2015 International Conference on Process Control (PC)
June 9–12, 2015, Štrbské Pleso, Slovakia

On-line Solution of System Constraints in Generalized Predictive Control Design

Convenient Way to Cope with Constraints

Květoslav Belda
Department of Adaptive Systems
Institute of Information Theory and Automation of the Czech Academy of Sciences
Pod Vodárenskou věží 4, Prague 8, Czech Republic
Email: belda@utia.cas.cz

***Abstract*—A specific efficient solution of constrained system outputs is presented. The solution is based on on-line weighting of individual system outputs or output components representing in fact a specific on-line tuning/applification of the components of weighting control parameters. Thus, the predictive control design retains flexibility without any increase of time-computational demands. The proposed solution represents only soft solution of the constraints, i.e. a compliance with the constraints is not fully guaranteed, but the result is satisfactory and not so far from results of hard-constraint solution computed e.g. by a quadratic programming. The solution predetermines the area of the use to the systems with fast dynamics requiring very high sampling rates as electric drives and the like. The solution is presented in the paper as a generalized solution for common use. It is demonstrated by simple simulative examples and one illustration from a real control of 3-phase electric drive.**

Keywords—Predictive control, incremental algorithms, system constraints, component tuning of control parameters, quadratic cost function, quadratic programming problem.

I. Introduction

Generalized predictive control (GPC) is an attractive control strategy due to its clarity of the design and flexibility in the solution including complex constrained multivariable control problems. The optimization of the objective or cost function is usually evaluated in each time-instant within a specific finite horizon. Then, the key question is about time-computational demands of the optimization procedure in the relation to a sampling period.

A usual solution applied in the predictive control design is an application of a quadratic programming (QP) [1]. For complex systems, which require a high sampling rate i.e. the very short sampling period to achieve adequate control, the solution of a QP problem is usually not computationally feasible. There exist approximative off-line approaches based on Multi-Parametric Toolbox [2], [3] in the literature. However, these generally usable approaches presume specific properties of controlled systems like low level of non-linearities or let us say homogenous or small number of heterogenous property areas in a state-system space including limits in variable dimensions describing the systems. From control design point of view, the controlled systems are usually multidimensional time-parameter-varying systems described possibly by linear or linearized model forms. A problem of the presence of the non-linearities follows mostly from physical features of the controlled systems. Standard GPC design relies on a linear or linearized dynamic model, respects all input and output constraints, and optimizes a quadratic cost function. Thus, the quadratic cost function together with various constraints corresponding to the system model description forms the basis for an expression of true user objectives for reliable performance of GPC.

The proposed solution in this paper follows from the questions: 'How to simply tune or to weigh individual components in the cost function?' or 'How to excite a redistributive optimization process of the cost function towards compliance with required constraints?' It takes into account individual components of controlled system inputs and possibly outputs to influence the weight of the term element containing channel-signal overshooting its constraining limits. Once some constraint is activated, the solution try to fluently manage the appropriate channel-signal closed to or up to given constraining limit. Principally, the principle consists in a specific power function with base depending on topical value of the signal and required value of the constraint and exponent as a specific additional tuning parameter.

In similar direction, there are definitely a lot of various practical solutions, which are in fact based on direct switching of individual elements of weighting penalization matrices. However, such solutions are limited by the form of individual terms involved in the cost function. Usually, only control error and its future predicted values are penalized. It does not enable optimization to gain the penalizing only system state or outputs at overshooting the constraints. In case of the term of the control actions is situation simple for positional (absolute) algorithms. However, if only increments of control actions are penalized, the problem is the same as at the control error term. Furthermore, the switching is roughly discontinued and it continues also during activated constraints without any smoothing re-tuning. In case of proposed solution, the result is not really pure result as from usual QP algorithms, but it is closed to them.

The paper is organized as follows. The problem of constraints is defined in Section II. Key elements of considered GPC design are defined in Section III. Section IV deals with the proposed on-line solution of constraints. Section V outlines suitable optimization way for GPC design. Section VI contains illustrations from simulations and real experiment. The paper concludes by Section VII, which summarizes achieved results.

II. Model and Constrains Defined for GPC Design

The problem of the GPC under specific constraints can be defined as follows. Let a linear discrete generally time-variant state-space model describing a controlled system is considered

$$\begin{aligned} \hat{x}_{k+1} &= A_k\, x_k + B_k\, u_k \\ y_k &= C_k\, x_k \end{aligned} \tag{1}$$

which is subject to the constraints

$$\begin{aligned} y_{min} &\leq y_j \leq y_{max},\ (C_j\, x_{min} \leq C_j\, x_j \leq C_j\, x_{max}) \\ u_{min} &\leq u_j \leq u_{max} \\ y_j &= r_j,\ (C_j\, x_j = r_j) \end{aligned} \tag{2}$$

at all time instants $j > 0$ and $k > 0$, where the index j gradually falls within finite intervals of the time instants $j = k, \cdots, k + N_p$, where k represents an initial time instant of the appropriate topical finite interval determined by prediction horizon N_p.

The state-space model (1) consists of the state-space matrix $A_k^{n\times n}$, input matrix $B_k^{n\times r}$ and output matrix $C_k^{m\times n}$. Individual variables in (1) and (2), $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^r$, $y_k \in \mathbb{R}^m$ and $r_k \in \mathbb{R}^m$ are the state, input, output and reference vectors, respectively. Sequentially, the constants $y_{min} \leq y_{max}$, $(C_j\, x_{min} \leq C_j\, x_{max})$ and $u_{min} \leq u_{max}$, appearing in the inequality constraints, have appropriate dimensions as system variables. These listed constants are usually given by physical features of controlled systems or they can follow from topical situations in the control process like detected abrupt obstacles or changes in a level of available input energy etc. They are crucial for safe, faultless and continuous operation. Finally, reference values r_j in equality constraints - requirements, important from user point of view, are given prior to real control process. They need not be planned by user in a compliance with physical limits involved in indicated inequality constraints.

Note that indicated alternative constrains in round brackets in (2) represents a general option of the state-space constrains instead of output constrains only. Thus, e.g. for the selection of the output matrix $C_j|_{m=n} = I^{n\times n}$, the GPC design turns to full state-space control.

To conclude this preliminary section, let us define furthermore time series or so called sequences $\Delta\hat{Y}_{k+1}$, $\hat{Y}_{k+1}$, R_{k+1}, ΔU_k, U_k and $\hat{E}_k$, which represent sequences of predictions (increments and values of future expected system outputs), references, control actions (increments and values of searched system inputs) and predictions of control errors within a given prediction horizon N_p:

$$\begin{aligned} \Delta\hat{Y}_{k+1} &= \left[\Delta\hat{y}_{k+1}, \cdots, \Delta\hat{y}_{k+N_p}\right]^T,\ \Delta\hat{y}_{k+1} = \hat{y}_{k+1} - y_k \\ \hat{Y}_{k+1} &= \left[\hat{y}_{k+1}, \cdots, \hat{y}_{k+N_p}\right]^T \\ R_{k+1} &= \left[r_{k+1},\ \cdots, r_{k+N_p}\right]^T \\ \Delta U_k &= \left[\Delta u_k, \cdots, \Delta u_{k+N_p-1}\right]^T, \quad \Delta u_k = u_k - u_{k-1} \\ U_k &= \left[u_k, \cdots, u_{k+N_p-1}\right]^T \\ \hat{E}_k &= \left[\bar{e}_k, \cdots, \hat{\bar{e}}_{k+N_p-1}\right]^T, \quad \bar{e}_k = \bar{e}_{k-1} + e_k \\ & \qquad\qquad\qquad\qquad\qquad e_k = r_k - y_k \end{aligned}$$

The definitions will usefully condensate the next explanation.

III. Key Elements of GPC Design

This section summarizes the key design elements of GPC algorithms considered in a proposed solution of constraints. The GPC algorithms are derived as discrete (digital) procedures as it is required for the digital implementation. They provide computation of control actions within one optimization calculation. Generally, the calculation employs predictions of expected future output values mathematically expressed by equations of predictions [4]. Those equations are closely related to the form of a cost function [5]. At a predictive control design, the quadratic cost function is used in different forms. In this paper, the basic absolute (positional) algorithm and two specific incremental algorithms [6] of GPC design are considered. They differ in the number of included integrators, i.e. none, one or two integrators. This is reflected in the mentioned key GPC elements: equations of predictions and corresponding cost functions.

A. Equations of Predictions

The equations of predictions express mathematically functional expressions of future system outputs in relation to unknown future control actions. Their composition influences significantly properties of computed control actions. The composition of these equations arises from discrete state-space model (1) and required control action properties. Let them are introduced successively:

- absolute (positional) algorithm

$$\hat{Y}_{k+1} = f_1\, x_k + G_1\, U_k \tag{3}$$

- incremental algorithms

$$\Delta\hat{Y}_{k+1} = f_1\, \Delta x_k + G_1\, \Delta U_k \tag{4}$$

$$\hat{Y}_{k+1} = f_{\mathrm{I}}\, y_k + f_2\, \Delta x_k + G_2\, \Delta U_k \tag{5}$$

$$\hat{E}_k = f_{\mathrm{I}}\, \bar{e}_k + R_s - f_{\mathrm{II}}\, y_k - f_3\, \Delta x_k - G_3\, \Delta U_k \tag{6}$$

where individual elements f_1, G_1; f_2, G_2; f_3, G_3; f_{I}, f_{II}; and R_s are as follows (without index $_k$ for simplification)

$$f_1 = \begin{bmatrix} CA \\ \vdots \\ CA^N \end{bmatrix}, \quad G_1 = \begin{bmatrix} CB & \cdots & 0 \\ \vdots & \ddots & \vdots \\ CA^{N-1}B & \cdots & CB \end{bmatrix} \tag{7}$$

$$f_2 = \begin{bmatrix} CA \\ \vdots \\ \sum\limits_{i=1}^{N} CA^i \end{bmatrix}, \ G_2 = \begin{bmatrix} CB & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum\limits_{i=1}^{N} CA^{i-1}B & \cdots & CB \end{bmatrix} \tag{8}$$

$$f_3 = [\ 0 \quad CA \quad \cdots \quad \sum_{i=1}^{N-1} (N-i)CA^i\ ]^T,$$

$$G_3 = \begin{bmatrix} 0 & \cdots & 0 \\ CB & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum\limits_{i=1}^{N-1} (N-i)CA^{i-1}B & \cdots\ \ CB & 0 \end{bmatrix} \tag{9}$$

$$f_{\mathrm{I}} = [\, I\ \cdots\ I\,]^T, \quad f_{\mathrm{II}} = [\, 0\ I\ 2I\ \cdots\ (N-1)I\,]^T \tag{10}$$

$$R_s = [\ 0 \quad r_{k+1} \quad r_{k+1} + r_{k+2} \quad \cdots \quad \sum_{i=1}^{N-1} \{r_{k+i}\}\ ]^T \tag{11}$$

B. Forms of the Quadratic Cost Functions

- GPC without integrator (positional algorithm)

$$J_k = (\hat{Y}_{k+1} - R_{k+1})^T Q_{YR} (\hat{Y}_{k+1} - R_{k+1}) + U_k^T \, Q_U \, U_k \quad (12)$$

where penalty matrices Q_{YW} and Q_U are defined as

$$Q_\diamond = \begin{bmatrix} Q_*^T Q_* & & 0 \\ & \ddots & \\ 0 & & Q_*^T Q_* \end{bmatrix} \left| \begin{array}{l} \text{subscripts } \diamond, *: \\ \diamond \in \{YR,\, U\} \\ * \in \{yr,\, u\} \end{array} \right. \quad (13)$$

It gives, by usual minimization, a control action vector:

$$U_k = (G_1^T Q_{YR} G_1 + Q_U)^{-1} \, G_1^T Q_{YR} \, (R_{k+1} - f_1 x_k) \quad (14)$$

- GPC with single integrator (1^{st} incremental algorithm)

$$\begin{aligned} J_k &= (\hat{Y}_{k+1} - R_{k+1})^T Q_{YR} (\hat{Y}_{k+1} - R_{k+1}) \\ &+ \Delta\hat{Y}_{k+1}^T Q_{\Delta Y} \Delta\hat{Y}_{k+1} + \Delta U_k^T Q_{\Delta U} \Delta U_k \end{aligned} \quad (15)$$

where penalty matrices $Q_{\Delta Y}$ and $Q_{\Delta U}$ are defined as

$$Q_\diamond = \begin{bmatrix} Q_*^T Q_* & & 0 \\ & \ddots & \\ 0 & & Q_*^T Q_* \end{bmatrix} \left| \begin{array}{l} \text{subscripts } \diamond, *: \\ \diamond \in \{\Delta Y,\, \Delta U\} \\ * \in \{\Delta y,\, \Delta u\} \end{array} \right. \quad (16)$$

It leads to the following control action increments vector:

$$\begin{aligned} \Delta U_k &= (G_2^T Q_{YW} G_2 + G_1^T Q_{\Delta Y} G_1 + Q_{\Delta U})^{-1} \\ &\times \Big[\; G_2^T Q_{YW} (W - f_{\mathrm{I}} y_k) \\ &- \big\{ G_2^T Q_{YW} f_2 + G_1^T Q_{\Delta Y} f_1 \big\} \, \Delta x_k \; \Big] \end{aligned} \quad (17)$$

- GPC with double integrator (2^{nd} incremental algorithm)

$$\begin{aligned} J_k &= (\hat{Y}_{k+1} - R_{k+1} - \hat{E})_k^T Q_{YR} (\hat{Y}_{k+1} - R_{k+1} - \hat{E}_k) \\ &+ \Delta\hat{Y}_{k+1}^T Q_{\Delta Y} \Delta\hat{Y}_{k+1} + \Delta U_k^T Q_{\Delta U} \Delta U_k \end{aligned} \quad (18)$$

It leads to the following control action increments vector:

$$\begin{aligned} \Delta U_k &= (G^T Q_{YW} G + G_1^T Q_{\Delta Y} G_1 + Q_{\Delta U})^{-1} \\ &\times \Big[\; G^T Q_{YW} \bar{e}_k + G^T Q_{YW} W + G^T Q_{YW} W_s \\ &- G^T Q_{YW} (f_{\mathrm{I}} + f_{\mathrm{II}}) y_k \\ &- \big\{ G_2^T Q_{YW} (f_2 + f_3) + G_1^T Q_{\Delta Y} f_1 \big\} \, \Delta x_k \; \Big] \end{aligned} \quad (19)$$

where $G = G_2 + G_3$.

IV. Solution of Constraints in GPC

As was mentioned in Section I, there exist generally two main directions in the constraint solutions: the first follows from Quadratic Programming (QP) or it is motivated by this, and the second represents heterogenous set of different ad hoc solutions based on specific penalization tuning or use specific properties of controlled systems as redundant actuation. Note, that second direction solves usually the problem partially, i.e. either state-space/output system constraints or input (control actions) constraints.

This section firstly briefly summarizes initial points of standard solution as QP-problem and then focusses on novel fast on-line solution, which belong more or less to the solution set of second direction, but it can be considered as more general and more universal for practical application to the systems with high dynamics requiring high sampling rates.

A. Standard Solution via Quadratic Programming

QP-problem is usually defined for GPC as follows

$$\min_{U_k} J_k \;=\; \min_{U_k} \frac{1}{2} U_k^T \, \mathrm{H}_k \, U_k \;+\; \mathrm{f}_k^T \, U_k \quad (20)$$

subject to

$$\begin{aligned} A_0 U_k &\le U_{max} \\ -A_0 U_k &\le -U_{min} \\ A_1 U_k &\le Y_{max} \; - b_1, (\;\; A_2 U_k \le \; X_{max} \; - b_2) \\ -A_1 U_k &\le -Y_{min} + b_1, (-A_2 U_k \le -X_{min} + b_2) \end{aligned}$$

where H_k is a quadratic matrix term and f_k^T is a linear vector term. E.g. for basic positional GPC, the appropriate parameters can be defined as follows

$$\begin{aligned} \mathrm{H}_k &= (G_1^T Q_{YR}^T \, G_1 + Q_U) \\ \mathrm{f}_k^T &= (f_1 x_k - R_{k+1})^T Q_{YR} \, G_1 \\ A_0 &= I^{(r N_p \times r N_p)}, \;\; A_1 = G_1, \;\; b_1 = f_1 x_k \end{aligned} \quad (21)$$

Thus, in general, the elements in (21) are the functions of the matrices of the state-space model (1) and input and output weighting penalization matrices. In addition, the matrices A_i and vectors b_i follow partly from initial inequality constraints (2) and partly are based on the state-space matrices from (1) included in the equations of predictions (3) - (6). Note that equality constraints (i.e. user references R_{k+1}) are preferably involved in the linear term f_k^T. This is initial definitions, and then, QP algorithms are applied.

B. Novel Fast On-line Solution of Constrains

Due to advantageous properties of the weighting control parameters (penalization matrices (13) and (16)), which balance the terms in quadratic cost functions (12), (15) and (18), the computation of control actions can be specifically tuned during real control process. Direct tuning of the parameters is suitable for determination of the GPC controller stiffness or can be used for smoothing of the signals in incremental definitions.

To solve constraints of the signals, let us consider a different idea based on a specific inverse tuning. It is possible due to proportional character and coupling in the cost function. Instead of the increase of the corresponding penalty matrix, when some real constraint is activated, it can be considered increase of individual channel of output or state possibly control action, i.e. specific element of their vectors. This inverse excitation artificially enlarges control error, which is naturally propagated through functional predictions over whole horizon. Thus, the appropriate constant penalization matrix responses by effort to balanced such artificial control error outlier by generation of the control actions, i.e. corresponding control reaction to the activated constraint, and the like. This idea eliminates complicated changing of the individual elements of the penalization matrices and gives more capability in constrained cases. Note that indicated idea cannot guarantee or compliance with all constraints. It represents rather a soft solution of the constraints. However, it is simple to implement in the control design.

The verbally described idea can be realized as follows. Let us consider some real system output or state-space element, which topically activates corresponding constraint and let such an element is marked by the index i, then for:

- symmetric constraints i.e. $y_{mix} = y_{max} = |y_{min}|$

$$yE(i)_k := y(i)_k;$$

$$if\ |y(i)_k| \geq y(i)_{mix} - |r(i)_{k+1}|\ k_s,$$

$$yE(i)_k := \left(\frac{|y(i)_k + r(i)_{k+1}\, k_s|}{y(i)_{mix}} \right)^{k_{sp}} y(i)_k; \qquad (22)$$

$$end$$

- asymmetric constraints for $y_{min} < y_{max}$

$$yE(i)_k := y(i)_k;$$

$$if\ y(i)_k \geq y(i)_{max} - |r(i)_{k+1}|\ k_s,$$

$$yE(i)_k := \left| \frac{y(i)_k + |r(i)_{k+1}|\, k_s}{y(i)_{max}} \right|^{k_{sp}} y(i)_k; \qquad (23)$$

$$elseif\ y(i)_k \leq y(i)_{min} + |r(i)_{k+1}|\ k_s,$$

$$yE(i)_k := \left| \frac{y(i)_k + |r(i)_{k+1}|\, k_s}{y(i)_{min}} \right|^{k_{sp}} y(i)_k; \qquad (24)$$

$$end$$

where coefficients k_s , $k_{sp} \in \mathbb{R}$, $k_s \geq 0$ and $k_{sp} \geq 1$ are, let us say, 'safety' and 'power' coefficients.

The main coefficient is a 'power' coefficient, which amplifies given values of the particular channel under its active constraint. The higher amplification the higher discrepancy in the appropriate term of the cost function cross the whole prediction horizon N_p. The higher discrepancy together with unchanged penalization matrix extensively overruns other values. As mentioned, it causes the excitation leading to the different redistribution of the searched control actions (input energy) within the optimization.

The second coefficient k_s relates to the level of the compliance with the activated constraint. In case of only small difference between given constraint and corresponding reference value or compliance of the reference with the constraint, it can be zero, i.e. $k_s = 0$. However, provided that big discrepancy between the required target (user required reference) and corresponding constraint level (e.g. sudden obstacle), the 'safety' coefficient adjusts safety zone in front of given constraint.

Returning back to the ground of the expressions (22) and (23) or (24), then the exponentiation of the base terms determines the amplification ≥ 1 of the original element value.

At real constraint activation, the signal is exponentially increased. Nevertheless, signal values (e.g. $yE(i)_k$) can be used as other unchanged particular input value signals (e.g. $y(i)_k$) to whichever suitable optimization procedure used in GPC design. Note that in the incremental algorithms, especially in the 2^{nd} incremental algorithm, an anti-wind up has to be solved, e.g. cumulated control error $\bar{e}_k$, increase of which has to be broken in, otherwise the controller or system stays stuck and cannot itself directly disengage.

V. Suitable Optimisation Way for GPC Design

To optimize the equations (12), (15) or (18), besides indicated simple results (14), (17) or (19), let us consider the following example, which is applicable on all indicated cost functions. It is based on square-root decomposition as follows

$$\min_U J_k \;=\; \min_U \mathbb{J}_k^T \mathbb{J}_k \quad \rightarrow \quad \min_U \mathbb{J}_k \qquad (25)$$

which indicates the possibility to optimize the square-root of the cost functions only, i.e.

$$\min_U \mathbb{J}_k \;=\; \min_U \begin{bmatrix} \mathbb{Q}_{YR} & 0 \\ 0 & \mathbb{Q}_U \end{bmatrix} \begin{bmatrix} \hat{Y}_{k+1} - R_{k+1} \\ U_k \end{bmatrix} \qquad (26)$$

where square-roots $\mathbb{Q}_{YR}$ and $\mathbb{Q}_U$ follow from $Q_{YR} = \mathbb{Q}_{YR}^T \mathbb{Q}_{YR}$ and $Q_U = \mathbb{Q}_U^T \mathbb{Q}_U$. The expression (26) can be interpreted by the corresponding system of the algebraic equations that way

$$\begin{aligned} \begin{bmatrix} \mathbb{Q}_{YR} \\ \mathbb{Q}_U \end{bmatrix} U_k \;&-\; \begin{bmatrix} \mathbb{Q}_{YR}(R_k - f_1 x_k) \\ 0 \end{bmatrix} = 0 \\ A \quad U_k \;&-\; b = 0 \\ Q^T A \quad U_k \;&=\; Q^T b \quad \text{with respect to} \quad A = QR \\ R_1 \quad U_k \;&=\; c_1 \end{aligned} \qquad (27)$$

Orthogonal matrix Q^T transforms the matrix A to upper triangular matrix R_1 as follows

$$\begin{bmatrix} \; A \; \end{bmatrix} \begin{bmatrix} U_k \end{bmatrix} = \begin{bmatrix} b \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \begin{bmatrix} U_k \end{bmatrix} = \begin{bmatrix} c_1 \\ c_z \end{bmatrix} \qquad (28)$$

where c_z is a loss vector. Its Euclidian norm $||c_z||_2 = \sqrt{J_k}$. Finally, the searched control action is

$$u_k = U_k(1\!:\!r)\,, \quad \text{or} \quad u_k = u_{k-1} + \Delta U_k(1\!:\!r) \qquad (29)$$

for positional algorithm and both incremental algorithms, respectively.

VI. Illustrative Examples

In this section, simple simulative examples for all proposed algorithms and a one illustrative representative example from real control process (drive speed control) are discussed. Both simulative examples and one real example demonstrate slightly different target constraints. However, they are solved similarly via proposed on-line solution in the Subsection IV.B.

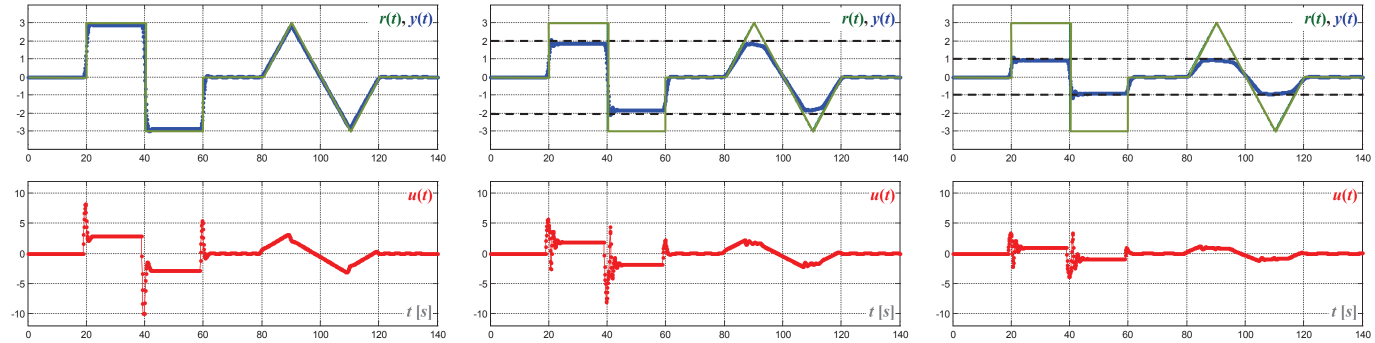
A. Simulative Examples

The simulative examples are based for simplicity on control results with a single-input single-output system of second order described by the following continuous transfer function

$$G_s(s) \;=\; \frac{1}{s^2 + 2s + 1} \qquad (30)$$

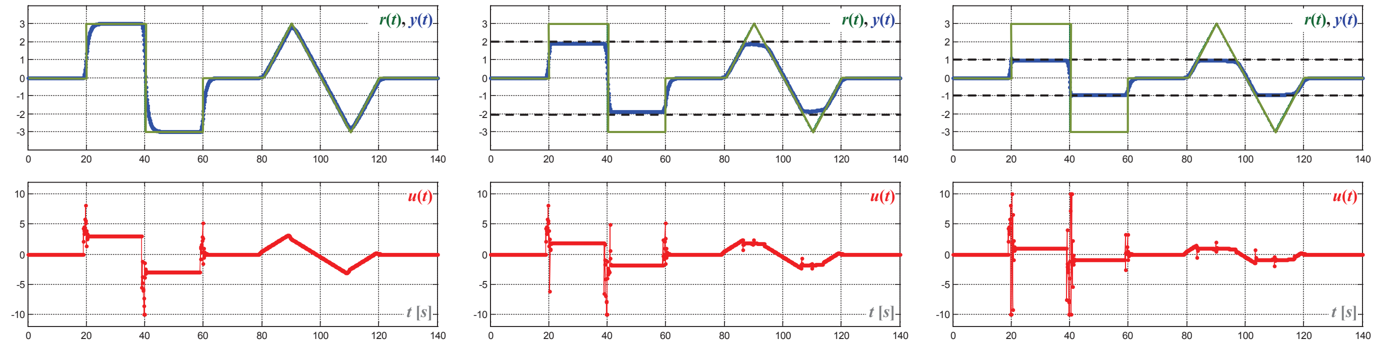
and corresponding transfer function in discrete-time domain

$$G_s(z^{-1}) = \left. \frac{0.0047 + 0.0044z^{-1}}{1 - 1.81z^{-1} + 0.82z^{-2}} \right|_{Ts=0.1s} \qquad (31)$$



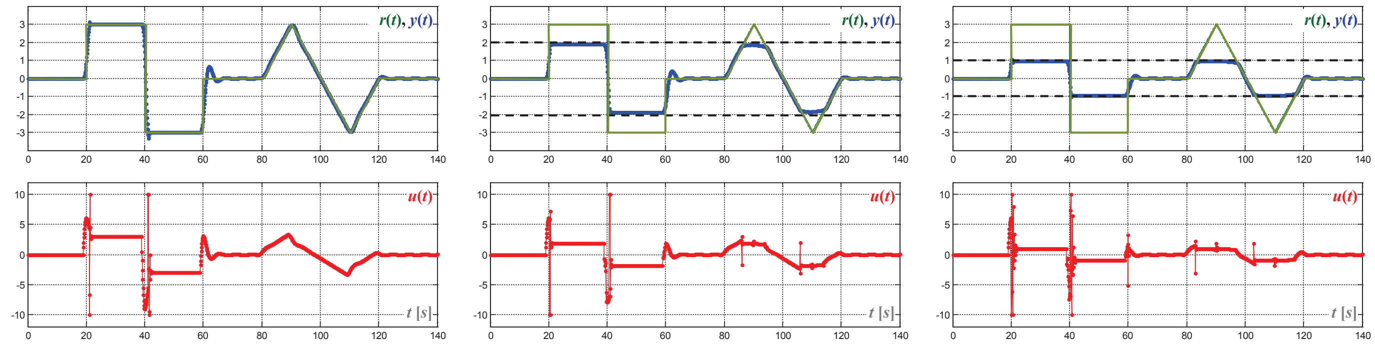
(a) (b) (c)

Fig. 1. GPC positional algorithm: (a) boundary constraints (ks = 0), (b) constraints y_{mix} = ±2, and (c) constraints y_{mix} = ±1.



(a) (b) (c)

Fig. 2. GPC 1^{st} incremental algorithm: (a) boundary constraints (ks = 0), (b) constraints y_{mix} = ±2, and (c) constraints y_{mix} = ±1.



(a) (b) (c)

Fig. 3. GPC 2^{nd} incremental algorithm: (a) boundary constraints (ks = 0), (b) constraints y_{mix} = ±2, and (c) constraints y_{mix} = ±1.

Simulative examples are in Fig.1 - Fig.3. The figures show behavior of all mentioned GPC algorithms for case, if user reference signals are close to constraint boundaries or they violate them completely. Such situations can happen both in limiting cases and mainly in sudden obstacles in a system workspace. The simulative experiments were adjusted identically for each GPC procedure:

- positional algorithm
  $N = 10, Q_{yr} = I, Q_u = diag(0.15);$
- 1^{st} incremental algorithm
  $N = 10, Q_{yr} = I, Q_{\Delta y} = diag(8), Q_{\Delta u} = diag(0.005);$
- 2^{nd} incremental algorithm
  $N = 10, Q_{yr} = I, Q_{\Delta y} = diag(50), Q_{\Delta u} = diag(0.05);$

The initial given (required) user reference signal, composed as $r(t) = [steps\,[0,\ 3, \text{-}3,\ 0], triangles\,[0,\ 3,\ 0, \text{-}3,\ 0]]$, entered control algorithms during whole control process unchanged. An obstacle, consisting in constrained operational range $(\pm 2, \pm 1)$, was a separate signal, which directly did not enter algorithms, but it participated in the inequality conditions of the solution of constrains - see Subsection B, Section IV.

Especially, at 2^{nd} incremental GPC algorithm (Fig.3a, 3b), differences are obvious in the overshoot suppression during activated constraint limits (3 in 20s, -3 in 40s) and no constraint activation (0 in 60s). In case of activation, the control actions are more boosted to decelerate motion of the system.

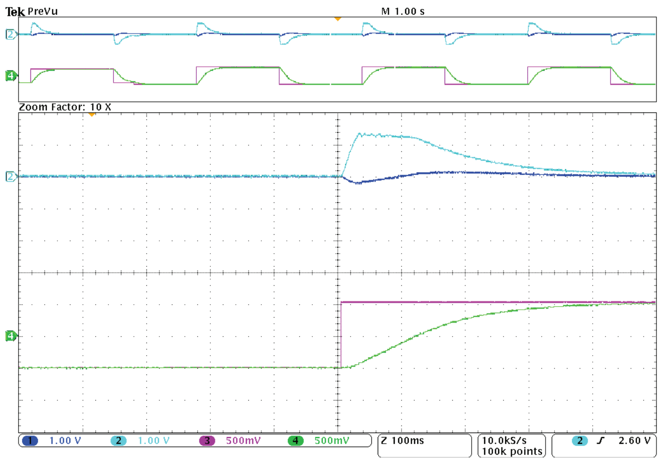


Fig. 4. Current limitation combined with field weakening by 1^{st} algorithm ($k_s = 0$) within speed control; command speed $\pm$1000rpm; ch1: i_{S_d} current (25A/1V), ch2: i_{S_q} current (25A/1V), ch3: command el. rotor speed (135Hz/1V), ch4: measured el. rotor speed (135Hz/1V).

B. Representative Example in Real Control

Representative example of the application of proposed constraint solution in real control was realized within a speed control task of 3-phase permanent magnet synchronous motor [7]. That motor represents multi-input multi-output system (two inputs: input voltages - $[u_{S_d}, u_{S_q}]^T$; four state variables: current components, angular el. speed and drive torque - $[i_{S_d}, i_{S_q}, \omega_e, \tau]^T$; and three controlled outputs: only selection from the state - $[i_{S_d}, i_{S_q}, \omega_e]^T$). The control process is limited by maximum admissible current levels.

In drive control, there is a problem with an overshooting of the physical current constraints. If it happens, then current guards disconnect power supply. It is an undesirable interruption of the process. Therefore, an effort is to avoid such interruptions by compliance of the control design with the current constraints. In this particular case, the current limit for individual components was 40A. For safety's sake, the really used limit was set for 36A. Due to the requirement for the smallest possible currents (i.e. constant zero current reference values), the coefficient k_s was set for zero only. It is given by the references, which are in the compliance with constraints $r_{i_{S_d}} = r_{i_{S_q}} \rightarrow 0 \ll i_{S_{d,q}max}$. The executive torque-current component was limited under the given constraint.

The detail of the Fig. 4 within time range 100ms shows current limitation of the i_{S_q} - channel 2 (upper part) at the step change of the speed reference (lover part). During the step change, due to maximum admissible supply voltage, a field-weakening of the current i_{S_d} - channel 1 (upper part) happened without any influence of the applied current limitation of i_{S_q}.

VII. Conclusion

The paper deals with the specific inverse tuning-'boosting' of the individual cost terms in the quadratic cost functions, which can influence a control energy redistribution during GPC optimization process in contrary to 'direct' tuning e.g. [8]. The proposed tuning idea can be profitably used at constrained cases with no increase of computation-time demands and no changing of the used GPC algorithm.

The proposed solution causes full propagation trough prediction horizon. However, the inverse tuning-'boosting' may influence a specific component and comply with its appropriate constraint without influence of the control design quality, if constraints are not activated. This procedure can be considered for system outputs, selected state variables or system inputs. In this paper, the constraints on system outputs (Subsection VI.A) and selected state variables (Subsection VI.B) were considered. Small differences between a specific constraint and corresponding reference value is caused by the activation instant. Proposed predictive control algorithms generate control actions till some constraint is activated. Immediately after activation, the computation of the control is influenced by the inverse tuning-'boosting'. Such tuning represents pushing system to comply with activated constraint, but it leads to small chatter around given constraint as discussed at the end of the Subsection IV.B.

Besides, the specific design of predictive control algorithms was presented for common reference signals. The proposed solution of constraints was involved into all those algorithms and the main algorithm features were discussed.

References

[1] N. Gould and P. L. Toint, "Preprocessing for quadratic programming." *Math. Programming, Series B*, vol. 100, pp. 95–132, 2004.

[2] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.

[3] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari, "Multi-Parametric Toolbox (MPT)," *Hybrid systems: computation and control*, pp. 448–462, 2004.

[4] A. Ordis and D. Clarke, "A state-space description for GPC controllers," *Int. J. Systems SCI.*, vol. 24, no. 9, pp. 1727–1744, 1993.

[5] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2009.

[6] K. Belda, "On offset free generalized predictive control," in *Proc. of 19th Int. Conf. on Process Control*, Slovak Republic, 2013, p. 6 pp.

[7] K. Belda and D. Vošmik, "Explicit GPC algorithms for speed control of PMSM drives," in *39th IEEE IES Conf.*, Austria, 2013, pp. 2833–2838.

[8] K. Belda, "On-line parameter tuning of model predictive control," in *Proceedings of the 18th IFAC World Congress*, Italy, 2011, pp. 5489–5494.