LMPM POSITION CONTROLLER PARAMETER OPTIMIZATION USING GENETIC ALGORITHM

Tatiana Radičová, Jakub Vonkomer, Milan Žalman, Ivan Sekaj

Institute of Control and Industrial Informatics,

FEI STU Ilkovičova 3, 812 19 Bratislava, Slovak Republic

Abstract

Genetic algorithms are one of the most well-known evolutionary computing methods using heuristic searches that mimic the process of natural evolution. These methods provide very useful results to optimization and search problems. There are several factors which have particular impact on algorithm to be successful. That means correct population initialization, sufficient simulation time and optimization function – fitness function that plays very important role in description of sought extreme. We used genetic algorithm to design high performance position controller for LMPM drive. The controller should have faster dynamics, smaller position error and improved noise immunity compared to controller designed by Pole placement method. Pole placement method is a standard method of controller design which compares calculated denominator of closed-loop system to desired denominator of equal powers. Easiness of using MATLAB and Genetic algorithm toolbox is demonstrated by solving the controller design problem. Finally, results gathered from simulation in MATLAB-Simulink are presented.

1 Introduction

Nowadays, linear motors types are still more and more preferred thanks to their matchless features. Although in comparison with rotary motors, they are henceforward financially demanding. It can be mentioned that a high speed trains, such as Maglev or Trans-rapid became famous by implementation these kind of motors. Linear motors however occur in various sectors, whether in electrotechnical or electronic production (drives for operating and position engineering, drives into machine tools …).

In this paper we will focus to position servo-drive control design of aircore LMPM in consideration of control performance with using of Genetic Algorithms (GA). GAs are one of the most famous and the most used representatives of evolutionary computing techniques with wide range of application [1]. Control performance possesses highly important function in servo-drives that is why we took advantage of GA to improve the overall performance. Result performance is compared to controller designed by Pole-placement method with lead-compensator described in (Radicova, Zalman) [2].

Main idea of designing controller parameters using GA has been publicly adopted in the 1990's, but remains popular in the present as well which is proven by number of papers in relevant journals [6][7]. Interesting is also attempt of PI position controller design of SMPM drive by Khater and others [5].

The goals of creating artificial intelligence and artificial life can be traced back to the very beginnings of the computer age. The earliest computer scientists - Alan Turing, John von Neumann, Norbert Wiener, and others were motivated in large part by visions of imbuing computer programs with intelligence, with the life-like ability to self-replicate and with the adaptive capability to learn and to control their environments. These early pioneers of computer science were as much interested in biology and psychology as in electronics, and they looked to natural systems as guiding metaphors for how to achieve their visions. It should be no surprise, then, that from the earliest days computers were applied not only to calculating missile trajectories and deciphering military codes, but also to modeling the brain, mimicking human learning and simulating biological evolution. These biologically motivated computing activities have waxed and waned over the years, but since the early 1980s they have all undergone resurgence in the computation research community. The first has grown into the field of neural networks, the second into machine learning, and the third into what is now called "evolutionary computation", of which genetic algorithms are the most prominent example [3].

Idea of evolutionary computing was introduced in the 1960s by I. Rechenberg in his work "Evolution strategies". His idea was then developed by other researchers. GAs were invented by John Holland and developed by him and his students and colleagues. This lead to Holland's book "Adaption in Natural and Artificial Systems" published in 1975. In 1992 John Koza has used genetic algorithm to evolve programs to perform certain tasks. He called his method "genetic programming" (GP). During reproduction, first occurs recombination (or crossover). Genes from parents form in some way the whole new chromosome. The new created offspring can then be mutated. Mutation means, that the elements of DNA are a bit changed. These changes are mainly caused by errors in copying genes from parents [8].

2 Position servo-drive, implementation block scheme

Position servo-drive can be performed by various algorithms. PID algorithm with lead compensator in comparison with genetic algorithm is applied, referring to the article [9]. The main asset is to execute synthesis of PID algorithm with lead compensator and evaluate better method of their parameters. The entire position servo-drive structure may bee seen in the Figure 1.

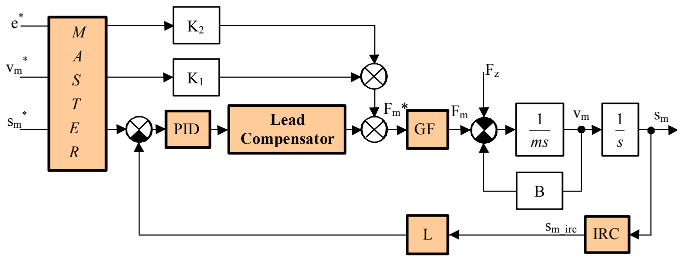

Figure 1: Entire diagram of position servo-drive (PID-proportional–integral–derivative controller, GF-Force generator, L-Luenberger observer, IRC-Incremental sensor)

2.1 Master-slave generator

Master serves as a generator of control state variables, whereby control vector can have larger dimension than number of measured values. Generator of this control vector is realized on the principal of feedback algorithm, whereby for this realization knowledge of the parameters of the control system “rough” model is needed. [4]. **Slave** contains controllers of state variables.

Master generator task is to generate curves from desired state variables which are kept under control within predefined constraints.

2.1.1 Calculation of precorrection parameters

By calculating precorrection coefficients (K_1, K_2) we are starting from condition for feed-forward control:

$$G_x(s) = \frac{1}{G_2(s)} \tag{1}$$

And dynamics of force generator is not considered for controller design calculations.

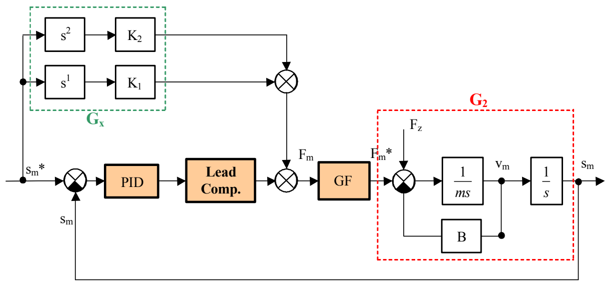

Figure 2: Position servo-drive block diagram with PID structure and marked precorrection Master-3D

From the Figure 2 results:

$$G_2(s) = \left(\frac{\frac{1}{ms}}{1 + \frac{B}{ms}} \right) \frac{1}{s} = \frac{1}{ms^2 + Bs} \tag{2}$$

After substitution in the formula (1):

$$G_x(s) = \frac{1}{G_2(s)} = ms^2 + Bs\,; \quad \begin{matrix} K_1 = B \\ K_2 = m \end{matrix} \tag{3}$$

where parameters K_1 a K_2 answer coefficients of precorrection.

2.2 Force generator

Position servo-drive (Figure 1) includes force generator LMPM. Force generator is one of the unavoidable blocks of linear servo-drive control structure, which works on a principal of vector frequency-current control synchronous motor with PM (Figure 3).

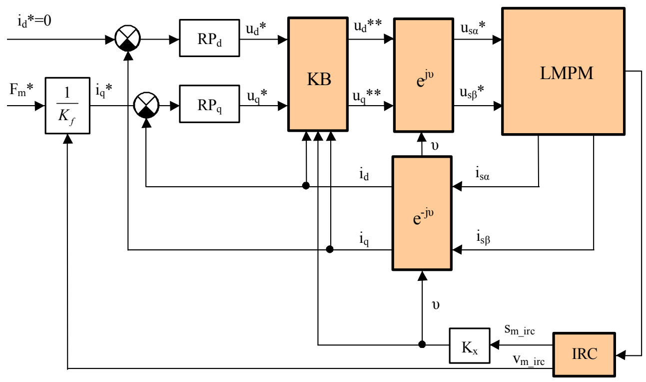

Figure 3: Force generator LMPM structure

2.3 Luenberger observer

Luenberger observer is an observer of velocity and acceleration. In general it may contain different algorithm structures for observing velocity and acceleration. In this paper we chose PID algorithm for controlling the third order system however.

We are using pole-placement method and comparing denominator of close-loop system N(s) with desired denominator $N_ž(s)$ by equal power.

$$N(s) = N_{ž}(s) \tag{4}$$

$$N(s) = s^3 + \left(\frac{D}{\tilde{m}}\right)s^2 + \frac{P}{\tilde{m}}s + \frac{I}{\tilde{m}} \tag{5}$$

$$N_{ž}(s) = \left(s^2 + 2\xi_1\omega_{01}s + \omega_{01}^2\right)\left(s + k_1\omega_{01}\right) \tag{6}$$

Parameters setup variables ξ, k and ω_0 are further explained in the Table 3.

3 Controller design methods

As mentioned in chapter 2, we are trying to compare 2 methods of PID controller design. PID controller is the most used controller in praxis and it contains 3 parallel connected sub-circuits. The first sub-circuit is proportional, which multiply controller input value with adjustable coefficient. The second parallel sub-circuit integrates and the third parallel sub-circuit derivates controller input value.

We do not consider dynamics of force generator in the synthesis of PID controller. Pole placement method is perfect for PID controller design, but it is impossible to design parameters of PID with lead compensator together. That is why we used genetic algorithms. Next chapters are describing both methods.

3.1 Pole placement

Pole placement is one of the most widely used methods of controller design. It compares denominator of close-loop system N(s) with desired denominator $N_{ž}(s)$ of equal powers.

$$N(s) = N_{ž}(s) \tag{7}$$

$$N(s) = s^3 + \left(\frac{D+B}{m}\right)s^2 + \frac{P}{m}s + \frac{I}{m} \tag{8}$$

$$N_{ž}(s) = \left(s^2 + 2\xi\omega_0 s + \omega_0^2\right)\left(s + k\omega_0\right) \tag{9}$$

And final relations/equations are:

$$P = m\omega_0^2\left(2\xi k + 1\right) \tag{10}$$

$$I = mk\omega_0^3 \tag{11}$$

$$D = \omega_0\left(2\xi + k\right)m - B \tag{12}$$

Parameters setup variables ξ, k and ω_0 are described closer in the Table 3.

Next step is the design of parameters for the lead compensator. Lead compensator is a dynamic system with a common well-known structure, which is often designed for unsuitable transfer poles or zeros of control system. In this case it is called “compensating lead compensator”.

$$G_{LC} = \frac{aT_1 s + 1}{T_1 s + 1} \tag{13}$$

Lead compensator design is not the main purpose of this paper and you can find it in paper (Radičová, Žalman)[2]. A task to design all 5 parameters (P, I, D, a, T_1) with Pole placement method in continuous system led to analytically easily unsolvable problem. So another solution for this task had to be found. It is seemed that genetic algorithms are right ones.

3.2 Genetic algorithms

GAs are one of the mostly used representatives of evolutionary computing. These algorithms are based on finding optimal solution for the given problem. Accordingly, fitness function is used and its design is the base rule for success. In this paper fitness function represents minimalization of position error using the following

$$Fitness = \sum |e| + a\sum |dy| \tag{14}$$

As a solving tool **Genetic algorithm toolbox** was used [10]. It is not a standard part of MATLAB distribution. The Toolbox can be used for solving of real-coded search and optimization problems. Toolbox functions minimize the objective function and maximizing problems can be solved as complementary tasks as well.

Process of searching

First of all, a random population is generated with a predefined number of chromosomes in one population within prescribed limits for controller parameters (for particular values see Table 1). Then two best strings according the fitness function were selected to the next generation. Bigger number of strings (18 in our case) was selected to the next generation by tournament. Then number of crossovers and mutations are applied to the population to achieve bigger chances to reach the global optimum.

This algorithm, using methods mentioned above, is able to design 6 parameters (6 because of discrete realization), for PID controller and lead compensator as well. Discrete transfer function for lead compensator is:

$$G_{LC} = \frac{a_1 z + a_2}{z + b_2} \tag{15}$$

Table 1: used parameters of genetic algorithm optimization

Number of generations	100
Number of chromosomes in one population	30
Number of genes in a string	3

Figure 4 shows algorithm progress, how was fitness function searching for optimal results.

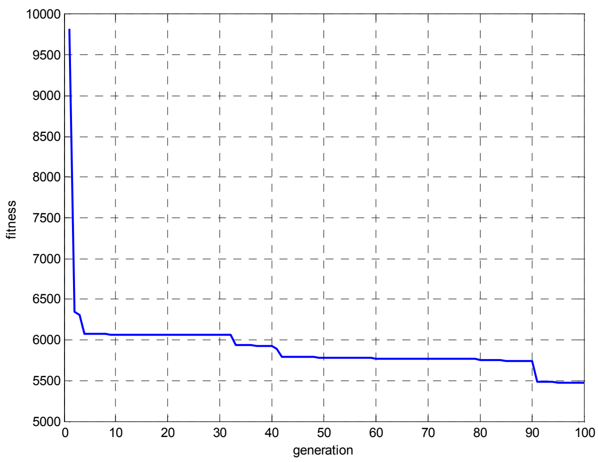

Figure 4: Convergence of genetic algorithm (GA)

4 Results

Genetic algorithms are a powerful tool for optimization purposes because they are able to find solution where the other methods fail.

First success was that position error shows lower values using GA for PID controller parameters design in comparison with pole placement method (Figure 5).

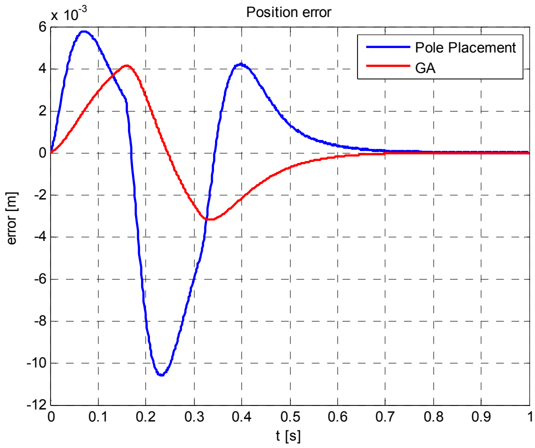

Figure 5: Position error characteristic comparing Pole placement with genetic algorithm as a calculating methods of PID controller design without precorrection constants

Next step was to use positive effect of precorrection constants and Luenberger observer and achieve more precise position of LMPM. Figure 6 represents position error of the LMPM system (Figure 1), where parameters of PID controller were designed by pole placement method with lead compensator (blue line) and parameters of PID controller and lead compensator designed by genetic algorithm (red line). Difference in the parameters obtained by different methods may be seen in Table 3. Big contribution of GA is that they are able to design 6 parameters (P, I, D, a_1, a_2, b_2) for PID controller with lead compensator and both in discrete forms.

As you can see below, value of position error is almost the same, but controller designed with genetic algorithm does not show ineligible ripple effects.

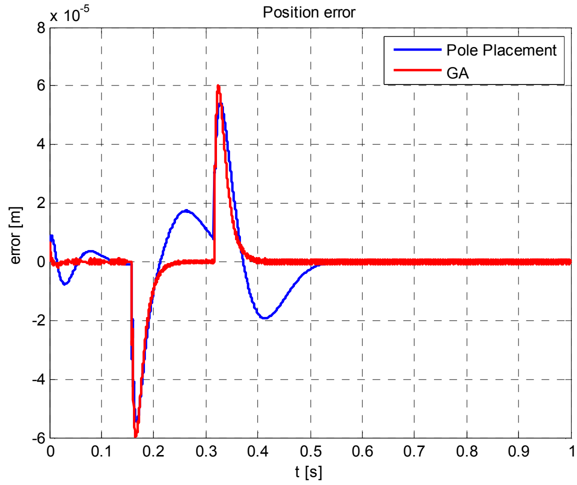

Figure 6: comparison of position error, obtained from controllers designed by Pole placement (blue) and Genetic Algorithms (red) with precorrection constants and Luenberger observer

Table 2: comparison of parameters obtained by two different methods

	P	**I**	**D**	**a_1**	**a_2**	**b_2**
Pole-placement	4737.4	99220	75.3882	20	-19.8425	-0.8425
GA	45100	4.8020	780.2274	1.5315	-1.4123	-0.4219

Conclusion

Genetic algorithm toolbox in collaboration with MATLAB is a very powerful tool for optimization and search problems. Provided figures show that within 100 generations, a solution better than analytical (Pole placement method) was found. At this point it can be said that genetic algorithm is capable to design more than 3 parameters in comparison with pole placement method what is a significant contribution in this area. Using positive effect of precorrection and Luenberger observer led to the achievement of more precise positioning of LMPM (Figure 6). Finally, we can allege that PID controller with lead compensator parameters designed by genetic algorithms possess significant impact on positioning precision.

Acknowledgement

This paper was written with support of the project APVV-0530-07.

Appendix

Table 3: Acronym table

Acronym	**Meaning**	**Value**
T	Sampling period	0.2 ms
Parameters for PID controller		
ξ	Damping index	1
k	Shift pole index	1
ω_0	Bandwidth	$2\pi f_0$
f_0	Frequency	10 Hz
Parameters for Luenberger observer		
ξ_1	Damping index	1
k_1	Shift pole index	1
ω_{01}	Bandwidth	$2\pi f_0$
f_{01}	Frequency	10 Hz
Parameters for precorrection		
K_1	Precorrection constant	B = 0.01 $kg.s^{-1}$
K_2	Precorrection constant	m = 0.4 kg
Parameters for IRC sensor		
N	Resolution	2 µm

References

[1] Ivan Sekaj, *Evolučné výpočty a ich využitie v praxi*. Iris, Bratislava 2005.

[2] Radičová T., Žalman M.; *Master-slave position servo-drive design of aircore linear motor with permanent magnets*, AT&P Journal Plus 1/2010

[3] Mitchell M.; *Introduction to genetic algorithms*, MIT Press, 1998

[4] Žalman M., *Akčné členy,* STU Bratislava 2003

[5] Khater, F.; Shaltout, A.; Hendawi, E.; Abu El-Sebah, M.; , "PI controller based on genetic algorithm for PMSM drive system," Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on , vol., no., pp.250-255, 5-8 July 2009 doi: 10.1109/ISIE.2009.5217925
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5217925&isnumber=5213059

[6] Solano, J.; Jones, D.I.; , "Parameter determination for a genetic algorithm applied to robot control," Control, 1994. Control '94. International Conference on , vol.1, no., pp.765-770 vol.1, 21-24 March 1994 doi: 10.1049/cp:19940229
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=327048&isnumber=7757

[7] Wang, Y.P.; Hur, D.R.; Chung, H.H.; Watson, N.R.; Arrillaga, J.; Matair, S.S.; , "A genetic algorithms approach to design an optimal PI controller for static VAr compensator," *Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference on* , vol.3, no., pp.1557-1562 vol.3, 2000 doi: 10.1109/ICPST.2000.898203
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=898203&isnumber=19429

[8] Introduction to Genetic Algorithms, Internet article,
http://www.obitko.com/tutorials/genetic-algorithms/biological-background.php

[9] Zhou Yun-fei, Song Bao, Chen Xue-dong : Position/force control with a lead compensator for PMLSM drive system, Springer-Verlag London Limited 2005, 18 November 2005

[10] I.Sekaj, M.Foltin. Matlab toolbox – Genetické algoritmy. Konferencia Matlab 2003. Praha, 2003

Ing. Tatiana Radičová
tatiana.radicova@stuba.sk

Ing. Jakub Vonkomer
jakub.vonkomer@stuba.sk

Prof. Ing. Milan Žalman, PhD.
milan.zalman@stuba.sk

Doc. Ing Ivan Sekaj, PhD.
ivan.sekaj@stuba.sk