Multi-Objective PI Controller Design with an Application to Speed Control of Permanent Magnet DC Motor Drives

C.Agees Kumar
Department of Electronics& Instrumentation Engineering
Noorul Islam College of Engineering
Kumaracoil, TamilNadu, India
sakthi_cak13@hotmail.com

N.Kesavan Nair
Department of Electrical & Electronics Engineering
Noorul Islam College of Engineering
Kumaracoil, TamilNadu, India
kesavannaireee@hotmail.com

Abstract—**This paper proposes a Multi-Objective Optimization (EMOO) algorithm is used to tune the Proportional Integral (PI) speed regulator in the Permanent Magnet DC (PMDC) motor drive system It aims at achieving good robustness and struck at global optima. Calculus based methods including gradient approaches mainly search for local optimum solutions, rather than global optimum. Due to the drawbacks that exist in calculus based methods, the proposed method is employed for multiobjective optimization problems. The robustness features of the proposed design approach are verified using multi-objective evolutionary Non-dominated Sorting Genetic Algorithm (NSGA-II) and Non-Dominated Sorting Particle Swarm Optimization (NSPSO). Conflicting objectives considered are rise time and settling time for multi-objective optimization. Speed control of motor of Permanent Magnet DC Motor is achieved with PI speed regulator tuned using NSGA-II and NSPSO. Finally, the Pareto Optimal solutions obtained show the effectiveness of NSPSO in comparison with NSGA-II in design of PID controllers for a highly non-linear process.**

Keywords- PI Controller; Multi-objective Optimization; NSGA-II; NSPSO;Pareto Optimal front; Speed control.

I. INTRODUCTION

Optimization in engineering design has always been of great importance and interest particularly in solving complex real-world design problems. Basically, the optimization process is defined as finding a set of values for a vector of design variables so that it leads to an optimum value of an objective or cost function. In such single-objective optimization problems, there may or may not exist some constraint functions on the design variables and they are respectively referred to as constrained or unconstrained optimization problems. There are many calculus-based methods including gradient approaches to search for mostly local optimum solutions. However, some basic difficulties in the gradient methods such as their strong dependence on the initial guess can cause them to find a local optimum rather than a global one. This has led to other heuristic optimization methods, particularly Genetic Algorithms being used extensively during the last decade. Such nature-inspired evolutionary algorithms differ from other traditional calculus based techniques. The main difference is that GA's work with a population of candidate solutions, not a single point in search space. This helps significantly to avoid being trapped in local optima as long as the diversity of the population is well preserved.

In multi-objective optimization problems, there are several objective or cost functions (a vector of objectives) to be optimized (minimized or maximized) simultaneously. These objectives often conflict with each other so that as one objective function improves, another deteriorates. Therefore, there is no single optimal solution that is best with respect to all the objective functions. Instead, there is a set of optimal solutions, well known as Pareto optimal solutions, which distinguishes significantly the inherent natures between single-objective and multi-objective optimization problems. V. Pareto was the French-Italian economist who first developed the concept of multi-objective optimization in economics. The concept of a Pareto front in the space of objective functions in multi-objective optimization problems (MOPs) stands for a set of solutions that are non-dominated to each other but are superior to the rest of solutions in the search space. Evidently, changing the vector of design variables in such a Pareto optimal solutions consisting of these non-dominated solutions would not lead to the improvement of all objectives simultaneously. Consequently, such change leads to a deterioration of at least one objective to an inferior one. Thus, each solution of the Pareto set includes at least one objective inferior to that of another solution in that Pareto set, although both are superior to others in the rest of search space.

The inherent parallelism in evolutionary algorithms makes them suitably eligible for solving MOPs. The early use of evolutionary search is first reported in 1960s by Rosenberg. Since then, there has been a growing interest in devising different evolutionary algorithms for MOPs. Basically, most of them are Pareto-based approaches and use the well-known non-dominated sorting procedure. Basically, both NSGA-II and NSPSO as Pareto-based approaches use the revolutionary non-dominated sorting procedure originally proposed by Goldberg.

Brush DC motor, AC induction motor and the brushless permanent magnet synchronous motor are used in today's industrial and commercial speed control systems. Permanent magnet (PM) motors are probably the most commonly used DC motors, which operate from a direct current power source. The brush DC motor is composed of two major components, the moving armature and the stationary field. Today, the overwhelming majority of smaller DC motors use permanent magnet fields. A simple, permanent-magnet dc motor is an essential element in a variety of products, such as toys, servomechanisms, valve actuators, robots, and automotive electronics. Furthermore, since the motor's field, created by the permanent magnet, is constant, the relationship between torque and speed is very linear. A Permanent Magnet motor can provide relatively high torque at low speeds and permanent magnet field provides some inherent self-braking when power to the motor is shutoff.

Due to decoupled nature of field and armature mmf's DC motors exhibit outstanding drive performance characteristics. By [1] using variable battery tapping's, variable supply voltage, resistors or electronic controls we can achieve speed control. The simplest and oldest speed control is variable speed control system. In negative resistance adding resistance causes more speed variation, then a series negative resistance should cause less speed variation to attain constant speed. In series resistance control adding a resistor in series with the motor can control speed – higher resistance will cause speed to drop and it is counter-productive for constant speed. Brushed DC motors, within certain constraints, can be controlled either with a linear or a switching amplifier. Four-quadrant thyristor converters are widely used in constant speed drives that use DC brushed motors. Linear amplifiers are used to control the speed of small, DC brushed motors. The operation of a PWM amplifier [2] in the generation of the switching waveform can also be used for achieving the speed control. Unlike SCR controls that switch at line frequency, PWM controls produce smoother current at higher switching frequencies.

From the several meta heuristics currently available, evolutionary algorithms that are based on the emulation of the mechanism of natural selection are among the most popular and they enable tradeoffs among conflicting optimization goals to be explored. While designing a control system we should consider a number of performance issues, such as system stability, the static and dynamic index, and system robustness. Now the creative combination of a variety of pre-existing control methodologies and Genetic algorithms have resulted in powerful tools such as [5] Multiobjective control, PID control, optimal control, robust control [7], intelligent control and adaptive control [3, 4].

The online tuning Genetic Algorithm tuning [6] of PI controller have been used in recent years. This paper deals with implementation of speed control of permanent magnet DC motor drive system with a pareto optimal proportional integral [10] speed regulator tuned using multi objective evolutionary [11] Non-Dominated Sorting Genetic Algorithm (NSGA-II). The results of PI controller tuning is obtained by the minimization of the two objectives functions which are controller performance parameters evaluated over a finite time needed to reach the steady state of motor speed:

Rise time

Settling time

The rest of this paper is organized as follows. Evolutionary Multiobjective algorithm NSGA-II is introduced in Section III. Controller tuning design problem is described in Section IV. Here, specification of the test rig, NSGA-II parameters and performance measures for the PI controller are stated. The tuning results are presented in Section V. Finally, the conclusions are given in Section VI.

II. MATHEMATICAL MODELING OF PMDC

The equivalent circuit of dc motor is shown in Fig 1.The torque, T is related to the armature current, i by a constant factor $K_t T = K_t I_a$. For the separately excited DC motor, the back emf, e is related to the rotational velocity by: $e = K_b u_T$

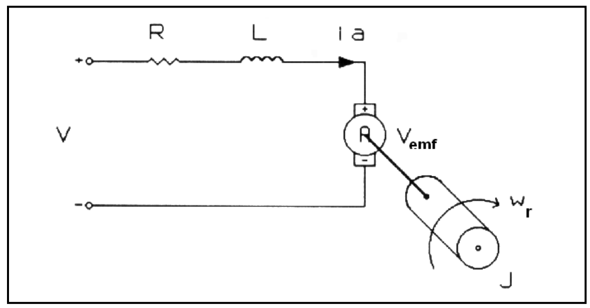

Figure 1. Equivalent representation of DC motor

In SI units K_t (armature constant) is equal to K_b (motor constant). From Fig. 2

V = input voltage (V)
R = nominal resistance (Ω)
L = nominal inductance (H)
J = Inertial load (kgm2/s2)
V_{emf} = back emf voltage (V)
b = damping constant (Nm.S)
τ = motor output torque (Nm)
θ = motor shaft angle (rad).

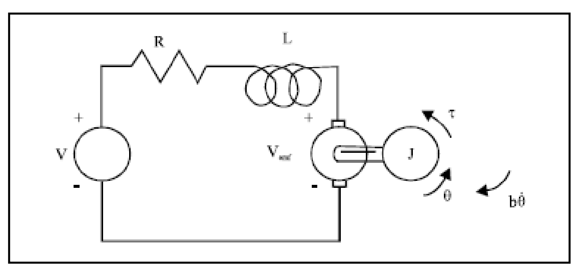

Figure 2. Electro-Mechanical Model of a DC motor

The DC motor equations based on Newton's law combined with Kirchoff's law:

$$J\frac{d\omega}{\partial t} + B\omega = K_t i_s - T_L \quad (1)$$

$$L_s \frac{di_s}{\partial t} + R_a i_a = V_a - K_b \omega_T \quad (2)$$

Where:

J = J -The moment of inertia

B = ω_T -Damping ratio of the mechanical system

R = R_a-The electrical resistance of the armature circuit

L = L_a -The electrical inductance of the armature circuit

In the state-space form, the equations above can be expressed by choosing the rotational speed and electric current as the state variables and the voltage as an input. The output is chosen to be the rotational speed.

$$\frac{d}{dt}\begin{bmatrix}\omega_T \\ i_a\end{bmatrix} = \begin{pmatrix} \frac{-B}{J} & \frac{K}{J} \\ \frac{-K}{L_a} & \frac{-R_a}{L_a} \end{pmatrix}\begin{pmatrix}\omega_T \\ i_a\end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{L_a}\end{pmatrix} V_a \quad (3)$$

$$\omega_T = \quad (1 \;\; 0)\begin{pmatrix}\omega_T \\ i_a\end{pmatrix} \quad (4)$$

III. Multi-objective Non Dominated Sorting Genetic Algorithm-II (NSGA-II) & Non Dominated Sorting Particle Swarm Optimization (NSPSO)

NSGA (Non-Dominated Sorting in Genetic Algorithms) is a popular non- domination based genetic algorithm for multi-objective optimization. It is a very effective algorithm but has been generally criticized for its computational complexity, lack of elitism and for choosing the optimal parameter value. A modified version, NSGA-II was developed, which has a better sorting algorithm, incorporates elitism and no sharing parameter needs to be chosen a priori approach. NSPSO combines the strengths of the these advanced operations (A fast non-dominated sorting approach, crowding distance ranking, elitist strategy, mutation and selection operations) with single-objective PSO search.

A. The main algorithm of NSGA-II

Step 1 : A random population is initialized.
Step 2 : Objective functions for all objectives and constraint are evaluated.
Step 3 : Front ranking of the population is done based on the dominance criteria.
Step 4: Crowding distance is calculated.
Step 5: Selection is performed using crowded binary tournament selection operator.
Step 6: Crossover and mutation operators are applied to generate an offspring population.
Step 7 : Parent and offspring populations are combined and a non-dominated sorting is performed.
Step 8 : The best members of the combined population replace the parent population**.**

B. The main algorithm of NSPSO

Step 1 : Initialize a random initial population.
Step 2 : Sort the population based on non- domination and crowding distance ranking
Step 3 : Front ranking of the population is done based on the dominance criteria.
Step 4: Assign each individual a fitness (or rank) equal to its non-domination level
Step 5: Selection is performed using crowded binary tournament selection operator.
Step 6: Randomly choose one individual as *gbest* for N times from the nondominated solutions, and modify each searching point.
Step 7 : Do mutation operator
Step 8 : Combine the offspring and parent population
Step 9 : Sort the extended population based on non-domination and fill the new population of size N with individuals from the sorting fronts starting to the best.
Step 10: Modify the $pbest_i$ of each searching point

C. Objective Functions used

1. Settling Time minimization

$$f_2(K_I, K_P, K_D) = \max\left(\frac{1}{1+T_N}\right) \quad (05)$$

2. Rise Time minimization:

$$f_3(K_I, K_P, K_D) = \max\left(\frac{1}{1+T_R}\right) \quad (06)$$

Where, T_N denotes Settling Time and T_R denote Rise time. These objectives are being simultaneously optimized and results are obtained.

IV. Simulation And Results

Conventional methods of tuning of controllers are proposed by Zeigler and Nichol's.This method can be employed only for lower order linear systems.Only a single value of $K_{p,}$ K_i and K_d can be obtained for a particular process.Simulations are carried out using MATLAB 8.0.

In case of Multiobjective Optimization, from the Pareto front, different values of $K_{p,}$ K_i and K_d can be obtained for a particular process based on different objectives.

Solutions from different points of Pareto front causes different results from the aspect of overshoot/undershoot, settling time and rise time. So one can select a single solution based on system conditions and requirements.

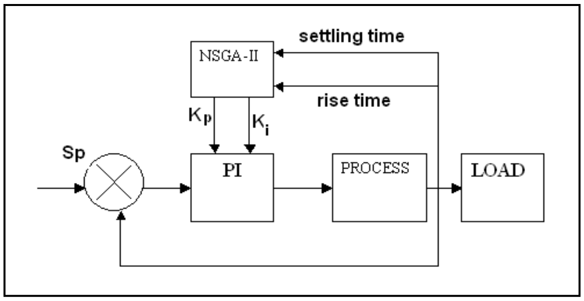

Figure 3. Functional Block Diagram of Speed control using NSGA-II

The optimal tuning of the PMDC speed regulator using evolutionary multi-objective algorithm was performed successfully. A Simulated Binary Crossover (SBX) with the crossover probability of 0.9 ensures a good mixing of genetic material, while the polynomial mutation is used in mutation probability of 0.33 guarantees that on average one parameter of each individual will be mutated in Table 1.

TABLE 1. Basic parameters of the NSGA-II algorithm.

ALGORITHM PARAMETER	VALUE
Population, N	60
Generations, G	50
Pool size, N/2	30
Tour size	2
Crossover probability	0.9
Mutation probability	0.33

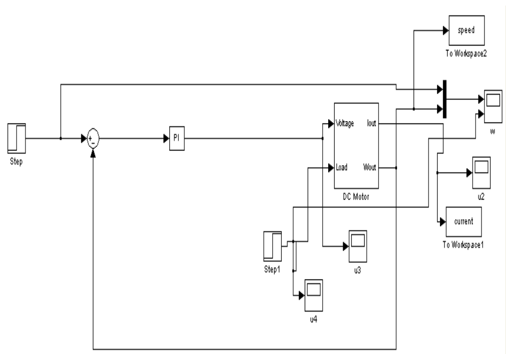

Figure 4. Multiobjective optimal speed control of PMDC is implemented in MATLAB/ Simulink.

Fig.4 shows Multiobjective optimal speed control of PMDC implemented in MATLAB 7.10 / Simulink Environment with 2GB RAM, 300 GB Hard Disk, Intel Core2 Duo Processor.

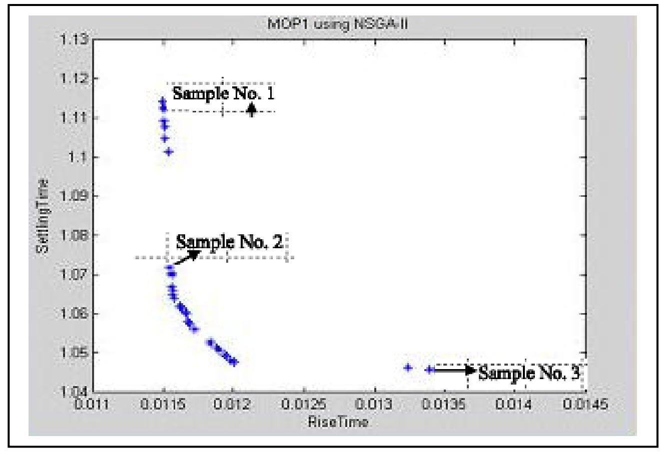

Figure 5. Pareto front 1 with settling time and Rise Time as Objectives with NSGA-II

Pareto optimal front with settling time and Rise Time as Objective functions is shown in Fig.2. From the Pareto Optimal front, three samples are taken. From these samples, corresponding K_p, K_i and K_d values are obtained as per the requirements of the user.

When the system needs low rise time, one can choose sample1, or needs low settling time, sample 3 will be fine.If a compromised solution of acceptable overshoot and settling time is required, sample 2 will be fine.Once sample is selected, appropriate $K_{p,}$ K_i and K_d values are obtained from Pareto Optimal front.

TABLE II. PID GAINS FOR THREE SAMPLES IN PARETO FRONT WITH NSGA-II

Sample	K_p	K_i	K_d
1	1.9913	0.1582	6.3874
2	5.0109	0.0991	2.8990
3	3.9932	0.1243	1.9765

The response of Proportional Integral (PI) controller is shown in Fig. 6 for a set point speed of 100 rpm with PI parameters (K_p= 5.0109 and K_i = 0.0991).It is inferred that the actual set point of 100 rpm is achieved with settling time of 400 sec before reaching the set point.

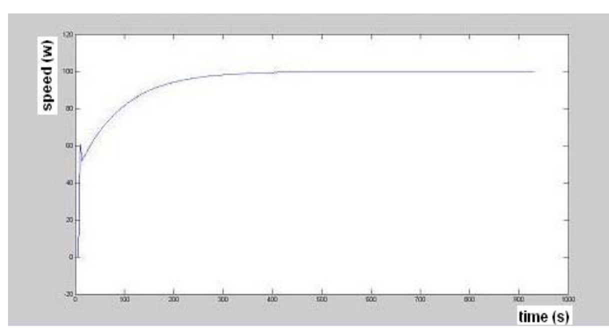

Figure 6. PI controller response (Time Vs Speed) for Sample 2 with NSGA-II

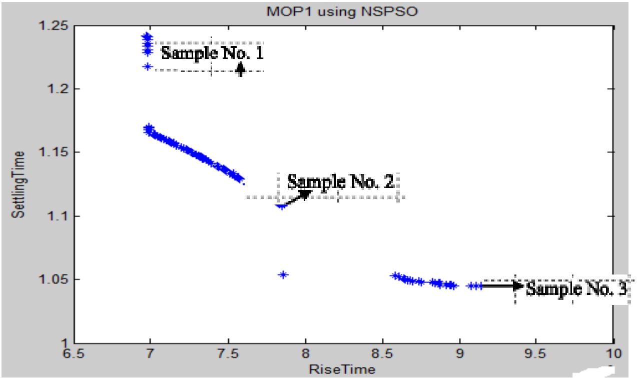

Figure 7 Pareto front with settling time and Rise Time as Objectives with NSPSO

Pareto optimal front with settling time and Rise Time as Objective functions is shown in Fig.7. From the Pareto Optimal front, three samples are taken. From these samples, corresponding K_p, K_i and K_d values are obtained as per the requirements of the user.

When the system needs low rise time, one can choose sample1, or needs low settling time, sample 3 will be fine.If a compromised solution of acceptable overshoot and settling time is required, sample 2 will be fine.Once sample is selected, appropriate $K_{p,}$ K_i and K_d values are obtained from Pareto Optimal front.

TABLE III. PID GAINS FOR THREE SAMPLES IN PARETO FRONT WITH NSPSO

Sample	K_p	K_i	K_d
1	4.3916	0.0965	3.7644
2	5.7145	0.3345	2.1243
3	3.6543	0.2543	1.8543

The response of Proportional Integral (PI) controller is shown in Fig. 8 for a set point speed of 100 rpm with PI parameters (K_p= 5.7145 and K_i = 0.0991).It is inferred that the actual set point of 100 rpm is achieved with settling time of 250 sec before reaching the set point.

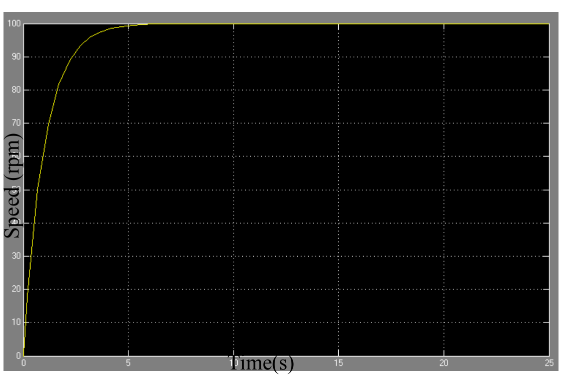

Figure 8. PI controller response (Time Vs Speed) for Sample 2 with NSPSO

V. CONCLUSION

Motor criteria such as durability, high performance, efficiency, easy and cheap control, low maintenance demands have led to a new type of speed control of DC motor excited by permanent magnets. In this work, a mathematical model for the Permanent Magnet DC Motor is developed and implemented in MATLAB/Simulink.

Speed control using Evolutionary Algorithms NSGA-II & NSPSO is explained and is simulated in MATLAB/Simulink environment. The Simulation results are analysed and found that the PI speed controller tuning using evolutionary algorithms allows us to find several members of the Pareto optimal set in a single run. The optimal tuning of the PMDC speed regulator using evolutionary multi-objective algorithm was performed successfully.

The main objective functions to be minimized are rise time and Settling time. The optimization solution results are a set of near optimal trade-off values which are called the Pareto front or optimality surfaces. Pareto front enables the operator to choose the best compromise or near optimal solution that reflects a trade-off between key objectives. In this paper more values of PID controller parameters ($K_{p,}$ K_i and K_d) can be obtained from a single Pareto Optimal front, so the designer has the flexibility to select a single solution based on rise time and settling time. The iterative simulation results show the effectiveness of the NSPSO in comparison with NSGA-II since it allows the operator to find a near optimal good compromise among the proposed goals, which is the best trade-off of low cost PID controller design. For Sample 2, Settling Time in NSPSO is very much reduced when compared to NSGA-II so that the process quickly reaches the set point.

REFERENCES

[1] S. Rajaram and S. Murugesan, "A New Method for Speed Measurement/ Control of DC Motors", IEEE Transactions on Instrumentation and Measurement, Volume 27, Issue 1, March 1978 pp99–102.

[2] H.F. Weber, "Pulse-Width Modulation DC Motor Control", IEEE Transactions on Industrial Electronics and Control Instrumentation, Volume IECI-12, Issue 1, March 1965, pp: 24–28.

[3] Astrom, K.J., "Adaptive feedback control", Proceedings of the IEEE, Volume 75, Issue 2, Feb. 1987, pp:185–217.

[4] P. Chevrel and S. Siala, "Robust DC-Motor Speed Control Without any Mechanical sensor", Proceedings of the 1997 IEEE International Conference on Control Applications, Hartford, CT, October 5–7, 1997, pp244–246.

[5] P. Wozniak, "Preferences in evolutionary multi-objective optimisation of electric motor speed control with hardware in the loop", Institute of Automatic Control, Technical University of Łód´z, 18/22 Stefanowskiego St., 90-369 Łód´z, Poland, Applied Soft computing ,(2009) 49-55.

[6] K. Deb, "Multi-objective Optimization Using Evolutionary Algorithms", Wiley, Chichester, 2001.

[7] M. Ahmad, L. Zhang, J.C. Readle, "On-line genetic algorithm tuning of a PI controller for a heating system", in: Proceedings of the 2nd International Conference on Genetic

Algorithms in Engineering Systems, Glasgow, 1997, pp. 510–515.

[8] P. Stewart, D.A. Stone, P.J. Fleming," Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop", Engineering Applications of Artificial Intelligence 17 (2004) 275–284.

[9] Y. Jin, J. Branke, "Evolutionary optimization in uncertain environments—a survey", IEEE Transactions on Evolutionary Computation 9 (2005) 303–317.

[10] P. Wozniak, "Preferences in evolutionary multi-objective optimisation with noisy fitness functions: hardware in the loop study", in: Proceedings of the 2nd International Symposium on Advanced in Artificial Intelligence, IMCSIT, Wisła, 2007, pp. 337–346.

[11] G.L.O. Serra, C.P. Bottura, "Multiobjective evolution based fuzzy PI controller design for nonlinear systems", Engineering Applications of Artificial Intelligence 19 (2006) 157–167.

[12] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. "A fast and elitist multi-objective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.

[13] J. Branke, K. Deb, "Integrating user preferences into evolutionary multi-objective optimization", in: Y. Jin (Ed.), Knowledge Incorporation in Evolutionary Computation,Springer, Berlin, 2005, pp. 461–477.

[14] P. Wo zniak, T. Sobieraj, "Multi-objective control design by experiment-based evolutionary methods", in: Proceedings of the Conference on Control for Power Electronics and Electrical Drives, SENE'07, Technical University of Łod z, Łod z, 2007, pp. 521–526.

[15] Thierry Castagnet and Jean Nicolai, "Digital Control for Brush DC Motor", IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, vol. 30, no. 4, July/August 1994, pp883–888.

[16] B. Lu, A. Monti, R. Dougal, "Real-time hardware-in-the-loop testing during design of power electronics controls", in: Proceedings of the IEEE IECON, IEEE, 2003, pp. 1840–1845.