Convenient optimization strategy implemented in multivariable predictive control

Marek Kubalcik[1,*], *Vladimir* Bobal[1], and *Tomas* Barot[2]

[1]Department of Process Control, Tomas Bata University in Zlin, Faculty of Applied Informatics, Nad Stranemi 4511, 76005 Zlin, Czech Republic
[2]Department of Mathematics with Didactics, University of Ostrava, Faculty of Education, Mlynska 5, 70103 Ostrava, Czech Republic

Abstract. A significantly important part of model predictive control (MPC) with constraints is a solution of an optimization task. The result of the optimization is a vector of future increments of a manipulated variable. The first element of this vector is applied in the next sampling period of MPC in the framework of a receding horizon strategy. In practical realization of a multivariable MPC, the optimization is characterized by higher computational complexity. Therefore, reduction of the computational complexity of the optimization methods has been widely researched. One suitable principle of precomputing operations was proposed by Wang, L. This general optimization strategy is further modified in this paper. Decreasing of the computational complexity of the optimization by using of the proposed modification is discussed.

1 Introduction

Model predictive control (MPC) [1]-[2] has been widely applied in controlling of industrial processes thanks to its ability to deal with control difficulties such as constrained variables [3], time-delay [4], nonlinearity [5] and non-minimum phase [6]. Theoretical research in the area of predictive control has a great impact on the industrial world and there are many applications of predictive control in industry [7]-[8]. Predictive control is also one of the most effective approaches for control of multi-input multi-output (MIMO) systems [9]. An advantage of model predictive control is that the multivariable systems can be handled in a straightforward manner.

A structure of a predictive controller can be divided into two parts: a predictor and an optimizer. The basic idea of MPC is to use a model of a controlled process to predict *N* future outputs of the process. A trajectory of future manipulated variables is given by solving an optimization problem incorporating a suitable cost function and constraints. Only the first element of the obtained control sequence is applied. The whole procedure is repeated in following sampling period. This principle is known as the receding horizon strategy [10]-[11]. One of the major advantages of predictive control is its ability to do on-line constraints handling in a systematic way. Almost all industrial applications hold constraints of input, output and state space variables.

A significantly important part of the constrained MPC is an optimization task. A frequently used type of optimization in MPC is the quadratic programming [13], where constraints are considered. Previous, current and predicted control variables of MPC are included in a cost function [14]. In case of constrained multivariable predictive control, many constraints are processed in the optimization problem. Therefore, a selection of an appropriate numerical method is a necessary condition for successful achievement of the vector of future increments of the manipulated variables.

The Hildreth method [15] has been widely used for purpose of solving of the quadratic programming problems in MPC. This approach can be categorized as a dual method [15], which manipulates with the Lagrangian multipliers [15]. Reduction of the computational complexity of the optimization methods has been widely researched. The reason is that in certain cases of predictive control of fast dynamics processes an optimization algorithm may not be feasible within the sampling period time.

A general modification of the Hildreth method, proposed by Wang, L., was published in [15] and has been frequently utilized in MPC algorithms [16]. Reduction of a computational complexity is based on testing of the occurance of a multidimensional extreme, which is computed in the current sampling period in MPC under all constraints.

In this paper, a modification of the optimization strategy proposed by Wang, L. [15] was further improved from the algoritmical point of view. As evaluation of all defined constraints is significantly time-demanding in multivariable MPC, the proposed modification of the Wang's approach can be advantageous due to significant reduction of numerical iterative operations required by the optimization algorithm.

[*] Corresponding author: kubalcik@utb.cz

2 Multivariable model predictive control

In the multivariable model predictive control [1]-[2], a system with two inputs and two outputs (TITO) will be further considered. The TITO processes are frequently encountered multivariable processes in practice [9]. A general transfer matrix [11] of a TITO system can be expressed as (1), where $\boldsymbol{U}$ and $\boldsymbol{Y}$ are vectors of the manipulated variables and the controlled variables.

$$\boldsymbol{G}(z)=\begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \tag{1}$$

$$\boldsymbol{Y}(z)=\boldsymbol{G}(z)\boldsymbol{U}(z) \tag{2}$$

$$\boldsymbol{U}(z)=\left[u_1(z),u_2(z)\right]^T,\ \boldsymbol{Y}(z)=\left[y_1(z),y_2(z)\right]^T \tag{3}$$

It may be assumed that the transfer matrix (1) can be transcribed to form (4) of the matrix fraction.

$$\boldsymbol{G}(z)=\boldsymbol{A}^{-1}\left(z^{-1}\right)\boldsymbol{B}\left(z^{-1}\right)=\boldsymbol{B}_1\left(z^{-1}\right)\boldsymbol{A}_1^{-1}\left(z^{-1}\right) \tag{4}$$

The model can be also written in form (5).

$$\boldsymbol{A}\left(z^{-1}\right)\boldsymbol{Y}(z)=\boldsymbol{B}\left(z^{-1}\right)\boldsymbol{U}(z) \tag{5}$$

As an example, a model with polynomials of second degree was chosen in (6)-(7). The model has sixteen parameters. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined as follows:

$$\boldsymbol{A}\left(z^{-1}\right)=\begin{bmatrix} 1+a_1z^{-1}+a_2z^{-2} & a_3z^{-1}+a_4z^{-2} \\ a_5z^{-1}+a_6z^{-2} & 1+a_7z^{-1}+a_8z^{-2} \end{bmatrix} \tag{6}$$

$$\boldsymbol{B}\left(z^{-1}\right)=\begin{bmatrix} b_1z^{-1}+b_2z^{-2} & b_3z^{-1}+b_4z^{-2} \\ b_5z^{-1}+b_6z^{-2} & b_7z^{-1}+b_8z^{-2} \end{bmatrix} \tag{7}$$

A widely used model in model predictive control is the CARIMA (Controller Autoregressive Integrated Moving Average) model which we can obtain by adding a disturbance model (8), where $\boldsymbol{n}$ is a non-measurable random disturbance that is assumed to have zero mean value and constant covariance and (9) in case of TITO system.

$$\boldsymbol{A}\left(z^{-1}\right)\boldsymbol{y}(k)=\boldsymbol{B}\left(z^{-1}\right)\boldsymbol{u}(k)+\boldsymbol{C}\left(z^{-1}\right)\boldsymbol{\Delta}^{-1}\left(z^{-1}\right)\boldsymbol{n}(k) \tag{8}$$

$$\boldsymbol{\Delta}\left(z^{-1}\right)=\begin{bmatrix} 1-z^{-1} & 0 \\ 0 & 1-z^{-1} \end{bmatrix} \tag{9}$$

For purposes of simplification, polynomial matrix $\boldsymbol{C}$ will be further supposed to be equal to the identity matrix [17].

The difference equations (10) of the CARIMA model are used for computation of predictions in predictive control. These equations can be further written into a matrix form (11)-(12).

$$\begin{aligned} y_1(k+1)&=(1-a_1)y_1(k)+(a_1-a_2)y_1(k-1)+a_2y_1(k-2)- \\ &-a_3y_2(k)+(a_3-a_4)y_2(k-1)+a_4y_2(k-2)+ \\ &+b_1\Delta u_1(k)+b_2\Delta u_1(k-1)+b_3\Delta u_2(k)+b_4\Delta u_2(k-1) \end{aligned} \tag{10}$$

$$\begin{aligned} y_2(k+1)&=(1-a_7)y_2(k)+(a_7-a_8)y_2(k-1)+a_8y_2(k-2)- \\ &-a_5y_1(k)+(a_5-a_6)y_1(k-1)+a_6y_1(k-2)+ \\ &+b_5\Delta u_1(k)+b_6\Delta u_1(k-1)+b_7\Delta u_2(k)+b_8\Delta u_2(k-1) \end{aligned}$$

$$\begin{aligned} \boldsymbol{y}(k+1)&=\boldsymbol{A}_1\boldsymbol{y}(k)+\boldsymbol{A}_2\boldsymbol{y}(k-1)+\boldsymbol{A}_3\boldsymbol{y}(k-2)+ \\ &+\boldsymbol{B}_1\Delta\boldsymbol{u}(k)+\boldsymbol{B}_2\Delta\boldsymbol{u}(k-1) \end{aligned} \tag{11}$$

$$\boldsymbol{A}_1=\begin{bmatrix} 1-a_1 & -a_3 \\ -a_5 & 1-a_7 \end{bmatrix},\ \boldsymbol{A}_2=\begin{bmatrix} a_1-a_2 & a_3-a_4 \\ a_5-a_6 & a_7-a_8 \end{bmatrix} \tag{12}$$

$$\boldsymbol{A}_3=\begin{bmatrix} a_2 & a_4 \\ a_6 & a_8 \end{bmatrix},\ \boldsymbol{B}_1=\begin{bmatrix} b_1 & b_3 \\ b_5 & b_7 \end{bmatrix},\ \boldsymbol{B}_2=\begin{bmatrix} b_2 & b_4 \\ b_6 & b_8 \end{bmatrix}$$

It was necessary to directly compute three steps-ahead predictions by establishing of previous predictions to later predictions. The model order defines that computation of one step-ahead prediction is based on the three past values of the system output:

$$\begin{aligned} \hat{\boldsymbol{y}}(k+1)&=\boldsymbol{A}_1\boldsymbol{y}(k)+\boldsymbol{A}_2\boldsymbol{y}(k-1)+\boldsymbol{A}_3\boldsymbol{y}(k-2)+ \\ &+\boldsymbol{B}_1\Delta\boldsymbol{u}(k)+\boldsymbol{B}_2\Delta\boldsymbol{u}(k-1) \end{aligned} \tag{13}$$

$$\begin{aligned} \hat{\boldsymbol{y}}(k+2)&=\boldsymbol{A}_1\boldsymbol{y}(k+1)+\boldsymbol{A}_2\boldsymbol{y}(k)+\boldsymbol{A}_3\boldsymbol{y}(k-1)+ \\ &+\boldsymbol{B}_1\Delta\boldsymbol{u}(k+1)+\boldsymbol{B}_2\Delta\boldsymbol{u}(k) \end{aligned}$$

$$\begin{aligned} \hat{\boldsymbol{y}}(k+3)&=\boldsymbol{A}_1\boldsymbol{y}(k+2)+\boldsymbol{A}_2\boldsymbol{y}(k+1)+\boldsymbol{A}_3\boldsymbol{y}(k)+ \\ &+\boldsymbol{B}_1\Delta\boldsymbol{u}(k+2)+\boldsymbol{B}_2\Delta\boldsymbol{u}(k+1) \end{aligned}$$

Computation of the predictions can be divided into recursion of the free response and recursion of the matrix of dynamics. The free response vector can be expressed as:

$$\boldsymbol{y}_0=\left[\begin{array}{cc} p_{11} & p_{12} \\ p_{21} & q_{22} \\ \hline p_{31} & p_{32} \\ p_{41} & p_{42} \\ \hline p_{51} & p_{52} \\ p_{61} & p_{62} \end{array}\right]\begin{bmatrix} \Delta u_1(k-1) \\ \Delta u_2(k-1) \end{bmatrix}+\left[\begin{array}{cc|cc|cc} q_{11} & q_{12} & q_{13} & q_{14} & q_{15} & q_{16} \\ q_{21} & q_{22} & q_{23} & q_{24} & q_{25} & q_{26} \\ \hline q_{31} & q_{32} & q_{33} & q_{34} & q_{35} & q_{36} \\ q_{41} & q_{42} & q_{43} & q_{44} & q_{45} & q_{46} \\ \hline q_{51} & q_{52} & q_{53} & q_{54} & q_{55} & q_{56} \\ q_{61} & q_{62} & q_{63} & q_{64} & q_{65} & q_{66} \end{array}\right]\left[\begin{array}{c} y_1(k) \\ y_2(k) \\ \hline y_1(k-1) \\ y_2(k-1) \\ \hline y_1(k-2) \\ y_2(k-2) \end{array}\right]=$$

$$=\begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{bmatrix}\Delta\boldsymbol{u}(k-1)+\begin{bmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} & \boldsymbol{Q}_{13} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} & \boldsymbol{Q}_{23} \\ \boldsymbol{Q}_{31} & \boldsymbol{Q}_{32} & \boldsymbol{Q}_{33} \end{bmatrix}\begin{bmatrix} \boldsymbol{y}(k) \\ \boldsymbol{y}(k-1) \\ \boldsymbol{y}(k-2) \end{bmatrix}=\boldsymbol{P}\Delta\boldsymbol{u}(k-1)+\boldsymbol{Q}\begin{bmatrix} \boldsymbol{y}(k) \\ \boldsymbol{y}(k-1) \\ \boldsymbol{y}(k-2) \end{bmatrix} \tag{14}$$

Coefficients of matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ for further predictions are computed recursively. Based on the three previous predictions it is repeatedly computed the next row of the matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ in the following way:

$$\boldsymbol{P}_4=\begin{bmatrix} p_{71} & p_{72} \\ p_{81} & p_{82} \end{bmatrix}=\boldsymbol{A}_1\boldsymbol{P}_{31}+\boldsymbol{A}_2\boldsymbol{P}_{21}+\boldsymbol{A}_3\boldsymbol{P}_{11} \tag{15}$$

$$\boldsymbol{Q}_{41} = \begin{bmatrix} q_{71} & q_{72} \\ q_{81} & q_{82} \end{bmatrix} = \boldsymbol{A}_1\boldsymbol{Q}_{31} + \boldsymbol{A}_2\boldsymbol{Q}_{21} + \boldsymbol{A}_3\boldsymbol{Q}_{11} \tag{16}$$

$$\boldsymbol{Q}_{42} = \begin{bmatrix} q_{73} & q_{74} \\ q_{83} & q_{84} \end{bmatrix} = \boldsymbol{A}_1\boldsymbol{Q}_{32} + \boldsymbol{A}_2\boldsymbol{Q}_{22} + \boldsymbol{A}_3\boldsymbol{Q}_{12} \tag{17}$$

$$\boldsymbol{Q}_{43} = \begin{bmatrix} q_{75} & q_{76} \\ q_{85} & q_{86} \end{bmatrix} = \boldsymbol{A}_1\boldsymbol{Q}_{33} + \boldsymbol{A}_2\boldsymbol{Q}_{23} + \boldsymbol{A}_3\boldsymbol{Q}_{13} \tag{18}$$

Recursion (19)-(20) of the matrix $\boldsymbol{G}$ is similar. The next element of the first column is repeatedly computed and the remaining columns are shifted. This procedure is performed repeatedly until the prediction horizon is achieved. If the control horizon is lower than the prediction horizon a number of columns in the matrix is reduced. Predictions can be written in a compact matrix form (21).

$$\boldsymbol{G}\Delta\boldsymbol{u} = \left[\begin{array}{cc|cc} g(1,1) & g(1,2) & 0 & 0 \\ g(2,1) & g(2,2) & 0 & 0 \\ \hline g(3,1) & g(3,2) & g(1,1) & g(1,2) \\ g(4,1) & g(4,2) & g(2,1) & g(2,2) \\ \hline g(5,1) & g(5,2) & g(3,1) & g(3,2) \\ g(6,1) & g(6,2) & g(4,1) & g(4,2) \end{array}\right] \left[\begin{array}{c} \Delta u_1(k) \\ \Delta u_2(k) \\ \hline \Delta u_1(k+1) \\ \Delta u_2(k+1) \end{array}\right] = \tag{19}$$

$$= \begin{bmatrix} G(1,1) & 0 \\ G(2,1) & G(1,1) \\ G(3,1) & G(2,1) \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}(k) \\ \Delta\boldsymbol{u}(k+1) \end{bmatrix}$$

$$\boldsymbol{G}_{41} = \begin{bmatrix} g_{71} & g_{72} \\ g_{81} & g_{82} \end{bmatrix} = \boldsymbol{A}_1\boldsymbol{G}_{31} + \boldsymbol{A}_2\boldsymbol{G}_{21} + \boldsymbol{A}_3\boldsymbol{G}_{11} \tag{20}$$

$$\begin{aligned} &\hat{\boldsymbol{y}}(k+j) = \boldsymbol{G}\varDelta\boldsymbol{u}(k+j-1) + \boldsymbol{P}\varDelta\boldsymbol{u}(k-1) + \boldsymbol{Q}\boldsymbol{y}(k-j+1) \\ &j \le N \end{aligned} \tag{21}$$

2.1. Implementation of MPC

In the framework of the optimization subsystem of MPC, the computation of a control law of MPC is particularly based on minimization of quadratic criterion (22). This specific form of the optimization problem is then related to quadratic optimization [13]-[14].

$$J(k) = \sum_{j=1}^{N} \boldsymbol{e}(k+j)^2 + \lambda \sum_{j=1}^{N_u} \Delta\boldsymbol{u}(k+j)^2 \tag{22}$$

where $\boldsymbol{e}(k+j)$ is a vector of predicted control errors, $\Delta\boldsymbol{u}(k+j)$ is a vector of future increments of the manipulated variable, N is a length of the prediction horizon, N_u is a length of the control horizon and λ is a weighting factor of control increments. A predictor in a vector form is given by:

$$\hat{\boldsymbol{y}} = \boldsymbol{G}\Delta\boldsymbol{u} + \boldsymbol{y}_0 \tag{23}$$

where $\hat{\boldsymbol{y}}$ is a vector of system predictions along the horizon of the length N, $\Delta\boldsymbol{u}$ is a vector of control increments, $\boldsymbol{y}_0$ is the free response vector. $\boldsymbol{G}$ is a matrix of the dynamics given by equation (24).

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_0 & 0 & \cdots & \cdots & 0 \\ \boldsymbol{G}_1 & \boldsymbol{G}_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \boldsymbol{G}_0 & 0 \\ \boldsymbol{G}_{N-1} & \cdots & \cdots & \cdots & \boldsymbol{G}_0 \end{bmatrix} \tag{24}$$

where sub-matrices $\boldsymbol{G}_i$ have dimension 2x2 and contain values of the step sequence.

The criterion (22) of the optimization problem can be written in a general vector form (25).

$$J = (\hat{\boldsymbol{y}} - \boldsymbol{w})^T (\hat{\boldsymbol{y}} - \boldsymbol{w}) + \lambda\Delta\boldsymbol{u}^T\Delta\boldsymbol{u} \tag{25}$$

where $\boldsymbol{w}$ is a vector of the reference trajectory. The criterion can be modified using the expression (25) to (26).

$$J = 2\boldsymbol{g}^T\Delta\boldsymbol{u} + \Delta\boldsymbol{u}^T\boldsymbol{H}\Delta\boldsymbol{u} \tag{26}$$

where the gradient $\boldsymbol{g}$ and the Hess matrix $\boldsymbol{H}$ are defined by following expressions:

$$\boldsymbol{g}^T = \boldsymbol{G}^T(\boldsymbol{y}_0 - \boldsymbol{w}) \tag{27}$$

$$\boldsymbol{H} = \boldsymbol{G}^T\boldsymbol{G} \tag{28}$$

In context of the quadratic programming optimization with constraints, general formulation of predictive control is as follows

$$\min_{\Delta\boldsymbol{u}} 2\boldsymbol{g}^T\Delta\boldsymbol{u} + \Delta\boldsymbol{u}^T\boldsymbol{H}\Delta\boldsymbol{u} \tag{29}$$

with respect to matrix inequality in a compact form:

$$\left.\begin{array}{c} \boldsymbol{M}\varDelta\boldsymbol{u} \le \boldsymbol{\gamma} \\ \boldsymbol{M} \in \mathcal{R}^{m,n}, \boldsymbol{\gamma} \in \mathcal{R}^{m,1} \\ \boldsymbol{M} = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{m1} & M_{m2} & \cdots & M_{mn} \end{bmatrix} \\ \boldsymbol{\gamma} = [\gamma_1 \quad \gamma_2 \quad \cdots \quad \gamma_m]^T \end{array}\right\} \tag{30}$$

2.2 Optimization algorithm widely used in MPC

In practical applications of MPC, a modification proposed by Wang, L. has been widely implemented [15]. This advantageous approach represents a new insight to the optimization strategy used in MPC. MPC is characteristic by a frequent occurance of a situation, when the quadratic programming problem can be completely substituted by a simple multi-dimensional extrem problem.

The main idea of the modification is based on a pre-computed vector of future increments of the manipulated variables in form of a multi-dimensional extrem (31). If the inequality (30) is fulfilled then the whole problem of quadratic programming is eliminated and the solution has form (31).

$$\boldsymbol{\Delta u} = -\boldsymbol{H}^{-1}\boldsymbol{b} \tag{31}$$

If the multi-dimensional extreme is achieved, then the computational complexity significantly decreases. Otherwise the quadratic programming problem has to be solved using the Hildreth method, which results in equation (32).

$$\boldsymbol{\Delta u} = \boldsymbol{-H^{-1}}(\boldsymbol{M^{\mathrm{T}}d} + \boldsymbol{b}^{T}) \tag{32}$$

3 Proposal of optimization strategy

For purposes of further decreasing of computational complexity of the optimization algorithm, the approach described in the previous section was further improved. The approach presented by Wang. L [15] spends a large amount of the computational time by evaluation of all conditions in (33).

$$\left.\begin{array}{r} \forall i, i \in \langle 1; m\rangle : \boldsymbol{m\Delta} u \le \gamma_i \,; \\ \boldsymbol{m} \in \mathcal{R}^{1,n};\ \boldsymbol{\Delta u} = -\boldsymbol{H}^{-1}\boldsymbol{b}; \\ \boldsymbol{m} = \boldsymbol{m}(i) = \begin{bmatrix} M_{i1} & M_{i2} & \cdots & M_{in}\end{bmatrix} \end{array}\right\} \tag{33}$$

The condition (33) can be effectively modified while maintaining the original advantages of the modification presented by Wang. L. A new form of the conditions is defined by (34). The testing of the conditions is progresivelly divided into partial operations.

$$\left.\begin{array}{c} \left\{\begin{array}{c} \exists\delta, 1 \le \delta < m : \widehat{\boldsymbol{M}}\boldsymbol{\Delta\hat{u}} > \widehat{\boldsymbol{\gamma}} \Rightarrow \boldsymbol{\Delta u} = \operatorname{argmin}(J; \boldsymbol{M\Delta} u \le \boldsymbol{\gamma}) \\ \exists\delta, \delta = m : \widehat{\boldsymbol{M}}\boldsymbol{\Delta\hat{u}} \le \widehat{\boldsymbol{\gamma}} \Rightarrow \boldsymbol{\Delta u} = -\boldsymbol{H}^{-1}\boldsymbol{b} \\ \exists\delta, \delta = m : \widehat{\boldsymbol{M}}\boldsymbol{\Delta\hat{u}} > \widehat{\boldsymbol{\gamma}} \Rightarrow \boldsymbol{\Delta u} = \operatorname{argmin}(J; \boldsymbol{M\Delta} u \le \boldsymbol{\gamma}) \end{array}\right. \\ \widehat{\boldsymbol{M}} \in \mathcal{R}^{\delta,n}, \widehat{\boldsymbol{\gamma}} \in \mathcal{R}^{\delta,1};\ \boldsymbol{\Delta\hat{u}} = -\boldsymbol{H}^{-1}\boldsymbol{b}; \\ \widehat{\boldsymbol{M}} = \widehat{\boldsymbol{M}}(\delta) = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{\delta 1} & M_{\delta 2} & \cdots & M_{\delta n} \end{bmatrix} \\ \widehat{\boldsymbol{\gamma}} = \widehat{\boldsymbol{\gamma}}(\delta) = \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_\delta \end{bmatrix}^T \end{array}\right\} \tag{33}$$

In case of the first failure the testing is terminated and the rest of the conditions is not evaluated. The quadratic programming problem then must be solved. This enables saving of the computational time. This is significant particularly in case of multivariable MPC where the number of operations is large. If all the conditions are fulfilled and the testing is completed, then the multidimensional extreme problem without constraints is solved.

4 Simulation results

The modification published by Wang, L. was compared with the method proposed in this paper by simulation of constrained multivariable predictive control in MATLAB. The comparison was based on a measurement of floating point operations [18] of a whole MPC algorithm which was applied for simulation control of a simulation controlled system defined by (35)-(36). A setting of further parameters of control is defined by (37).

$$\boldsymbol{A}(z^{-1}) = \begin{bmatrix} 1\text{-}1.3264z^{-1} + 0.3271z^{-2} & 0.024z^{-1} - 0.0029z^{-2} \\ \text{-}0.0711z^{-1} + 0.0759z^{-2} & 1\text{-}1.0911z^{-1} + 0.134z^{-2} \end{bmatrix} \tag{35}$$

$$\boldsymbol{B}(z^{-1}) = \begin{bmatrix} 0.2983z^{-1}\text{-}0.097z^{-2} & 0.093z^{-1} + 0.0682z^{-2} \\ 0.1755z^{-1} + 0.0688z^{-2} & 0.1779z^{-1} + 0.1065z^{-2} \end{bmatrix} \tag{36}$$

$$u_{\min} = -1.7,\ u_{\max} = 1.75,\ \Delta u_{\max} = 0.07 \tag{37}$$

Constraints of the manipulated variables and increments of the manipulated variables were considered which is obvious from definition (37). Setting of constraints has an appropriate form, as can be seen in (38). Where ***I*** is an identity matrix [17] and ***E*** is a unit matrix [17].

$$\boldsymbol{M} = \left[\begin{array}{cccc} -\boldsymbol{E}^{2,2} & 0 & \cdots & 0 \\ -\boldsymbol{E}^{2,2} & -\boldsymbol{E}^{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\boldsymbol{E}^{2,2} & -\boldsymbol{E}^{2,2} & \cdots & -\boldsymbol{E}^{2,2} \\ \hline \boldsymbol{E}^{2,2} & 0 & \cdots & 0 \\ \boldsymbol{E}^{2,2} & \boldsymbol{E}^{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{E}^{2,2} & \boldsymbol{E}^{2,2} & \cdots & \boldsymbol{E}^{2,2} \\ \hline \boldsymbol{I}^{2,2} & 0 & \cdots & 0 \\ 0 & \boldsymbol{I}^{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{I}^{2,2} \end{array}\right], \boldsymbol{\gamma} = \left[\begin{array}{c} \boldsymbol{u}(k-1) + \begin{bmatrix} u_{\min} \\ u_{\min} \end{bmatrix} \\ \vdots \\ \boldsymbol{u}(k-1) + \begin{bmatrix} u_{\min} \\ u_{\min} \end{bmatrix} \\ \hline -\boldsymbol{u}(k-1) + \begin{bmatrix} u_{\max} \\ u_{\max} \end{bmatrix} \\ \vdots \\ -\boldsymbol{u}(k-1) + \begin{bmatrix} u_{\min} \\ u_{\min} \end{bmatrix} \\ \hline \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{array}\right] \tag{38}$$

As can be seen in Table 1, a significantly lower number of floating point operations [18] was achieved when using the proposed approach. The complexity variable μ equal to a maximum horizon N was incrementaly increasing.

Table 1. Decreasing of number of operations using proposed approach.

	MPC without Proposal	**Proposed Approach**
μ	F [flops]	F^{*}[flops]
5	46115895	46101100
10	356752161	356697015
15	1311763707	1311643086
20	3952627076	3952398870
25	8412199693	8411825659
30	15642495861	15641915446
35	25789861477	25789050231

An order of the computational complexity can be expressed by using a function O=$O(\mu)$ [18]. F^{*} is the number of flops and O^{*} expresses the order of the complexity function for the proposed approach. F and O

are the number of flops and the expression of complexity function for the method proposed by Wang, L. In equations (39) and (40), the results were obtained using a non-linear regression [19].

$$\left.\begin{aligned} F &= 746126\mu^3 - 5\times10^6\,\mu^2 - 2\times10^7\,\mu + 2\times10^8 \\ O &= O(\mu) = 746126\mu^3 \end{aligned}\right\} \quad (39)$$

$$\left.\begin{aligned} F^* &= 746122\mu^3 - 5\times10^6\,\mu^2 - 2\times10^7\,\mu + 2\times10^8 \\ O^* &= O^*(\mu) = 746122\mu^3 \end{aligned}\right\} \quad (40)$$

Simulation results can be seen in Fig.1 and Fig.2.

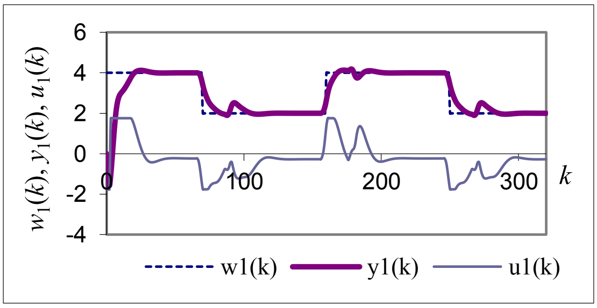

Fig. 1. Simulation of MPC - 1st variables of control.

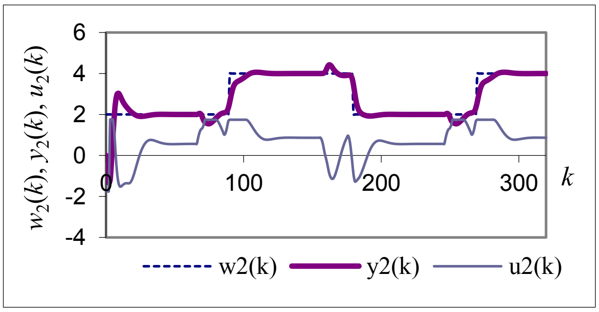

Fig. 2. Simulation of MPC - 2nd variables of control.

5 Conclusions

The widely used modification of the optimization strategy in MPC, published by Wang, L., was further modified and improved in this paper. Advantages of the proposed modification were demonstrated and proved by simulations in MATLAB. The proposed approach was successfully implemented for the multivariable constrained MPC. Multivariability and considered constraints in MPC significantly increase a computational complexity of the optimization. Therefore, the proposed approach can be advantageous for multivariable MPC with constraints.

References

1. J.P. Corriou, *Process control: theory and applications* (Springer, 2004)
2. W.H. Kwon, *Receding horizon control: model predictive control for state models* (Springer, 2005)
3. E.F. Camacho, C. Bordons, *Model predictive control* (Springer, 2007)
4. M. Kubalcik, V. Bobal, MATEC Web Conf. **125,** 02023, (2017)
5. J. Novák, P. Chalupa, Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. Springer (2013)
6. T. Barot, M. Kubalcik, 15th International Carpathian Control Conference (ICCC) (2014)
7. D. Ingole, J. Holaza, B. Takacs, M. Kvasnica, 20th International Conference on Process Control (PC). IEEE (2015)
8. D. Honc, F. Dusek, Control and Monitoring Algorithms in Process Automation Applications. Shaker Verlag (2012)
9. P. Navratil, L. Pekar, International Journal of Circuits, Systems and Signal Processing, **7**(6) (2013)
10. S. Huang, *Applied predictive control* (Springer, 2002)
11. J.B. Rawlings, D.Q. Mayne, *Model Predictive Control Theory and Design*, (Nob Hill Pub, 2009)
12. J.A. Rossiter, *Model Based Predictive Control: a Practical Approach* (CRC Press, 2003)
13. G.M. Lee, N.N. Tam, N.D. Yen, *Quadratic Programming and Affine Variational Inequalities: A Qualitative Study* (Springer, 2005)
14. Z. Dostal, *Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities* (Springer, 2009)
15. L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB* (Springer, 2009)
16. T. Barot, M. Kubalcik, Advances in Intelligent Systems and Computing, **348**, Springer (2015)
17. A.M. Robert, *Linear Algebra: examples and applications* (Hackensack, 2005)
18. R. Hunger, Technical Report [online]. Technische Universität München, Associate Institute for Signal Processing (2007) Available from: https://mediatum.ub.tum.de/doc/ 625604/625604.pdf
19. I. Krivy, J. Tvrdik, R. Krpec,. Computational Statistics and Data Analysis, **33** (12), 277 (2000)