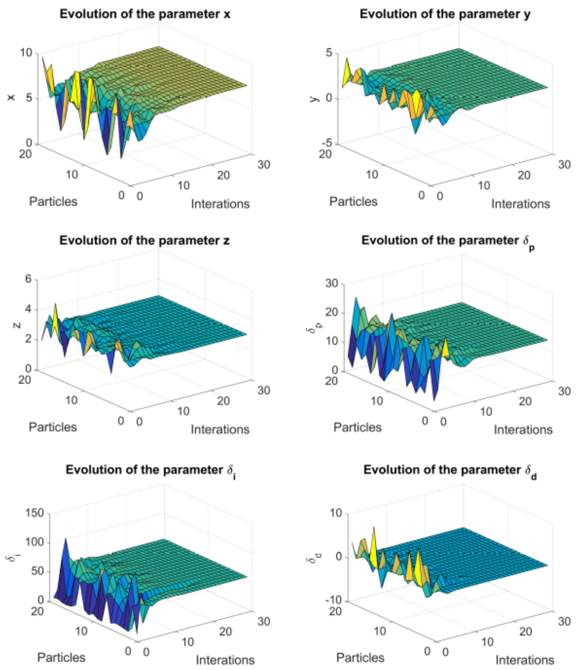

Figure 4: Evolution of the parameters of the PSO optimization.

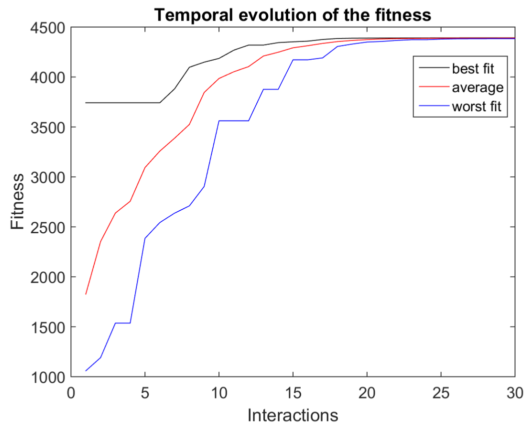

Figure 5: PSO temporal evolution

5 Results

The prototype of the Buck converter that has been built for experimentation is shown in figure 6. The specifications of the converter are shown in table 2.

The controller was implemented in an Altera Terasic DE0-nano development kit that runs at 50 MHz with eight 12-bit ADC channels and 5 μs sample time. This converter delivers a 20 V output from a 50 V source.

For programming and simulation, the Altera DSP Builder tool has been employed. This tool comprises Matlab/Simulink blocks that serve for simulation and for code generation. The system model in Simulink using DSP Builder blocks is shown in Figure 7. After validating the operation of the entire system by means of simulations, the same model can be compiled and generates the proper code to be embeded into the FPGA.

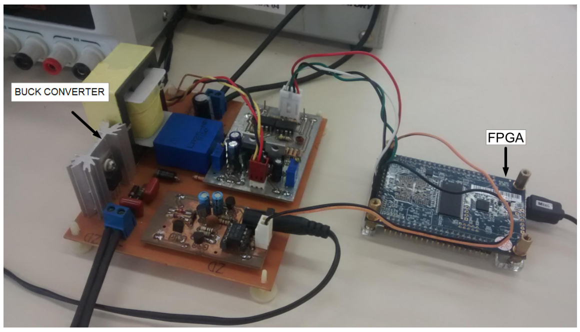

Figure 6: Buck converter

Table 2: Buck converter parameters

Description	Symbol	Value
Capacitor	C	100 μF
Max. capacitor voltage ripple	$\Delta V_c\%$	10%
Capacitor resistance	R_C	0,2 Ω
Inductor	L	2,54 mH
Max. inductor current ripple	$\Delta I_L\%$	10%
Inductor resistance	R_L	0,81 Ω
Switch on-resistance	R_{on}	0,55 Ω
Forward voltage	V_d	1 V
Frequency	f	50 kHz
Input voltage	V_i	50 V
Output voltage	V_O	20 V
Load	R	10 Ω

The results obtained by simulations are presented next.

The designed PID control parameters are presented in Table 3 (left). Based on these parameters, the PSO optimization was conducted resulting in the optimized parameters for the GAPID control, presented in Table 3 (right).

As previously stated, the linked parameters case were employed in order to comply with the same design requirements of the linear PID.

In figure 8 the output responses of linear PID

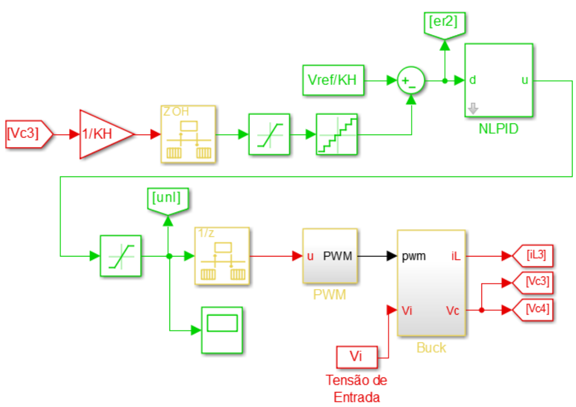

Figure 7: Model of Buck controller using DSP Builder in Simulink.

Table 3: Control parameters

PID param.	Value
k_p	6.5E-3
k_i	21.9
k_d	6.5E-6

GAPID Param.	Value
x	7.37
y	2.16
z	2.89
δ_p	13.4
δ_i	55.2
δ_d	6.12E-4

Table 4: Performance comparison.

Parameter?	PID	GAPID	Enhancement
T_s	1.82 ms	0.983 ms	46%
$M(\%)$	2.71%	1.85%	31.7%

and GAPID are presented, for a rated load of 10 Ω. A significant performance enhancement of 46% over the linear PID was obtained, and with 31.7% lower overshoot, which are summarized in table 4. The most important point to be noted here is that this performance is achieved by GAPID with parameters linked to the PID parameters. Other PID designs can be done and then the optimization process used to obtain the linked parameters of the GAPID and, again, performance enhancements can be obtained.

In order to achieve such enhancements, the adaptive gains takes action during the startup transient, as shown in the first three waveforms of figure 9, and these gains induce the control action shown in the fourth waveform, which is responsible for the GAPID output.

Experiments using a Buck prototype were carried out with the view to verify the behavior of the controller in a real-world system. The DSP Builder model shown in figure 7 was compiled and embedded into the FPGA.

The inductor current and output voltage waveforms for the linear PID are presented in figure 10. The measured setling-time is 1.82 ms, about the same obtained by simulation.

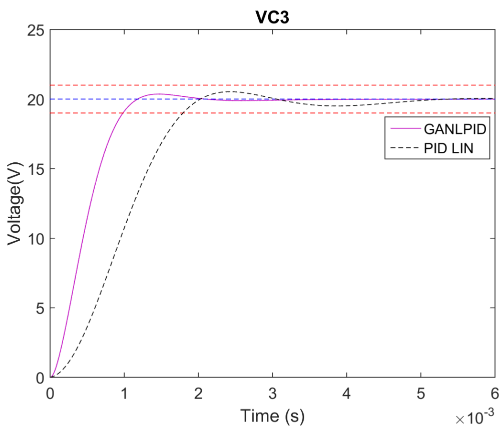

Figure 8: Transient responses for the linear PID and GANPID.

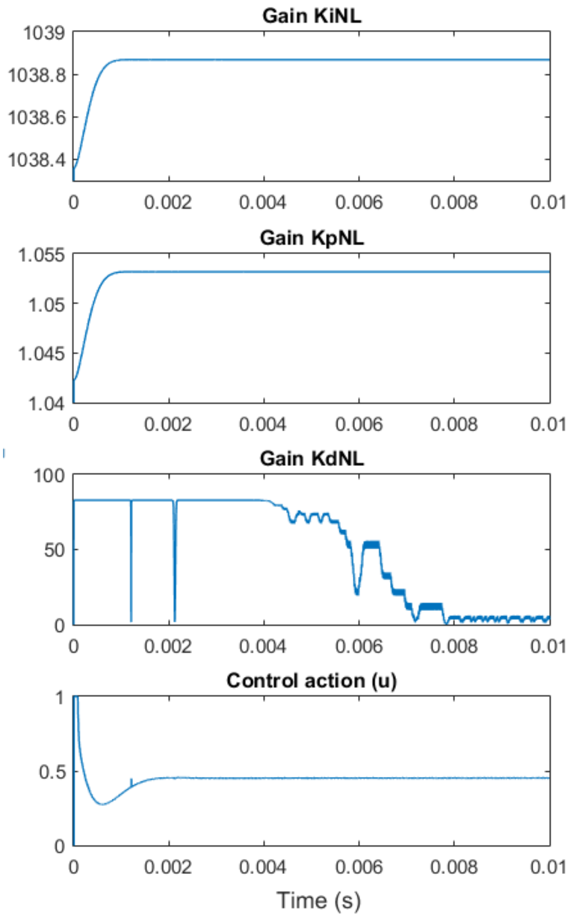

Figure 9: Variation of adaptive gains $k_p(\delta)$, $k_i(\delta)$ and $k_d(\delta)$ and resulting control signal during the startup transient.

The waveforms for the GAPID are presented in figure 11. The measured setling-time is 0.936 ms, very close to the value obtainded by simulation.

Unfortunately, the visual comparison is not possible because the waveforms were captured in different horizontal time ranges.

The experimental results show that the model built for simulation is very accurate. As a result, the obtained results are very similar and demonstrate the effectiveness of the GAPID in the Buck converter.

6 Conclusions

This paper presented the proposal of a PID-based nonlinear adaptive control strategy based on Gaussian functions. The GAPID parameters linked to the PID parameters case was choosen because it presented better results related to robustness and a trend curve for load variations less "randomic" when compared to the free parameters case. Also, four alternatives for cost functions were tested and the results were favorable to IAE.

The PSO algorithm were employed in order to optimize the GAPID parameters.

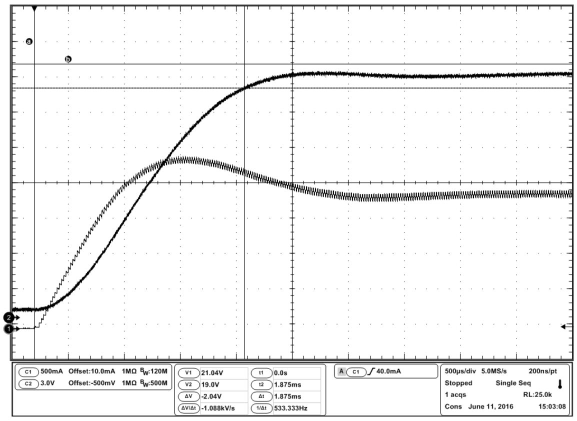

Figure 10: Output waveforms for the linear PID in the prototype (Ch.1: 500mA/div; Ch.2: 3V/div, Time: 500μs/div).

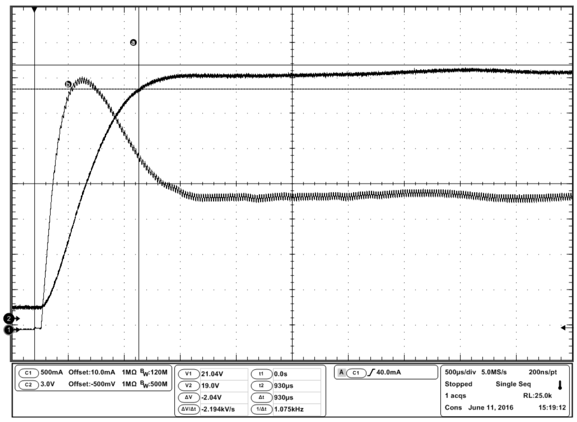

Figure 11: Output waveforms for the GAPID in the prototype (Ch.1: 500mA/div; Ch.2: 3V/div, Time: 500μs/div).

The work presented the simulation and experimental results using the optimized parameters. The results obtained by GAPID showed significant enhancements to PID (46% shorter setling-time and 31.7% less overshoot). The results obtained both in simulation and in experimentation demonstrate the efficiency of the new control strategy.

As future works, the trend curves of converter parameters variation can be performed in order to suggest a design methodology using the GAPID in Buck-like converters as well as explore the application to other converter topologies.

References

Castro, L. N. (2009). *Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications*, Chapman & Hall/CRC.

Ceglia, G., Grau, V., Guzman, V., Sanchez, C., Ibanez, F., Walter, J., Millan, A. and Gimenez, M. I. (2004). A new multilevel inverter topology, *Devices, Circuits and Systems, 2004. Proceedings of the Fifth IEEE International Caracas Conference on*, Vol. 1, pp. 212–218.

Chen, W. C., Wang, C. S., Su, Y. P., Lee, Y. H., Lin, C. C., Chen, K. H. and Du, M. J. (2013). Reduction of equivalent series inductor effect in delay-ripple reshaped constant on-time control for buck converter with multilayer ceramic capacitors, *IEEE Transactions on Power Electronics* **28**(5): 2366–2376.

Costa, L. F., Mussa, S. A. and Barbi, I. (2013). Multilevel boost dc-dc converter derived from basic double-boost converter, *Power Electronics and Applications (EPE), 2013 15th European Conference on*, pp. 1–10.

do Nascimento, F. A. F., Arce, J. E., Dias, A. N., Filho, A. F., de Magalhães Miranda, G. and Carnieri, C. (2013). Meta-heurística otimização por enxame de partículas aplicada ao planejamento florestal, *in* H. S. Lopes, L. C. de Abreu Rodrigues and M. T. A. Steiner (eds), *Meta-Heurísticas em Pesquisa Operacional*, 1 edn, Omnipax, Curitiba, PR, chapter 22, pp. 355–366.

Hwu, K. I. and Yau, Y. T. (2010). Performance enhancement of boost converter based on pid controller plus linear-to-nonlinear translator, *IEEE Transactions on Power Electronics* **25**(5): 1351–1361.

Isayed, M. B.; Hawwa, M. A. (2007). A nonlinear pid control scheme for hard disk drive servosystems, *Proceedings of the Mediterranean Conference on Control and Automation*, Athens, Greece.

Kaster, M. S., Cunha, G. D., Burgardt, I. and Nascimento, C. B. (2011). Análise do uso de controle pid não linear aplicado a conversor buck, *Proceedings of the X Conferência Brasileira de Dinâmica, Controle e Aplicações*.

Kaster, M. S., Lucas, R., Oliveira, R. M. and Nascimento, C. B. (2015). Performance analysis of an adaptive gaussian nonlinear pid control applied to a step-down cc-cc converter, *Proceedings of the 24th International Symposium on Industrial Electronics (ISIE 2015)*, p. 1022.

Kennedy, J. and Eberhart, R. C. (2001). *Swarm Intelligence*, Morgan Kaufmann.

León-Aldaco, S. E. D., Calleja, H. and Alquicira, J. A. (2015). Metaheuristic optimization methods applied to power converters: A review, *IEEE Transactions on Power Electronics* **30**(12): 6791–6803.

Romaneli, E. F. R. and Barbi, I. (2000). A new dc-dc converter with low current ripple characteristics, *Telecommunications Energy Conference, 2000. INTELEC. Twenty-second International*, pp. 560–566.

Wang, W. and m. Bai, Z. (2010). Performance analysis of an improved single neuron adaptive pid control, *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*, pp. 22–25.