Evolutionary Design of Polynomial Controller

R. Matousek, S. Lang, P. Minar and P. Pivonka

***Abstract*—**In the control theory one attempts to find a controller that provides the best possible performance with respect to some given measures of performance. There are many sorts of controllers e.g. a typical PID controller, LQR controller, Fuzzy controller etc. In the paper will be introduced polynomial controller with novel tuning method which is based on the special pole placement encoding scheme and optimization by Genetic Algorithms (GA). The examples will show the performance of the novel designed polynomial controller with comparison to common PID controller.

***Keywords*—**Evolutionary design, Genetic algorithms, PID controller, Pole placement, Polynomial controller

I. Introduction

EVOLUTIONARY algorithms or generally various soft-computing methods provide very robust tools usable in tasks of mathematical optimization. In this paper shall be presented new method for polynomial controller design which is based on pole placement encoding scheme and Genetic Algorithms (GA) optimization. As optimization task the optimal setting of PID controller vs. Polynomial Controller for two dynamic systems has been chosen. From the point of view of optimization it is a nontrivial task of setting the optimum parameters of dynamic system. In general it is a task nonlinear and multi-criterion. From point of view of genetic algorithms (generally optimization soft-computing algorithms) this type of task can be viewed as challenge which solution has very practical implementations [3], [4].The general concept of the negative feedback loop to control of the plant is in Fig. 1.

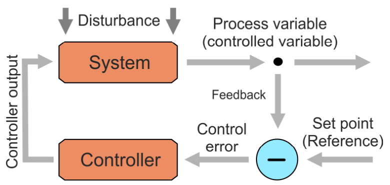


Fig. 1 The general concept of the negative feedback loop to control the dynamic behaviour of the system with major parts description

PID controller can be viewed as common tool usable for controlling of industrial and non-industrial processes. Controller can be used to control velocity, revolutions, etc.

Presented polynomial controller design which is based on special pole placement encoding scheme is generalization of the controller possibility. Commonly the PID algorithm or our presented Polynomial controller are in the process of control implemented using PLCs (Programmable Logic Controllers), DCSs (Distributed Control Systems), IPCs (Industrial PC Control Systems) or single loop or standalone controllers.

In this paper a novel optimal polynomial controller tuning approach based on the special pole placement encoding scheme and genetic algorithms optimization is proposed. For the next interpretation we will denote this controller as ZPK controller (zero-pole-gain). There is willful similarity with name of Matlab function "zpk". The Matlab and the GA implementation in Global Optimization Toolbox were used for this research.

All authors are from Brno University of Technology, Faculty of Mechanical Engineering, Dept. of Applied Computer Science, Technicka 2896/2, 61669 Brno, Czech Republic, corresponding author have e-mail matousek@fme.vutbr.cz

This work was supported by the research projects of MSM 0021630529 "Intelligent Systems in Automation", GACR No.: 102/091668 "Evolutionary Control Design", and IGA (Internal Grant Agency of Brno University of Technology) FSI-S-11-31 "Application of Artificial Intelligence".

II. Polynomial controller

A. PID Controller

The theory of control deals with methods which leads to change of behavior of controlled dynamic system (further only system). The desired output of a system is called the reference or set point. When one or more outputs of the system need to follow a certain reference over time then a controller modifies the inputs of system to obtain the desired value on the output of the system.

A PID controller is a generic control loop feedback mechanism which is the most commonly used feedback controller [2]. The PID controller has three separate constant parameters: Proportional (P), Integral (I) and Derivative (D). It can be said the P depends on present error, I on accumulation of past errors and D is prediction of future errors based on rate of change. Basic block diagram of PID controller is based on parallel circuit [8]. The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining $u(t)$ as the controller output, the general form of the PID algorithm is:

$$u(t) = K_P e(t) + \frac{1}{T_I}\int_0^t e(t)dt + T_D \frac{de(t)}{dt} \tag{1}$$

where constant K_P is gain and T_I resp. T_D are the integrative resp. derivative time constants. In our case we have used for testing the simplified variant of PID controller given by equation (2).

$$u(t) = K_P \left[e(t) + K_I \int_0^t e(y)dy + K_D \frac{de(t)}{dt} \right] \tag{2}$$

B. Polynomial controller and system Poles and Zeros

In our point of view we defined in the paper the polynomial controller (also similarly ZPK controller) as controller, which is represented by rational function [1]. I.e. the controller is implemented as ratio of polynomials where numerator represents zeros of transfer and denominator the poles, Fig. 2.

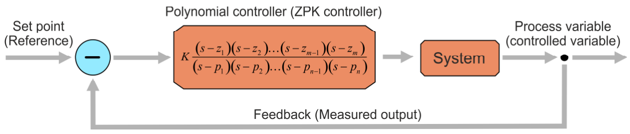


Fig. 2 The polynomial controller (ZPK controller) in control loop

Transfer is naturally defined in s-plane system. This polynomial transfer function provides a ground for determining important system response characteristics without solving the complex differential equation. As definition, the transfer function of the polynomial controller is a rational function in the complex variable $s = \sigma + j\omega$, that is

$$C(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} \tag{3}$$

For our next work is convenient to factor the polynomials in the numerator and denominator, and to write the transfer function in terms of those factors:

$$C(s) = \frac{N(s)}{D(s)} = K\frac{(s-z_1)(s-z_2)\ldots(s-z_{m-1})(s-z_m)}{(s-p_1)(s-p_2)\ldots(s-p_{n-1})(s-p_n)} \tag{4}$$

where the numerator $N(s)$ and denominator $D(s)$ of polynomials have real coefficients defined by the system's differential equation and $K = b_m / a_n$. As we can see from (4) the z_i's are the roots of the equation $N(s)$=0, and are denoted to be the system Zeros, and the p_i's are the roots of the equation $D(s)$=0 and are denoted to be the system Poles.

There are in (4) the factor in the numerator and denominator written so that when $s = z_i$ the numerator $N(s)$=0 and the transfer function vanishes, that is

$$\lim_{s \to z_i} C(s) = 0 \tag{5}$$

and similarly when $s = p_i$ the denominator polynomial $D(s)$=0 and the value of the transfer function becomes unbounded by .

$$\lim_{s \to p_i} C(s) = \infty \tag{6}$$

all of the coefficients of polynomials $N(s)$ and $D(s)$ are real, therefore the poles and zeroes must be either purely real, or appear in complex conjugate pairs. The influence of the location of the roots in s-plain to control process, is in Fig. 3.

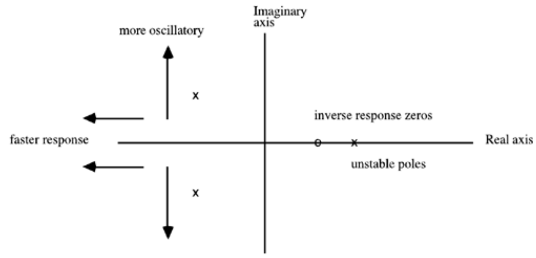


Fig. 3 Influence of the roots (poles, zeros) in case of stability. Poles in the positive part (real axis) of s-plane mean unstable process. The imaginary part of roots mean oscillations. The poles further from zero has influence on the fast response of the system. Special example is zero in the positive part of s-plane, in this case the system is non-minimal and the first response has inverse direction

III. Evolutionary Design

A. Genetic algorithm (GA)

The Genetic Algorithm (GA) is a well known optimization technique inspired by biological principles of natural selection (Darwin's theory of selection of species) and genetics (Mendel's theory of heredity). GA belongs to a larger class of Evolutionary Algorithms (EAs), or more generally EA belong to the main domain of soft computing or artificial intelligence [3], [7]. EA generate solution to optimization problems using techniques inspired by natural evolution, such as selection, crossover, mutation, inheritance. A GA computer implementation is heuristics, which means it estimates a solution. An effective GA representation and significant fitness (objective function) evaluation are the keys of the success in GA implementations. The appeal of GAs comes from their simplicity and elegance as robust global search algorithms as well as from their power to discover good solutions rapidly for high-dimensional problems. The common flow chart of GA optimisation of polynomial controller is in the Fig. 4.

We used GA implementation which is included in Matlab global optimisation toolbox in case of our polynomial controller parameters' design. The selected GA parameters are: size of population is 50 individuals (chromosomes), SAS selection scheme, n-points crossover, and adaptive mutation.

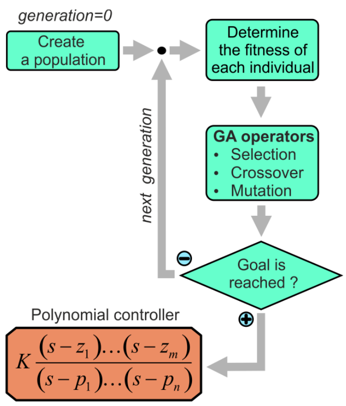


Fig. 4 The GA optimization flow chart with result

B. Zero-pole-gain encoding scheme (ZPK encoding)

Using the Laplace transform we can denote the polynomial controller, also ZPK controller, as ration of two polynomials (3). In this case we consider polynomials of degree higher than 2. Important note is that positive coefficients for polynomials of order 3 and higher are necessary condition of stability but not sufficient condition [1], [5].

For this reason the optimization of polynomial's coefficients for optimal polynomial controller setting is relatively ineffective. The reason is, many unstable solutions emerge. If the controller is unstable the whole control circuit is unstable. Another important disadvantage is sensitivity to change of parameter when changing one parameter generates change of all roots of given polynomial (numerator, denominator).

Major contribution of this paper is new system of coding of polynomial controller which eliminates introduced disadvantages in connection with evolutionary optimization using Genetic Algorithm. Optimization of controller is realized in plane of Laplace transform, i.e. in *s*–plane. Optimized is the location of roots of polynomial which is synthesized after. From theory we know that if we place roots only to left half-plane of *s*–plane the controller is stable transfer element of the system. This way we will eliminate huge part of prior unstable solutions given by influence of developed controller. Our new designed way of coding roots of polynomial moreover partially optimize the structure of controller which is important as well.

For further definition we consider a polynomial controller with real roots (poles and zeros). Further it is necessary to comply with following conditions: PZK controller must be stable, causal, and also we will demand zeros of transfer were in left half-space of *s*-plane. According to dynamics of given plant it is then necessary to choose suitable range of values of roots, i.e. optimized parameters of controller. For example for given plant will be suitable range of values [-1000; 0]. The value of root -1000 corresponds to 1 ms in dynamic time interpretation. For optimization algorithm we choose range of values [-1000; 1000]. This way we will create a set of possible roots, which will not be used for roots' values optimization (they don't comply with the stability condition), but they will ensure the possible structural variability of polynomial controller. For clarification of this principle see Fig. 5.

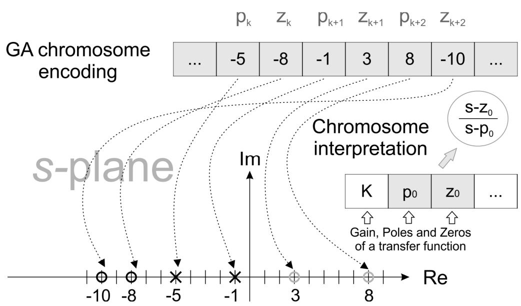


Fig. 5 Example of encoding polynomial controller and it's transform to transfer form with zeros and poles

Values of individual (chromosome) generated by the genetic algorithm which are positive are excluded from polynomial. This way the structural changes of controller's transfer are achieved. A problem can arise if order of polynomial in the numerator is higher than order of polynomial in the denominator of ZPK controller. In this case the controller would not show causal behavior. Given problem is solved by adding realization constants with fast dynamics into the denominator. These constants affect the resulting dynamic behavior of plant only minimally and simultaneously ensure causal behavior of controller.

Generalization of introduced practice to complex area is obvious. Whereas we should note the complex roots with nonzero imaginary component exist only as complex conjugates and therefore it is sufficient to code them only to one parameter of GA individual (chromosome), see Fig. 6.

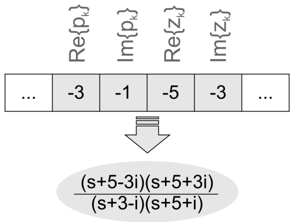


Fig. 6 Example of interpretation of coding of general polynomial controller with real and complex roots. The presence of the negative values in the imaginary part of the GA chromosome is interpreted as complex conjugation of the imaginary numbers

Also in the case of general encoding scheme, where is possible to obtain complex form of transfer function, one can reduce the structure and imaginary part of the complex transfer function using positive values in given part of the individual (GA chromosome). The positive value of the imaginary part of the complex root means elimination of complex root, i.e. we obtain only real part of the root. The negative value of the imaginary part of the complex root means roots with complex conjugation. The positive value of real part of the complex root means elimination of the given complex root, i. e. structural modification of the complex transfer function. An influences of the positive values in the GA chromosome are shown in the Fig. 7.

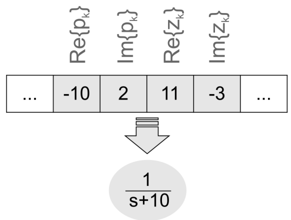


Fig. 7 Example of interpretation of coding of general polynomial controller with real roots. In this case the positive imaginary value p_k encoding process eliminate imaginary part of complex root, i.e. pole and positive value z_k means elimination of the root, i.e. structural modifications of the complex transfer function

International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:5, No:11, 2011

IV. Experimental Systems

For our tests we have designed two dynamic systems. The test system one is given in time domain by (7) and in s domain by transfer function (8) or more practically by transfer function in gain-time constant form by (9). This is open loop stable system of third order with one step response by Fig. 8.

$$y_1(t) = \frac{3}{2}e^{-\frac{1}{2}t} - 6e^{\frac{1}{4}t} + \frac{9}{2}e^{-\frac{1}{6}t} \tag{7}$$

$$G_1(s) = \frac{6}{48s^3 + 44s^2 + 12s + 1} \tag{8}$$

$$G_1(s) = \frac{6}{(2s+1)(4s+1)(6s+1)} \tag{9}$$

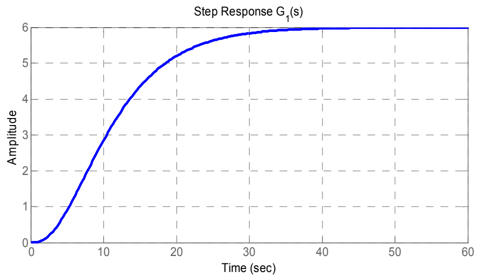


Fig. 8 Unit step response of LSI system $G_1(s)$ with zero initial state

The test system two is given in time domain by (10) and in s domain by transfer function (11) or more practically by transfer function in gain-time constant form by(12). This is open loop oscillating system of third order with one step response by Fig. 9.

$$y_2(t) = \frac{2}{5}\sin\left(\frac{1}{4}t\right) - \frac{1}{5}\cos\left(\frac{1}{4}\right) + \frac{1}{5}e^{-\frac{1}{2}t} \tag{10}$$

$$G_2(s) = \frac{2}{32s^3 + 16s^2 + 2s + 1} \tag{11}$$

$$G_2(s) = \frac{2}{(2s+1)(4s+i)(4s-i)} \tag{12}$$

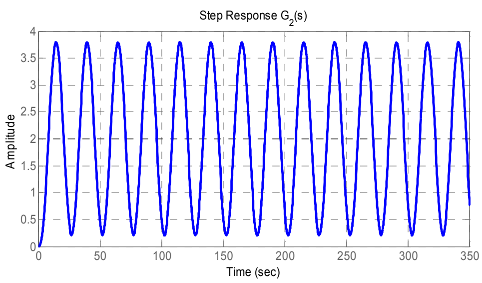


Fig. 9 Unit step response of LSI system $G_2(s)$ with zero initial state

V. Test and Results

All tuned parameters of the polynomial controllers (ZPK controller) for our test suite were designed by means of GA optimization process, i.e. all roots, gain and in consequence the structure of the polynomial controller. As criteria of optimality (fitness, objective function) the ITAE integral criteria (13) was used.

$$f_{ITAE}(K, \mathbf{p}, \mathbf{z}) = \int_0^t t\left|e(t)\right| dt \tag{13}$$

The basic models which were used for the polynomial controller design are given in Table 1. There the maximum number of roots, i.e. Zeroes and Poles means the maximum degree of polynomial. This fact is not limited and of course some coefficients can be missed in the final polynomial controller design. There we can remark that PID controller was extended in transfer function form for small time constant (0.001 s) due to derivative part of the controller as one can see on the following presented solution.

TABLE I
The Base Models of Polynomial Controllers and the Labels

Label of the controller model	Number system	Max degree of polynomial function*	Number of optimized parameters**
GAPID	*Real*	*2*	*3*
GAR3C	*Complex*	*6*	*13*
GAR5	*Real*	*5*	*11*
GAR5C	*Complex*	*10*	*21*

** It is means the max number of roots of the models generated by GA.*
*** Sum of optimize parameters, i.e. numerators, denominators and gain.*

As example we can show the GAR3C polynomial controller rational function in general form (14). In practical implementation the one pole is added by force (15) in order to reach zero steady state error of the control process (integrated part in the controller can guarantee the zero steady state error).

$$C_{GAR3C}(s) = K\frac{(s-z_1)(s-z_2)(s-z_3)}{(s-p_1)(s-p_2)(s-p_3)} \tag{14}$$

$$C_{GAR3C+}(s) = K\frac{(s-z_1)(s-z_2)(s-z_3)}{(s-p_1)(s-p_2)(s-p_3)(s-0)} \tag{15}$$

GA is the heuristic search algorithm which uses random modification of its own behavior. Therefore, all of presented results are based on the 30 test runs of the GA optimization process.

A. Test System $G_1(s)$

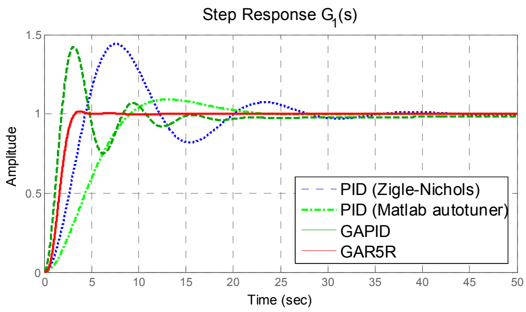
The best results of optimal tuning of the polynomial controller (ZPK controller) based on fitness are given in Table II. There are descriptive characteristic (popular performance criteria) of the unit step response for better objectiveness and possibility of comparing our results with another.

TABLE II
THE PERFORMANCE CHARACTERISTICS OF THE ZPK CONTROLLERS FOR $G_1(S)$

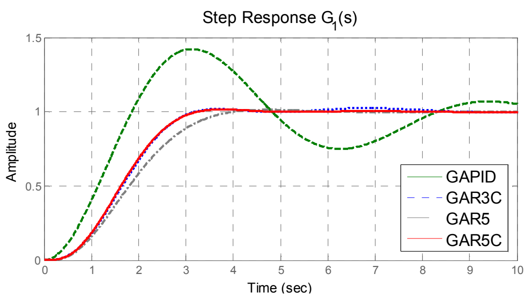
Model of controller	Fitness (ITAE)	Setting* time (±2 %)	Rise Time (90 %)	Overshoot
GAPID	15.7	38.12 s (40.72, 8.31)	1.71 s	0.420
GAR3C	7.5	7.49 s (14.1, 16.69)	2.61 s	0.025
GAR5	7.8	3.69 s (5.33, 4.15)	3.07 s	0.013
GAR5C	6.5	3.04 s (5.29, 15.21)	2.60 s	0.016

* *There are also median value and standard deviation in the brackets.*

The comparison between suggested models of polynomial controller and classical PID controller solution in case of unit step response is in Fig. 10. There one can see the huge differences between GARx solutions and another ones. The Fig. 10 show differences of solutions for all polynomial controllers given in Table II as well.



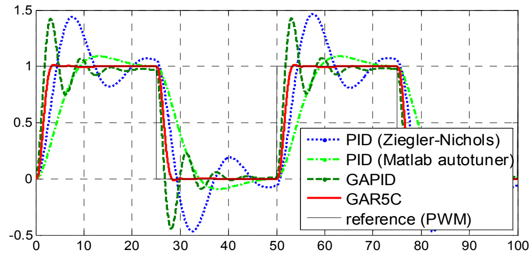
(a)



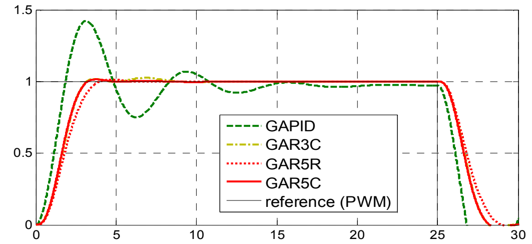
(b)

Fig. 10 Unit step response of close loop system $G_1(s)$ and GARx models of polynomial controllers. The comparison between GAR5 model of polynomial controller and classical PID controller solutions (above). Detail of unit step response characteristics for all presented GARx models (bottom)

The comparison between suggested models of polynomial controller and classical PID controller solution in case of PWM reference signal can be also interested in case of practical implementation, Fig. 11. There one can see the huge differences between GAR5C solution and another ones. The Fig. 11 clearly show the differences of solutions for the variants of polynomial controllers (ZPK controllers).



(a)



(b)

Fig. 11 PWM response of close loop system $G_1(s)$ and GARx models of polynomial controllers. The comparison between GAR3C and GAR5C show approximately similarity of the results

B. Test System $G_2(s)$

The system $G_2(s)$ is an oscillator in fact (10). The system also include complex roots. Due to this system is complicated for the standard PID structure tuning. The polynomial controller has bigger advantage in this case. Results of optimal tuning of the polynomial controller (ZPK controller) based on fitness are given in Table III.

TABLE III
THE PERFORMANCE CHARACTERISTICS OF THE ZPK CONTROLLERS FOR $G_2(S)$

Model of controller	Fitness (ITAE)	Setting* time (±2 %)	Rise Time (90 %)	Overshoot
GAPID	92.7	34.84 s (35.12, 2.29)	5.78 s	0.042
GAR3C	19.2	9.61 s (10.84, 39.88)	1.03 s	0.615
GAR5	18.4	9.04 s (9.73, 4.71)	1.13 s	0.632
GAR5C	15.9	7.90 s (9.23, 1.42)	1.03 s	0.644

* *There are also median value and standard deviation in the brackets.*

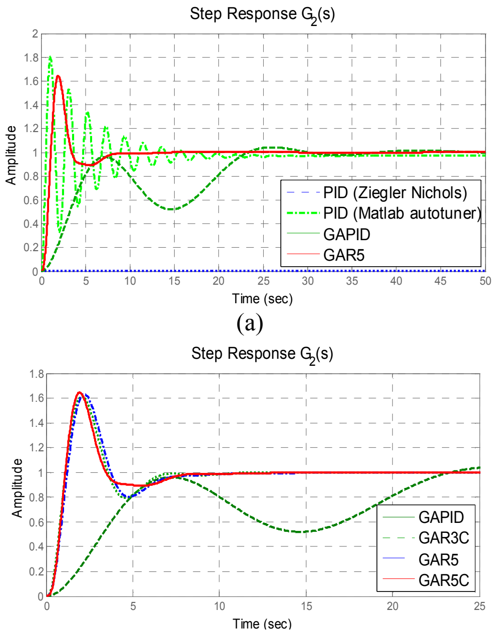


Fig. 12 Unit step response of close loop system $G_2(s)$ and GARx models (GAR3C, GAR5 and GAR5C) of polynomial controllers. The comparison between GAR5 model of polynomial controller and classical PID controller solutions (above). Detail of unit step response characteristics for all presented GARx models (bottom)

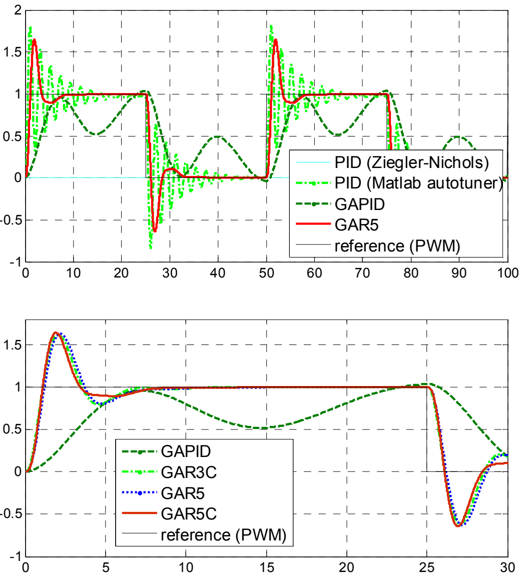


Fig. 13 PWM response of close loop system $G_2(s)$ and GARx models (GAR3C, GAR5 and GAR5C) of polynomial controllers. The comparison between GAR3C and GAR5C show approximately similarity of the results. This figures clearly display difficulty of control design for standard PID controller and advanced solution obtained using polynomial controllers

VI. Conclusion

This paper clearly shows the new design method in case of polynomial controller. The method is based on the special encoding scheme of the GA. Optimal polynomial controller design is possible by means of this method. In our case we used the ITAE integral criteria and numerical solutions using genetic algorithms. Many future modifications are possible [6]. The presented solution of the polynomial controllers is shown as examples in (16) and (18) for $G_1(s)$ system a and in (17) and (19) for $G_2(s)$ system. As one can see, the structural changes was made by this proposed method. From our results (Table I., Table II.) we can derive that the GAR5C is the best. On the other side the price for the using complex roots in case of GAR3C and GAR5C can be very high considering deviation of the solution and degree of the searched polynomials. The best image about the results and the comparison with standard PID controller is presented in the Fig. 10 to Fig. 13.

$$C_{GAR3C+,G_1}(s) = \frac{16.81s^4 + 47.38s^3 + 41.03s^2 + 10.79s + 0.89}{s^4 + 4.51s^3 + 7.68s^2 + 8.69s} \quad (16)$$

$$C_{GAR3C+,G_2}(s) = \frac{0.000317}{s^5 + 1.006s^4 + 0.416s^3 + 0.097s^2 + 0.012s} \quad (17)$$

$$C_{GAR5C+,G_1}(s) = \frac{13s^5 + 91s^4 + 245s^3 + 194s^2 + 48s + 4}{s^5 + 6s^4 + 24s^3 + 39s^2 + 40s} \quad (18)$$

$$C_{GAR5C+,G_2}(s) = \frac{284s^5 + 676s^4 + 608s^3 + 257s^2 + 51s + 4}{s^5 + 8s^4 + 21s^3 + 20s^2 + 2s} \quad (19)$$

References

[1] Aström, K. J., Bernardson,, B., Ringdhal, A.: Solution using robust adaptive pole placement. In: Proc. European Control Conference (ECC'91), Grenoble 1991, pp. 1341–2346

[2] Aström, K. J., Wittenmark, B.: Computer Controlled Systems, Theory and Design. Prentice Hall, Englewood Cliffs, N.J. 1990

[3] D Goldberg, "Genetic Algorithm. in Search, Optimization and Machine Learning", Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[4] R. Grepl. J. Vejlupek. V. Lambersky. M. Jasansky. F. Vadlejch. P. Coupek. P. "Development of 4WS/4WD Experimental Vehicle: platform for research and education in mechatronics". IEEE International Conference on Mechatronics. ICM 2011-13-15, 2011.

[5] T. Gusner, J. Adamy, "Controller Design for polynomial Systems with Input Constraints", In *Joint 48th IEEE Conference on Decision Control and 28th Chinese Control Conference*, Schanghai, P.R.China, 2009.

[6] Z. Oplatkova, R. Senkerik, I. Zelinka, "Synthesis of Control Rule for Synthesized Chaotic System by means of Evolutionary Techniques", in 16th Int. Conference on Soft Computing, pp.91-98, Brno, 2010.

[7] R. Matousek, "GAHC, Improved Genetic Algorithm", in the Springer book series (Eds.: Krasnogor, et al.) Nature Inspired Cooperative Strategies for Optimization (NICSO 2007), Volume 129, 2008, XIV, pp. 114-125., ISSN 1860-949X, Springer Berlin, 2008.

[8] P. Pivonka, V. Veleba, M. Seda, P. Osmera, R. Matousek, "The short Sampling Period in Adaptive Control", in Proc. of the IAENG Int. conference WCECS 2009, pp.724-729, San Francisco, 2009.