# Thermal process control using neural model and genetic algorithm

Daniel Honc, Petr Doležel and Jan Merta

University of Pardubice, Pardubice 532 10, Czech Republic
daniel.honc@upce.cz

**Abstract.** Predictive Controller of a laboratory thermal process is presented in the paper. Process model is approximated by a neural network. On-line optimization is done by a genetic algorithm. Control algorithm is tested on the laboratory thermal process and compared to the standard control methods like predictive controller with the transfer and state-space linear model and the quadratic programming optimization method or a PI controller.

**Keywords:** Predictive Controller, Neural Network, Genetic Algorithm, PI controller, thermal process.

## 1 Introduction

Model Predictive Control (MPC) is one of the Advanced Process Control (APC) methods used in the industry. In simple words, it is an optimization algorithm placed above the standard Digital Control Systems (DSC) ensuring optimal control with respects to the technological but also to the economic conditions and restrictions. DSC layer (mostly PID controller) is present because of the reliability, accuracy and safety of the control. The main task for APC methods is to find the optimal working point (nonlinear static optimization) and to drive the system to this point (dynamical optimization) especially for Multi-Input Multi-Output (MIMO) systems. Usually the APC generates the set-points to DSC.

MPC strategy consists of finding the optimal control actions for the whole prediction horizon based on the optimization and the dynamical model of the controlled system [1-3]. The model is crucial for the successful application. It could be an open loop control strategy. From practical point of view a receding horizon concept is introduced – the feedback is used according to the new measurements (actualization of the state of the controlled system and measurements of the actual disturbances if available). Every sample time the whole procedure is repeated and only the first (actual) control action from the vector of calculated optimal control actions is used. The criterion defining the control goals is tailored to the given control task. Typical requirement in industry is to maximize profit, production volume, to minimize input costs, amount of by-products etc. Nonlinear optimization method must be used if the model is nonlinear [4-6]. In most of the real situations the linear approximation is usually sufficient for the industrial applications. Quality of the model is good around the given working point and worst in the remote areas. Even if the process is more nonlinear it depends how tight control is required. For slower control (relatively to the controlled process behavior), the approximate gain and the dominant time constant can be enough. Linear models used in MPC are usually the final impulse and step response models, the transfer functions and state-space models. The result model being linear is that the originally nonlinear optimization problem can be solved as a linear or quadratic programming task with very effective algorithms [1-3].

Key parts of the MPC are the cost function formulation, the dynamical process model and the numerical optimization method. Increasing of the on-line optimization speed caused by the computational power increase and new numerical methods allows to use MPC for bigger and faster processes. There is also potential for using the non-linear models, complex cost functions and nonlinear optimization methods. Theoretically, there is no need of DCS layer – the MPC controller can calculate directly the control actions instead of the set-point for PID controllers.

Authors have experience with the predictive control of a nonlinear plant using piecewise-linear neural model and quadratic programming [7]. A little bit of curiosity and for educational reasons as well, authors want to try to use a nonlinear neural model of the controlled system and a genetic algorithm to minimize the quadratic cost function. Applications with the genetic algorithm and predictive controller can be found e.g. in [8-12]. We will control a real laboratory thermal process (heated metal bar by Peltier element), discuss results and compare them with the standard MPC methods or PI controller.

The paper is structured as follows. Introduction is in chapter 1. Control strategy is formulated in chapter 2. In chapter 3 controlled system is described. Identification experiments are presented in chapter 4 and control experiments are in chapter 5. Conclusions are given in chapter 6.

## 2 Control strategy

The paper describes MPC design and laboratory application with process model in the form of neural network and on-line optimization done by the genetic algorithm.

### 2.1 Cost function

Standard cost function is considered with the penalization of future control errors and control increments

$$J = \sum_{j=1}^{N_2} r_j \left( \hat{y}(k+j) - w(k+j) \right)^2 + \sum_{j=1}^{N_u} q_j \left( \Delta u(k+j-1) \right)^2 \tag{1}$$

where

- $N_2$ is prediction horizon,
- $N_u$ is control horizon,
- $r_j, q_j$ are penalization parameters,
- $\hat{y}(k+j)$ is prediction of controlled variable,

$w(k+j)$ is desired value (set-point) and
$\Delta u(k+j-1)$ is control increment (change, move).

The control horizon can be shorter than the horizon for the set-point following. We can reduce the computational complexity by supposing some control increments at the end of the control horizon to be zero – the control variable will freeze at the last calculated control action. Penalization of control increments is considered to avoid the steady-state control error – this would be the case if the absolute control actions were penalized.

### 2.2 Process model and its identification

In the context of this contribution, identification is considered as a statistical approach that provides a model of a dynamical process from the measured data. Many different identification techniques are available and the suitable one must be chosen according to the specific requirements.

Usually, linear models in the form of ARX (Auto-Regressive model with eXogenous input) or ARMAX (Auto-Regressive-Moving Average model with eXogenous input) equation are used for MPC [13]. However, different approach is provided in this paper.

Dynamical neural models are generally able to approximate even highly complex and nonlinear processes very precisely since feedforward neural networks (FFNNs) are proven to be universal approximators [14]. On the other hand, a black-box-like structure of the neural model restricts a bit the conventional use for the controller design. Since the optimization is performed by a variant of genetic algorithm, no strict conditions are required to the model form. Therefore, a dynamical neural model can be preferably implemented for this case.

There are several possibilities available to design a neural model of the process. One of them assumes the process description by the following nonlinear discrete-time difference equation

$$\begin{aligned} y_S(k) = \psi[&y_S(k-1), \ldots, y_S(k-n), \\ &u(k-1), \ldots, u(k-m)],\ m \le n \end{aligned} \tag{2}$$

where
$\psi(.)$ is a general nonlinear function,
$y_s$ is calculated output from the process,
$u$ is input to the process and
$n$ is the order of the difference equation.

The aim of the identification is to design FFNN, which approximates the function $\psi(.)$. In other words, FFNN must be designed and trained to provide outputs as close to process outputs as possible. A general diagram of a dynamical neural model is illustrated in Fig. 1.

A dynamical neural model design generally consists in training and testing set acquisition, neural network training and pruning, and neural model validating. These steps have already been defined comprehensively many times e.g. in [15-16].

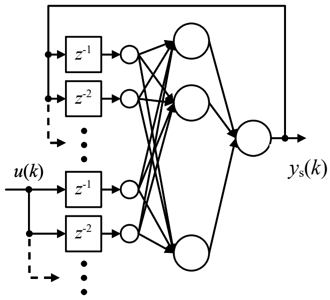


**Fig. 1.** Dynamical neural model.

### 2.3 Optimization method

Genetic algorithms are suitable for constrained, nonlinear optimization and non-convex optimization problems [8]. Genetic algorithms belong to evolutionary algorithms, they use population of candidate solutions. They are inspired by the Darwinian evolution using the crossover and mutation genetic operators. A more detailed description of this technique can be found e.g. in the book [17].

Papers where genetic algorithms have been used for MPC control are e.g. [8-10]. Authors suggested a variety of methods to deal with the time constraint of the predictive controller, such as using shifted individuals from previous time steps. Authors applied genetic algorithms to find a feasible descent solution rather than the optimal solution at each sampling time in [9]. Genetic algorithm for thermal process control is published in [11]. Genetic algorithms were used also in predictive control to select the control and prediction horizons [12].

Authors used a population-based genetic algorithm with real alphabet to optimize the cost function at each time step. The genes of individuals directly represent the sequence of control actions. The fitness function equals to the cost function.

When generating the initial population, we were inspired by the method of shifted individuals suggested in [8] and [9]. We do not generate individual control actions from the full range of possible control actions, but we derived individuals of the initial population from the shifted best individual from the previous time step. From the previous time step we remove the first control action to move the entire sequence of the control actions. Missing last control action is duplicated from the previous control action. The remaining individuals of the initial population are generated as random deviations of the individual genes of this modified individual within a given range.

As a selection mechanism we used tournament selection. We used the one-point crossover and the mutation that adds a random deviation to the selected gene within a given range. For termination of the run of the genetic algorithm we use maximum number of generations and maximum elapsed time because of time constraint of the controller.

## 3 Controlled system

Proposed algorithm is tested at a temperature control system GUNT RT040 [18] – see Fig. 2.

**Fig. 2.** RT 040 Training system [6].

The metal rod is heated or cooled using the Peltier element (signal Y). The bar temperature is measured in three positions by the temperature sensors (X1, X2 and X3 signals). The fan (signal Z) on the Peltier element cooler can be switched on or off – see Fig. 3.

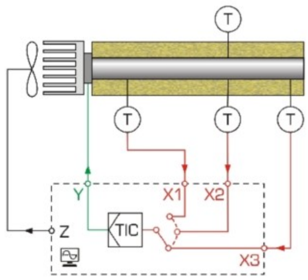


**Fig. 3.** RT 040 Training system diagram [6].

Voltage (control signal of the Peltier element) in the rage from -5 to 5 V is used as a control action $u$ (manipulated variable). Farthest sensor X3 (from Peltier element) is used as a temperature measurement point for the controlled variable $y$. RT 040 uses USB multifunction LabJack U12 data acquisition card [19] with drivers support to different environments. Measurement and control software is written as a MATLAB script and run on a PC – see Fig. 4. The same script is used for the simulation and for the control experiments. The simulation can be done in real time or as fast as possible. The real-time simulation allows to test whether the sample time is enough long for the certain algorithm and used hardware – load means what percentage of the sampling period is consumed with the measurement, control algorithm, actuation and plotting. The predicted controlled variable and the calculated future optimal control action are displayed – red line starting from the current time instant. The user can check whether the algorithm with given parameter works fine or must be stopped and retuned.

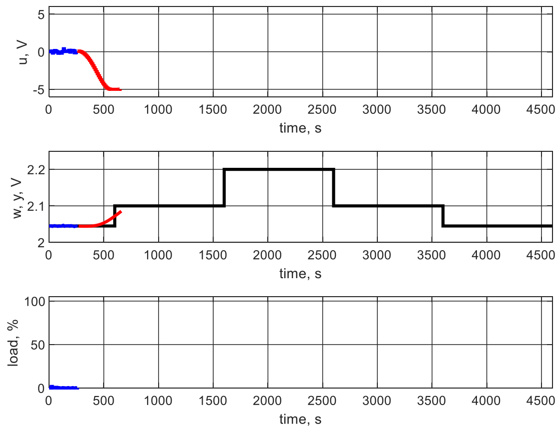


**Fig. 4.** Control software window in MATLAB.

# 4 Identification experiments

Identification experiments are carried out – the temperature response to 14 step changes, each 1 hour long is measured – blue lines in Fig. 5. Fan is on during all the experiments. Positive $u$ means cooling, negative $u$ heating. From the response, it can be seen that the gain by the cooling is approx. 10 time smaller than by the heating. This is usual nonlinearity of the thermal processes – efficiency of the cooling is smaller than of the heating.

The process is approximated with the dynamical neural model – red line in Fig. 5. The procedure of the neural model design is not described here, since it is a standard procedure. At the end of this procedure, the neural network inside the neural model consists of four inputs ($m = 2$, $n = 2$), five neurons with the hyperbolic tangent activation functions in the hidden layer, and one output neuron with the linear (identical) activation function. The sampling time is set to 10 s.

The model uses absolute $u$ and $y$. This model is used by all simulations like a simulation model and by the neural network and genetic algorithm predictive controller for predictions – they are calculated in a cycle and used for the cost function evaluation.

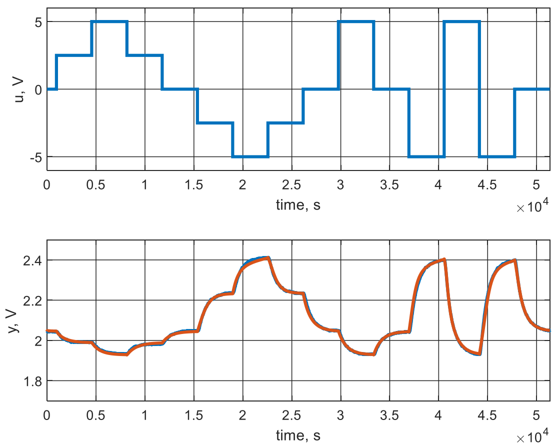


**Fig. 5.** Identification with neural network model.

Second order continuous-time transfer function parameters are identified for the predictive controllers with transfer-function and state-space model – equation (3) and red line in Fig. 6.

$$F(s) = \frac{-0.048}{(758s+1)(91s+1)} \tag{3}$$

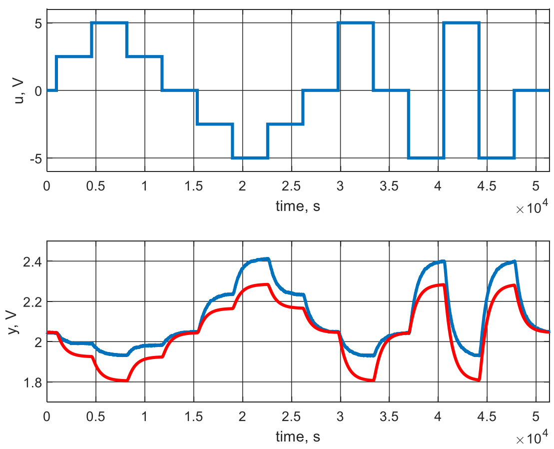


**Fig. 6.** Identification with linear model.

Working point $u = 0$ V, $y = 2.045$ V is used by the identification. The system has negative gain and practically one dominant time constant. Continuous-time transfer function is transformed into the discrete-time transfer function and state-space model for the predictive controllers.

# 5 Control experiments

Four control experiments were carried out. The most important experiment is the experiment with the neural network and genetic algorithm – see Fig. 7. For comparison with the standard control methods another three experiments are measured – predictive controller with the transfer function model and quadratic programming – Fig. 8, predictive controller with the state-space model and quadratic programming – Fig. 9 and PI controller – Fig. 10. Algorithms for both predictive controllers can be found in [20]. The parameters of the controllers are in Tab. 1.

**Table 1.** Parameters of the controllers.

| Symbol | Value | Dimension | Meaning |
|---|---|---|---|
| $T_S$ | 10 | s | sample time |
| $N_2$ | 40 | s | prediction horizon |
| $N_u$ | 40 | s | control horizon |
| $r$ | 1 | - | control error penalization |
| $q$ | 0.001 | - | control increment penalization |
| $C$ | $(1-0.8z^{-1})^2$ | - | filtering polynomial |
| $P$ | [0.8 0.8] | - | poles of the observer |
| $r_0$ | -100 | - | PI controller gain |
| $T_i$ | 300 | s | PI controller integral time constant |

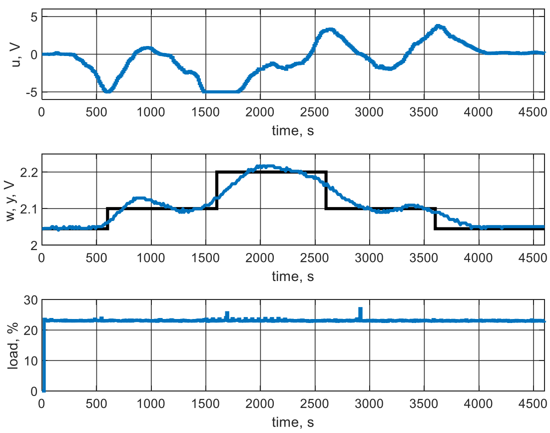


**Fig. 7.** Control with neural network and genetic algorithm.

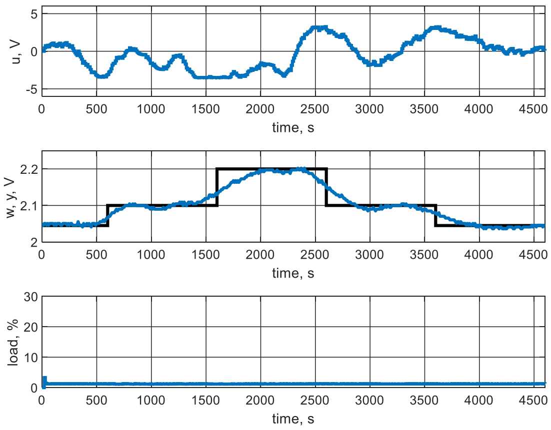


**Fig. 8.** Control with transfer function and quadratic programming.

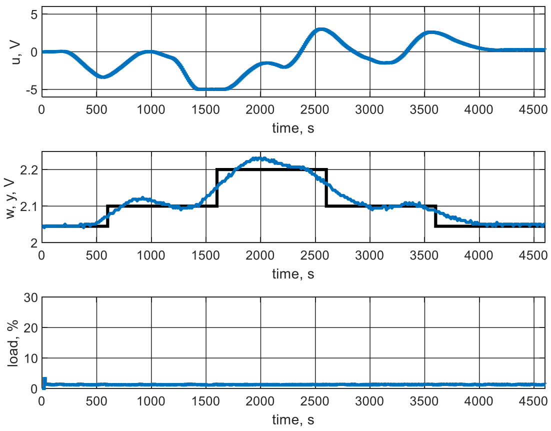


**Fig. 9.** Control with state-space and quadratic programming.

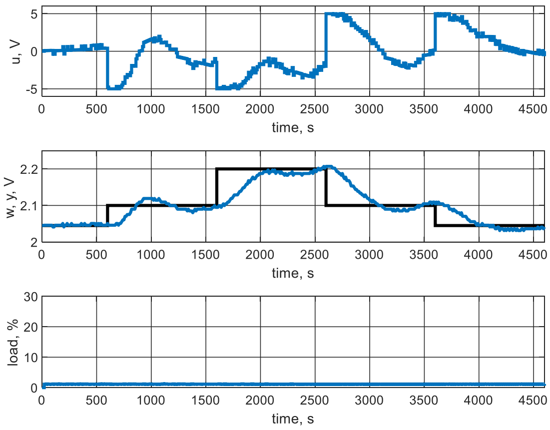


**Fig. 10.** Control with PI controller.

Integral Absolute control Error (IAE) and Integral Square control Error (ISE) is calculated as the control quality measures.

**Table 2.** Control quality measures.

| Controller | IAE | ISE |
|---|---|---|
| Neural network and genetic algorithm | 61.0 | 1.75 |
| Transfer function and quadratic programming | 51.8 | 1.32 |
| State-space and quadratic programming | 58.4 | 1.43 |
| PI controller | 101.1 | 5.53 |

# 6 Conclusion

Predictive controller with the neural network model and genetic algorithms as the optimization method is presented and tested at a laboratory thermal process. Authors were curious whether the nonlinear model with nonlinear on-line optimization will be better or worse than some standard control methods – predictive controllers with the linear models and quadratic programming and the PI controller. The result is, that in our case, the nonlinear model does not bring visible improvement even if the identification was much better than with the linear model. This can be caused by the relative simplicity of the controlled process – controlled process is nonlinear but practically first order system with the dominant time constant. The nonlinear controller burdened the computer more than the linear predictive controller which was comparable to the PI controller. Sample time 10 s was at the border of the fastest samplings we can use – horizon length is given by the controlled process and increasing of the sample time leads to the need to use longer control horizon and hence the higher computational demands. We should compare the solution with some standard nonlinear MPC control algorithm. The nonlinearity of our process was not crucial, so we used the linear

MPC. It would be interesting to control more complex, more difficult to control, more nonlinear process because there should be better to see the advantage of the NMPC over the other methods. Our system was too easy to control. On the other hand, the advantage of a good model was clear because the laboratory experiments took quite a long time and there was no space to iterate with the controller parameters tuning. Therefore, the simulation of the control responses with the neural model and expected measurement noise was very beneficial in the control design and testing phase.

## Acknowledgment

This research was supported by Institutional support of The Ministry of Education, Youth and Sports of the Czech Republic at University of Pardubice and SGS grant at Faculty of Electrical Engineering and Informatics.

## References

1. Rossiter, J.A.: Model-based Predictive Control – A Practical Approach, CRC Press (2003).
2. Camacho, E.F., Bordons, C.: Model Predictive Control, Springer (2007).
3. Maciejowski, J.: Predictive Control with Constraints, Prentice Hall (2002).
4. Allgöwer, F., Findeisen, R., Nagy, Z.K.: Nonlinear Model Predictive Control: From Theory to Application, J. Chin. Inst. Chem. Engrs., Vol. 35, No. 3, 299-315 (2004).
5. Magni, L., Raimondo, D.M., Allgöwer, F.: Nonlinear Model Predictive Control - Towards New Challenging Applications, Lecture Notes in Control and Information Sciences, vol. 384, Springer (2009).
6. Qin, S.J., Badgwell, T.A.: An overview of nonlinear model predictive control applications, In.: Nonlinear Predictive Control, pp. 369–392. Springer (2000).
7. Honc, D., Doležel, P., Gago, L.: Predictive control of nonlinear plant using piecewise-linear neural model, In.: Proceedings of the 21st International Conference on Process Control, pp. 161-166 (2017).
8. Onnen, C., Babuška, R., Kaymak, U., Sousa, J.M., Verbruggen, H.B., Isermann, R.: Genetic algorithms for optimization in predictive control, Control Engineering Practice, 5(10), 1363-1372 (1997).
9. Chen, W., Li, X., Chen, M.: Suboptimal nonlinear model predictive control based on genetic algorithm, In.: Intelligent Information Technology Application Workshops, IITAW'09, Third International Symposium on. IEEE, p. 119-124 (2009).
10. Blasco, X., Martinez, M., Senent, J., Sanchis, J.: Generalized predictive control using genetic algorithms (GAGPC). an application to control of a non-linear process with model uncertainty, In.: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp. 428-437, Springer (1998).
11. Stojanovski, G., Stankovski, M.: Model predictive controller employing genetic algorithm optimization of thermal processes with non-convex constraints, In.: Genetic Algorithms in Applications. InTech (2012).
12. Mohammadi, A., Asadi, H., Mohamed, S., Nelson, K., Nahavandi, S.: Optimizing model predictive control horizons using genetic algorithm for motion cueing algorithm, Expert Systems with Applications, 92, 73-81 (2018).

13. Bobál, V.: Digital self-tuning controllers: algorithms, implementation and applications, London: Springer (2005).
14. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators, Neural Networks, vol. 2, no. 5, pp. 359-366 (1989).
15. Haykin, S.: Neural Networks: A Comprehensive Foundation, Prentice Hall (1994).
16. Korbicz, J., Janczak, A.: A neural network approach to identification of structural systems, In.: Proceedings of IEEE International Symposium on Industrial Electronics, Poland, pp. 98-103, vol.1, (1996).
17. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning, Machine learning, 3(2), 95-99 (1988).
18. RT 040 Training system: temperature control, HIS, on/line at https://www.gunt.de/en/products/ mechatronics/automation-and-process-control-engineering/simple-process-engineering-control-systems/training-system-temperature-control-hsi/080.04000/rt040/glct-1:pa-148:ca-83:pr-1045 (cited 12. 1. 2019).
19. U12 Series, on-line at https://labjack.com/products/u12 (cited 12. 1. 2019).
20. Honc, D., Sharma, K.R., Abraham, A., Dušek, F., Pappa, N.: Teaching and Practicing Model Predictive Control, IFAC-PapersOnLine, 49 (6), pp. 34-39 (2016).