Master Slave LMPM Position Control Using Genetic Algorithms

Tatiana Radičová and Milan Žalman

Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and Information Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia
{tatiana.radicova,milan.zalman}@stuba.sk

Abstract. Recently, in the era of high speed computers, nanotechnology and intelligent control; genetic algorithms belong to the essential part of this high tech world. Therefore, this paper sticks two actual topics together - linear motor and genetic algorithm. It is generally known that linear motors are maintenance free and they are able to evolve high velocity and precision which is why we made closer look on this topic. To make the linear motor more precise, genetic algorithm was applied. The GA role was to design optimal parameters for PID regulator, lead compensator and Luenberger observer to ensure the most precise positioning. Eventually, some experiments were done to demonstrate the impact of Luenberger observer and it will be also shown responses of position, velocity, force, and position error, which were gained from the experiment using GA.

Keywords: linear motor, genetic algorithm, master-slave control, Luenberger observer.

1 Introduction

Linear motors with permanent magnets (LMPM) found their right place within all kinds of motors covering the applications where the standard rotary motors have no chance to succeed. Thanks belong to their matchless features like better positioning and precision, no maintenance and no bearing ware. Therefore these motors can be found in various sectors, whether in electro-technical or electronic production – drives used in the elevators, conveyors, pumps, compressors, paper machines, robots, etc. Above all it has to be mentioned that linear motors purchase the popularity mainly by their implementation in the velocity trains such as Maglev or Trans-rapid.

This paper will be focused on position servo-drive control design of LMPM comparing two methods: Genetic Algorithm (GA) and Pole Placement Method with 4D Master slave control and Luenberger observer. It will be unveiled the positive effect of Luenberger observer in one of the experiments, as well.

GA is 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. GA is able to design the parameters for PID controller, lead compensator and Luenberger observer at the same time, 9 parameters,

V. Snasel et al. (Eds.): SOCO Models in Industrial & Environmental Appl., AISC 188, pp. 73–82.
springerlink.com

which is the main reason GA was applied. There is no problem to design these parameters independently with Pole-placement method, but it is naturally time consuming and mathematical techniques have to be well-known.

The 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].

2 LMPM Position Control

2.1 Position Servo-Drive, Implementation Block Scheme

Position servo-drive can be performed by various algorithms (PID, PIV, P+PI…). PID algorithm with lead compensator is applied, referring to the article [2]. However, the entire block diagram consists of Luenberger observer and 4D master generator in addition.

The entire position servo-drive structure may be seen in the Fig.1.

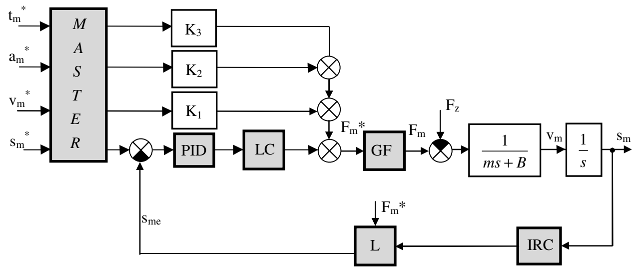

Fig. 1. Entire diagram of position servo-drive (PID-proportional–integral–derivative controller, LC-Lead Compensator, GF-Force generator, L-Luenberger observer, IRC-Incremental sensor)

2.2 Force Generator

Force generator GF is one of the most important blocks of linear servo-drive control structure and works on a principal of vector frequency-current control synchronous motor with PM (Fig.2). It contains blocks of Park's transformation, compensation block, IRC sensor, two current controllers (CC_d, CC_q) and block LMPM – particular servo-drive realized by following equations

$$\begin{gathered} u_d = R_s i_d + \frac{d\psi_d}{dt} - \omega_s \psi_q \\ u_q = R_s i_q + \frac{d\psi_q}{dt} + \omega_s \psi_d \\ \psi_f = L_{md} i_f \\ \psi_d = L_d i_d + \psi_f \\ \psi_q = L_q i_q \\ F_m - F_z = m\frac{dv_m}{dt} \\ \omega_s = K_x v_m \ ; \qquad \upsilon = K_x s_m \end{gathered} \tag{1}$$

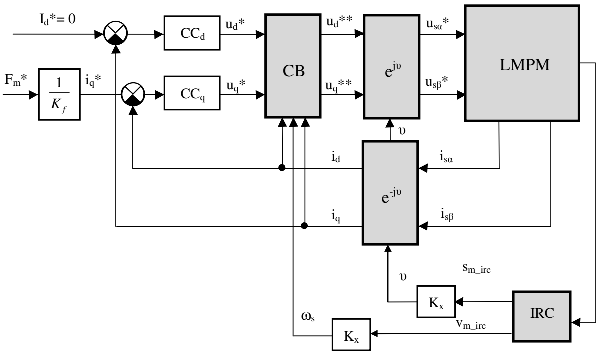

Fig. 2. Force generator LMPM structure

Equation (2) represents a relation between rotary and linear parameters issued from the physical interpretation.

$$v_m = 2\tau_p f_s \tag{2}$$

τ_p – Pole spacing [m]
f_s – Power supply frequency [Hz]

Then generally holds the equation (3).

$$K_x = \frac{\omega_s}{v_m} = \frac{\pi}{\tau_p} \tag{3}$$

2.3 Master-Slave Control

Master slave control can be assign to the status control or model control [4]. Its significant advantage is the inutility of knowing the exact mathematical model of controlled system. The quality model of regulating system is highly sufficient providing that you are familiar with the scale of the main parameters.

2.3.1 Master-Slave Generator

Master serves as a generator of control state variables and surprisingly the control vector can be greater than number of measured variables. Its task is to generate desired waveforms of state variables – control vectors which shape can be rectangular, trapezoidal or sinusoidal, either 3-dimensional or 4-dimensional. In this paper is used 4D master slave generator and it generates state variables of the position, velocity, acceleration and jerk (Fig.3).

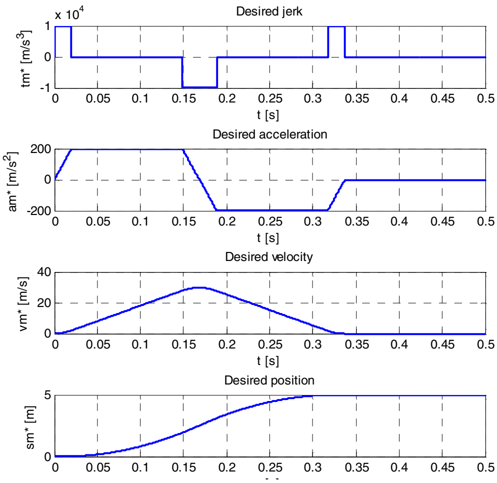

Fig. 3. Time responses of Master slave output values

2.3.2 Precorrection Constants

Among indispensable parts in Master-slave control belong precorrection constants. Their task is to enhance position accuracy and consequently lower the position error. Calculating precorrection coefficients (K_1, K_2, and K_3) starts from the condition for feed forward control and force generator dynamics is considered.

$$G_x(s) = \frac{1}{G_2(s)} \tag{4}$$

$$G_2(s) = \frac{1}{T_{gm}s+1}\frac{1}{ms+B}\frac{1}{s} = \frac{1}{T_{gm}ms^3 + \left(T_{gm}B+m\right)s^2 + Bs}$$

$$G_x(s) = \frac{1}{G_2(s)} = T_{gm}ms^3 + \left(T_{gm}B+m\right) + Bs$$

$$K_1 = B;\ K_2 = T_{gm}B + m;\ K_3 = T_{gm}m$$

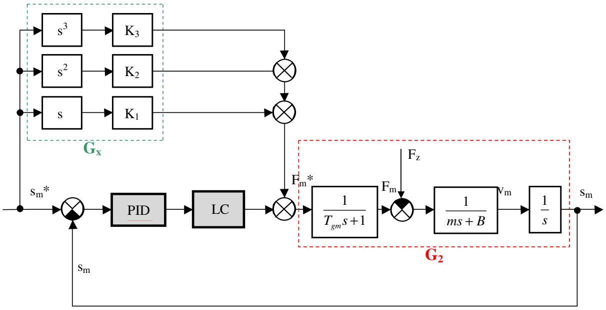

Fig. 4. Force Position servo-drive block diagram with PID structure and marked precorrection Master-3D (PID-proportional–integral–derivative controller, LC-Lead Compensator, GF-Force generator,)

2.4 Luenberger Observer

Observers are algorithms that combine sensed signals with other knowledge of the control system to produce observed signals [2]. They can enhance the accuracy and reduce the sensor-generated noise. Consequently, among various observers, Luenberger observer was chosen. Basically it is the observer of velocity and acceleration. In general, it may contain different algorithm structures. In this paper is chosen PID algorithm for controlling the third order system, though.

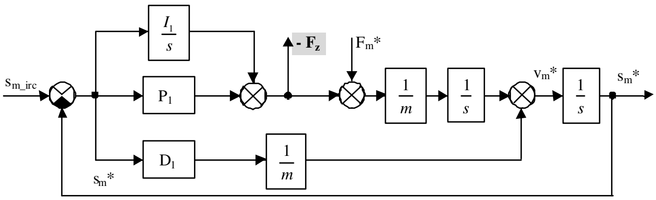

Fig. 5. Luenberger observer block diagram

Pole-placement method is applied. It compares denominator of close-loop system N(s) with desired denominator $N_D(s)$ by equal power.

$$
\begin{gathered}
N(s)=s^3+\left(\frac{D_1}{\tilde{m}}\right)s^2+\frac{P_1}{\tilde{m}}s+\frac{I_1}{\tilde{m}} \\
N_D(s)=\left(s^2+2\xi_1\omega_{01}s+\omega_{01}^2\right)\left(s+k_1\omega_{01}\right) \\
N(s)=N_D(s) \\
P_1=\tilde{m}\omega_{01}^2\left(2\xi_1k_1+1\right) \\
I_1=\tilde{m}k_1\omega_{01}^3 \\
D_1=\omega_{01}\left(2\xi_1+k_1\right)\tilde{m}
\end{gathered}
\tag{5}
$$

Parameters setup variables ξ_1, k_1 and ω_{01} are further explained in the Table 3.

3 Controller Design Methods

The comparison of two controller design methods will be presented. Pole Placement method is used for designing the PID controller.

3.1 Pole Placement Method

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 $N_D(s)$ by equal power. Accordingly, controller parameters are designed (P, I, D), however force generator GF dynamics is not considered.

$$\begin{gathered} N(s)=N_D(s) \\ N(s)=s^3+\left(\frac{D+B}{m}\right)s^2+\frac{P}{m}s+\frac{I}{m} \\ N_D(s)=\left(s^2+2\xi\,\omega_0 s+\omega_0^2\right)(s+k\omega) \\ P=m\omega_0^2(2\xi k+1) \\ I=mk\omega_0^3 \\ D=\omega_0(2\xi+k)m-B \end{gathered} \tag{6}$$

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

Lead compensator coefficients are design by well-known method using relation (lead - lag).

$$G_{LC}=\frac{aT_1s+1}{T_1s+1} \tag{7}$$

The lead compensator design is not the main purpose of this paper and you can find it in (Radičová, Žalman)[12]. A task to design parameters for PID controller together with lead compensator by Pole-placement method led to analytically unsolvable problem. Therefore, another solution for this task had to be found.

However, the design of PID and lead compensator parameters were continuous, realization was discreet.

3.2 Genetic Algorithm

GA, one of the mostly used representatives of evolutionary computing, is based on finding optimal solution (optimal structure and controller parameters) for the given problem. Accordingly, the base rule for success is the precise fitness function design. Hence the fitness function represents minimization of position error using the following

$$Fitness=\sum|e|+a\sum|dy| \tag{8}$$

Genetic algorithm toolbox was used as a solving tool [10]. It is not the 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.

The process of searching is adjusted very sophistically. First of all, a random population is generated with a predefined number of chromosomes in one population within prescribed limits for controller, lead compensator and Luenberger observer parameters (for relevant values see Table 1). To achieve preferable parameters, fitness function, which minimized position error from Luenberger observer, was applied.

$$Fitness_{Luenberger} = \sum |e| \tag{9}$$

Then two best strings according the first fitness function were selected to the next generation. Bigger number of strings 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 progress is the same for the fitness function used for Luenberger observer. Finally, the best parameters are chosen.

Table 1. Table of parameters extracted from GA

Number of generations	100
Number of chromosomes in one population	30
Number of genes in a string	3
Parameter "a" weight	0.7

This algorithm, using the method mentioned above, is able to design 9 parameters at once. Therefore, GA belongs to the very effective algorithms which employ easiness compared to the incredible mathematic severity. Eq. 10 represents transfer function of parameters obtained from GA for the lead compensator, the PID controller and the representation of Luenberger observer (Fig.5). Table 2 shows concrete parameters designed with GA and Pole Placement method according to Fig.1.

$$\begin{aligned} G_{LC}(z) &= \frac{a_1 z + a_2}{z + b_2} \\ G_{PID}(z) &= P + I\frac{Tz}{z-1} + D\frac{z-1}{Tz} \\ G_{LO}(z) &= P_1 + I_1\frac{Tz}{z-1} + D_1\frac{z-1}{Tz} \end{aligned} \tag{10}$$

Table 2. Table of parameters extracted from GA

	P	I	D	a_1	a_2	b_2	P_1	I_1	D_1
GA	6193	61.55	138.4	8	-0.987	-0.293	3917	251080	129.25
Pole Placement	4737	99220	75.39	20	-19.95	-0.9454	4737	992200	301.59

4 Simulation Results

At the beginning, it has to be mentioned that following experiments were performed on the simulation model in Matlab Simulink environment using Luenberger observer and precorrection constants according to the Fig.1.

First experiment (Fig.6, Fig.7) compares the behavior of force and position error with/without Luenberger observer. Table 3 shows used simulation parameters.

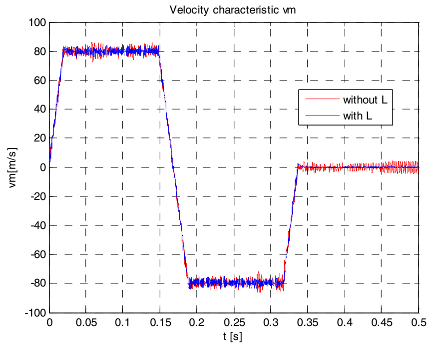

Fig. 6. The time response of force comparing the response with/without Luenberger observer

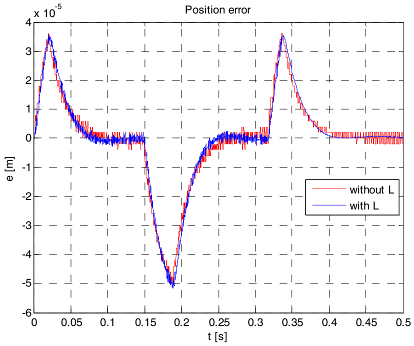

Fig. 7. The time responses of position error comparing the response with/without Luenberger observer

Table 3. Table of acronyms

Acronym	Meaning	Value
T	Sampling period	0.2 ms
T_{gm}	Time constant of GF	0.5 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	10
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

Second experiment represents linear servo-drive behavior using parameters gained from GA and Pole Placement method with the parameters listed in the Table 3 and Table 4 (Fig.8, 9, 10, 11).

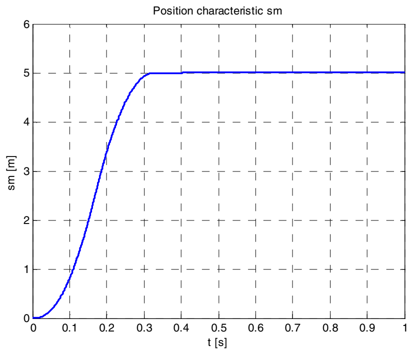

Fig. 8. The time response of position

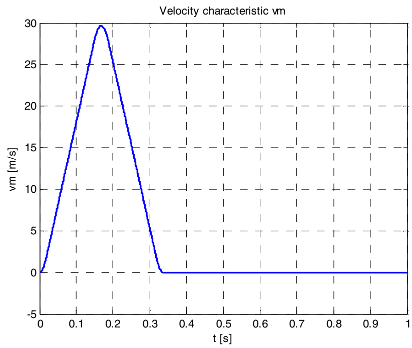

Fig. 9. The time response of velocity

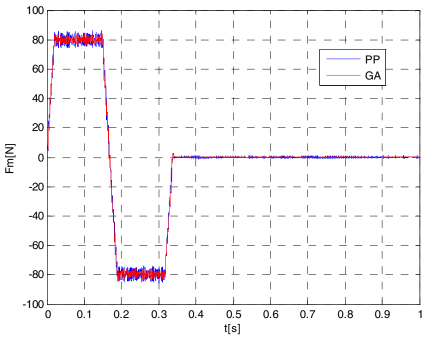

Fig. 10. The time response of force

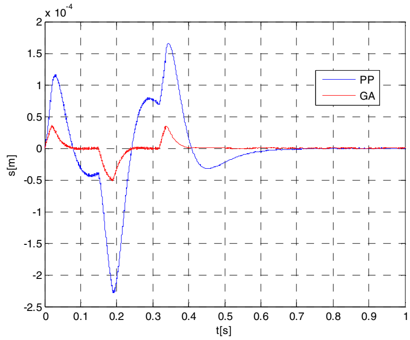

Fig. 11. The time response of position error

5 Conclusion

It has been performed two experiments which confirm that genetic algorithm toolbox in connection with MATLAB is a very powerful tool for optimization and search problems. GA was able to design nine optimal parameters for linear servo drive that is the significant contribution to this area. In addition, as can be seen in the Fig. 11, the positioning accuracy and dynamics of the system is very high using GA. Eventually, it has to be mentioned that using Luenberger observer and precorrection constant led to the more precise positioning of LMPM.

Acknowledgment. This work was supported by the Slovak Research and Development Agency under the contract No. VMSP-II-0015-09.

References

[1] Sekaj: Evolučné výpočty a ich využitie v praxi. Iris, Bratislava (2005)
[2] Ellis, G.H.: Observers in Control Systems. Academic Press (2002)
[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. In: IEEE International Symposium on Industrial Electronics, ISIE 2009, July 5-8, pp. 250–255 (2009), `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5217925&isnumber=5213059`, doi:10.1109/ISIE, 5217925
[6] Solano, J., Jones, D.I.: Parameter determination for a genetic algorithm applied to robot control. In: International Conference on Control 1994, March 21-24, vol. 1, pp. 765–770 (1994), `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=327048&isnumber=7757`, doi: 10.1049/cp:19940229
[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. In: International Conference on Power System Technology, Proceedings, PowerCon 2000, vol. 3, pp. 1557–1562 (2000), `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=898203&isnumber=19429`, doi:10.1109/ICPST.2000.898203
[8] Introduction to Genetic Algorithms, Internet article, `http://www.obitko.com/tutorials/genetic-algorithms/biological-background.php`
[9] Zhou, Y.-F., Song, B., Chen, X.-D.: Position/force control with a lead compensator for PMLSM drive system. Springer-Verlag London Limited (November 18, 2005)
[10] Sekaj, I., Foltin, M.: Matlab toolbox – Genetické algoritmy. Konferencia Matlab 2003. Praha (2003)
[11] Radičová, T., Žalman, M.: LMPM Position Control with Luenberger Observer Using Genetic Algorithms. In: ELEKTRO 2012, Rajecké Teplice (2012)
[12] Radičová, T., Žalman, M.: Master-slave position servo-drive design of aircore linear motor with permanent magnets. AT&P Journal Plus (January 2010)