Modifications of Optimization Algorithms Applied in Multivariable Predictive Control

MAREK KUBALCIK, VLADIMIR BOBAL
Department of Process Control
Faculty of Applied Informatics
Tomas Bata University in Zlin
Nad Stranemi 4511, 76005, Zlin
CZECH REPUBLIC
kubalcik@fai.utb.cz, bobal@fai.utb.cz
http://fai.utb.cz/en

TOMAS BAROT
Department of Mathematics with Didactics
Faculty of Education
University of Ostrava
Frani Sramka 3, 709 00, Ostrava
CZECH REPUBLIC
Tomas.Barot@osu.cz
http://pdf.osu.eu

Abstract: - Non-linear optimization, particularly quadratic programming (QP), is a mathematical method which is widely applicable in model predictive control (MPC). It is significantly important if constraints of variables are considered in MPC and the optimization task is then computationally demanding. 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. Besides the generally used numerical Hildreth's method of QP, a possible suitable modification is based on precomputing operations proposed by Wang, L. This general optimization strategy is further modified. Two modifications, which could be applied separately each, were interconnected in this paper. The first modification was published previously; however, its application can be more efficient in connection with the second proposed approach, which modifies precomputing operations. Decreasing of the computational complexity of the optimization by using of the proposal is discussed and analyzed by measurements of floating point operations and control quality criterions using hypotheses tests – paired T-test and Wilcoxon test.

Key-Words: - Model Predictive Control; Multivariable Control; Optimization; Quadratic Programming; Hildreth's Method; Constraints.

1 Introduction

Model predictive control (MPC) [1]-[2] has been widely applied in controlling of industrial processes with respect 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 has a great impact on the industrial world in this area of predictive control. There are many applications of predictive control in industry [7]-[8]. Research of the predictive control has been significantly related to industrial practice. Predictive control is also one of the most effective approaches for control of multivariable systems (MIMO) [9]. An advantage of model predictive control is that the multivariable systems can be handled in a straightforward manner.

A predictive controller can be divided into two subsystems, a predictor and an optimizer, which are cooperating on a receding horizon strategy [10]. In the MPC, the basic idea is to use a model of a controlled process to predict future outputs of the process [11] and a trajectory of future manipulated variables is given by solving an optimization problem incorporating a suitable cost function with constraints [12]. Only the first element of the obtained control sequence is applied in a framework of the receding horizon strategy. One of the advantages of the predictive control is its ability to do on-line constraints handling in a systematic way, which frequently appears in industrial applications.

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's 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]. Its modifications applied in MPC have not been widely described in literature.

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

In this paper, an observed modification [18] of the optimization strategy following Wang, L. [15] is further improved by interconnecting together with an efficient approach to Hildreth's method [17], which is based on from algorithmic 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.

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). This model proved to be effective for control of several TITO laboratory processes [7]-[8], where controllers based on a model with polynomials of the first degree failed. 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 [19].

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:

$$\begin{aligned} \hat{\boldsymbol{y}}(k+1) &= \boldsymbol{A}_1\boldsymbol{y}(k) + \boldsymbol{A}_2\boldsymbol{y}(k-1) + \boldsymbol{A}_3\boldsymbol{y}(k-2) + \\ &+\boldsymbol{B}_1\Delta\boldsymbol{u}(k) + \boldsymbol{B}_2\Delta\boldsymbol{u}(k-1) \\ \hat{\boldsymbol{y}}(k+2) &= \boldsymbol{A}_1\boldsymbol{y}(k+1) + \boldsymbol{A}_2\boldsymbol{y}(k) + \boldsymbol{A}_3\boldsymbol{y}(k-1) + \\ &+\boldsymbol{B}_1\Delta\boldsymbol{u}(k+1) + \boldsymbol{B}_2\Delta\boldsymbol{u}(k) \\ \hat{\boldsymbol{y}}(k+3) &= \boldsymbol{A}_1\boldsymbol{y}(k+2) + \boldsymbol{A}_2\boldsymbol{y}(k+1) + \boldsymbol{A}_3\boldsymbol{y}(k) + \\ &+\boldsymbol{B}_1\Delta\boldsymbol{u}(k+2) + \boldsymbol{B}_2\Delta\boldsymbol{u}(k+1) \end{aligned} \tag{13}$$

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:

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

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

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).

$$\boldsymbol{G}\Delta\boldsymbol{u} = \begin{bmatrix} g(1,1) & g(1,2) & 0 & 0 \\ g(2,1) & g(2,2) & 0 & 0 \\ g(3,1) & g(3,2) & g(1,1) & g(1,2) \\ g(4,1) & g(4,2) & g(2,1) & g(2,2) \\ g(5,1) & g(5,2) & g(3,1) & g(3,2) \\ g(6,1) & g(6,2) & g(4,1) & g(4,2) \end{bmatrix} \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k+1) \end{bmatrix} = \begin{bmatrix} G(1,1) & 0 \\ G(2,1) & G(1,1) \\ G(3,1) & G(2,1) \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}(k) \\ \Delta\boldsymbol{u}(k+1) \end{bmatrix} \tag{19}$$

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

$$\hat{\boldsymbol{y}}(k+j) = \boldsymbol{G}\Delta\boldsymbol{u}(k+j-1) + \boldsymbol{P}\Delta\boldsymbol{u}(k-1) + \boldsymbol{Q}\boldsymbol{y}(k-j+1) \quad j \le N \tag{21}$$

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:

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

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 occurrence of a situation, when the quadratic programming problem can be completely substituted by a simple multi-dimensional extreme 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 extreme (31). If the inequality (30) is fulfilled, then the whole problem of quadratic programming is eliminated and the solution has form (31).

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

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

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

3 Proposal of Interconnection of Convenient Optimization Strategies

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).

This approach was published in [18]. It was focused on improving of precomputing operations based on constraints in QP.

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

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 progressively divided into partial operations.

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

In case of the first failure the testing is terminated and the rest of the conditions is not evaluated. The multi-dimensional extreme problem is then 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 quadratic programming problem must be solved.

The second approach, focused on improving the Hildreth's method, was published in [17]. This method was primarily based on improving the numerical algorithm of the Hildreth's method. The main principles are bound with including of a new exit condition of the iterative algorithm. This modified exit condition did not significantly affect quality of control, as can be seen in simulation results in [17]. Instead the comparison of equality of last computed results in main numerical cycle, further condition of fulfilling constraints (30) is being tested [17, p. 78].

In this paper, modification [17] is denoted as the first modification. As the second modification, the methodological extension [18] is entitled.

4 Simulation Results

MPC with both modifications was compared with the MPC without modifications by simulation of constrained multivariable predictive control in MATLAB. The comparison was based on a measurement of floating point operations [20] 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).

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

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

$$u_{\min}=-1.7,\ u_{\max}=1.75,\ \Delta u_{\max}=0.07 \quad (37)$$

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 $\boldsymbol{I}$ is an identity matrix [19] and $\boldsymbol{E}$ is a unit matrix [19].

$$\boldsymbol{M}=\begin{bmatrix}-\boldsymbol{E}^{2,2} & 0 & \cdots & 0\\ -\boldsymbol{E}^{2,2} & -\boldsymbol{E}^{2,2} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ -\boldsymbol{E}^{2,2} & -\boldsymbol{E}^{2,2} & \cdots & -\boldsymbol{E}^{2,2}\\ \hline \boldsymbol{E}^{2,2} & 0 & \cdots & 0\\ \boldsymbol{E}^{2,2} & \boldsymbol{E}^{2,2} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{E}^{2,2} & \boldsymbol{E}^{2,2} & \cdots & \boldsymbol{E}^{2,2}\\ \hline \boldsymbol{I}^{2,2} & 0 & \cdots & 0\\ 0 & \boldsymbol{I}^{2,2} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{I}^{2,2}\end{bmatrix}, \boldsymbol{\gamma}=\begin{bmatrix}\boldsymbol{u}(k-1)+\begin{bmatrix}u_{\min}\\ u_{\min}\end{bmatrix}\\ \vdots\\ \boldsymbol{u}(k-1)+\begin{bmatrix}u_{\min}\\ u_{\min}\end{bmatrix}\\ \hline -\boldsymbol{u}(k-1)+\begin{bmatrix}u_{\max}\\ u_{\max}\end{bmatrix}\\ \vdots\\ -\boldsymbol{u}(k-1)+\begin{bmatrix}u_{\min}\\ u_{\min}\end{bmatrix}\\ \hline \Delta u_{\max}\\ \vdots\\ \Delta u_{\max}\end{bmatrix} \quad (38)$$

Floating point operations of particular cases of MPC were computed using rules in [20]. The complexity variable μ, which is equal to a maximum horizon N, was incrementally increasing. As can be seen in Table 1, a significantly lower number of floating point operations F^* was achieved when using MPC with both modifications. In F, a maximum number of operations was achieved for case of non-modified MPC. Application of only the first modification of MPC was active in small horizons μ=10 and μ=15 with a number of operations F_1. In other cases with horizons greater than μ=15, the second modification got better results, as can be seen in F_2. Interconnection of both modifications caused decreasing of floating point operations.

Simulation results of MPC without modifications can be seen in Fig. 1-2 and with proposed modifications in Fig. 3-4.

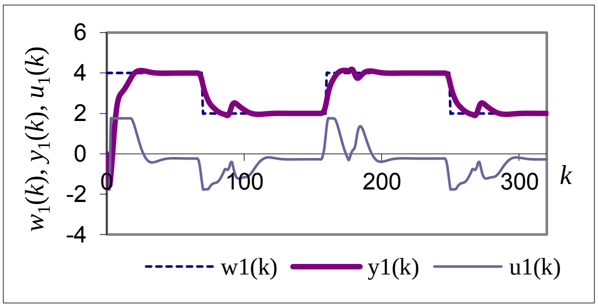

Fig. 1 Simulation of MPC - 1st Variables of Control without Modifications

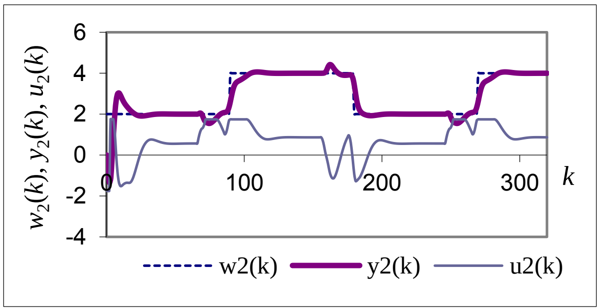

Fig. 2 Simulation of MPC - 2nd Variables of Control without Modifications

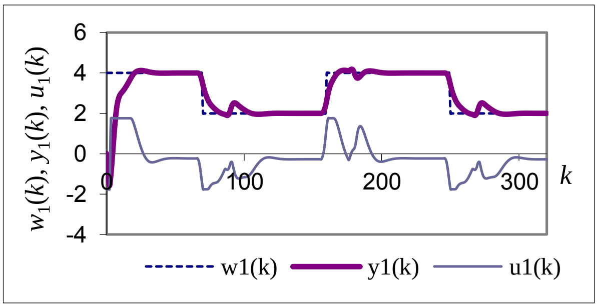

Fig. 3 Simulation of MPC - 1st Variables of Control with Both Proposed Modifications

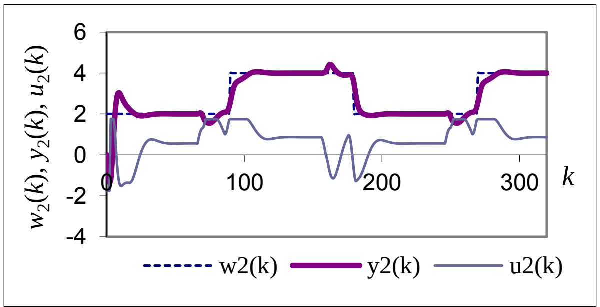

Fig. 4 Simulation of MPC - 2nd Variables of Control with Both Proposed Modifications

Table 1: Analysis of Number of Floating Point Operations using Proposed Modifications in MPC

	MPC without Modification	MPC with 1st Modification	MPC with 2nd Modification	MPC with Both Modifications
μ	F	F_1	F_2	F^*
10	20488351	19881556	20487679	19880947
15	70640181	40643698	70638817	40642365
20	169256574	169256574	169254155	169254155
25	332731867	332731867	332728093	332728093
30	577460460	577460460	577455031	577455031
35	919836753	919836753	919829369	919829369
40	1376255146	1376255146	1376245507	1376245507
45	1963110039	1963110039	1963110039	1963097845

An order of the computational complexity can be expressed by using a function $O=O(\mu)$ [20]. F^* is the number of flops and O^* expresses the order of the complexity function for the proposed approach with both modifications. O is a complexity function for the case of MPC without modifications. In equations (39) and (40), the results are obtained using a non-linear regression [21].

$$\left.\begin{aligned} F &= 21859\mu^3 - 14376\,\mu^2 + 6666.1\,\mu \\ O &= O(\mu) = 21859\mu^3 \end{aligned}\right\} \tag{39}$$

$$\left.\begin{aligned} F^* &= 20267\,\mu^3 + 103648\mu^2 - 2\times10^6\,\mu \\ O^* &= O^*(\mu) = 20267\mu^3 \end{aligned}\right\} \tag{40}$$

Results of analyses of floating point operations for MPC without modifications F and for MPC with both modifications $F*$ can be seen in Fig.5 and Fig.6.

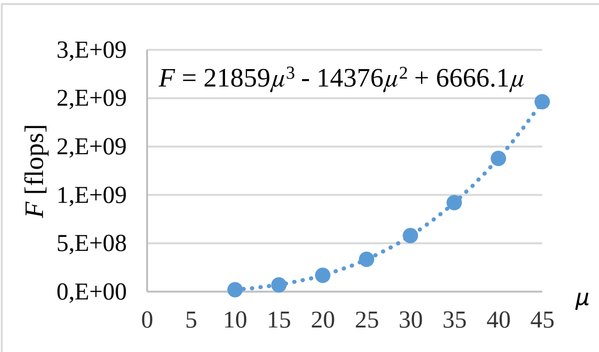

Fig. 5 Analysis of Floating Point Operations for MPC without Modifications

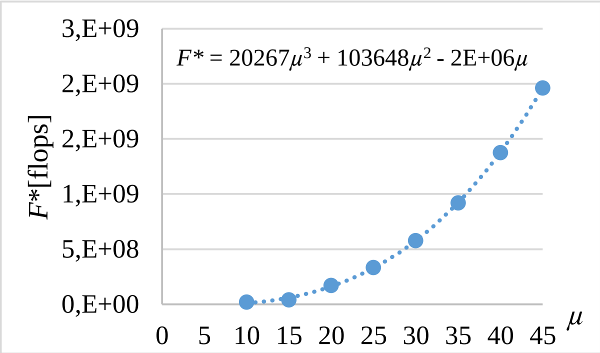

Fig. 6 Analysis of Floating Point Operations for MPC with Both Modifications

Further, quality of control was analyzed using criterions (41) and (42) generally used in research focused on control engeneering e.g. [4], [17].

$$J_1 = \sum_k [\Delta u_1(k)]^2 + \sum_k [\Delta u_2(k)]^2 \qquad (41)$$

$$J_2 = \sum_k [w_1(k) - y_1(k)]^2 + \sum_k [w_2(k) - y_2(k)]^2 \qquad (42)$$

These criterions of quality (41)-(42) were measured in whole MPC control in case with both proposed modification and without modifications depending on horizons defined by μ, as can be seen in Table 2 and Table3. Where both criterions J_1 and J_2 are depended on a number of horizons μ.

Table 2: Analysis of Control Quality using Criterion J_1=$J_1(\mu)$

	MPC without Modification	MPC with 1st Modification	MPC with 2nd Modification	MPC with Both Modifications
μ	J_1	J_1	J_1	J_1
10	58,3219	58,3219	58,3219	58,3219
15	56,0261	56,0260	56,0261	56,0260
20	55,0907	55,0907	55,0907	55,0907
25	55,6287	55,6286	55,6287	55,6286
30	58,1919	58,1917	58,1919	58,1917
35	58,2704	58,2704	58,2704	58,2704
40	58,3068	58,3068	58,3068	58,3068
45	58,4951	58,4951	58,4951	58,4951

Table 3: Analysis of Control Quality using Criterion J_2=$J_2(\mu)$

	MPC without Modification	MPC with 1st Modification	MPC with 2nd Modification	MPC with Both Modifications
μ	J_2	J_2	J_2	J_2
10	195,4043	195,4043	195,4043	195,4043
15	196,0840	196,0841	196,0840	196,0841
20	197,6620	197,6621	197,6620	197,6621
25	196,7332	196,7333	196,7332	196,7333
30	193,0959	193,0962	193,0959	193,0962
35	192,9479	192,9479	192,9479	192,9479
40	192,7495	192,7495	192,7495	192,7495
45	192,1464	192,1464	192,1464	192,1464

According to procedures for testing hypotheses, described e.g. in [22]-[23], the measured criterions in Table 2 and Table 3 were further tested.

At first, testing the normality of data [22] was provided using a generally applied method Shapiro-Wilk described in [24]. Table 4 contains results of testing normality of data. Testing zero-hypotheses, which express fulfilling of the normality, were confirmed on the significance level α in software PAST [25].

Table 4: Testing Normality of Data of Criterion J_1=$J_1(\mu)$ using Shapiro-Wilk Test

		MPC without Modif.	MPC with 1st Modif.	MPC with 2nd Modif.	MPC with Both Modif.
		p-value	*p*-value	*p*-value	*p*-value
		0,009781	0,009785	0,009781	0,009785
Result of Testing Zero-Hypothesis (Fail to Reject / Reject)	α=0,05	Fail to Reject	Fail to Reject	Fail to Reject	Fail to Reject
	α=0,01	Fail to Reject	Fail to Reject	Fail to Reject	Fail to Reject
	α=0,001	Reject	Reject	Reject	Reject

Table 5: Testing Normality of Data of Criterion J_2=$J_2(\mu)$ using Shapiro-Wilk Test

		MPC without Modif.	MPC with 1st Modif.	MPC with 2nd Modif.	MPC with Both Modif.
		p-value	*p*-value	*p*-value	*p*-value
		0,245	0,2451	0,245	0,2451
Result of Testing Zero-Hypothesis (Fail to Reject / Reject)	α=0,05	Fail to Reject	Fail to Reject	Fail to Reject	Fail to Reject
	α=0,01	Fail to Reject	Fail to Reject	Fail to Reject	Fail to Reject
	α=0,001	Fail to Reject	Fail to Reject	Fail to Reject	Fail to Reject

Existence of statically significant differences between J_1 and J_2 with and without modification was tested by paired T-test or Wilcoxon test. The paired T-test can be applied for data comparison, which indicate normality behaviour. The fulfilling of normality is obvious from Table 4 or Table 5. In other cases, the Wilcoxon test has to be used for non parametrical data.

Depending on testing normality of all data in Table 4 and Table 5, zero hypothesis about non-existence of statistically significant differences were performed, as can be seen in Table 6 and Table 7. Hypotheses, which were failed to reject, express situation with similarities of both data sets (before and after modifications in MPC).

In results of testing hypotheses, similarities in criterions J_1 and J_2 were found. Therefore modifications of MPC, proposed in this paper, have not statistically significant influence on the control quality. These conclusions can be verified using Table 2 and Table 3, where differences on the 4th decimal places can be seen.

Table 5: Testing Hypotheses on Non-Existence of Differences between Data of Criterion $J_1=J_1(\mu)$ before and after Applied Modification in MPC

Result of Testing Zero-Hypothesis (Fail to Reject / Reject)	MPC without Modif.	MPC with Both Modif.	Applied Test
	p-value		
α=0,05	*p*= 0,1036: Fail to Reject		Paired T-test
α=0,01	*p*= 0,1036: Fail to Reject		Paired T-test
α=0,001	*p*= 0,10881: Fail to Reject		Wilcoxon

Table 6: Testing Hypotheses on Non-Existence of Differences between Data of Criterion $J_2=J_2(\mu)$ before and after Applied Modification in MPC

Result of Testing Zero-Hypothesis (Fail to Reject / Reject)	MPC without Modif.	MPC with Both Modif.	Applied Test
	p-value		
α=0,05	*p*= 0,0796: Fail to Reject		Paired T-test
α=0,01	*p*= 0,0796: Fail to Reject		Paired T-test
α=0,001	*p*= 0,0796: Fail to Reject		Paired T-test

7 Conclusion

Interconnection of modifications of optimization numerical methods was applied in constrained multivariable MPC in this paper. Advantages of the proposed approach were demonstrated and proved by simulations in MATLAB. 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. Analysis of saving of floating point operations and influence of the proposed approach on quality of control was performed. By analysis of the simulation results it was proved that application of the proposed modification significantly saves floating point operations and concurrently does not affect quality of control. It was then proved that the proposed method can be successfully applied.

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, Predictive control of multivariable time-delay systems, *MATEC Web Conf.*, Vol.125, 2017.

[5] J. Novák, P. Chalupa, Nonlinear State Estimation and Predictive Control of pH Neutralization Process, *Nostradamus 2013*: *Prediction, Modeling and Analysis of Complex Systems*, 2013.

[6] T. Barot, M. Kubalcik, Predictive Control of Non-Minimum Phase Systems, *15th International Carpathian Control Conference (ICCC)*, 2014.

[7] D. Ingole, J. Holaza, B. Takacs, M. Kvasnica, FPGA-Based Explicit Model Predictive Control for Closed Loop Control of Intravenous Anesthesia, *20th International Conference on Process Control (PC)*, 2015.

[8] D. Honc, F. Dusek, State-Space Constrained Model Predictive Control, *27th European Conference on Modelling and Simulation*, 2013. Control of Intravenous Anesthesia, *20th International Conference on Process Control (PC)*, 2015.

[9] P. Navratil, L. Pekar, Possible approach to control of multi-variable control loop by using tools for determining optimal control pairs,

International Journal of Circuits, Systems and Signal Processing, Vol.7, No.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*, Spinger, 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, Predictive control of systems with fast dynamics using computational reduction based on feedback control information, *Advances in Intelligent Systems and Computing 348*, Springer, 2015.
[17] M. Kubalcik, V. Bobal, T. Barot, Modified Hildreth's Method Applied in Multivariable Model Predictive Control, *HELIX 2018 International Conference on Innovation, Engineering and Entrepreneurship*, 2018.
[18] M. Kubalcik, V. Bobal, T. Barot, Convenient Optimization Strategy Implemented in Multivariable Predictive Control, *MATEC Web Conf.*, Vol.210, 2018.
[19] A.M. Robert, *Linear Algebra: examples and applications*, Hackensack, 2005.
[20] R. Hunger, *Floating Point Operations in Matrix-Vector Calculus (Version 1.3)*, Technische Universität München, Associate Institute for Signal Processing, 2007.
[21] I. Krivy, J. Tvrdik, R. Krpec, *Stochastic algorithms in nonlinear regression*, Computational Statistics and Data Analysis, Vol.33, No.12, pp. 277-290.
[22] M. Vaclavik, Z. Sikorova, T. Barot, Particular Analysis of Normality of Data in Applied Quantitative Research, *2nd Computational Methods in Systems and Software 2018*, 2018.
[23] K. Sulovska, S. Belaskova, M. Adamek, Gait patterns for crime fighting: Statistical evaluation. *Proceedings of SPIE - The International Society for Optical Engineering*, Vol.8901, 2013.
[24] H.A. Noughabi, Two Powerful Tests for Normality, *Ann. Data. Sci.*, Vol.3, No.2, 2016, pp. 225-234.
[25] O. Hammer, D.A.T. Harper, P.D. Ryan, PAST: Paleontological statistics software package for education and data analysis, *Palaeontologia Electronica*, Vol.4, No.1, 2001.