Robust model predictive control using neural networks

Krzysztof Patan[1], Piotr Witczak[2]

Abstract— **The paper deals with robust model predictive control designed using recurrent neural network. A dynamic neural network is trained to act as the one-step ahead predictor, which is then used successively to obtain k-step ahead prediction of the plant output. Based on the neural predictor, the control law is derived solving a constrained optimization problem. The robustness of the considered predictive scheme is derived using the concept of an error model. Based on the developed robust model, a optimization problem is redefined. Two solutions are portrayed. The first one is to change the cost function in order to consider the robust model of the plant, while the second one is to impose constraints on the process output using derived uncertainty bands.**

I. INTRODUCTION

Model Predictive Control (MPC) is the subject of intensive research for the last three decades [3], [6], [15]. This research effort has succeeded in many practical applications [12], [5], [1]. The attractiveness of predictive control algorithms comes from its natural ability to consider process and technological constraints imposed on input, output or state variables. The second very important reason is that the operating principles are understandable and relatively easy to explain to practitioners which seems to be a crucial aspect during implementation of a new control scheme in the industry.

A crucial question concerning MPC is its robustness against model uncertainty and noise. Robustness of a control system is referenced to a specific uncertainty range and specific stability and performance criteria. In spite of a rich literature devoted robust control of linear systems, very little is known about the robust control of linear systems with constraints as well as nonlinear systems. In general, if we talk about the robustness issues we assume that the uncertainty of the model follows from two main sources [2], [14]: (i) unmodelled dynamics of a plant, (ii) unmeasured noise/disturbances which enter the plant. In the framework of linear time-invariant systems different approaches have been proposed, e.g. impulse/step responses, a polytopic uncertainty or bounded input disturbances. Generally speaking, the existing methods can be divided into two classes: structured and unstructured uncertainties [3], [14]. These uncertainty descriptions, however, are very useful in the case of linear time-invariant systems, especially using H_∞ paradigm. Unfortunately, they cannot be used in the framework of nonlinear systems.

In this paper the nonlinear predictive control based on Generalized Predictive Control (GPC) concept is considered. This kind of predictive control is well suited for single variable control as well as for adaptive control. To deal with a nonlinearity of a plant a dynamic neural network is applied. Neural networks provide an interesting and valuable alternative to classical methods, because they can easily deal with the most complex situations which are not sufficiently defined for deterministic algorithms to execute. They are especially useful when there is no mathematical model of a plant. Moreover, neural networks provide an excellent tool for dealing with nonlinear problems [9]. These features allow for designing adaptive control systems for complex, unknown and nonlinear dynamic processes [7], [8], [10]. As the neural network is a nonparametric model, it represents a plant dynamics as well as nonlinearity in a distributed way using all weight parameters. Thus, common approaches to robust MPC synthesis listed in the literature cannot be used here as they are devoted to linear systems with parametric representation [14]. Recently, neural networks has been applied to robust MPC synthesis [16]. However, is such an approach, recurrent neural networks are used to solve minimax oprimal control problem.

The purpose of the paper is to propose a method which is able to cope with the problem of the uncertainty associated with the neural network model in the framework of robust predictive control. The presented approach is based on the Model Error Modelling (MEM) portrayed in the paper of Reinelt and co-workers [13], but the MEM version in time domain was described in the previous work of the author [10]. The idea is to design a robust model of a plant using available data. This data should represent well possibly wide range of operating conditions of the plant. Using the robust model, the uncertainty region is derived, which is then used to redefine the open-loop optimal control problem. Two solutions are proposed. The first one is to change the cost function in order to consider the robust model of the plant, while the second one is to impose constraints on the process output using derived uncertainty bands. The process output trajectory will be included inside the uncertainty band when the same input is applied, in spite of uncertainties.

The paper is organized as follows. After Introduction, in Section II, uncertainty modelling is presented. Then, in Section III basics about nonlinear model predictive control based on a neural network predictor is portrayed. The next section contains propositions and solutions for robust model predictive control synthesis. Some experimental results are shown and discussed in Section V. The last section contains conclusions and final remarks.

[1]K. Patan is with Institute of Control and Computation Engineering, University of Zielona Góra, Poland, k.patan@issi.uz.zgora.pl

[2] P. Witczak is a Ph.D. student in Faculty of Electrical Engineering, Computer Science and Telecommunication, University of Zielona Góra, Poland.

II. Modelling and uncertainty description

Let assume that the process is represented by the following prediction form of the nonlinear difference equation:

$$\hat{y}(k+1) = f(y(k), \ldots, y(k-n_a+1), u(k), \ldots, u(k-n_b+1)), \quad (1)$$

where f is a nonlinear mapping function, $y(k)$ is the output of the process at time k, n_a and n_b are the number of past outputs and inputs considered by the model, respectively. To design a nonlinear model of the process artificial neural networks can be successfully used. Neural networks proved their usefulness in modelling of nonlinear dynamic processes [8], [4], [10]. In the field of neural modelling the simplest solution is to use the feedforward networks with external dynamics [10], [8]:

$$\hat{y}(k+1) = f(\boldsymbol{x}) = \sigma_o(\boldsymbol{W}_2 \sigma_h(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2), \quad (2)$$

where

$$\boldsymbol{x} = [y(k), \ldots, y(k-n_a+1), u(k), \ldots, u(k-n_b+1)]^T,$$

$\boldsymbol{W}_1 \in \mathbb{R}^{n_a+n_b \times v}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{v \times 1}$ are weight matrices of hidden and output layers, respectively, $\boldsymbol{b}_1 \in \mathbb{R}^v$ and $\boldsymbol{b}_2 \in \mathbb{R}^1$ are bias vectors of hidden and output units, respectively, $\sigma_h : \mathbb{R}^v \to \mathbb{R}^v$ is the activation function of the hidden layer, and $\sigma_o : \mathbb{R}^1 \to \mathbb{R}^1$ is the activation function of the output layer and v stands for the number of hidden neurons. Weight matrices as well as bias vectors are subject of training based on historical data recorded in a plant. Mostly, training is carried out off-line.

A perfect identification procedure does not exist. Every modelling procedure either for linear or nonlinear processes suffers from the so called the model mismatch, c.f. the model of the system is not a faithful replica of plant dynamics. On this basis, the uncertainty can be seen as a measure of unmodelled dynamics, noise and disturbances. Frequently, model uncertainty is considered as located in parameters. However, in such a case perfect decoupling residuals from uncertainties is limited by the number of available measurements. Alternative way is to propagate uncertainty into residuals. Then, all modelling errors are represented by the function $w(k)$ and the plant is represented by the following family of models:

$$\bar{y}(k+1) = \hat{y}(k+1) + w(k), \quad (3)$$

where $w(k) \in \mathcal{W}$ represents the additive uncertainty and $\mathcal{W}$ is a compact set. Then, the model uncertainty is defined as

$$\underline{w}(k) \leqslant w(k) \leqslant \overline{w}(k),$$

and all possible trajectories are included in bands that depend on lower $\underline{w}(k)$ and upper $\overline{w}(k)$ uncertainty estimates. This kind of uncertainties if often called global uncertainties [3]. The only assumption made here is that they are bounded. In general $w(t)$ may be a function of past inputs and outputs. To estimate model uncertainty model error modelling can be used. MEM employs prediction error methods to identify a model from input-output data [13]. The first step illustrated in Fig. 1 is to model a process without uncertainty

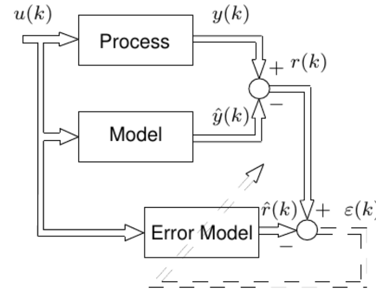

Fig. 1. Deriving of the model uncertainty.

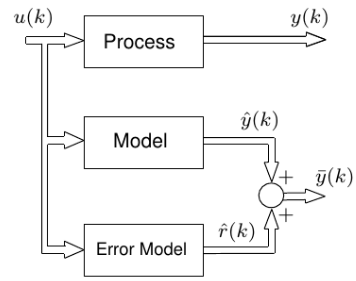

Fig. 2. Construction of the robust model.

considerations. Then, the uncertainty model is designed by analyzing the residual $r(k)$ evaluated from the inputs (Fig. 2). The residual is a signal defined as a difference between the process output and the output of a model:

$$r(k) = y(k) - \hat{y}(k). \quad (4)$$

Modelling of residuals provides the so-called *error model*. To identify an error model it is required to collect a set of the data $\{u(i), r(i)\}_{i=1}^N$. This can be done during testing of the process model, preferably in closed-loop control. The error model constitutes an estimate of the error due to undermodelling. To design of the error model one can use well known linear models such as the Finite Impulse Response (FIR) model of the high order or the AutoRegressive with eXogenous input (ARX) model [13]. If the accuracy of these models is not acceptable one can use nonlinear models, e.g. dynamic neural networks presented earlier in this section. From general point of view the error model is described by the following difference equation:

$$\hat{r}(k+1) = f_e(r(k), \ldots, r(k-n_{n_a}+1), u(k), \ldots, u(k-n_{n_b}+1)), \quad (5)$$

where $\hat{r}(k+1)$ represents an estimate of the residual at the time instant $k+1$, n_{n_a} and n_{n_b} represents the number of past residuals and inputs needed for designing the error model, respectively. Final representation of a robust model is given as:

$$\bar{y}(k) = \hat{y}(k) + \hat{r}(k). \quad (6)$$

Assuming that $\hat{r}(k)$ has the normal distribution, statistical properties of the error model output can be used to estimate the uncertainty of the model. The confidence region forms uncertainty bands around response of the robust model: the upper band

$$\overline{w}(k) = \bar{y}(k) + t_\alpha \sigma \tag{7}$$

and the lower band

$$\underline{w}(k) = \bar{y}(k) - t_\alpha \sigma, \tag{8}$$

where t_α is a tabulated value of the normal distribution $\mathcal{N}(0,1)$ assigned to $1-\alpha$ confidence level, σ is the standard deviation of $\hat{r}$ calculated over a testing set of data. The centre of the uncertainty region is the signal $\bar{y}(k)$ [10].

III. Nonlinear model predictive control

Let us consider a nonlinear version of generalized predictive control. Let introduce the cost function:

$$J = \sum_{i=N_1}^{N_2} e^2(k+i) + \rho \sum_{i=1}^{N_u} \Delta u^2(k+i-1), \tag{9}$$

where $e(k+i) = y_r(k+i) - \hat{y}(k+i)$ is the tracking error, $y_r(k+i)$ is the reference at time $k+i$, $\hat{y}(k+i)$ is the prediction of future process outputs, $\Delta u(k+i-1) = u(k+i-1) - u(k+i-2)$, $u(k)$ is the control signal at time k, N_1 is the minimum prediction horizon, N_2 is the prediction horizon, N_u is the control horizon, ρ represents a factor penalizing changes in the control signal. Additionally, the following constraints are considered:

- constraints on control moves

$$\Delta u(k+i) = 0, \quad N_u \leqslant i \leqslant N_2 - 1,$$

- constraints on process variables

$$\underline{v} \leqslant v(k+j) \leqslant \overline{v}, \quad \forall j \in [N_{v1}, N_{v2}],$$

  where N_{v1} and N_{v2} are the lower and upper constraint horizons, respectively, and $[\underline{v}, \overline{v}]$ defines the allowed space for a variable v,
- terminal constraints

$$e(k+N_p+j) = 0, \quad \forall j \in [1, N_c],$$

  where N_c is the terminal constraint horizon.

A nonlinear model predictive control can be defined as the following open-loop optimization problem:

$$\boldsymbol{u}(k) \triangleq \arg\min J \tag{10a}$$

$$\text{s.t.} \quad e(k+N_2+j) = 0, \quad \forall j \in [1, N_c], \tag{10b}$$

$$\Delta u(k+N_u+j) = 0, \quad \forall j \geqslant 0, \tag{10c}$$

$$\underline{u} \leqslant u(k+j) \leqslant \overline{u}, \quad \forall j \in [0, N_u - 1], \tag{10d}$$

$$\underline{y} \leqslant \hat{y}(k+j) \leqslant \overline{y}, \quad \forall j \in [N_1, N_2], \tag{10e}$$

where $\underline{u}$ and $\overline{u}$ are lower and upper control bounds, $\underline{y}$ and $\overline{y}$ limit the allowed space for the predictions of the plant output. The objective of an optimization procedure is to find the optimal control sequence satisfying all constraints but the first element of this sequence is used for the control purposes. The optimization procedure should assure fast convergence and numerical robustness. Second-order optimization algorithms based on Newton and Levenberg-Marquardt methods seem to be a reasonable choice [8], [11]. Moreover, they allow relatively easy to consider constraints imposed on process variables, i.e. using a penalty cost.

IV. Robust MPC synthesis

A. Unmeasured disturbances

To deal with unmeasured disturbances discussed in Section II, the model of a process can be equiped with the additional term $d(k)$. This term stands for the disturbance model defined in the following way:

$$d(k) = Kr(k) + d(k-1), \tag{11}$$

where $r(k)$ is a residual defined by (4), K represents the gain of the disturbance model. As the disturbance model (11) includes an integrator, the offset-free steady-state behaviour of the control system can be achieved [3], [15]. Frequently, K is assumed to be equal to 1 and $d(k)$ is assumed to be constant within the prediction horizon [3]. Another idea is to use very simple form based on the residual [3]:

$$d(k) = Kr(k). \tag{12}$$

Considering unmeasured disturbances $d(k)$ the predictor (2) can be rewritten in the form:

$$\hat{y}(k+1) = f(\boldsymbol{x}) + d(k), \tag{13}$$

and assuming that $d(k)$ is constant within the prediction horizon, implementation of the optimization procedure of the problem (10) does not change. The only problem here is to find a proper description of the unmeasured disturbances. Representations (11) and (12) are only propositions.

B. Modified cost function

Assuming the uncertainty description of the model discussed in Section II, the robust MPC can be achieved by modifying the cost function in such a way that instead of the output of the nominal model $\hat{y}(k)$ the cost uses the output of the robust model $\bar{y}(k)$. The cost function becomes:

$$\bar{J} = \sum_{i=N_1}^{N_2} (y_r(k+i) - \bar{y}(k+i))^2 + \rho \sum_{i=1}^{N_u} \Delta u^2(k+i-1). \tag{14}$$

In order to define the cost (14) it is needed to determine i-step ahead predictions of the error model $\hat{r}(k+i)$. These predictions can be easily calculated by successive recursion of the one-step ahead prediction $\hat{r}(k+1)$. However, it should be kept in mind that residuals are available up to time k, then one should substitute residual predictions for actually calculated residuals for $i > 1$ as follows:

$$r(k+i) = \hat{r}(k+i), \quad \forall i > 1.$$

As the uncertainty description is included in the robust model itself the output constraints (10e) are not used during optimization. However, the optimization routine becomes slightly more complex due to the requirement of calculating partial derivatives $\frac{\partial \hat{r}(k+i)}{\partial \boldsymbol{u}(k)}$ and $\frac{\partial^2 \hat{r}(k+i)}{\partial \boldsymbol{u}^2(k)}$. If the neural network is used to design the error model, these derivatives are calculated analogously to derivatives $\frac{\partial \hat{y}(k+i)}{\partial \boldsymbol{u}(k)}$ and $\frac{\partial^2 \hat{y}(k+i)}{\partial \boldsymbol{u}^2(k)}$ assigned to the nominal model of a process. Detailed formuale for determining these derivatives can be found in [8] on pages: 183–186.

C. Output constraints

Another way to achieve robustness of MPC is to properly define output constraints. In this case, the cost function is based on the nominal model of the process (9) but robustness of the control system is achieved by using output constraints which include the uncertainty description according to (7) and (8). Then, the inequality constraint (10e) can be represented in the following way:

$$\overline{g}_i(\boldsymbol{u}) \;=\; \hat{y}(k+i) - \overline{w}(k+i), \tag{15}$$

$$\underline{g}_i(\boldsymbol{u}) \;=\; \underline{w}(k+i) - \hat{y}(k+i), \tag{16}$$

where $i \in [N_1, N_2]$. A popular approach for considering constraints is to transform the original problem to its alternative unconstrained form using a penalty cost. Let us use constraints representation (15) and (16), then the penalty cost function can be defined as follows:

$$\tilde{J}(k) = J(k) + \lambda \sum_{i=N_1}^{N_2} \overline{g}_i^2(\boldsymbol{u}) S(\overline{g}_i(\boldsymbol{u})) + \lambda \sum_{i=N_1}^{N_2} \underline{g}_i^2(\boldsymbol{u}) S(\underline{g}_i(\boldsymbol{u})), \tag{17}$$

where $S(x) = 1$ if $x > 0$ and $S(x) = 0$ otherwise. The function $S(x)$ makes it possible to consider a set of active inequality constraints at the current iterate of the algorithm. The principle of operation in this case is as follows. Before the optimization begins, the uncertainty bands $\underline{w}(k+i)$ and $\overline{w}(k+i)$ are determined based on the current control $\boldsymbol{u}(k)$. Then the optimization procedure starts in order to determine a new control sequence subject to constraints. During the optimization, $\underline{w}(k+i)$ and $\overline{w}(k+i)$ are independent on the variable $\boldsymbol{u}(k)$. Based on this, optimization of the penalty function does not require to calculate additional partial derivatives. Partial derivatives of the penalty terms use the already defined partial derivatives $\frac{\partial \hat{y}(k+i)}{\partial \boldsymbol{u}(k)}$ and $\frac{\partial^2 \hat{y}(k+i)}{\partial \boldsymbol{u}^2(k)}$. Taking into account a computation burden this approach is less complex than the solution presented in the previous section. Thus, all active inequality constraints are taken into account during the optimization. Now, the objective is to solve the following unconstrained problem:

$$\boldsymbol{u}(k) \triangleq \arg\min \bar{J}(\boldsymbol{u}). \tag{18}$$

D. Robust performance

The performance of the proposed robust control schemes is tested using the multiplicative output uncertainty portrayed in Fig. 3. The presented scheme can be regarded as the parametric uncertainty of the plant gain. Parameter v representing the gain is bounded within a region $[v_{min}, v_{max}]$ and can be represented as:

$$v = \bar{v}(1 + \gamma\Delta), \tag{19}$$

where $\bar{v}$ is the nominal (mean) parameter value, Δ is any real scalar satisfying $|\Delta| \leqslant 1$, and γ represents the relative uncertainty in the parameter v:

$$\gamma = \frac{v_{max} - v_{min}}{v_{max} + v_{min}}. \tag{20}$$

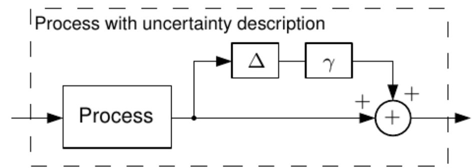

Fig. 3. Process with output multiplicative uncertainty.

V. Experiments

As the illustrative example, a pneumatic servomechanism for control the position of the mass is used [8]. The pneumatic servomechanism is a system, in which power is supplied and transmission of signals is carried out through the medium of compressed air. Pneumatic servos have the advantages of low cost, high power to weight ratio, easy of maintenance, cleanliness and a readily available and cheap power source. The system consists of the double acting cylinder lifting an inertial weight. The cylinder is fed from a set of four adjustable air valves. The pneumatic servomechanism is a nonlinear system. The main nonlinear behaviour follows from nonlinear friction forces, deadband due to stiction and dead time due to the compressibility of air. The scheme of the process in shown in Fig. 4 and technical data is listed in Table I. The valves are operated in such a way that for input signal $u \geqslant 0$ the valves S_1 and S_4 are open and for $u < 0$ valves S_2 and S_3 are open. All valves open proportionally to their control signals. The sampling frequency is set to 10Hz. The considered pneumatic servo can be viewed as a SISO system with the control signal of valves as the input and the piston position as the output. The specificity of the process (process is poorly damped and includes integration action) makes it difficult to generate data for neural network training. Then it is necessary to operate in closed-loop control with the P controller. The gain of the controller was set to $K_p = 10$.

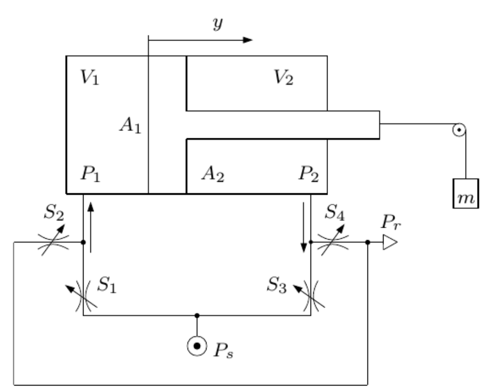

Fig. 4. Scheme of pneumatic servomechanism

TABLE I

PROCESS SPECIFICATION

Symbol	Description	Value/unit
V_1	cylinder volume	0.4908 liter
V_2	cylinder volume	0.4123 liter
A_1	chamber 1 area	19.63 cm^3
A_2	chamber 2 area	16.49 cm^3
P_s	supplied pressure	6 bar
P_r	exhaust pressure	1 bar
m	load mass	20 kg
y	piston position	[-0.245,0.245] m

A. Modelling

Training data was collected using the reference signal in the form of random steps with levels covering possible piston positions from the interval $(-0.245, 0.245)$. Changes of the reference were also triggered randomly. Additionally, the response of the process was contaminated by the white noise with the magnitude equal to 5% of the output signal. The order of the model was selected after analysing the physical properties of the process. Finally, a model of the fourth order ($n_a = n_b = 4$) was used. After some trails the number of hidden neurons was found to be equal to 8 ($v = 8$). Hidden neurons consist of hyperbolic tangent activation function and the output neuron has the linear activation function. Modelling results are presented in Fig. 5, where the output of the plant is marked with the solid line, and the output of the model with the dashed one.

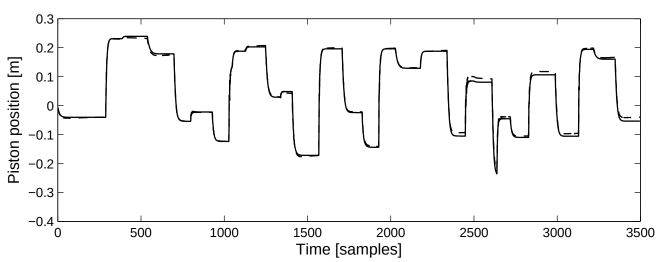

Fig. 5. Modelling results: outputs of process (solid) and model (dashed).

B. Uncertainty modelling

To estimate uncertainty associated with the neural model the model error modelling is applied. Firstly, the classical linear models (FIR and ARX) were applied without success. Then, the nonlinear version of ARX model realized by means of the neural network (2) was applied. To design the error model the training data was recorded in closed loop control where the predictive controller used the nominal model of the servo. Additionally, the gain uncertainty has been modelled according to scheme presented in Fig. 3 with $\gamma = 0.2$ and Δ generated in a random manner every 10 time steps. During research, the best error model was found to have 10 hidden neurons with the hyperbolic tangent activation function and one linear output neuron. The number of past residuals n_{n_a} was equal to 2 and the number of past inputs n_{n_b} was equal to 10. Uncertainty modelling results are presented in Fig. 6, where the output of the plant is marked with the solid line, the output of the nominal model with the dashed one, and the output of the robust model (6) with the dotted line.

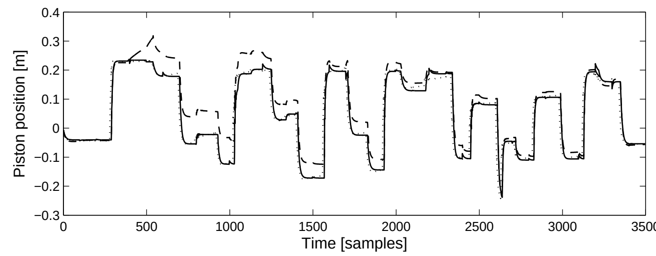

Fig. 6. Uncertainty modelling: outputs of process (solid), model (dashed) and error model (dotted).

C. Control and robust performance

During experiments four predictive schemes have been investigated. The first one is the model predictive control without robustness considerations. After thorough tuning, the MPC controller parameters are set as follows: $N_1 = 1$, $N_2 = 10$, $N_u = 2$, $\rho = 0.003$. Such experiment settings were found using trial and error procedure and assure pretty good performance of the fundamental control system. The control sequence is constrained with the upper control bound $\overline{u} = 4$, and the lower control bound $\underline{u} = -4$. Such control range was used during training. Additionally, three robust schemes are investigated. The first one used the disturbance model (12) to consider unmeasured disturbances affecting the system. After some trials the parameter $K = 0.01$. The second robust scheme used the output constraints discussed in Section IV-C. In this case the $\rho = 0.001$ and $\lambda = 0.1$. The third robust scheme is a combination of the previous two. The performance of the proposed control schemes has been tested using:

(i) different reference signals, namely random steps, modified ramp and sinusoidal signals,
(ii) parametric gain uncertainty (19), with $\gamma = 0.2$ and Δ generated randomly every 10 time steps,
(iii) white noise affecting the process output.

The quality of control is represented by an index Q_{SSE} defined as a sum of squared errors between the reference and process output. Abbreviations used in Tables mean: MPC – model predictive contol without robustness, MPCD – model predictive control with disturbance model (12), RMPC – robust model predictive control considered in Section IV-C, RMPCD – robust model predictive control with disturbance model. The considered predictive schemes are also compared with the P controller. Figures 7–9 show control results in normal operating conditions achieved by the P controller (dashed line) and predictive controllers (dotted line) for different reference signals considered. For clarity of presentation only results for RMPCD controller are presented there. In all cases the predictive schemes performed much better that the classical P controller. It is important to say that it was assumed that within the prediction horizon the reference signal is unknown. Tables II–IV include results of control for different reference signals and different conditions

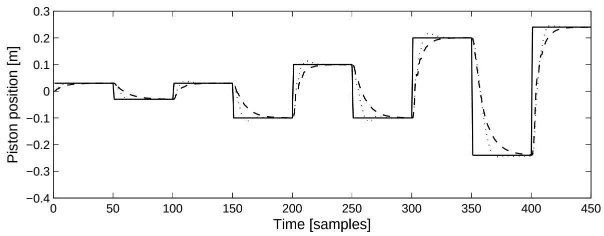

Fig. 7. Control results: reference (solid), P controller (dashed) and MPC (dotted).

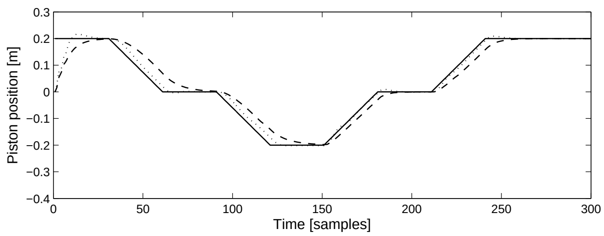

Fig. 8. Control results: reference (solid), P controller (dashed) and RMPCD (dotted).

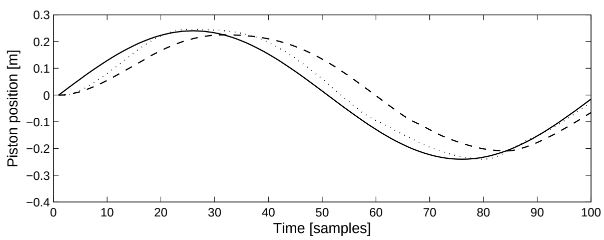

Fig. 9. Control results: reference (solid), P controller (dashed) and RMPCD (dotted).

TABLE II

CONTROL RESULTS FOR RANDOM STEPS REFERENCE.

Controller type	nominal work	parameter variation	noise
MPC	2.4019	2.2632	2.3848
MPCD	2.3011	2.2555	2.2926
RMPC	2.2545	2.1151	2.2612
RMPCD	2.2455	2.1091	2.2394

TABLE III

CONTROL RESULTS FOR MODIFIED RAMP REFERENCE.

Controller type	nominal work	parameter variation	noise
MPC	0.1977	0.2751	0.2277
MPCD	0.1704	0.2483	0.2004
RMPC	0.1599	0.2364	0.1908
RMPCD	0.1589	0.2364	0.1895

(parameter uncertainty and external disturbances). The robust version of MPC works pretty well. In all cases a fusion of robust control with disturbance model gave the best results, marked with frames.

VI. CONCLUSIONS

The purpose of the paper was to propose a new method for robust nonlinear model predictive control synthesis. Two solutions were described. The first one uses the modified cost function in order to consider the robust model of the plant, while the second one imposes constraints on the process output using derived uncertainty bands. Both proposition are based on the model error modelling which uses two neural network models: fundamental model of the plant and the so-called error model. At the moment only the second solution was tested using different operating condition of the plant giving promising results. The future work will be focused on the implementation of the first solution proposed in Section IV-B and to compare it with these considered in Section V.

TABLE IV

CONTROL RESULTS FOR SINUSOIDAL REFERENCE.

Controller type	nominal work	parameter variation	noise
MPC	0.128	0.1463	0.1398
MPCD	0.0852	0.0976	0.0968
RMPC	0.0856	0.0997	0.097
RMPCD	0.0849	0.0973	0.0957

REFERENCES

[1] A. Alessandri, C. Cervellera, and M. Gaggero, "Predictive control of container flows in maritime intermodal terminals," *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 1423–1431, 2013.

[2] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control*, ser. Lecture Notes in Control and Information Sciences, A. Garulli and A. Tesi, Eds. London: Springer–Verlag, 1999, pp. 207–226.

[3] E. F. Camacho and C. Bordóns, *Model Predictive Control*. Berlin: Springer-Verlag, 2004.

[4] S. Haykin, *Neural Networks. A Comprehensive Foundation, 2nd Edition*. New Jersey: Prentice-Hall, 1999.

[5] A. Koerber and R. King, "Combined feedback-feedforward control of wind turbines using state-constrained model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 1117–1128, 2013.

[6] J. Maciejowski, *Predictive Control with Constraints*. Harlow: Prentice-Hall, 2002.

[7] O. Nelles, *Nonlinear System Identification. From Classical Approaches to Neural Networks and Fuzzy Models*. Berlin: Springer–Verlag, 2001.

[8] M. Norgard, O. Ravn, N. Poulsen, and L. Hansen, *Networks for Modelling and Control of Dynamic Systems*. London: Springer-Verlag, 2000.

[9] K. Patan, "Approximation of state-space trajectories by locally reccurent globally feed-forward neural networks," *Neural Networks*, vol. 21, pp. 59–64, 2008.

[10] ——, *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*, ser. Lecture Notes in Control and Information Sciences. Berlin: Springer–Verlag, 2008.

[11] K. Patan and J. Korbicz, "Nonlinear model predictive control of a boiler unit: A fault tolerant control study," *Applied Mathematics and Computer Science*, vol. 22, no. 1, pp. 225–237, 2012.

[12] I. Prodan, S. Olaru, and S.-I. N. C. Stoica, "Predictive control for trajectory tracking and decentralized navigation of multi-agent formations," *Applied Mathematics and Computer Science*, vol. 23, no. 1, pp. 91–102, 2013.

[13] W. Reinelt, A. Garulli, and L. Ljung, "Comparing different approaches to model error modeling in robust identification," *Automatica*, vol. 38, pp. 787–803, 2002.

[14] S. Skogestad and P. I., *Multivariable feedback control. Analysis and design*. New York: John Wiley & Sons, 2005, 2nd edition.

[15] P. Tatjewski, *Advanced Control of Industrial Processes*. London: Springer, 2007.

[16] Z. Yan and J. Wang, "Robust model predictive control of nonlinear systems with unmodelled dynamics and bounded uncertainties based on neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 457–469, 2014.