Number 2, Volume VII, July 2012

POSSIBILITY OF SMART CAR SPEED CONTROL USING SOFT COMPUTING

Petr Doležel[1] Ivan Taufer[2]

Summary: Smart speed control of a car is the focus of the paper. There is introduced the possibility how to automatically control the car speed smoothly in the daily traffic. The technique uses nonlinear neural car model together with differential evolution search technique to determine continuously convenient throttle power so that the car speed is optimal with respect to defined cost function – the cost function can consider actual speed limits as well as speed smoothness.

Key words: Speed Control, Smart Car, Differential Evolution.

PREFACE

These days, a car is developing to operate more and more automatically. It collects many new achievements of academic subjects such as automatic control, sensor technologies, artificial intelligence, image processing etc. One of potential car improvements is the possibility of smart car speed control. The classical cruise control (sometimes known as autocruise) developed by Frank J. Riley (1) can be replaced by more sophisticated control system respecting full car dynamics including driving position engaged as well as actual speed limit. The method is fully described in next sections.

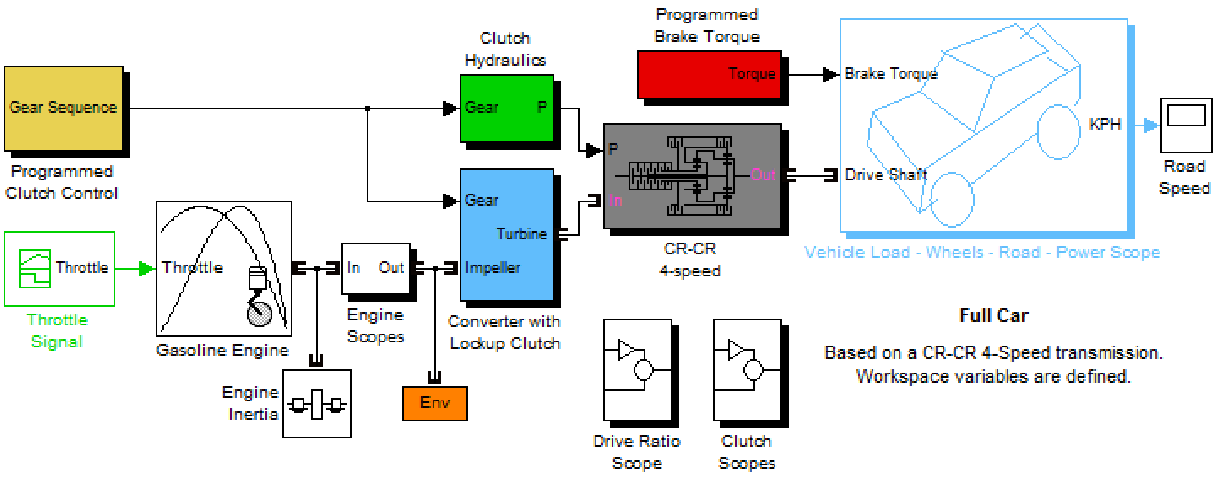
1. CAR MODEL

Instead of real car, smart car speed control is simulated here on using model with automatic transmission. This model is considered as single input-single output (SISO) plant with throttle pedal press as input and speed as output.

Matlab-Simulink full car model is chosen for control simulation. This model serves as SimDriveline library features demonstration and it includes engine and transmission models and a simplified model of the drivetrain-wheel-road coupling. Simulink scheme of the model is depicted in Fig. 1.

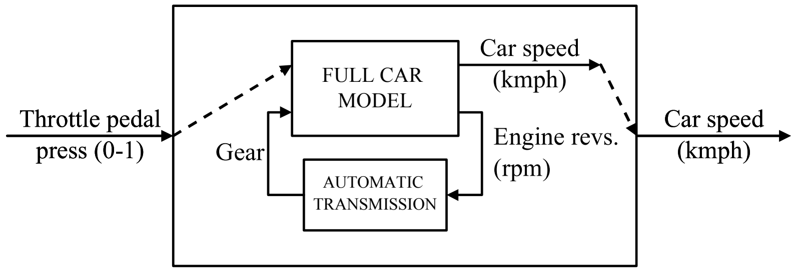
The model is improved on automatic transmission working this way: lower gear is engaged in case of less than 2000 engine revolutions per minute (rpm) and higher gear is engaged in case of more than 4000 rpm. Brakes usage is not considered in case of smooth drive – see Fig. 2

[1] Ing. Petr Doležel, Ph.D, University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control, Nám. Čs. legii 565, 532 10 Pardubice, Tel.: +420 466 037 504, E-mail: petr.dolezel@upce.cz
[2] Prof. Ing. Ivan Taufer, DrSc., University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control, Nám. Čs. legii 565, 532 10 Pardubice, Tel.: +420 466 037 123, E-mail: ivan.taufer@upce.cz



Source: Matlab

Fig. 1 - Full car model



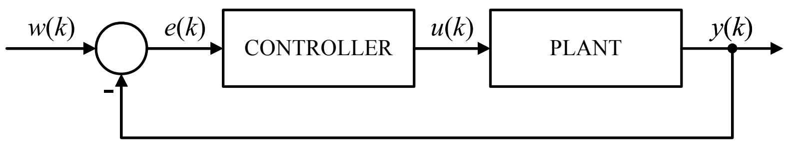
Source: Author

Fig. 2 - Car model for simulations

2. CONTROL METHOD

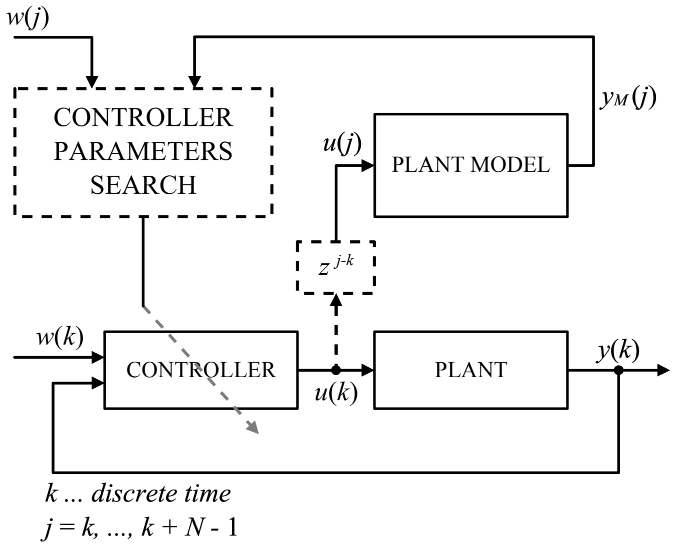
If advanced automatic control is required, it is always necessary to design suitable controlled plant model. One possibility is to use artificial neural network, because it represents effective tool for even highly nonlinear plants modelling. However, possibilities of neural model usage in process control are limited because control techniques in use cannot employ neural models.

On the other hand, the most classical and trustworthy control techniques are based on PID controllers, which are the widest spread controllers in industry (more than 95% of the control loops are PID type – see (2)). PID controllers are to be tuned to operate properly and there are many techniques to tune them, either theoretically based or experimental ones. However, all of them suppose linear controlled system. The method explained here aims to tune any discrete controller online (not only PID like). It expects knowledge of controlled system model (it shall use the neural model) and reference variable course over known future finite horizon. The method amplifies the basic feedback control loop connection illustrated in Fig. 3. Its structure is illustrated in Fig. 4, where $w(k)$, $u(k)$, $y(k)$ are reference variable, manipulated variable and controlled variable, respectively.



Source: Author

Fig. 3 – Feedback control loop



Source: Author

Fig. 4 – Feedback control loop with self-tuning discrete controller

So the premise is an availability of controlled system neural model and knowledge of reference variable course over future horizon *N*. Then there are chosen the parameters of any discrete controller repeatedly every discrete time instant so that the control response computed via the neural model over future horizon is optimal (according to chosen performance criterion).

2.1 Search algorithm

It is clear that the crucial problem is to choose a search algorithm. The search of discrete controller parameters has to run repeatedly in every single step of sampling interval, which lays great demands on computing time of the search algorithm. Naturally, usage of some iterative optimization algorithm with only one (or several) iteration realization every time instant is suggested. Gradient descent techniques seem inconvenient because of neural model usage. Neural model is black-box-like model so it is generally not possible to determine gradient descent analytically. On the other hand, evolutionary search techniques (genetic algorithm – see (3), (4), differential evolution – see (5), (6), etc.) appear to be suitable because these techniques do not require any particular information about search problem. The other indisputable advantage is its operating principle. In each iteration, evolutionary search techniques explore not only one value of input variables but whole set of them (one generation of individual solutions), which lowers significantly troubles with initial parameters random choice.

For this particular case, differential evolution is chosen. The reasons are, among others, that differential evolution is computationally rather less demanding algorithm, it works with decimal input values (contrary to genetic algorithm) and population of possible solutions is kept more diversified – see (6). Last but not least, differential evolution is chosen because of authors' recent validating experiments (7).

2.2 Controller type

The control method, which is described here, does not require any special form of discrete controller. After some experiments (8), (9), controller form

$$u(k) = p_0 w(k) - p_1 y(k) - p_2 y(k-1) - p_3 y(k-2) + u(k-1) \tag{1}$$

is considered to be convenient. For some $p_0 \ldots p_3$ parameters combinations, controller (1) acts like discrete PID controller (10). In general, however, it has one additional independent parameter. There can be obtained really suitable control performances by well-tuned controller (1).

2.3 Control method summary

Whole algorithm of described control method is compiled in following points:

1. Create dynamical neural model of controlled system – see (11)
2. Choose future horizon length N
3. Choose differential evolution parameters (number of individual solutions in one generation NP – any solution represents one particular quaternion of controller parameters $p_0 \ldots p_3$, crossover constant CR, mutation constant F) and their initial values
4. Measure controlled variable $y(k)$
5. Perform one iteration of differential evolution (based on the knowledge of controlled variable $y(k)$, course of its reference $w(k)$ till $w(k+N\text{-}1)$ and neural model of controlled system)
   a) perform control simulation with discrete controller and the neural model over future horizon N and evaluate cost function for all the individual solutions from current generation
   b) Apply cross-over and mutation (see (2)) so that offspring generation of solutions is bred
   c) Evaluate cost functions of offspring (see step a))
   d) Choose the best individual solution from the offspring generation
6. Evaluate manipulated variable $u(k)$ with discrete controller determined by the best individual solution obtained in step 5d)
7. $k = k + 1$, go to step 4

Future horizon length N is important parameter of the whole algorithm. There are no exact rules how to choose it. Too short horizon does not provide sufficient data to differential evolution. However, too long one brings data so distant from the current state that this data should not influence the next controller output value. It has to be mentioned that tall future horizon length causes prolonged computing time (computing time is one of key troubles).

Suitable definition of cost function is

$$J = \frac{1}{N} \cdot \sum_{i=k}^{k+N-1} \left| e(i) \right| + \frac{h_1}{N-1} \cdot \sum_{i=k+1}^{k+N-1} \left| \Delta u(i) \right| + h_2 \cdot \left| e(k+N-1) \right|, \tag{2}$$

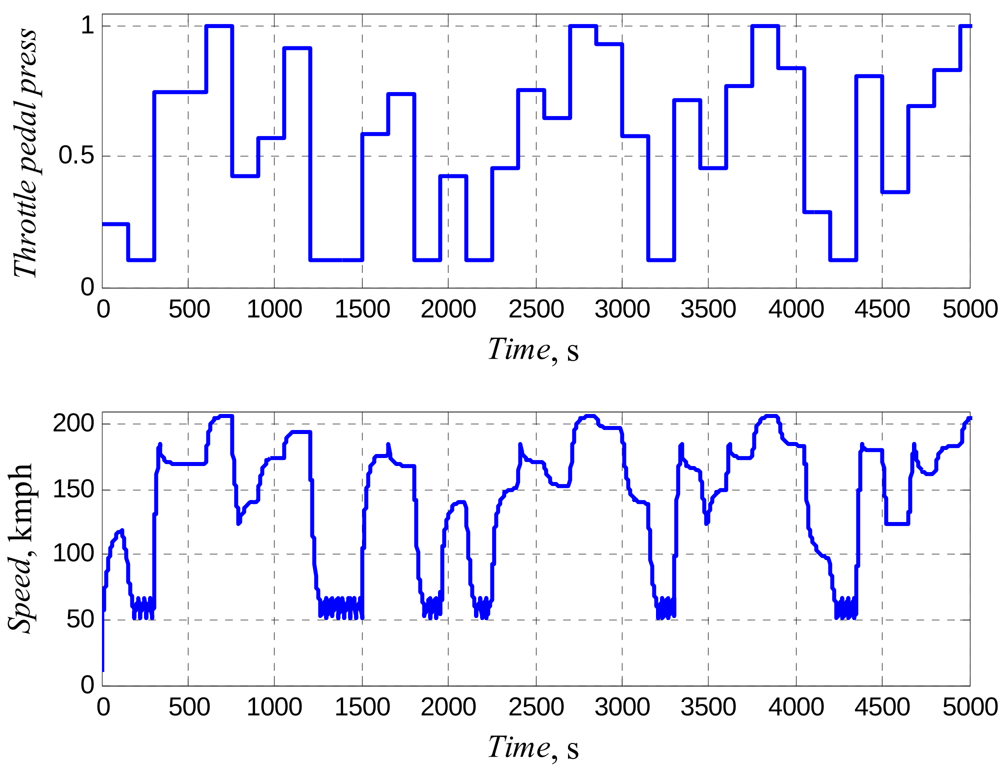
where $\Delta u(i) = u(i) - u(i-1)$, $e(i)$ is control error, h_1 is function parameter influencing manipulated variable differences, h_2 is function parameter influencing the state on the end of future horizon, N is future horizon length and $w(i)$ is reference variable.

Eventually, most of real controlled systems inputs and states are constrained. It is useful to include that limitation to control simulation (step 5a)) in order to influence discrete controller parameters search.

3. SMART CAR SPEED CONTROL

According to the paragraph 2, it is necessary to get the plant model. Thus, discrete neural model (sampling period 1s) is designed.

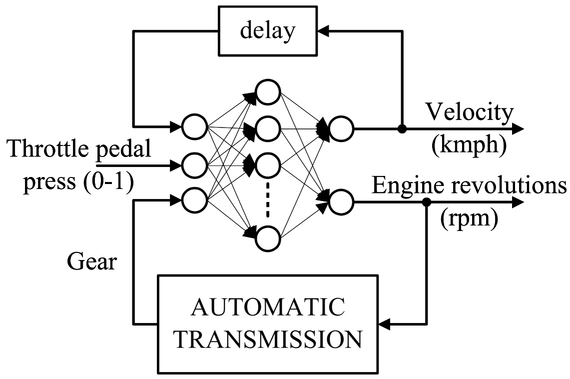
First, full car model (paragraph 1) is used for training set data acquisition – see Fig. 5.



Source: Author

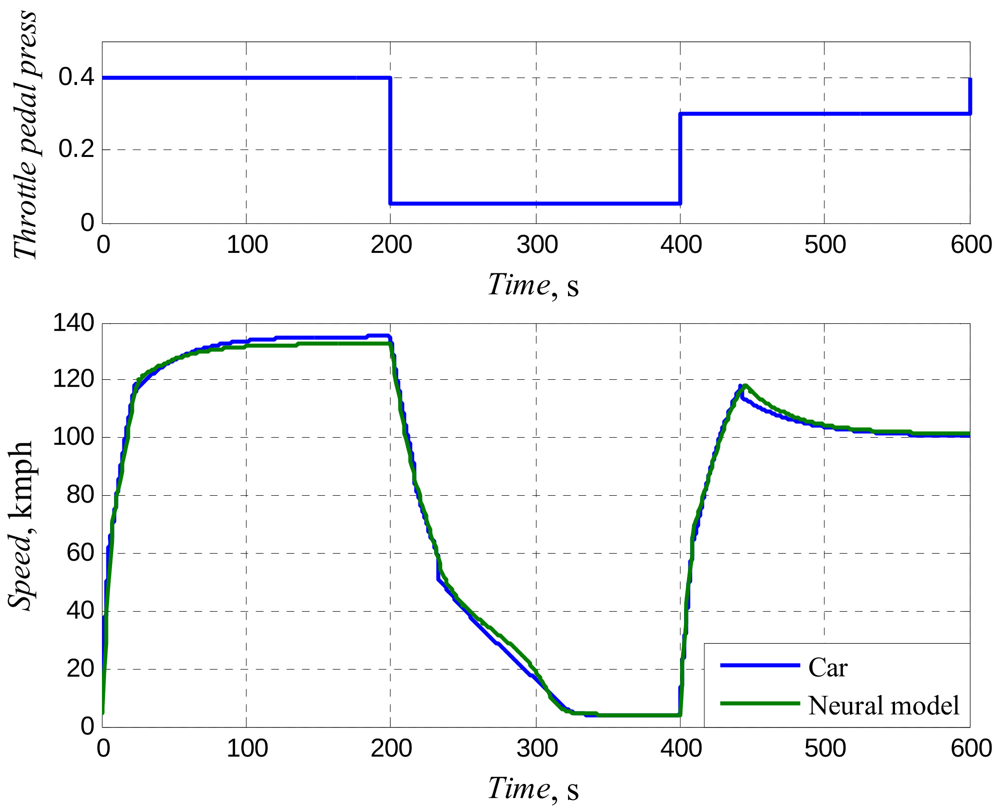
Fig. 5 – Training set for neural model

Then, artificial neural network is designed (topologically redundant neural network is trained offline by Levenberg-Marquardt training algorithm repeatedly while pruning is applied). Final neural network topology is 3 inputs, 10 neurons with sigmoid transfer functions in hidden layer and 2 output neurons with linear transfer functions. Comprehensive dynamical neural model design procedure can be found e.g. in (11). Compact discrete neural model of the plant is shown in Fig. 6, while fragment of its verification for defined input course is depicted in Fig. 7.



Source: Author

Fig. 6 – Neural model



Source: Author

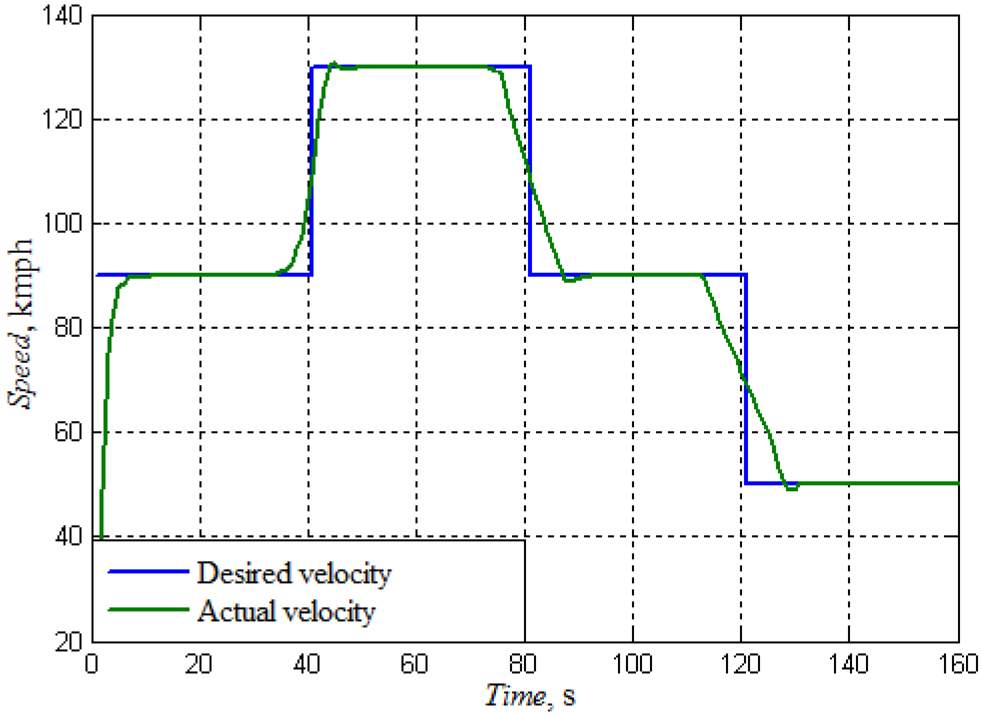
Fig. 7 – Fragment of neural model verification

Then, control loop (see Fig. 4) can be put together. Eligible parameters are chosen after some experiments partially according to (6):

Number of solutions $NP = 1000$
Crossover constant $CR = 0.85$

Mutation constant $F = 0.6$
Horizon length $N = 9$
Differential evolution iterations every time instant: 2

Reference variable course is determined as step function with magnitudes 50 kmph, 90 kmph and 130 kmph, since those are the speed limits for road vehicles in the Czech Republic. For real cars, reference variable can be obtained e.g. from car navigation system during route planning. Cost function is defined by the equation (2) where $h_1 = 20$, $h_2 = 0.5$. Control performance is plotted in Fig. 8.



Source: Author

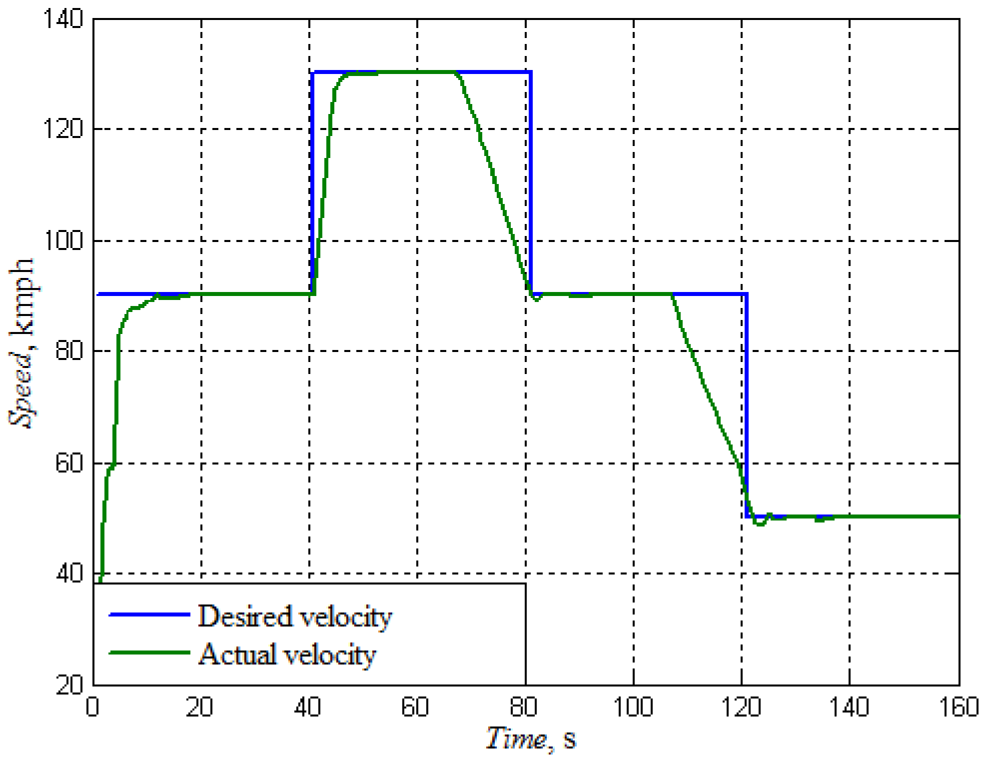
Fig. 8 – Control performance 1

Control performance in Fig. 8 is similar to control performance to be obtained by any of model predictive control techniques (10). The controller attempts to minimize the criterion (2), however the final performance is influenced by the eligible parameters values, indeed.

That control performance cannot be used practically. If the reference variable is defined by the speed limits, it is not suitable for actual speed to exceed that reference. However, cost function for differential evolution can be defined freely, it is decent advantage. Thus, cost function (2) is modified so that negative control error (desired speed lower than actual speed) is included in the criterion with tenfold heavier weight than positive control error. The purpose is clear – it is necessary to penalize those control performances, where actual car speed exceeds speed limit.

Control performance with cost function modified in this way is depicted in Fig. 9. The only parameter, which value is changed, is horizon length N. In this case $N = 15$.

It is obvious, that desired speed is conveniently followed by actual speed, but speed limits (identical to desired car speed) are not exceeded.



Source: Author

Fig. 9 – Control performance 2

CONCLUSION

The paper is focused on smart car speed control. It introduces sophisticated control technique based on online optimization of control action using differential evolution and dynamic neural model of the car. Control technique allows shaping speed course as necessary and control simulations show some decent features of the technique.

However, it is clear that online optimization using differential evolution can be hardly used in real traffic. Still, the possibility is to use this technique to determine huge sets of tested control action courses sorted to lookup-table since those sets can offer optimal control action according to actual car state. This approach can guarantee stability and safety of the control and it will be tested on laboratory models in near future.

ACKNOWLEDGEMENT

The work has been supported by the funds No. SGFEI03/2012 of the University of Pardubice, Czech Republic. This support is very gratefully acknowledged.

REFERENCES

(1) RILEY, F. J. *Constant Speed Regulator*, United States Patent US 2714880, March 24, 1950

(2) ASTROM, K. J., HAGGLUND, T. *PID controllers: theory, design and tuning*. Durham: International Society for Measurement and Control, 1995. 351 p. ISBN 1-55617-516-7.

(3) HOLLAND, J. *Adaptation in Natural and Artificial Systems (second edition).* Ann Arbor: MIT Press, 1992. 228 p. ISBN 978-0-262-58111-0.

(4) GOLDBERG, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning.* Boston: Addison-Wesley Longman Publishing Co., 1989. 372 p. ISBN 0201157675.

(5) STORN, R.M., PRICE, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization,* Vol. 11, 1997, pp. 341 -359. ISSN 0925-5001.

(6) PRICE, K., STORN, R.M., LAMPINEN J.A. *Differential Evolution: A Practical Approach to Global Optimization.* Berlin: Springer, 2005. 292 p. ISBN 978-3-540-20950-8.

(7) MAREŠ, J., DOLEŽEL, P., HRNČIŘÍK, P. *Artificial Neural Networks - Industrial and Control Engineering Applications*. 1st ed. Rijeka: Intech, 2011. Artificial Neural Network – Possible Approach to Nonlinear System Control, s. 359 – 376. ISBN 978-953-307-220-3.

(8) DOLEŽEL, P., MAREŠ, J. Reactor Furnace Control Using Artificial Neural Networks and Genetic Algorithm. In *Applied Electronics,* Plzeň, 9.-10. 9. 2009. Plzeň: University of West Bohemia, 2009, pp. 99-102. ISBN 978-80-7043-781-0 ISSN 1803-7232.

(9) DOLEŽEL, P., TAUFER, I. PSD controller tuning using artificial intelligence techniques. In *Proceedings of the 17th International Conference on Process Control '09,* Štrbské Pleso, 9.-12. 6. 2009. Bratislava: STU, 2009, pp. 120-124. ISBN 978-80-227-3081-5.

(10) BOBÁL, V., BÖHM, J., PROKOP, R. *Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace.* Brno: VITIUM a VUT v Brně, 1999. 244 p. ISBN 80-214-1299-2.

(11) HAYKIN, S. *Neural Networks: A Comprehensive Foundation.* New Jersey: Prentice Hall, 1999. 842 p. ISBN 0-13-273350-1.

(12) CAMACHO, E. F., BORDONS, C. *Model Predictive Control 3rd*. London: Springer – Verlag.. 208 p. ISBN 3-540-76241-8.