Robust model predictive control using neural networks

Krzysztof Patan and Piotr Witczak

Institute of Control and Computation Engineering
University of Zielona Góra

Introduction

- Model Predictive Control (MPC) – modern control strategy
- Neural networks – useful when dealing with nonlinear problems
- Robustness against model uncertainty and noise – a crucial question
- Robustness of nonlinear control system – still a challenge
- Open problems – how to deal with robustness of neural network based MPC
- Possible solution, min-max optimization, is time-consuming
- Purpose of the paper – to cope with model uncertainties using Model Error Modelling (MEM) and properly redefine the open-loop optimal control problem using uncertainty definition provided by MEM

Modelling and uncertainty estimation

Neural predictor

- One-step ahead prediction

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

where n_a and n_b represent number of past outputs and inputs, respectively

- Function f can be realized using dynamic neural network

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

where

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

$\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ – weight matrices, σ_h and σ_o – activation functions

- i-step ahead prediction

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

- Measurements of the output are available up to time k – one should substitute predictions for actual measurements since these do not exist

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

Uncertainty description

- Uncertainty of the model is a measure of unmodelled dynamics, noise and disturbances
- Plant is represented by the family of models

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

  where $w(k) \in \mathcal{W}$ – the additive uncertainty, $\mathcal{W}$ – a compact set
- All possible trajectories are bounded by lower $\underline{w}(k)$ and upper $\overline{w}(k)$ uncertainty estimates

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

- $w(k)$ may be a function of past inputs and outputs

Robust model

- Model uncertainty estimation – Model Error Modelling
- MEM analyzes residual signal

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

- Nonlinear form of the error model

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

where $\hat{r}(k+1)$ – an estimate of the residual at the time instant $k+1$

n_{n_a} and n_{n_b} – the number of past residuals and inputs, respectively

- Final representation of a robust model

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

- The upper band

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

- The lower band

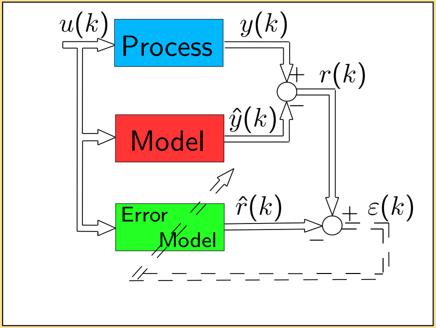
$$\underline{w}(k) = \bar{y}(k) - t_\alpha \sigma$$

where t_α – $\mathcal{N}(0, 1)$ tabulated value assigned to $1 - \alpha$ confidence level

σ – the standard deviation of the error model output

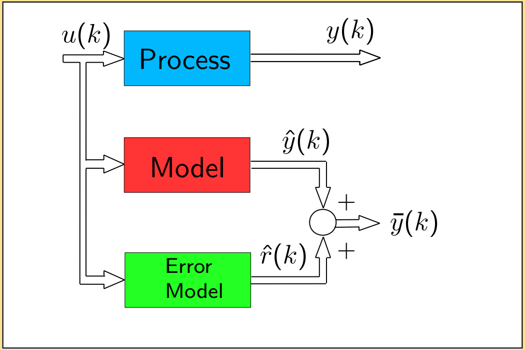
MEM procedure - step 1

1. collect the data $\{u(i), r(i)\}_{i=1}^{N}$ and identify an error model using these data. This model constitutes an estimate of the error due to under modelling, and it is called model error model



MEM procedure - step 2

2. construct a model along with uncertainty using both nominal and model error models



Nonlinear MPC

- Cost based on the GPC criterion

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

where $e(k+1) = y_r(k+i) - \hat{y}(k+i)$
$\Delta u(k+i-1) = u(k+i-1) - u(k+i-2)$
$y_r(k+i)$ – the future reference signal
$\hat{y}(k+i)$ – the prediction of future outputs
$u(k)$ – the control signal at time k
$\Delta u(k+i-1)$ – control change
ρ – the factor penalizing changes in the control signal

- Constraints on control moves

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

- Constraints on process variable v

$$\underline{v} \leqslant v(k+j) \leqslant \overline{v}, \quad \forall j \in [0, N_v]$$

where N_v – constraint horizon
$\underline{v}$ – lower limits
$\overline{v}$ – upper limits

- Terminal constraints, e.g.

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

where N_c – terminal constraint horizon

Problem definition

Let us redefine the nonlinear model predictive control based on the following open-loop optimization problem

$$\boldsymbol{u}(k) \triangleq \arg\min J \tag{1a}$$

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

$$\Delta u(k+N_u+j)=0, \quad \forall j \geqslant 0 \tag{1c}$$

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

$$\underline{y} \leqslant \hat{y}(k+j) \leqslant \overline{y}, \quad \forall j \in [N_1, N_2] \tag{1e}$$

where $\underline{u}$, $\overline{u}$ – lower and upper control bounds

$\underline{y}$, $\overline{y}$ – lower and upper bounds for output predictions

k.patan@issi.uz.zgora.pl

Robust MPC synthesis

- A possible way to achieve robust MPC – defining output constraints
- Then, the inequality constraint (1e) can be represented in the following way:

$$\underline{w}(k+1) \leqslant \hat{y}(k+i) \leqslant \overline{w}(k+i)$$

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

- Transformation of the original problem to its alternative unconstrained form – using a penalty cost:

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

  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 objective is to solve the following unconstrained problem:

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

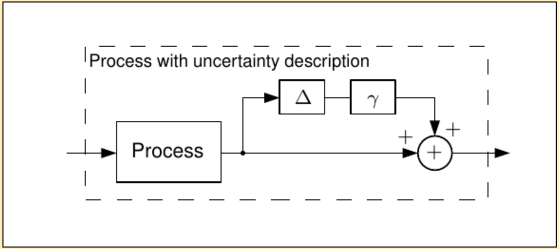
- The principle of operation:
  - before the optimization begins, the uncertainty bands $\underline{w}(k+i)$ and $\overline{w}(k+i)$ are determined based on the current control $\boldsymbol{u}(k)$
  - the optimization procedure starts in order to determine a new control sequence subject to constraints
  - during the optimization, $\underline{w}(k+i)$ and $\overline{w}(k+i)$ are independent on the variable $\boldsymbol{u}(k)$; consequently, optimization of the penalty function does not require to calculate additional partial derivatives.

Unmeasured disturbances

- To deal with unmeasured disturbances, the model of a process can be equipped with the additional term $d(k)$
- Considering unmeasured disturbances $d(k)$ the neural predictor can be rewritten in the form:
$$\hat{y}(k+1) = f(\boldsymbol{x}) + d(k) \tag{2}$$
- Frequently, $d(k)$ is assumed to be constant within the prediction horizon
- assuming that $d(k)$ is constant within the prediction horizon, implementation of the optimization procedure does not change
- The only problem here is to find a proper description of the unmeasured disturbances, e.g.
$$d(k) = Kr(k) \tag{3}$$
where $r(k)$ – the residual, K – the gain of the disturbance model

Performance checking

➢ Multiplicative output uncertainty scheme



➢ Representation of the gain

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

where $\bar{v}$ is the nominal (mean) parameter value

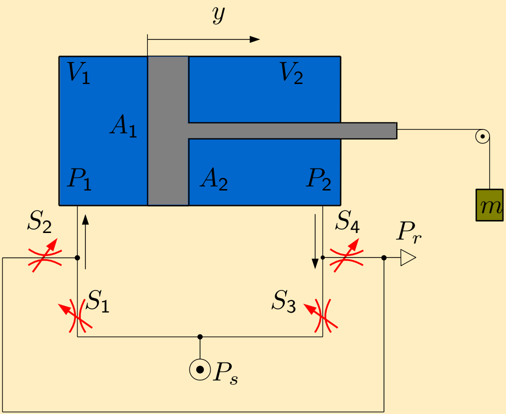
Δ – any real scalar satisfying $|\Delta| \leqslant 1$

γ – the relative uncertainty in the parameter v:

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

Illustrative example

Pneumatic servomechanism

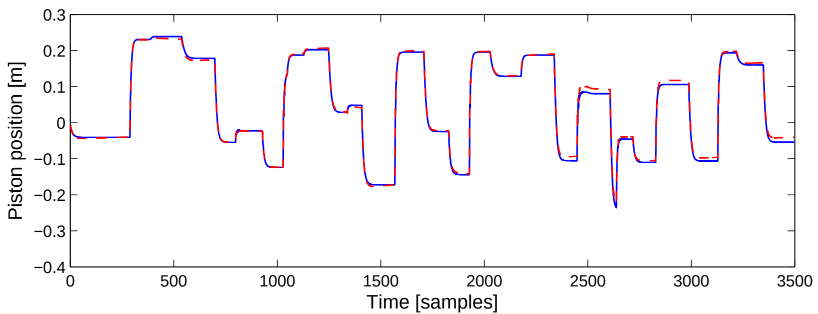


V_1, V_2 – cylinder volumes
A_1, A_2 – chamber areas
P_1, P_2 – chamber pressures
P_s – supplied pressure
P_r – exhaust pressure
m – load mass
y – piston position
$S_1, \dots, S_4$ – operating valves
u – control signal

S_1 and S_4 are open for $u \geqslant 0$
S_2 and S_3 are open for $u < 0$

Modelling

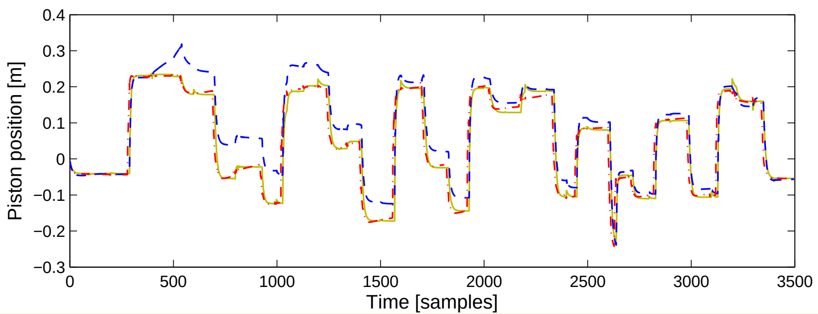
- Training data
  - input in the form of random steps with levels from the interval $(-0.245, 0.245)$
  - output was contaminated by the white noise with the magnitude equal to 5% of the output signal
- Neural model of the fourth order ($n_a = n_b = 4$) was used, 8 tangensoidal neurons in the hidden layer, one linear output neuron



Process output (solid/blue) and model output (dashed/red)

Uncertainty modelling

- Training data recorded in closed loop control
  - predictive controller with nominal model of the plant
  - gain uncertainty with $\gamma = 0.2$ and Δ generated randomly every 10 s
- Neural model specification: $n_{n_a} = 2$, $n_{n_b} = 10$, 10 hidden neurons with hyperbolic tangent activation function, one linear output neuron

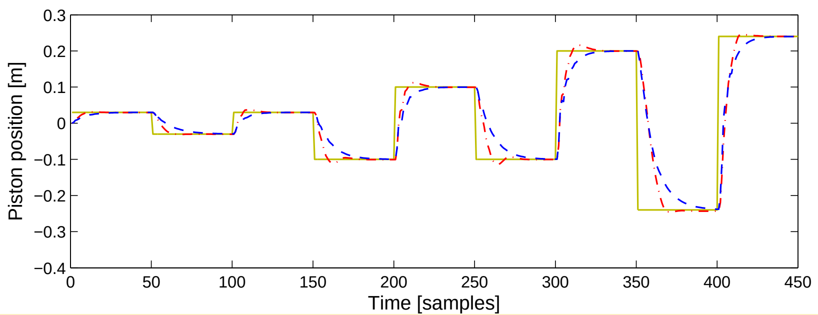


Outputs: process (solid/green), model (dashed/blue), robust model (dotted/red)

Control settings

- Predictive controller set up (MPC): $N_1 = 1$, prediction horizon $N_2 = 10$, control horizon $N_u = 2$, control moves penalty $\rho = 0.003$
- MPC with disturbance model (MPCD): gain $K = 0.01$
- Robust predictive control (RMPC): control moves penalty $\rho = 0.001$, output constraints penalty $\lambda = 0.1$
- Robust predictive control with disturbance model (RMPCD)
- Testing conditions:
  1. nominal work with different reference signals: random steps, ramp signal, sinusoidal signal
  2. parameter uncertainty: $\gamma = 0.2$, Δ generated every 10 time steps
  3. white noise affecting the output
- Quality index – Sum of Squared Errors (SSE) calculated on tracking error

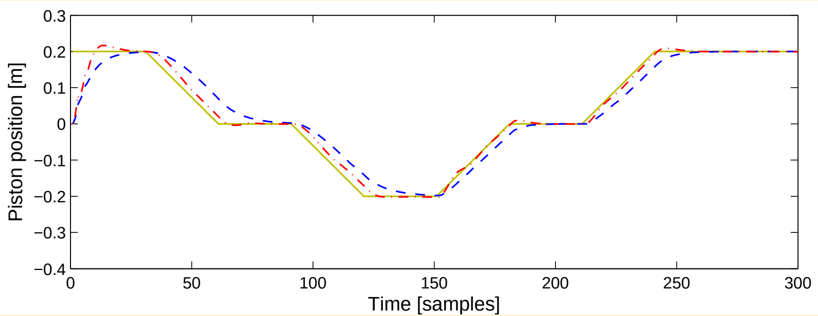
Results for random steps reference



Control: reference (solid/green), P controller (dashed/blue) and robust MPC (dotted/red)

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	$\boxed{2.2455}$	$\boxed{2.1091}$	$\boxed{2.2394}$

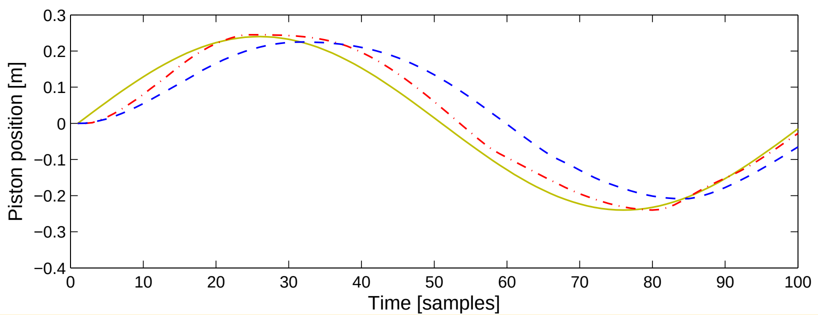
Results for modified ramp reference



Control: reference (solid/green), P controller (dashed/blue) and robust MPC (dotted/red)

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

Results for sinusoidal reference



Control: reference (solid/green), P controller (dashed/blue) and robust MPC (dotted/red)

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	$\boxed{0.0849}$	$\boxed{0.0973}$	$\boxed{0.0957}$

Concluding remarks

- A new method for robust nonlinear model predictive control was proposed
- The approach uses model error modelling carried out by means of dynamic neural networks
- The proposed numerical solution is very simple to implement and no time consuming
- The solution was tested on the pneumatic servomechanism using different working conditions of the plant with promising results
- The future work will be focused on the implementation of the robust MPC where the cost function is redefined 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 $\bar{y}(k)$