

PREDICTIVE CONTROL OF PROCESSES WITH UTILIZATION OF ARTIFICIAL INTELLIGENCE ELEMENTS

Marta Blahova, Vaclav Mach & Jan Valouch



This Publication has to be referred as: Blahova, M[arta]; Mach, V[aclav] & Valouch, J[an] (2020). Predictive Control of Processes with Utilization of Artificial Intelligence Elements, Proceedings of the 31st DAAAM International Symposium, pp.xxxx-xxxx, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-xx-x, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/31st.daaam.proceedings.xxx

Abstract

Predictive process control is a method of regulation suitable for controlling various types of systems, which is based on the idea of using the prediction of future system behavior and its optimization. Normally, a system model is used to predict behavior, and therefore it is necessary for the correct function of predictive control to make its correct selection and determine its parameters so that the controlled system is described as accurately as possible. Another advantage of predictive control is the possibility of including signal restrictions directly in the controller. The result is the application of some elements of artificial intelligence in suitable areas of predictive control, especially the use of simple evolutionary algorithms in optimization and neural networks as nonlinear models. One of the chapters of the article describes the possibilities of using these elements. It is proved that in addition to classical optimization algorithms, it is also possible to use simple evolutionary algorithms for optimization of prediction, while the computational complexity can be comparable depending on the type of solved problem and settings. The article describes a suitable selection of model systems with slow dynamics, their derivation, and the creation of nonlinear models in the form of scalable neural networks. The potential advantage of this approach for the control of systems that are difficult to describe or for the control of systems whose mathematical-physical description is not known. The chapter of the article also deals with the possibility of using the found models on real systems and determining the necessary conditions and requirements for their application.

Keywords: Predictive control; discrete dynamic models; artificial intelligence; neural networks; evolutionary algorithms.

1. Resources in crisis management software

Predictive control is a method of regulation, its main idea is to use the predictive behavior of the system to optimize the action. With this method, it is possible to combine systems with the advantages of discrete models. [1] The dynamic continuous model is usually expressed using differential equations. Mathematical operations with these examples are very difficult. Discrete models using differential equations to describe the dynamic behavior of a system, which uses a

limited number of samples of past values of system inputs and outputs, show current output values. The extension of this model by future value can be expected to predict the future behavior of the system. This extension is then called predictor. [2]

The shape of the predictor depends on the size of the prediction horizon. Building system behavior can be optimized to achieve set points. The optimizer and the predictor together form a controller. [3] Behavior prediction is from the system model, and therefore this model should describe the behavior of the system as accurately as possible. It is possible to design a controller with the lowest possible computational complexity, especially for systems with very fast dynamics. Predictive control can be used on different types of systems, even those that are often difficult to control. These are non-minimal phase systems, traffic delay systems, or non-linear systems. [4] System model with an identification system [5], either mathematical-physical calculation or experimental identification. [6] The disadvantage of these models is their high computational complexity, so they are suitable mainly for systems with slow dynamics.

Quadratic programming is usually used in the optimization process because the optimized problem is often quadratic in shape. Another advantage is the ability to include constraint conditions directly in the optimization process. It is then possible to use several types of algorithms to solve the optimization process, where simplex algorithms are often used. [7] In addition to classical methods, it is possible to use some types of evolutionary algorithms or algorithms based on them. If these algorithms are simple in principle, it is possible to achieve computational complexity comparable to the complexity of classical methods, while some of these evolutionary algorithms can be applied to some optimization problems. [8]

An example of the use of predictive control can be controlled by the liquid level in a system of three interconnected cylindrical tanks with two inputs, two outputs, and three states with applications of state and input-output models, where linearization around the running point will be performed using Taylor expansion line members. On a system of six interconnected tanks, the possibilities of a modeling and the use of experimental model identification using stages (stairs) of input-technical characteristics for the derivation of first and second-order transmission were demonstrated, extended predictive control, again using linearization using Taylor expansion to obtain six transmissions. [8]

The subject of research is the use of elements of artificial intelligence in some parts of predictive control, where the use of evolutionary algorithms will be demonstrated at least on a subsystem, and this algorithm will be compared with classical methods. Furthermore, model systems of tanks with functional geometries will be created, which are generalized to each other by other shapes of real systems occurring in practice or in nature. For applicability and generality, it is necessary to derive models by the classical procedure with a non-dimensional quantity. At the same time, the effect of hydrostatic pressure, which is the same in any tank in the tank and used only and only in height, will be feasible back into the controller when switching to the actual system. General models of neural networks will be created for these model systems, which will be used in the front controller. The models and regulators obtained in this way should therefore be diverse and their possibilities of use in a specific application-wide, always for a real tank of a given type of shape. The possible impact of computational complexity in online optimization with a sample period, ie about the dynamics of the controlled system. [9]

2. Predictive control

Predictive control is a method of control based on the simple idea that it is possible to predict the behavior of the system within the current control period, and then optimize this prediction to achieve the desired state of the system and possibly meet other control requirements. The MPC (Model Predictive Control) method uses its model to predict the behavior of the system. To achieve the best possible control, it is therefore necessary that the selected model best describes the behavior of the controlled system, and therefore it is necessary to select it, identify the system and determine the parameters of the model as best as possible [9].

The controller is then based on the use of a predictor and optimizer, where the predictor uses the obtained system model to predict output and control variables (or state variables) and the optimizer then optimizes the prediction to achieve the desired value and other required properties for system control. The main advantage of predictive control is its ability to deal with the limitations of action, controlled and state variables. Another advantage is its applicability to systems that are difficult to control by other methods. In particular, these are many-dimensional systems, non-minimal phase systems or systems with traffic delays. The disadvantages are greater computational complexity and the need to design a suitable predictor and optimizer [10].

3. Artificial Intelligence

The term artificial intelligence refers to systems or machines that exhibit a certain degree of intelligent behavior. Often these systems are inspired by natural processes. However, the field of artificial intelligence (AI) is very broad and therefore the article deals with only two areas of AI, neural networks (NN) and evolutionary algorithms. One of the

reasons to consider the possibility of deploying UI elements in predictive control is the fact that these systems can solve complex or general problems. One of the main limits of UI deployment can often be higher computational complexity compared to classical methods. Therefore, areas that are suitable for this deployment will be selected [11].

4. The experimental and methodological part

No real experiments were needed to achieve the sub-goals. The following software was used using the following literature, which is linked to this software [12].

- Matlab / Simulink software in version R2012b
- Wolfram Mathematica software in current versions.

Preliminary analysis and verification of the use of artificial intelligence elements

Own code in SW Wolfram Mathematica was used to solve preliminary tasks of purpose function analysis, quadprog () function and two own codes with "hill climbing" algorithm and "evolutionary-gradient" algorithm in Matlab SW were used to verify the deployment of evolutionary algorithms, and the results were processed in SW Wolfram Mathematica. To verify the deployment of the neural network, a custom neural network code was created in SW Matlab and the nonlinear optimization NMinimize () in SW Mathematica was used for network learning [13].

Choice of the system for predictive control and creation of a model system

The liquid flow tank was treated as a suitable test system for predictive control by abstraction and reduced to basic geometric shapes - model systems - representing selected situations, physical factors were also maximally reduced with respect to the necessary generality of the solution sought. Model systems were created by design on paper, then they were created as S-functions in SW Simulink, to which were added corrective S-functions solving problems of singularities, saturation, and the like, so that simulation in SW Simulink provides a realistic model within given constraints and generalization [13].

Creating a predictor

The predictor was created using a recursive approach and was implemented as a procedure within the S-Function of the controller in the Simulink simulation environment. The controller reads the file with the learned network and passes it to the predictor as a parameter [13].

Design of transfer of obtained results to a real system

Data from the real system were not available, but it was still possible to establish procedures and rules for the transfer of acquired knowledge to the real system. On the one hand, the scalability of the model was used, on the other hand, experience with the behavior of the used models and processes was used, and also in the direction of concretization and back extrapolation to physical influences, from which it was initially abstracted when choosing model systems [13].

Derivation of the CARIMA model predictor and optimizer

After obtaining the system model, it is possible to proceed to the derivation of the predictor. This will then be obtained by extending the differential equations by future values [14].

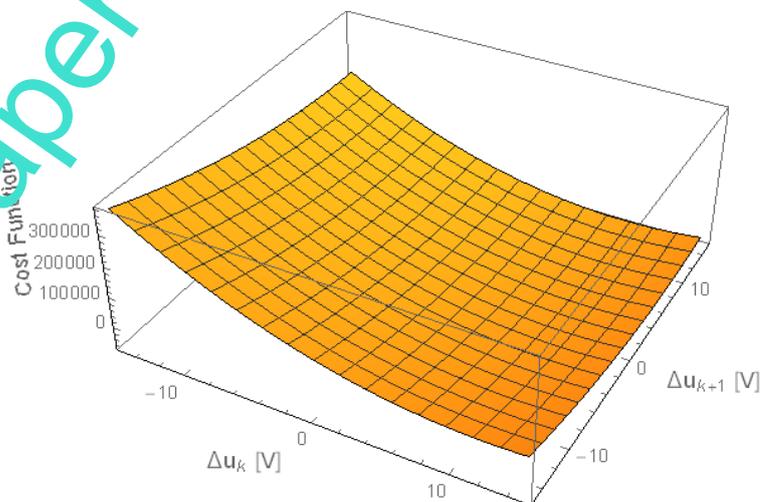


Fig. 1 - Purpose function of the optimizer for the CARIM

Model selection

The nonlinearity of the system is evident so that one of the nonlinear types of the model can be used for control purposes. For LV deployment options, the RBF network model was applied. In the interest of simplicity, structural optimization is omitted and some parameters are fixed to reduce degrees of freedom. The other parameters are then determined by sub-optimization [14].

Model identification and verification

To identify the RBF model, it is necessary to learn the network well. A training set and a set of test sets were created for this purpose. To do this, you can use the simulation of a real system in the SIMULINK environment and the application of typical input signals in the selected work area. The output of the system is then the monitored level and the input is the volume flow of the pump. Schemes of creating training and test set in SIMULINK, which can be seen in the Figure 2.

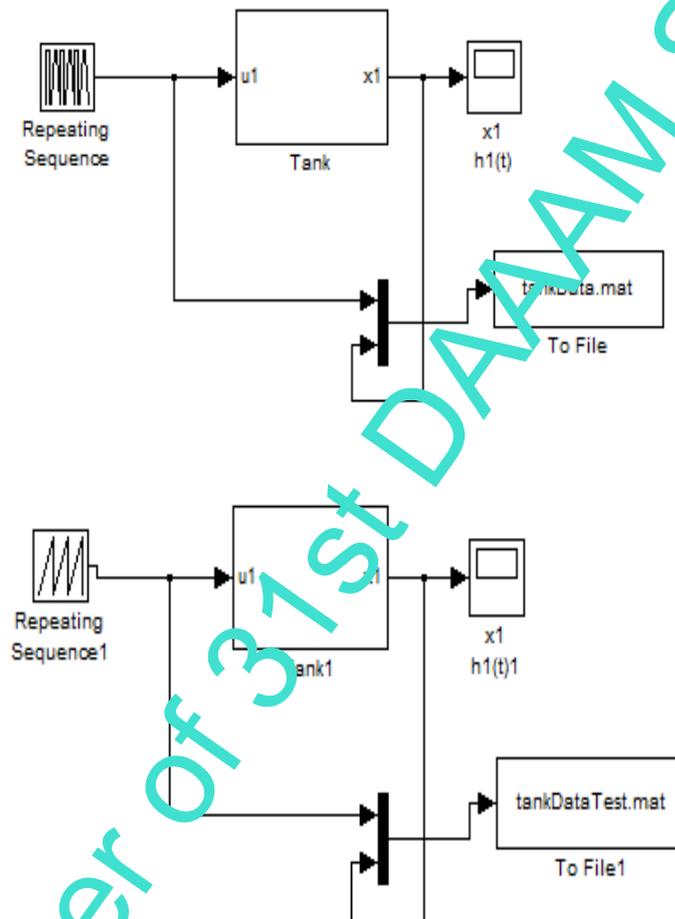


Fig. 2 - Creating a training and test set

There are several strategies for training the RBF network, but in this case, the strategy was created by combining the chosen procedures into the following specific form:

1. Selection of centers by random selection of input data
2. Center-based weight optimization (linear layer)
3. Optimization of centers based on determined weights (nonlinear layer)

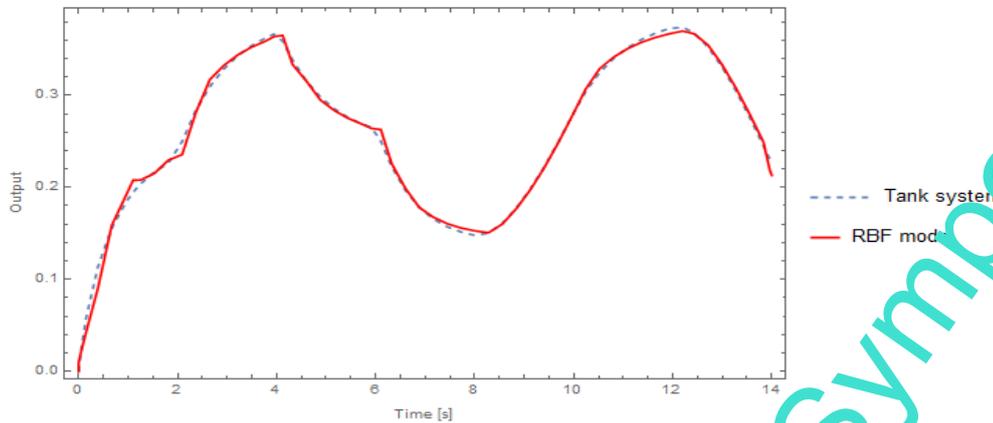


Fig. 3 - RBF network training, own source

The training is successful if the output of the RBF network and the real system match (input signal not displayed). Good network learning is key to the correct creation of the model, so the training sets should contain typical input signals (patterns) because only in this way is it possible to create a model that describes the real system with sufficient accuracy. Neural network learning usually consists of two steps: training and testing. Therefore, a set of test inputs was applied to the input of the real system, which can be used to test the success of model creation. After applying the test sets, the real system (S-Function) and the RBF network are compared again [15].

Management optimization.

The controller consists of two parts - a predictor and an optimizer. The optimizer uses a predictor to predict system behavior. The size of the horizon was chosen as follows $\{N_1, N_2, N_u\} = \{1, 3, 2\}$, to achieve low computational complexity. The control process was simulated in the Simulink environment, Figure 4.

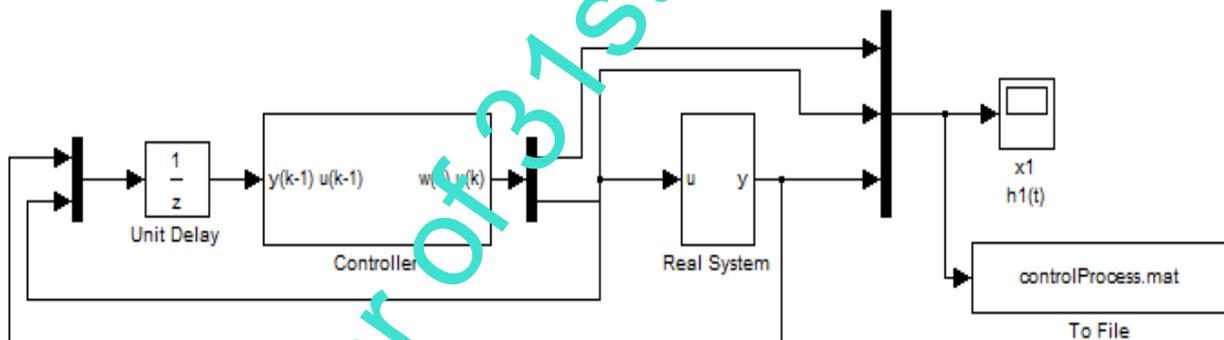


Fig. 4 - Simulation of predictive control of model systems

5. Estimation rules

Nevertheless, the following estimation rules can legitimately be formulated:

1. The tank filling/emptying time is covered by time scaling completely.
2. The volume of the tank is completely scaled.
3. The setpoint correction curve is completely scaled.
4. The model valid for one body (tank) always applies to all similar bodies (tanks).
5. However, the scaling effect is wider than the previous point, because the tank conversion factor (ie the ratio of diameter and height) has dropped from the equations, and therefore this ratio does not matter for the given shape of the level change rate equation, which is true if the volume is tanked linearly dependent on this ratio. In other words, each volume is given by the product of three dimensions of the body in mutually perpendicular directions (e.g. a.b.c). If one depends on the other two, then the conversion factor of the tank is not important for the transfer of the model into practice within the same type of geometry.
6. If the shape is completely independent in all three dimensions, the fact applies that of all the ratios between height and diameter, the one that encloses the relatively largest volume was chosen for the model systems used; it is a ratio of 1:1, which represents the smallest extreme on both sides and can therefore be considered a good mean estimate, which

also applies to shapes more complex than cylinder shells, cones, spheres, and other centrosymmetric and rotationally symmetric bodies, if we choose shape - correct - a type of model.

7. In practice, it will also be necessary to know and use the conversion function between the real action variable (eg the height of the sluice, angle turned by the tap, etc.) and the model action variable, ie the inverse characteristic of the device (outlet) must be known.

8. The effect of hydrostatic pressure is the same in a tank of any shape and depends only on height, and is retrofitable to the controller until the specific dimensions of the tank are known unless it is already implicitly included in the conversion function of the control device.

6. Conclusion

The obtained models apply to real systems only under certain conditions. Derived models are fully valid only if the above assumptions are met. The differential equations describing the systems contain the inflow and outflow quantities, where the inflow minus the outflow is determined for accumulation, neglecting the hydrostatic pressure and the physical limits of the fluid flow in general. It is therefore assumed that these quantities can be directly controlled, or at least measured, which can be done in practice, for example, using regulators, pumps or flow meters. Furthermore, it should be noted that the derived model systems were non-dimensional. Therefore, it is necessary to remap the real ranges so that they are within the ranges valid for model systems.

$$kq=1\Delta t_{max} \Rightarrow \Delta t_{max}=10,075= 13,3 \quad (6.50)$$

The sampling period in the simulations was chosen $\Delta t = 0.1$. By the mutual ratio of these values, it is possible to obtain the ratio of real values. In the case of a real sampling period other than 1: 133, it is necessary to resample to a frequency corresponding to this ratio. Alternatively, it is possible to retain the model for a new period, which is, however, a much more demanding procedure, and its deployment would be practical only if the real sampling frequency is lower than necessary. When applying the model to real systems, the occurrence of failure scenarios associated with the occurrence of singularities will be suppressed. On the other hand, situations associated with saturation are in practice solved realistically by overflow or closure of the outlet. These models can be used for systems that are similar in shape to model systems, but whose selection covers a significant number of cases. It would always be best to train the models directly from the data of the real system. However, this requires a large number of experiments, in the entire scope of the work area, or to have a long historical data series, from which it would be possible to select or supplement the training set. Due to the unavailability of real data, the application of these models to real systems is beyond the scope of research. Please prepare your paper and data exactly according to the instructions. That is the easiest and the most efficient way to have: a good published paper, clear presentation and successful symposium.

Conclusions should state concisely the most important propositions of the paper as well as the author's views of the practical implications of the results. Be brief and give most important conclusions from your paper. Do not use equations and figures here.

In the experimental part, an algorithm using the principles of classical and evolutionary methods was created and compared with the classical method of quadratic programming. It has been shown that similar computational complexity (depending on the settings) can be achieved as with the QuadProg () method. It was later even found that the computational complexity could be further reduced through parallelization, but the solution to this issue is already beyond the scope. Nevertheless, the algorithm passed even in an unparallelized version. Furthermore, mathematical shapes of model systems (tanks) were derived, the shape of which is similar to the shapes of systems occurring in practice and nature. Besides, these model systems have been non-dimensionalized, which allows their scalability and deployment to different types of systems with different ranges while meeting the conditions for their applicability. Furthermore, models of systems based on the RBF neural network were created. Training sets were created and the parameters of these models were determined by training on model systems. Furthermore, their calibration curves for stable output values were determined. Because the original model systems were non-dimensionalized and scalable, the created models and calibration curves also have these properties. If the conditions of applicability are met, it is therefore possible to use these models to control real systems. Due to the unavailability of data and the impossibility of experimenting with a real system such as water reservoirs, the possibilities of transferring the obtained models into practice are analyzed and the principles of how to proceed are formulated, thus fulfilling the last of the objectives.

7. Acknowledgments

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Program Project No. LO1303 (MSMT-7778/2014) and also by the European Regional Development Fund under the project CEBIA-Tech ED2.1.00/03.0089 and by the Internal Grant Agency of Tomas Bata University under the project No. IGA/CebiaTech/2020/003, project No. IGA/FAI/2020/003 and Institute of Security Engineering, Faculty of Applied Informatics.

8. References

- [1] Kluever, Craig A. (2015). Dynamic systems: modeling, simulation, and control. Hoboken, NJ: John Wiley and Sons, Inc. ISBN 9781118289457.
- [2] Rossiter, J. (2003). Model-based Predictive Control: A Practical Approach. CRC Press.
- [3] Khaled, N.; Bibin Pattel. (2018). Practical design and application of model predictive control: MPC for MATLAB and Simulink users. Kidlington, Oxford: Butterworth-Heinemann, an imprint of Elsevier. ISBN 9780128131196.
- [4] Rawlings, J. B.; David Q. Mayne. (2009). Model predictive control: theory and design. Madison, Wis.: Mob Hill Pub. 533 s. ISBN 9780975937709.
- [5] Mikleš, J.; Fikar M. (2004). Modeling, identification and process control 2. Identification and optimal control. STU Press, Bratislava. 260 pp. ISBN 80-227-2134-4.
- [6] Bobál, V. (2008). Adaptive and predictive control. Edition 1. Zlin: Tomas Bata University in Zlin., 134 pp. ISBN 978-80-7318-662-3.
- [7] Ploskas, N.; Nikolaos Samaras. (2017). Linear programming using MATLAB. Cham: Springer. ISBN 9783319659190.
- [8] Kochendefter, M. J. and Tim A. Wheeler. (2019). Algorithms for optimization. ISBN 9780162039420.
- [9] Wang, L. (2009). Model predictive control system design and implementation using MATLAB. London: Springer. 375 s. ISBN 9781848823303.
- [10] Lee Gue Myung, Tam N.N., Yen Nquyen Dong. (2005). Quadratic Programming and Affine Variational Inequalities: A Qualitative Study. Springer US. ISBN 978-0-387-24278-1.
- [11] Back, T.; B. Fogel David; Michalewicz Zbigniew. (1997). Handbook of Evolutionary Computation. Oxford University Press. 988 pp. ISBN 0750303927.
- [12] Mirjalili, S. (2018). Evolutionary algorithms and neural networks: theory and applications. ISBN 9783319930244; 1860-949X.
- [13] Simon, D. (2013). Evolutionary optimization algorithms: Biologically-Inspired and population-based approaches to computer intelligence. Hoboken, New Jersey: John Wiley & Sons, Inc. ISBN 9780470937419.
- [14] Anonymous. (2018). Selecting a Model Structure in the System Identification Process. Available from: <http://www.ni.com/product-documentation/4028/en/>.
- [15] Russell, S. J.; Peter Norvig. (2010). Artificial intelligence: a modern approach. Upper Saddle River, N.J.: Prentice Hall. 1132 pp. ISBN 0132071487.