Journal of Cybernetics and Informatics

published by

Slovak Society for
Cybernetics and Informatics

Volume 6, 2006

http://www.sski.sk/casopis/index.php (home page)

ISSN: 1336-4774

Journal of Cybernetics and Informatics 6 (2006)
http://www.sski.sk/casopis/index.php

THE GENETIC ALGORITHMS FOR SPEED CONTROLLER DESIGN OF INDUCTION MOTOR DRIVE

Marián Jančovič and Milan Žalman

Slovak University of Technology, Faculty of Electrical Engineering and Information Technology
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
e-mail: milan.zalman@ stuba.sk

Abstract: This paper aims at demonstrating possibilities of genetic algorithms application in speed controller design for a vector control of induction motor. The basic procedure to develop a genetic algorithm is described and the examples of its application using different quality criterions are introduced. Then a possibility to consider various disturbances in calculating optimal controller parameters, e.g. change of load torque is analysed. Finally, a possibility to use genetic algorithm for a design of parameters of IP speed controller is described.

Keywords: genetic algorithm, induction motor, vector control, speed control

1 INTRODUCTION

Genetic algorithms present universal optimization methods, which use global stochastic search algorithms. Owing to their universality these algorithms can be used to solve wide scope of optimization problems. One of them is the choice of optimal controller parameters in control structures of any complexity. In these cases an objective function consists of two parts. Firstly, it is necessary to simulate dynamical system and then calculate the value of appropriate quality criterion. Minimization of this quality criterion yields required system performance. The system complexity and number of searched parameters influence the number and time demands on simulation of dynamical processes. These properties also causes that genetic algorithms are significantly time demanding, and therefore a solution sometimes cannot be obtained in a reasonable time horizon. However, usually suboptimal solution is sufficient for the given control aim.

In comparison with conventional optimization approaches, the genetic algorithms are advantageous in that they enable to check a lot of control structures in one design cycle. Final controller design is then made based on a choice of various structures and parameter values. Another advantage is that genetic algorithm is applied directly on resulting simulation model and avoids introducing further simplifications required in classical design methods which at the same time introduce other inaccuracies into the system. Genetic algorithms enable to consider influence of various disturbances, noise and unfavourable events, which are taken into account in parameter evolution.

Stability of the solution follows implicitly from minimization of the quality criterion, since unstable solutions achieve extreme values of the quality criterion and are removed during evolution. Besides that, an explicit stability test can be included into the quality criterion if necessary.

A possibility to use GA for vector control of induction motor (IM) will be shown on optimal choice of speed controller parameters.

2 SPEED SERVO SYSTEM OF INDUCTION MOTOR IN THE DIRECT VECTOR CONTROL

In the next section the speed controller design in direct vector control of IM is analysed. Direct vector control uses feedback control of torque, modulus of vector of a rotor magnetic flux and components of vector of a stator current [1]. This method is robust to parameter changes and high quality dynamics is inherent. Considering the information on rotor magnetic flux the concepts with direct and indirect rotor field-oriented control can be distinguished. The most frequent are concepts based on evaluation of the rotor magnetic rotor flux and torque from the model of induction motor using observers. The direct vector control is more robust than the indirect vector control but its performance depends on the type of flux observer used. The block diagram of the torque generator (see Fig. 1) contains torque controller C_T, flux controller C_Ψ and linear controllers C_P for torque i_{S2} and flux i_{S1} components of current.

The classical structure of speed servo system is speed control with PID controller shown in the fig. 2. The speed controller output is the required torque T_m^*, which inputs into the described structure of the torque generator – TG in Fig. 1.

The other types of controllers which can be used in the speed structures is IPD speed controller [2]. The advantage of the control loop with this controller is in that the speed control closed loop transfer function has no zeros. Control response performance can be tuned using the only parameter – band-pass. Fig. 3 shows speed control loop using IP controller with state feedback from T_m. This feedback shifts a real pole of the speed control loop to the left and in this way damps the influence of oscillations due to complex conjugate poles and increases the band-pass.

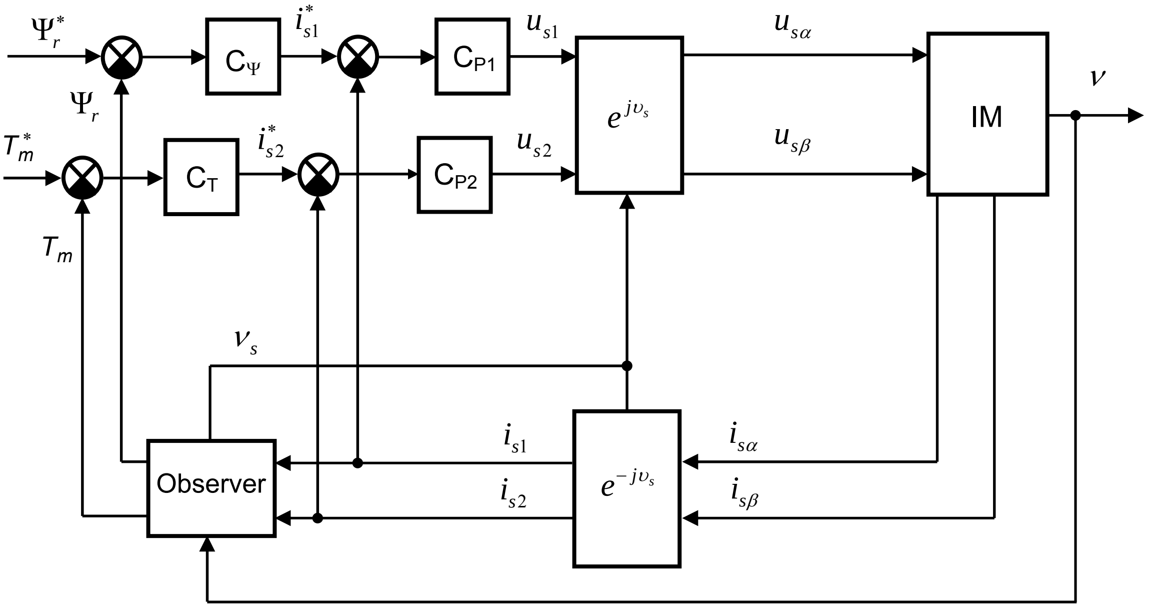


Fig. 1 Block diagram of the torque generator with direct vector control

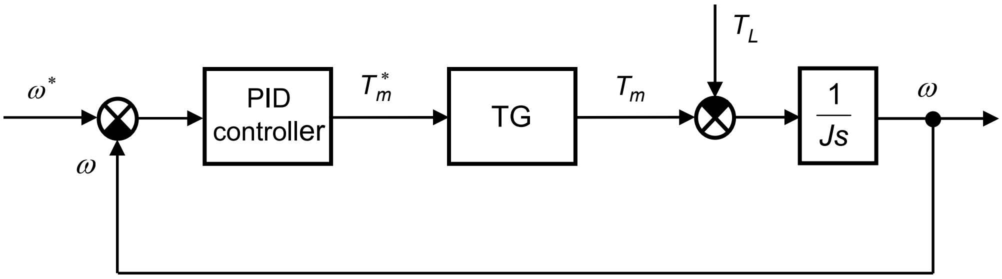


Fig. 2 Closed-loop speed control system with PID controller

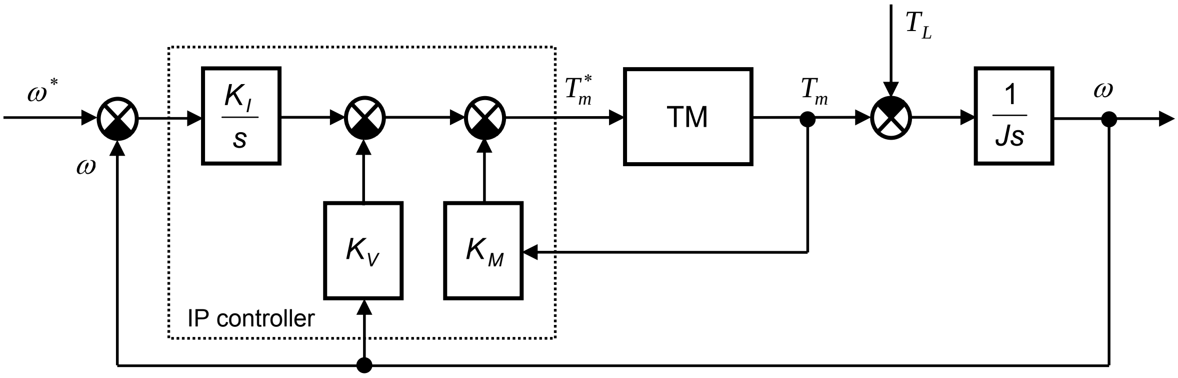


Fig. 3 Speed control loop using IP controller with state feedback from T_m

3 SPEED CONTROLLER DESIGN USING GENETIC ALGORITHMS

Genetic algorithm can be used to obtain controller parameters aiming at certain control design goal [3, 4, 5]. Successive generations of searched solutions are tested on controlled model and then the genetic operations are applied to generate new generations of solution which provide better and better qualities. After a fixed number of generations the best solution can be recorded and the appropriate candidate can be chosen.

In classical PID control structures three parameters – "genes" - are demanded – known as proportional gain (K_p), integration gain (K_i) and derivation gain (K_d). In our case we will search parameters of a discrete-time PS controller $G_C(z)$:

$$G_C(z) = K_P\left(1 + \frac{T}{T_I}\frac{1}{z-1}\right) \tag{1}$$

where K_P is proportional gain, T_I is integral time constant and T is sampling period.

The initial decision in the design of controller parameters using GA is how they are represented in a chromosome. Since we search two PS controller parameters, the values of these parameters have to be coded in chromosome representation. Chromosome is then in the form:

$$r = (K_P, T_I) \tag{2}$$

In the case of IP controllers (Fig. 3) three parameters should be designed. These parameters create the chromosome of a genetic algorithm:

$$r = \left(K_I, K_V, K_M\right) \tag{3}$$

To represent these parameters we choose real number values. In using real number code in comparison with binary code the procedure of the respective solution is more stable, since the values of real numbers change continuously, proportionally to the required value of change.

The next task is constraining the search space of solution, i.e. setting the feasible values intervals for each gene of the string. In the case of PS controller the space is represent by intervals ($K_{P\min}$; $K_{P\max}$), ($T_{\mathrm{Im}in}$; $T_{\mathrm{Im}ax}$). For IP controller the space of solutions is determined by the intervals ($K_{\mathrm{Im}in}$; $K_{\mathrm{Im}ax}$), ($K_{V\min}$; $K_{V\max}$), ($K_{M\min}$; $K_{M\max}$). The narrower the search space the quicker the solution.

Applying genetic operators a genetic algorithm provides candidates for controller parameters. Each candidate represents controller parameters. The population size can depend on particular case. In most cases it is recommended to choose the size between 10 and 100, most frequently between 20 and 50. Small population does not provide enough space for diversity of genetic information, too big population does not provide better effect and the solution is much longer.

The aim of objective function is to test each candidate and to evaluate its fitness values, based on the evaluation of the process performance respecting to its controller parameters (K_P, T_I). To evaluate the qualities of the designed controller the simulation of control loop is applied. Then the objective function consists of two steps. The first one is the implementation of parameters into controller model and the following simulation of the respective controlled system. After a step change of input signal, the system response is recorded in each sampling period, this response is then returned into the quality criterion as a vector with instantaneous response values. The next step is the calculation of the appropriate criterion to evaluate the qualities of the given controller (Fig. 4).

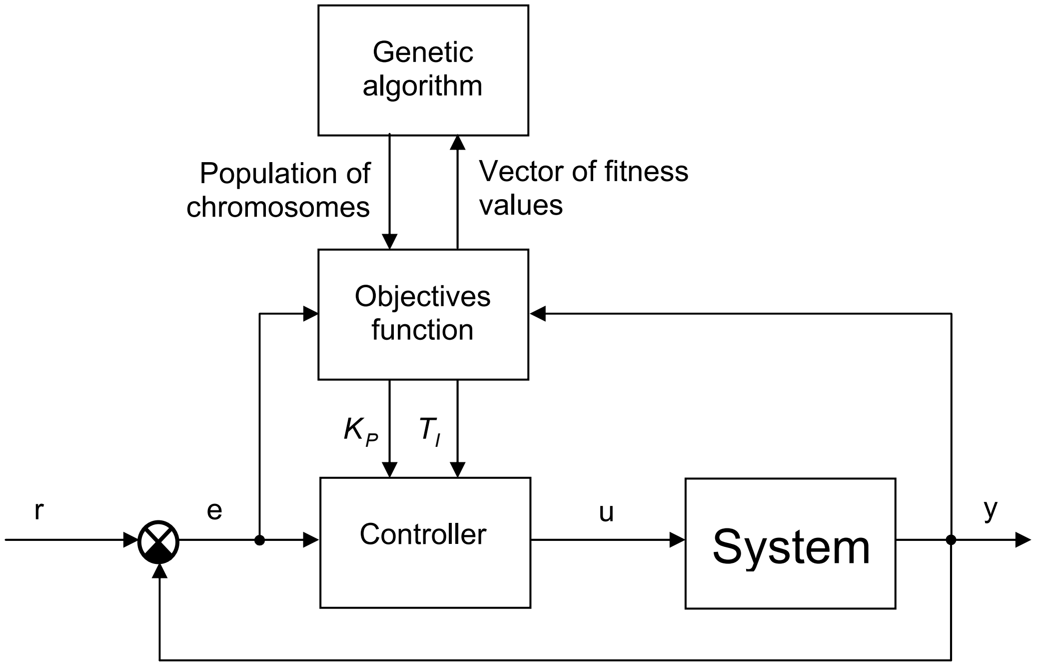


Fig. 4 Design of PS controller using genetic algorithm

The quality criterion uses the process response to calculate the error function for each member of population and returns the vector including the fitness values for each member of population. The required goal is to minimize the process error depending as well on the qualities of the used quality criterion. The genetic algorithm uses performance values to evaluate fitness for the population and then the genetic operators are applied on chromosomes of a new population created by stronger individuals. This procedure is repeated for the final number of generations.

In the case of control design the integral quality criterions can be used as the absolute error area (IAE - integral of absolute errors):

$$J_{IAE} = \int_{0_1}^{T} |e(t)| dt = \int_{0}^{T} |w(t) - y(t)| dt \qquad (4)$$

where e is the control error and T is the evaluated time interval. This criterion yields fast control responses with some small overshoots. If necessary to damp the overshoot or the oscillations, the terms comprising absolute error derivation of the first or second order are included into integral according to the following formulae:

$$J = \int_{0}^{T} \alpha|e| + \beta|e'| + \gamma|e''| dt \qquad (5)$$

where α, β, γ are weight coefficients. Control error derivates can be substituted by output derivates. Increasing values of β, γ with respect to α the overshoot and oscillations are damped.

Another criterion which can be used in the design of controller parameters is the one minimizing overshoot and settling time:

$$J = \alpha\eta + (1-\alpha)t_s \qquad (6)$$

where η is maximal overshoot, t_s is the settling time and α is the weight coefficient from the interval (0; 1). Settling time is the time when a step response for the last time enters the $\pm 5\%$ band around the required value.

4 DESIGN OF ROBUST SPEED CONTROLLER

In the robust controller design various approaches can be adopted. One possible way is to implement into simulation model a change of parameter in certain time instant. In this case the procedure to choose parameter values is the same as in previous sections. The only difference is in simulation model.

Another way to design a robust controller uses several chosen representative working points of the considered system, where an individual control loop model is created for each perturbed system. Controller parameters are common for all cases. Controller design procedure is the same as for the controller without a change of parameter; the only difference is that simulation is repeated several times for each working point. Quality criterion is the sum of particular quality criterions respective to working points:

$$J = \int_0^T \sum_{i=1}^{N} \delta_i J_i dt \tag{7}$$

where *N* is the number of specified working points and coefficients δ_i can used to weight the importance of individual working points.

5 SIMULATION RESULTS

To verify our proposed genetic algorithm the Matlab/Simulink environment is used. Parameters of IM are listed at the end of the paper.

For PS controller design we defined the feasible values intervals. Since controller parameters are positive numbers, the lower limit is set to zero. The upper limit for controller gain can be set according to its critical gain, when the oscillations appear. In our case the gain cannot be bigger than 1. This yields that the feasible values interval for the gain is ($K_{P\min}$; $K_{P\max}$) = (0; 1). For integration time constant the interval is set to ($T_{\mathrm{Im}in}$; $T_{\mathrm{Im}ax}$) = (0; 10000), since it appears in denominator and bigger values provide negligible numbers.

The GA procedure can be described as follows:

1. Initially, a population of 20 chromosomes generated at random in the given interval.
2. Successively, all 20 chromosomes are applied into the simulation model, where after motor exciting, in time 0.3 second a step change of required speed from 0 to 100 rads^{-1} is realized. After simulation taking *1* second a value of chosen criterion is evaluated. The GA aims at its minimization.
3. The chromosomes with the smallest and second smallest fitness value are copied into a new population.
4. We copy into the working group: three times the chromosome with the smallest fitness value, twice the chromosome with the second smallest and once the chromosome with the third smallest fitness value.
5. 12 chromosomes are randomly copied into the working group.
6. The crossover operator is applied on the working group, where the choice of parents for the crossover is random.
7. The multiplicative mutation operator is applied on the working group, where the value of the chosen gene is multiplied by a random number from the range (0, 2), with a mutation probability 0.5.
8. Then, the work group is moved to the new generation.
9. The algorithm is repeated with the next generation population until the 200 generations is reached. Then the best string is assessed as the best solution.

We tested application of various performance criterions in GA. Using quality criterion (4) the following PS speed controller is obtained:

$$G_C(z) = 0.3687\left(1 + \frac{1}{40.1290}\frac{T}{z-1}\right)$$

This criterion leads to fast control responses with some small overshoots, which can be seen from the response of the speed in Fig. 5 (dotted line).

To improve damping oscillation and overshoot the quality criterion (5) was used, including the derivative of control error in the integral as well. Then for the chosen weight coefficients $\alpha = 1$, $\beta = 0.5$ and $\gamma = 0$ we obtained the resulting speed controller:

$$G_C(z) = 0.1608\left(1 + \frac{1}{137.7576}\frac{T}{z-1}\right)$$

From the step response of the speed in Fig. 5 (dashed line) it can be seen that the appropriate ratio $\alpha : \beta$ provides sufficient damping the oscillation of the response without the derivative of the control error of the second order.

Finally, the use of criterion (6) was tested, using this criterion the overshoot and settling time can be minimized. With chosen coefficient $\alpha = 0.1$ the following speed controller was obtained:

$$G_C(z) = 0.1734\left(1 + \frac{1}{684.0619}\frac{T}{z-1}\right)$$

In the Fig. 5 (solid line) it can be seen that the obtained response of the speed is similar to the one with previous criterion.

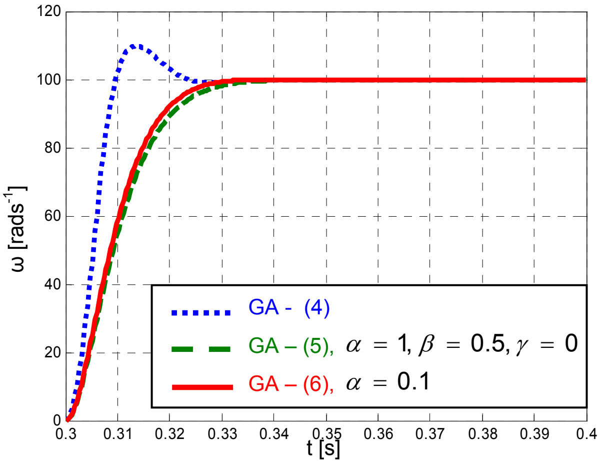


Fig. 5 Comparison of speed responses for different quality criterions

Comparison of speed responses in Fig 5 shows that using quality criterion (4) yields a quick response with 10% overshoot which is undesirable in speed structures. Using criterions (5) or (6) and appropriate tuning of their weighting coefficients enables to decrease this overshoot, which on the other hand gives slower control response. The final speed controller parameters were optimized for ideal conditions. Therefore the proposed structures do not provide good response to disturbances as the change of load torque. Such influences should be considered in the controller design by their implementing into the simulation model.

One possible way is to implement into simulation model a step change of load torque in certain time instant. In this case the procedure to choose parameter values is the same as in previous case. The only difference is in simulation model. In our test we have implemented into the model a step change of load torque from 0 to 1 Nm in time 1 s, where simulation time is prolonged to 2 s. Then for criterion (5) with coefficients $\alpha = 1$, $\beta = 0.1$ a $\gamma = 0.0005$ we have obtained the resulting speed controller:

$$G_C(z) = 0.1205\left(1 + \frac{1}{0.1285}\frac{T}{z-1}\right)$$

The speed response in Fig. 6 shows a good reaction to the change of load torque. Motor torque response is shown in Fig. 7.

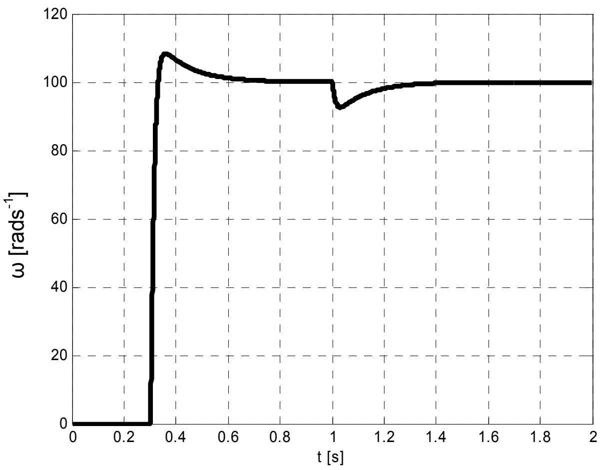


Fig. 6 Step response of speed using criterion (5) for $\alpha = 1$, $\beta = 0.1$ and $\gamma = 0.0005$

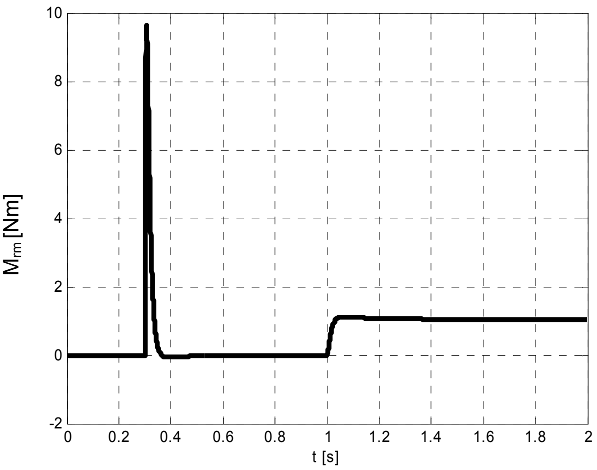


Fig. 7 Motor torque response using criterion (5) for $\alpha = 1$, $\beta = 0.1$ and $\gamma = 0.0005$

Another way to design a robust controller uses several chosen representative working points of the considered system, where an individual control loop model is created for each perturbed system. Controller parameters are common for all cases. Controller design procedure is the same as for the controller without a change of parameter; the only difference is that simulation is repeated several times for each working point. Quality criterion is the sum of particular quality criterions respective to working points (7).

We have considered quality criterion (5) for two working points. The first one corresponds to simulation of the loop under normal conditions and the second simulation includes additional load torque 1 Nm. Then, the fitness function is:

$$J = \int_0^T \left(\delta_1\left(\alpha|e| + \beta|e'| + \gamma|e''|\right) + \delta_2\left(\alpha|e| + \beta|e'| + \gamma|e''|\right)\right)dt \tag{8}$$

where $\delta_1 = 1$ is coefficient for the calculation of the first simulation, $\delta_2 = 1$ is coefficient for the calculation of the second simulation and $\alpha = 1$, $\beta = 0.1$ a $\gamma = 0$. In this case the following speed controller has been designed:

$$G_C(z) = 0.2082\left(1 + \frac{1}{0.2078}\frac{T}{z-1}\right)$$

The respective speed response in Fig. 8 shows the response to the change of the load torque of 1 Nm in the time 1s. Motor torque response is shown in Fig. 9.

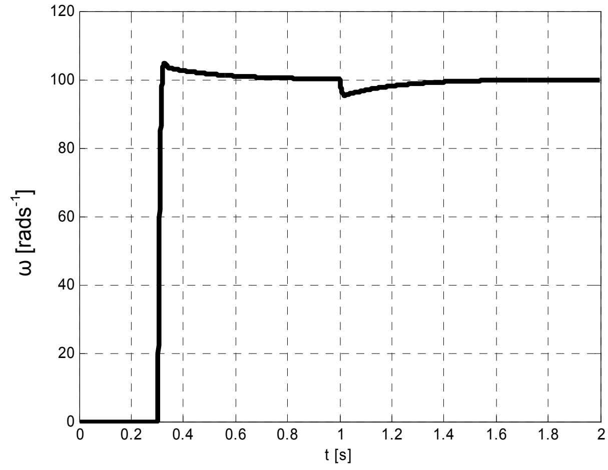


Fig. 8 Step response of speed using (5) for two working points

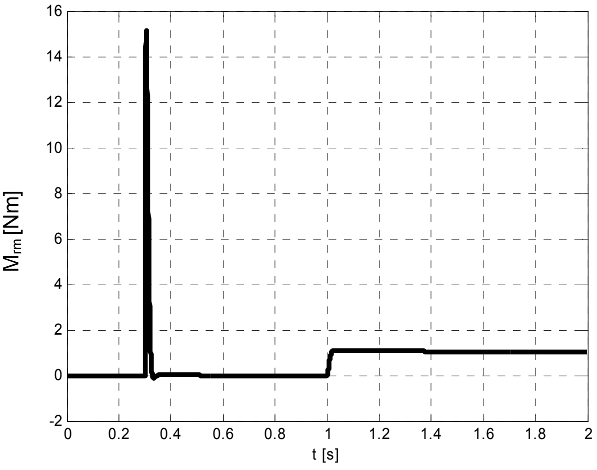


Fig. 9 Motor torque response using criterion (5) for two working points

Genetic algorithm includes a possibility to choose a control structure for noisy signals as well. In this case the influence of a noise can be included into the simulation model. The evolution procedure remains the same as was given in the first speed controller design. We have used a white noise with the gain 1.10^{-6} as a noise source in the simulation model. Using performance criterion (5) with weighting coefficients $\alpha = 1$, $\beta = 0.1$ a $\gamma = 0$, the following speed controller has been designed:

$$G_C(z) = 0.1243\left(1 + \frac{1}{27.0004}\frac{T}{z-1}\right)$$

The respective speed response in Fig. 10 shows the closed loop response to the change of the required speed from 0 to 10 rads^{-1} in the time 1s.

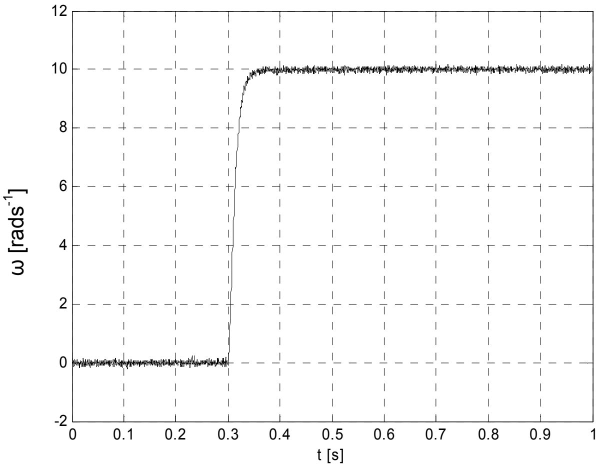
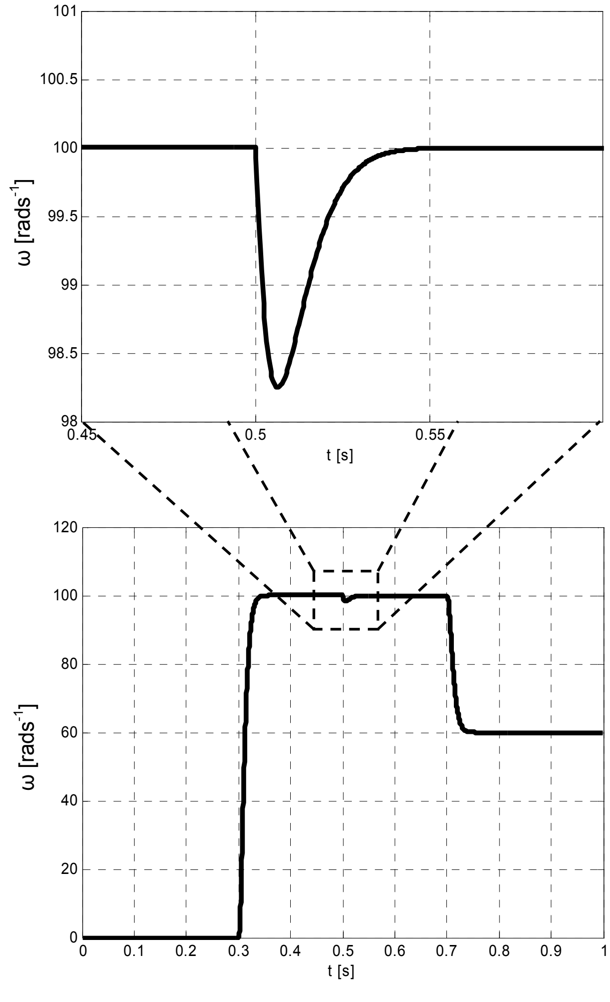


Fig. 10 Step response of speed considering a noise in evolution

A genetic algorithm can be used in searching parameters also for the IP speed controller (Fig. 3). In this case we determined the search space of solutions by the following feasible intervals of values for each gene of the string: ($K_{\mathrm{Im}in}$; $K_{\mathrm{Im}ax}$) = (0; 10000), ($K_{V\min}$; $K_{V\max}$) = (0; 10000),

($K_{M\min}$; $K_{M\max}$) = (0; 10000). To find the optimal solution the same genetic algorithm is used as in the case of PS speed controller, the difference is that instead of one point crossover in this case the two point crossover is used, since the chromosome includes three elements instead of two ones.

Using performance criterion (5) with weighting coefficients $\alpha = 1$, $\beta = 0.5$ and $\gamma = 0.0001$ the IP speed controller with parameters: $K_I = 135.0583$, $K_V = 1.7064$, $K_M = 2.7331$, has been obtained. Figures 11 and 12 shows speed and torque responses to the change $\omega^* = 100 rads^{-1}$ in the time $t = 0.3s$ and speed reverse $\omega^* = 50 rads^{-1}$ in the time $t = 0.7s$. In the time $t = 0.5s$ the load torque step change is realized $M_z = 1Nm$. The figures shows that the proposed IP speed control loop provides very good system response to the load torque change even without considering this disturbance in optimal controller parameters design.



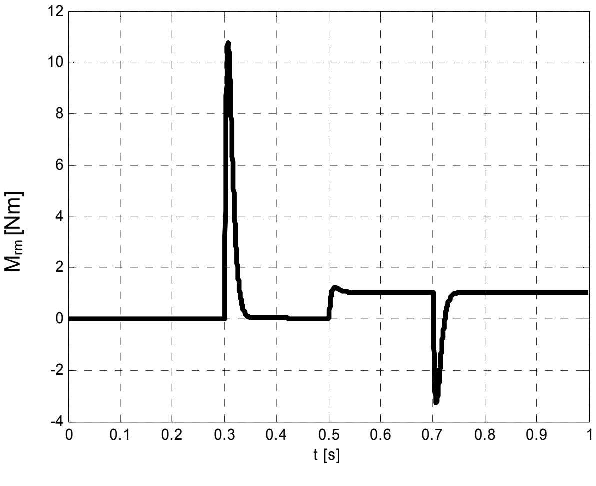


Fig. 11 Speed response using IP controller for $\omega^* = 100 rads^{-1}$ in the connecting time $t = 0.3s$, step change $M_z = 1Nm$ in time $t = 0.5s$ and $\omega^* = 50 rads^{-1}$ in connecting time $t = 0.7s$

Fig. 12 Motor torque response using IP controller for $\omega^* = 100 rads^{-1}$ in the connecting time $t = 0.3s$, step change $M_z = 1Nm$ in time $t = 0.5s$ and $\omega^* = 50 rads^{-1}$ in connecting time $t = 0.7s$

Besides the load torque the speed response is influenced by a change of moment of inertia as well. The influence of 50% increase and decrease of moment of inertia on the IP control is shown in Fig. 13 and 14. The change of moment of inertia influences in transient time the value of motor torque as well as the value of speed overshoot.

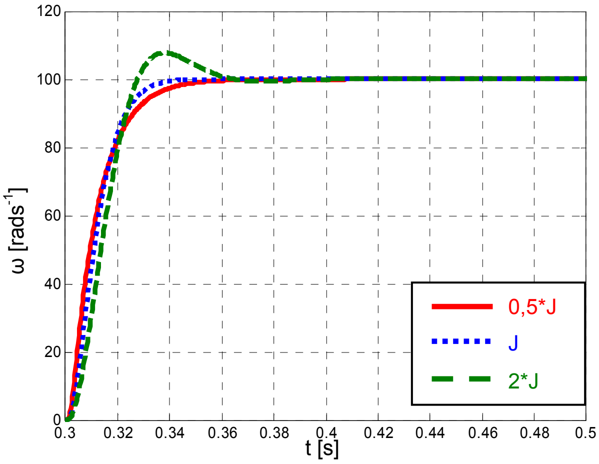


Fig. 13 Speed responses of the closed loop with IP controller for various values of moment of inertia

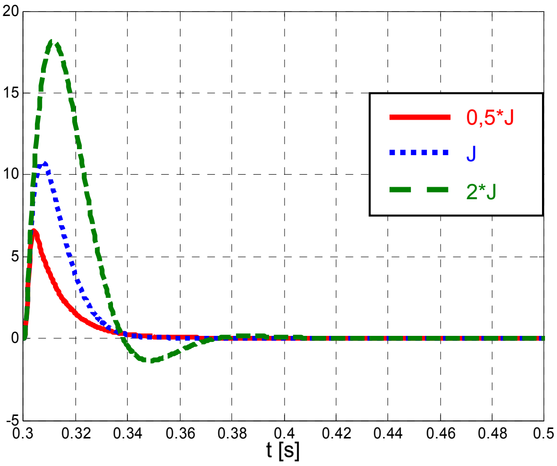


Fig. 14 Motor torque responses of the closed loop with IP controller for various values of moment of inertia

To make the simulation model as close as possible to the real device, the speed can be evaluated by speed estimator of IRC sensor (Fig. 15). Fig. 16 shows that the proposed IP controller provides the response of good quality even with this deterioration of output signal.

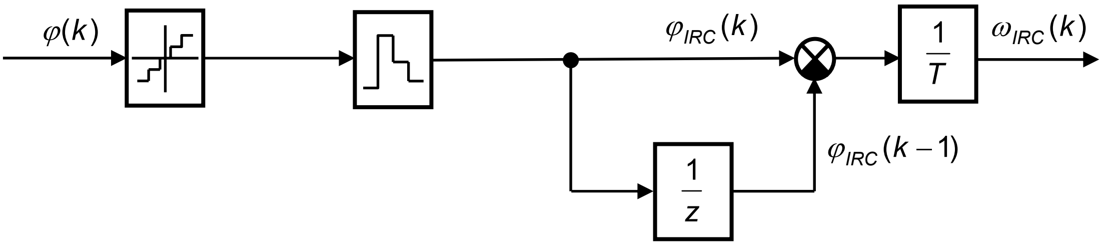


Fig. 15 Model of speed estimator of IRC sensor

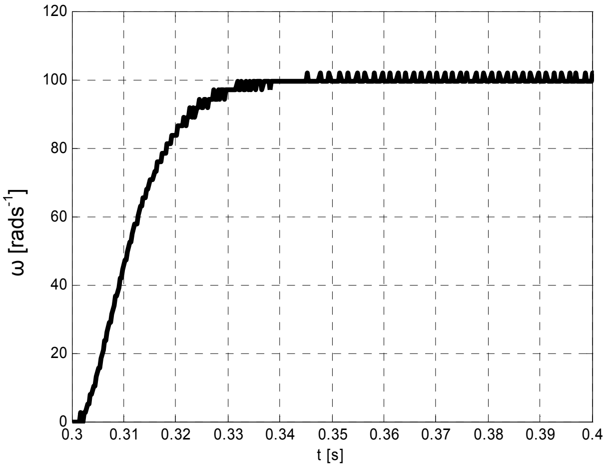


Fig. 16 Speed responses of the loop with IP controller and speed estimator of IRC sensor

Speed estimator of IRC sensor can be considered in the control loop as soon as the controller parameters are under the design. In this case the inaccuracy of output value influences the resulting form of the controller. Using criterion (5) with weighting coefficients $\alpha = 1$, $\beta = 0.5$ and $\gamma = 0$ we obtained the IP speed controller with parameters: $K_I = 210.1975$, $K_V = 4.4785$, $K_M = 4.5311$. The respective speed response in Fig. 17 shows that the genetic algorithm can find a solution even in these worsened conditions, though the results are a bit inferior in comparison with the previous case.

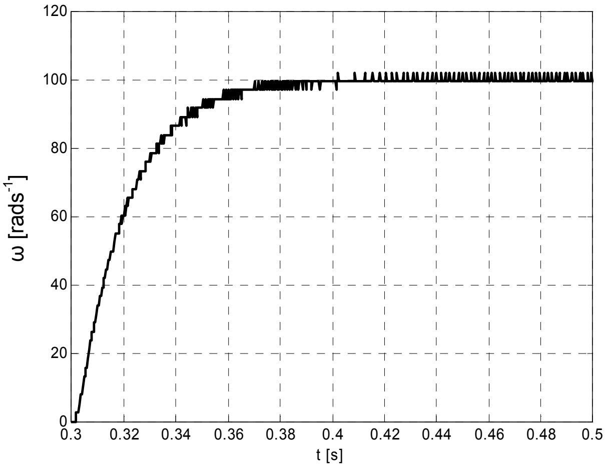


Fig. 17 Speed responses of the loop with IP controller and speed estimator of IRC sensor

To verify the resulting speed controllers design using the genetic algorithm, the obtained results are compared with responses of a closed loop with a one with a controller designed by classical method. One of the appropriate classical methods to design the IM speed controller is the standard form of Whiteley. Applying this method we have obtained the following PS speed controller:

$$G_C(z) = 0.1373\left(1 + \frac{1}{0.0964}\frac{T}{z-1}\right)$$

This controller is compared with PS controller robust with respect to the load change and IP speed controller, both designed using genetic algorithm.

In Fig. 18 and 19 the speed and motor torque responses are shown for a step change $\omega^* = 100 rads^{-1}$ in time $t = 0.3s$ and change of load torque $M_z = 1Nm$ in time $t = 1s$. The responses show that the genetic algorithm provides similar results as the Whiteley standard form method (WSF). However, the best response has been obtained with IP controller designed by the proposed genetic algorithm.

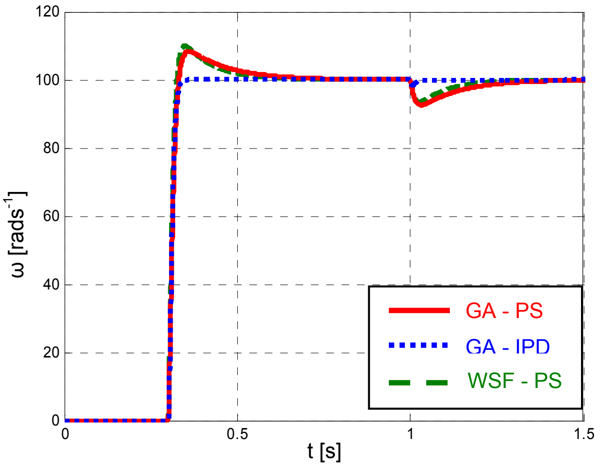


Fig. 18 Comparison of speed responses for $\omega^* = 100 rads^{-1}$ in connecting time $t = 0.3s$ and step change $M_z = 1Nm$ in time $t = 1s$

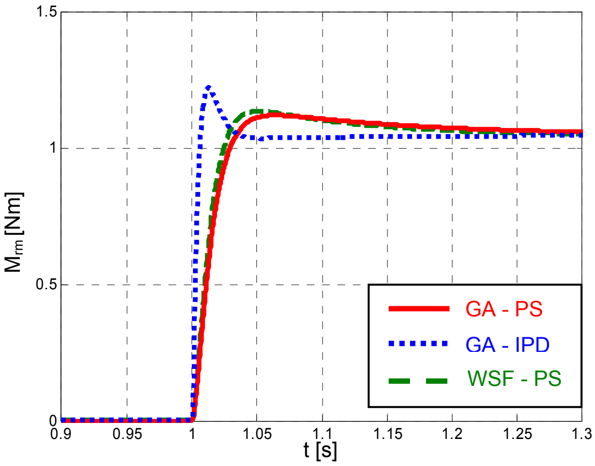


Fig. 19 Comparison of the motor torque responses for the step change $M_z = 1Nm$ in time $t = 1s$

6 CONCLUSION

The results of our experiment show that genetic algorithms can contribute to the methods for a speed controller design for vector control of induction motor. The result can be influenced by an appropriate choice of the performance criterion. One of the drawbacks of the genetic algorithms is their time demand that follows mainly from a number of required control loop simulations.

The advantage of genetic algorithms is that various changes of motor parameters can be considered as well as various constraints, disturbances and noises. These changes can be included in the objective function in simulations and so the solution that meets given requirements can be obtained. In this way the controller can be optimized e.g. for various values of load torque. The overall quality criterion is in this case equal to the sum of partial quality criterion in particular working points. However, the prolongation of the solution time must be taken into account in this case.

Genetic algorithms can be used to find optimal parameters for various speed controller structures not only for a classical PID controller. It can be e.g. IPD structure, which provides very good control loop qualities for load torque changes even without the necessity to consider this change in controller parameter design.

Genetic algorithms can be used also in on-line adaptation of control parameters. However, in this case the high computational demands bring certain limitation.

PARAMETERS OF IM

$P_n = 1.1kW$, $n = 2840 \min^{-1}$, $J = 0.0017 kg.m^{-2}$, $R_s = 7.608\Omega$, $R_r = 3.700\Omega$, $p' = 1$, $L_s = 0.6015H$, $L_r = 0.6015H$, $L_m = 0.5796H$, $T_r = 0.1626s$, $\sigma = 0.0715$

REFERENCES

[1] Žalman, M., Jovankovič, J.: Sytéza regulátorov vektorovo riadeného asynchrónneho motora metódou inverznej dynamiky. AT&P Journal, 2, 1998, 70–73.

[2] Žalman, M., Jovankovič, J., Uriček, M.: Master-Slave riadenie polohových servopohonov. AT&P Journal, 2, 3, 4, 2005.

[3] Sekaj, I.: Riešenie problémov pomocou genetických algoritmov. Automatizace, 9, 2004, 552-554.

[4] Sekaj, I.: Genetické algoritmy pri návrhu regulátorov a statickej optimalizácii procesov. Automatizace, 10, 2004, 617-620.

[5] Sekaj, I.: Genetic Algorithms for Control System Design Applications, SSKI, 1, 2003, 31–37.